PHD

An investigation into the control of automated venetian blinds

Skelly, Mark J.

Award date:
2002

Awarding institution:
University of Bath

An Investigation into the Control of Automated Venetian Blinds

Submitted by Mark J. Skelly
for the degree of Ph.D.
of the University of Bath

ProQuest LLC
789 East Eisenhower Parkway
P.O. Box 1346
Ann Arbor, MI 48106-1346

Abstract

In addition to other benefits, the venetian blind offers a building occupant protection against thermal discomfort, glare and visual intrusion. However if controlled incorrectly, the operation of blinds can have a detrimental effect on the daylight contribution within a room and lead to the unnecessary use of electric lighting. In an attempt to overcome this problem, many designers have chosen to automate the operation of venetian blinds in order to integrate their control within complex environmental control strategies. Unfortunately, recent occupancy surveys have shown that automated blind control systems do not often meet the expectations of their users and that there is a need for further development.

This thesis starts by investigating some of the shortcomings of current systems. Firstly by re-examining the interaction between the individual and the blind, and secondly by reviewing the technology of blind automation. It considers the factors that influence an occupant's decision to alter their blinds, and identifies the importance of a complex series of site-specific physical and contextual constraints that are often inter-twined with relatively unpredictable individual psychological parameters. Such complexity makes it difficult for a traditional fixed automatic control system to accurately pre-empt any action that an individual might need to take to alleviate discomfort. For example, the position and orientation of an occupant's task area and the shading of trees are among some of the factors that are either difficult or too expensive to assess on an individual building scale.

The thesis uses these initial findings to offer a fresh approach to blind control. It proposes a new framework that enables an energy conscious blind control system to adjust its original control mappings in response to patterns in individual user over-rides. Such a framework would give a system the opportunity to adapt to individual conditions that could not have been foreseen when the algorithms were written. Thus making it flexible and able to adapt to a variety of scenarios, building types and individual occupants.

Contents

List of Figures

Chapter One

Chapter Two

Chapter Three

Chapter Four

Chapter Five

Chapter Six

Chapter Seven

Chapter Eight

Chapter Nine

Chapter Ten

Appendix I

Appendix II

Appendix IV

Appendix V

Appendix VI

Appendix VII

Appendix VIII

Appendix X

List of Tables

Chapter Four

Chapter Eight

Appendix I

Appendix IV

Appendix V

Appendix VI

Appendix VII

Appendix VIII

Acknowledgements

The author would like to thank the following individuals for their encouragement and support during this research:

My supervisors
Martin Wilkinson of the School of Architecture and Civil Engineering
Stephen Ledbetter of the Centre for Window and Cladding Technology

I would also like to thank
Max Fordham, my mentor
and
Stephanie, my wife
for their continual support.

Chapter 1: Introduction

1.1 The Venetian Blind

Venetian blinds are traditionally manually operated shading devices and one of the more common forms of solar control in the UK today. Essentially they are horizontal louvre blinds, whose overlapping slats tend to be made from either wood, aluminium or perforated aluminium (see Figure 1.1). They can be mounted either internally, externally or within glazing elements and are available in a variety of slat sizes, ranging from a 15mm micro-blind, used predominately in sealed glazed units, through to 80-90mm blinds, used largely in external or double skin applications.

Figure 1.1: Picture showing a typical venetian blind system with 80mm aluminium slats, used for external and mid-pane applications

The venetian blind can offer the user two modes of operation. The first mode, the lift mode, is used to raise or lower the slats depending on whether solar protection is necessary, thus enhancing the daylight contribution from the glazed elements of the facade on overcast days. The second mode, the tilt mode, is used to adjust the angle of the slats to ensure that the sun's rays are obstructed whilst still admitting a certain amount of daylight and allowing partial views through the blinds. Figure 1.2 shows an example of the internal mechanisms that allow these two modes of control to operate on a standard venetian blind. The two outer cords (ladder cords) control the tilt and the central cord (the lift cord) controls the lift.

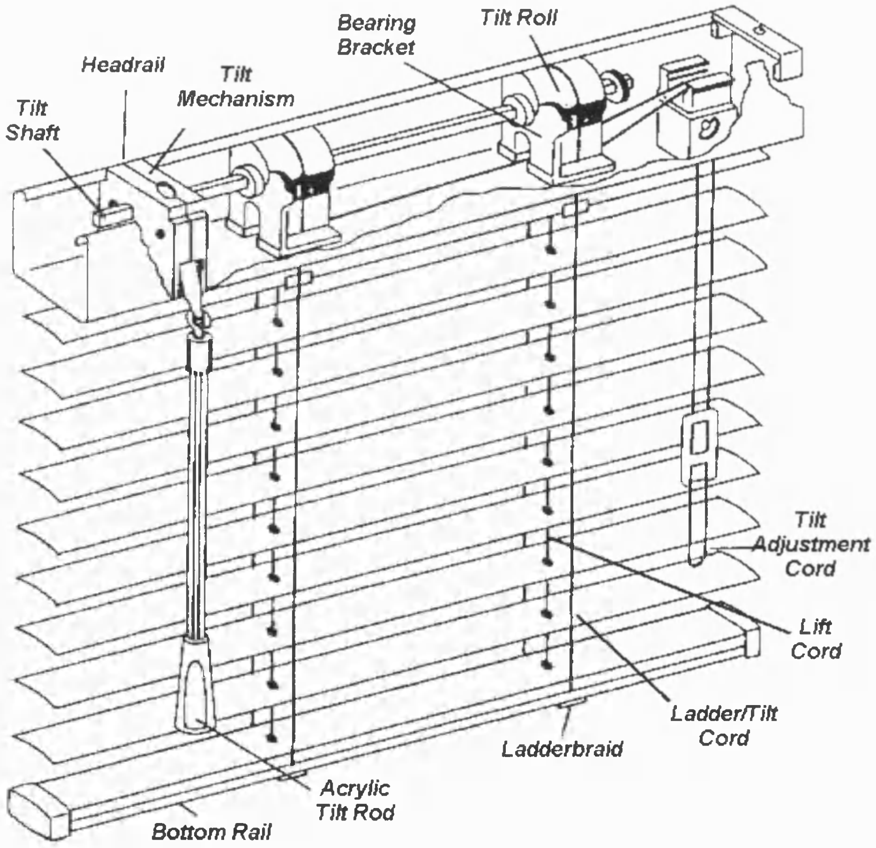


Figure 1.2: Diagram showing the major components of a manually operated dual control venetian blind.[1] [2]

---

[1] Diagram adapted from figure in The British Blind & Shutter Association (BBSA) "Specifiers' Guide to Blinds and Shutters for 1998-1999",March 1998, BBSA, UK.

[2] Definitions of components taken from prEN 12216, "Terminology and Definitions for Blinds and Shutters", 1995. British Standards Institute, London.

The tilt mode of the venetian blind enables the slats to be placed in any position within a 180 degrees range, between the outer edge of the blind pointing downwards, through horizontal, to the outer edge of the blind pointing upwards. This large extent of motion coupled with the fact that the blind slats are often curved results in a device that is optically and thermally very complex and difficult to model.[3] [4]

Not all venetian blind systems provide control of both the lift and the tilt modes of operation. For example, some mid-pane micro-blinds and large external blinds only offer tilt adjustment and cannot be raised. However, the availability of both modes is advantageous from both an energy and comfort perspective.

1.1.1 The Role of The Venetian Blind

Historically, the primary role of a venetian blind is to offer occupants protection against visual intrusion, glare and solar penetration. However, following a gradual growth in environmental awareness over the last few decades, the blind is now viewed by many building designers as an additional tool for utilising free energy from the external environment to help achieve a comfortable and energy efficient internal environment.

Despite only offering modest reductions in solar heat gains,[3] [5] [6] [7] the use of the most common form of venetian blind, the internal venetian blind, can lead to considerable energy savings, if controlled correctly and adjusted several times a day. [8] [9] [10] [11] [12]

---

[3] Klems J.H. and Warner J.L., 1997, "Solar Heat Gain Coefficient of complex fenestrations with a venetian blind for differing slat tilt angles", *ASHRAE Transactions*, Vol.103, pt.1, pp1026-1034.

[4] Papamichael K. and Selkowitz S., 1986, "The luminous performance of vertical and horizontal slat-type shading devices", 1986 Illuminating Engineering Society of North America Conference, Boston, Massachusetts, pp232-282.

[5] Nicol J.F., 1966, "Radiation transmission characteristics of blind systems", *Building Services Journal*, pp1-5.

[6] Jones R.H.L., 1980, "Solar radiation through windows - Theory and equations", *Building Services Engineering Research & Technology*, 1, pp83-91.

[7] Fisk D.J., 1981, "Thermal control of buildings", Applied Science, London.

[8] Lee E.S. and Selkowitz S.E., 1995, "The design and evaluation of integrated envelope and lighting control strategies for commercial buildings", ASHRAE Transactions, Vol.101, Pt.1, pp326-342.

[9] Mills L.R. and McCluney W.R., 1993, "The benefits of using window shades", *ASHRAE Journal*, Nov 1993, pp20-27.

Both the raising and lowering of the blinds and the adjustment of blind tilt angle can help balance the solar energy exchange through the facade depending on seasonal needs. Indeed the blind can serve a series of functions, to block the solar heat gain, to balance the daylight contribution within the room, and to reflect daylight onto the ceiling and deeper into the space,[13] all whilst providing the option of a view to the outside.

Unfortunately, field studies show that occupants very rarely adjust blinds, especially the slat angle.[14] [15] [16] [17] Such behaviour leads to the unnecessary use of artificial lighting and this fact has stimulated engineers to develop automatic photoelectric switching controls for lighting and blind systems.[10] [18] [19] [20]

---

[10] Newsham G.R., 1994, “Manual control of window blinds and electric lighting: Implications for comfort and energy consumption”, *Indoor Environment*, 3, pp135-144.

[11] Cho S.H., Shin K.S. and Zaheer-Uddin M., 1995, “The effect of slat angle of windows with venetian blinds on heating and cooling loads of buildings in South Korea”, *Energy*, Vol.20, No.12, pp1225-1236.

[12] Selkowitz S.E. et al, 1994, "Envelope and lighting technology to reduce electric demand: Final Report - Phase II", Lawrence Berkeley Laboratory, University of Califoria, Berkeley, CA, January 14 1994.

[13] Eames P.C. and Norton B., 1994, "A window blind reflector system for the deeper penetration of daylight into rooms without glare", International Journal of Ambient Energy, Vol.15, No.2, pp73-77.

[14] Rubin A.I., Collins B.L. and Tibbott R.L., 1978, “Window blinds as a potential energy saver: A case study,” NBS Building Science Series No.112, Washington.

[15] Collins B.L., 1979, “Window management: An overview”, *ASHRAE Transactions*, Vol.85, pt.2, pp633-637.

[16] Rea M.S., 1984, “Window blind occlusion: A pilot study”, *Building and Environment*, Vol.19, No.2, pp133-137.

[17] Inoue T. et al., 1988, “The development of the optimal control system for window shading devices based on investigations in office buildings”, *ASHRAE Transaction*, Vol.94, pp1034-1049.

[18] Owens P.G.T., 1978, "The energy implications of comfort criteria: Visual aspects of comfort affecting energy consumption", Proceedings of Man, Environment and Buildings, Loughborough University, 28 Sept 1978, ppE1-E10.

[19] Papamichael K., Rubinstein F., Selkowitz S. and Ward G., 1986, “The integration of operable shading systems and electric lighting controls”, Proceedings of the 1986 International Daylighting Conference, Nov 5-7, Long Beach CA, pp111-121.

[20] Shavit G. and Wruck R., 1993, "Energy conservation and control strategies for integrated lighting and HVAC systems", *ASHRAE Transactions*, Vol.99, pp785-789.

1.2 The Automated Venetian Blind and 'The Intelligent Facade'

1.2.1 Blind Automation

The automation of a venetian blind requires two major components. Firstly a motor, to drive the tilt and the lift cord motion and secondly a controller, to make decisions on when to adjust the tilt or lift.

Blind motors tend to be either 24V DC or 240V AC motors that are mounted in the head rail of the blind. For smaller blinds, a single tubular motor can be used to control both blind lift and tilt (see Figure 1.3). In larger blinds, separate motors are often used to perform both functions, although in recent years single, but complex, digital motors have started to be developed.

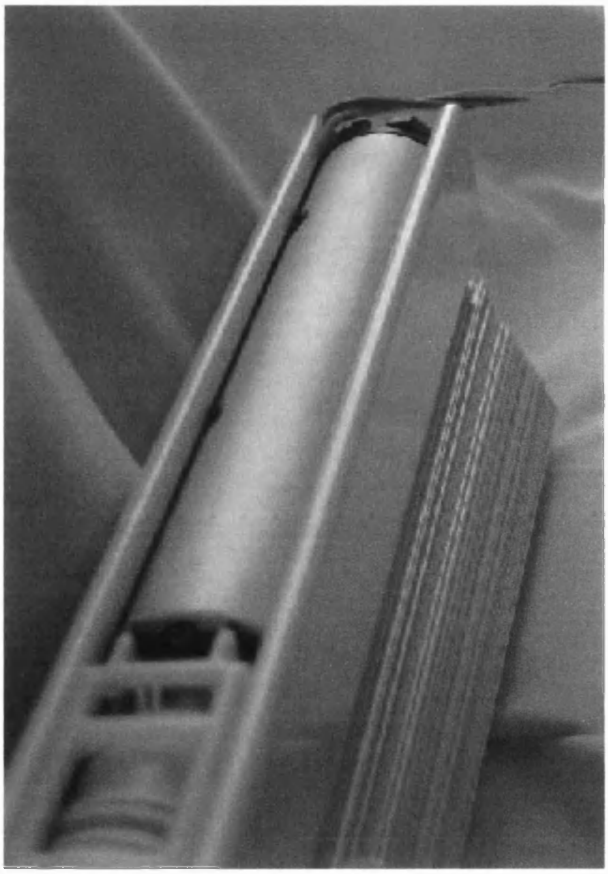


Figure 1.3: Picture of a 24V tubular motor located in the head-rail of a 25mm venetian blind. The motor is used to drive both the tilt and the lift motion.

The blind controller simply uses a control algorithm to process a number of inputs to produce the desired control action. Various types of control algorithms with different levels of complexity will be reviewed in Chapter Three. The inputs can originate from a single or variety of different sensor types, depending on the complexity of the system.

On its own, blind automation does not offer a guarantee of low energy consumption. Appropriate lighting and heating controls are also required to dim lights gradually as daylight levels increase, and to adjust temperature set points in response to solar energy exchange through the facade. In addition, it has been shown that the use of zoned occupancy sensors in such systems can also yield significant energy savings. [21] [22] [23] [24]

By integrating the relatively simple automated venetian blind with all of these different systems and their associated sensors, the blind has evolved further to form part of a complex environmental control system, termed by many 'The Intelligent Facade'.

1.2.2 'The Intelligent Facade'

The origins of the concept of the 'Intelligent Facade' and its role in the 'Intelligent Building' are reviewed in Appendix I. For the purposes of this introduction it is suffice to say that the 'Intelligent Facade' uses each of its components, either individually or accumulatively, to adjust the characteristics of the facade in response to a series of environmental variations. By doing so a dynamic envelope, when integrated with the building services systems, can achieve significant reductions in annual and peak energy demand. This goal requires the operational control algorithm

---

[21] Deloe D., 1986, "Sensors light the way from security to energy savings", *Energy Management Technology*, Vol.10, No.1, pp48-49.

[22] Jankowski W., 1993, "Occupancy sensors yield 25-75 percent energy savings", *Facilities Design and Management*, Vol.12, No.1, p25.

[23] Mahdavi A. et al., 1995, "Effects of lighting, zoning and control strategies on energy use in commercial buildings", *Journal of the Illuminating Engineering Society*, Winter 1995, pp25-35.

[24] Richman E.E., Dittmer A.L., Keller J.M., 1996, "Field analysis of occupancy sensor operation: Parameters affecting lighting energy savings", *Journal of the Illuminating Engineering Society*, Winter 1996, pp83-92.

to be designed to balance the energy use resulting from cooling and lighting, and to accommodate occupant comfort and individual preferences (see Figure 1.4).

In theory, 'The Intelligent Facade' is a building skin, analogous to the human skin, that adapts to and learns from its surroundings in order to optimise its performance. In practice it is simply a set of basic automated components, responding to a set of standard environmental inputs, showing no signs of intelligence at all. Indeed, despite the many predictions of the future capabilities of the building skin, and the fact that the foundations for its technological advancement exist, there are still a series of fundamental hurdles to its development that must first be overcome before such prophecies can be realised.

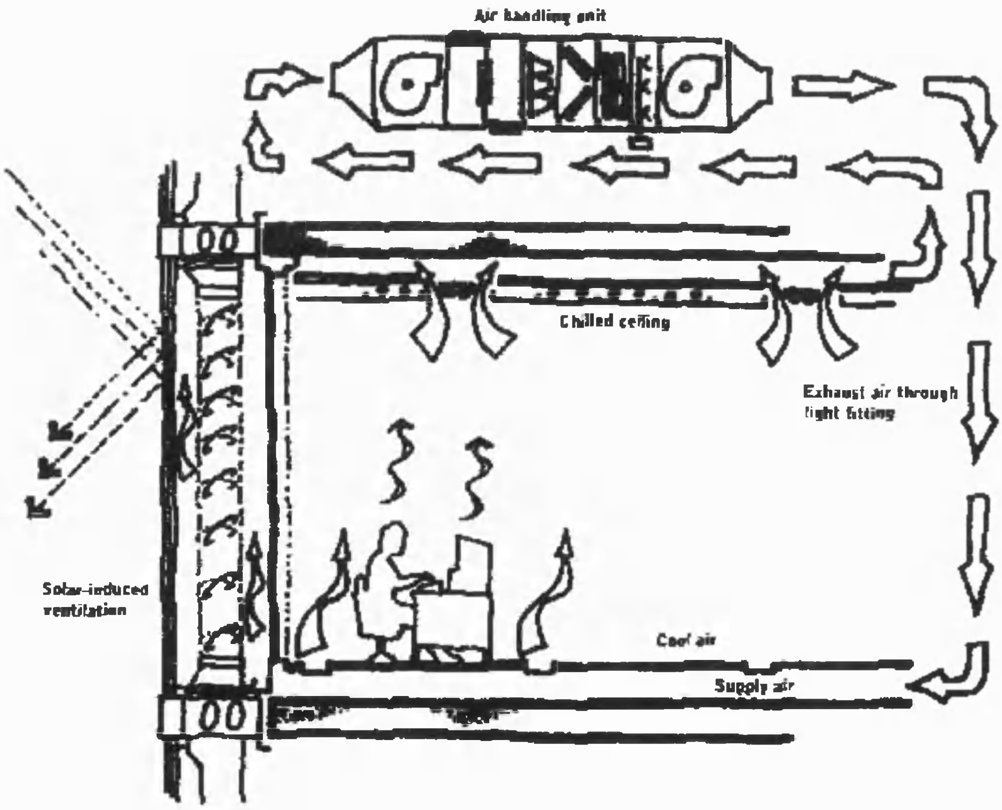


Figure 1.4: A sketch produced by Ove Arup and Partners showing the concept of the double skin "Intelligent Facade" at the Helicon Building in London. Courtesy of Arup Facade Engineering.

1.3 The Individual and 'The Intelligent Façade'

Until now the evolution of the venetian blind, from a simple manually controlled shading device to an energy saving 'Intelligent Facade' device, has largely neglected the needs of the user. This is largely due to the fact that many designers felt that the only way that they could eliminate the uncertainty of human behaviour in their carefully balanced, automated, environmental systems was to abolish all user intervention. Overlooked in the desire to save energy was one of the prime considerations of the end user, namely "will the people be comfortable?"[25]

1.3.1 The Individual and The Automated Environmental Control System

Post occupancy analyses of occupants' reactions to automated systems has found that when technology was applied to a local environmental control system without sufficient user over-ride, or where the systems had to be over-ridden constantly, the users became frustrated with those systems.[26] [27] This frustration led to discomfort, which in turn can lead to a decrease in productivity and eventually stress, despite the physical environmental parameters being well within predetermined comfort limits.

Economically small increases in productivity can easily outweigh gains from reduced energy consumption. [28] Thus it is important to remember that the term energy efficiency means using our energy effectively. In other words meeting occupant requirements for comfort and productivity as well as reducing energy consumption. [29] Therefore, any energy saving measures introduced to automated control systems should not adversely affect user productivity and the needs of individual occupants.

---

[25] Holz R. et al, 1997, "Effects of standard energy conserving measures on thermal comfort", *Building and Environment*, Vol.32, No., pp31-43.

[26] Bordass W., Leaman A., 1997, "From feedback to strategy", Buildings in use '97: How buildings really work. Commonwealth Institute, London.

[27] Veitch J.A. and Gifford R., 1996, "Choice, perceived control, and performance decrements in the physical environment", *Journal of Environmental Psychology*, Vol.16, pp268-276.

[28] Wyon D.P.,1993, "The economic benefits of a healthy indoor environment", Proceedings of Healthy Air '94, Italy, pp.405-416.

[29] Oseland N.A. and Williams A., 1997, "How best practice can improve productivity: The relationship between energy efficiency and staff productivity", Creating the Productive Workplace Conference, Workplace Comfort Forum, London.

1.3.2 The Individual and The Automated Venetian Blind

When considering the venetian blind, the evidence suggests that the application of energy conserving measures to an automated blind control strategy has led to a decrease in the user's perception of control and ownership of that system; both factors that could be attributed to the manually controlled blind's popularity. A study by Bordass, Bromley and Leaman in 1993 [30] identified that people were:

> *"....adversely affected by automated systems which make abrupt and seemingly capricious changes"..... "and found considerable occupant hostility to automatically controlled venetian blinds for this reason."*

It went on to say:

> *"Those advocating fully automatic control of natural light and glare should proceed with caution."*

Similar findings are noted within a post occupancy survey carried out by Ure and Dunham [31] which found that, in the building studied, 80% of the occupants were unhappy with the window blinds, which formed part of a daylight linked lighting system.

An earlier more detailed study, undertaken by Inoue et al in Japan, found that only 50% of the building's occupants considered automatic control worthwhile.[17] Typical comments written by the staff in researcher's questionnaires included:

> *"Blinds actuate even when one feels it is not required"* and *"Blinds do not actuate even when one feels it is required"*

---

[30] Bordass W., Bromley K. and Leaman A., 1993, "User and Occupant Controls in Office Buildings", Prepared for a conference entitled "Building Design, Technology and Occupant Well-being in Temperate Climates", Brussels, February 1993.

[31] Ure J.W. and Dunham S.J., 1997, "Lighting Control and Management Systems for Offices: Achieving User Satisfaction and Lower Operating Costs", Proceedings of the CIBSE National Conference 1997, Vol. II, Alexandra Palace, 5-7 Oct 1997.

More recently, a study by Stevens also found low levels of satisfaction in a series of buildings that incorporated automated blind systems.[32] Some of these building systems did have over-ride facilities, but although these systems didn't rate highly, they were shown to be more acceptable than systems without occupant over-rides. These findings reinforce the belief that satisfaction with an automated system is often associated with the perception of control over that system.

While in all of these studies the sample size is small and the analyses incomplete, the results are interesting and the findings are supported by a certain amount of anecdotal evidence. The studies highlight the fact that building users are intolerant of automated blinds and that as a result, there is a need to improve automated blind control strategies. The suggestion seems to be that this is achievable through user centred design, for example by including the occupant in the primary control loops of such systems. A few of the studies also identified the importance of changeable organisational issues in the design of control systems and how there is a need to provide flexibility in control system design and maintenance.[26] [30] [31]

1.3.3 Proposed Hypothesis, Methodology and Aims for a Study into Improving Automated Blind Control Systems.

This thesis describes the problem solving research methodology formulated to tackle the long-standing quandary of how automated blinds should be improved to better meet users' needs. The aim was to propose a new multidisciplinary approach to the problem that could lead to its solution by: firstly identifying the primary factors that influence a user's perception of and interaction with the automated system; and secondly finding a means of incorporating and balancing these basic requirements with the driving force behind the development of automated blinds to date, namely energy conservation.

---

[32] Stevens S., 1998, Occupants reactions to automated glass facades, Unpublished poster paper presented at Conf. Intelligent Buildings: Realising the Benefits, 6-8 October 1998 at BRE, Watford, UK.

The study begins by investigating the reasons why people adjust blinds and uses this information to better understand the shortcomings of current control techniques and possible areas for improvement. It highlights that the factors affecting user interaction are complex, but illustrates that incorporating these factors into control algorithms by making those algorithms more complex can result in a system that is unmanageable, difficult to commission correctly and user-unfriendly. The work goes on to suggest that a more demanding definition of "intelligent control" is needed if we are to successfully integrate users' needs with automated control.

Therefore the hypothesis for the thesis is:

Can existing control strategies be developed to incorporate factors that have been identified as being important to an 'individual' user, without the need for a complex management intensive structure?

The methodology used to explore this hypothesis evolved largely from the onus placed on the fact that users are individuals within individual contexts, and not an average of a number of individuals within standard contexts. This new and bold approach to the problem can be justified, in this case, by the huge environmental impact the slightest blind adjustment has on the internal environment and thus occupant satisfaction.

Traditional control system development tends to utilise two basic methods of information-gathering for determining the required control system functionality: experimental data or expert knowledge. This study uses the author's own experiences as an individual to gain enough multidisciplinary expert knowledge and crucially individual user experience in an individual context to propose and develop a solution. Although some data was gathered in the study, its collection was not the primary concern, as the studies duration was thought to be too short. This methodology was considered the only valid approach to tackling the problem as a whole.

Chapter 2: The Factors Affecting the Way Individuals Adjust their Blinds

2.1 Introduction

Many factors influence the way in which occupants adjust blinds (see Figure 2.1) and most contain a high degree of uncertainty for the building designer. This chapter uses the findings of a wide-ranging literature survey to divide these variables into three main categories:

(i) *Comfort* - no two individuals are alike: Each has different preferences depending on past experience and various other physical and psychological factors;

(ii) *Climate* - no two days are the same: In fact air temperatures and solar radiation levels can vary significantly from hour to hour and from minute to minute;

(iii) *Context* - no two locations are the same: Even in the same office the microclimate of one desk position can be dramatically different to another.

The few studies that have been undertaken on the factors affecting the use of blinds, all seem to suggest that on average the majority of user adjustments are triggered by a need to alleviate discomfort caused by glare, direct solar penetration or visual intrusion.[14] [15] [16] [17] [33] A decision on whether conditions are comfortable or uncomfortable for an individual is largely dependent on various climatic and contextual factors. In addition, certain climatic variables, particularly microclimatic variables, are also dependent on context. From this we can see that the interactions within and between these three categories are complex and closely related.

---

[33] Lindsay C.R.T. and Littlefair P.J., 1992, "Occupant use of Venetian Blinds in Offices", PD 233/92, Building Research Establishment, Watford, October 1992.

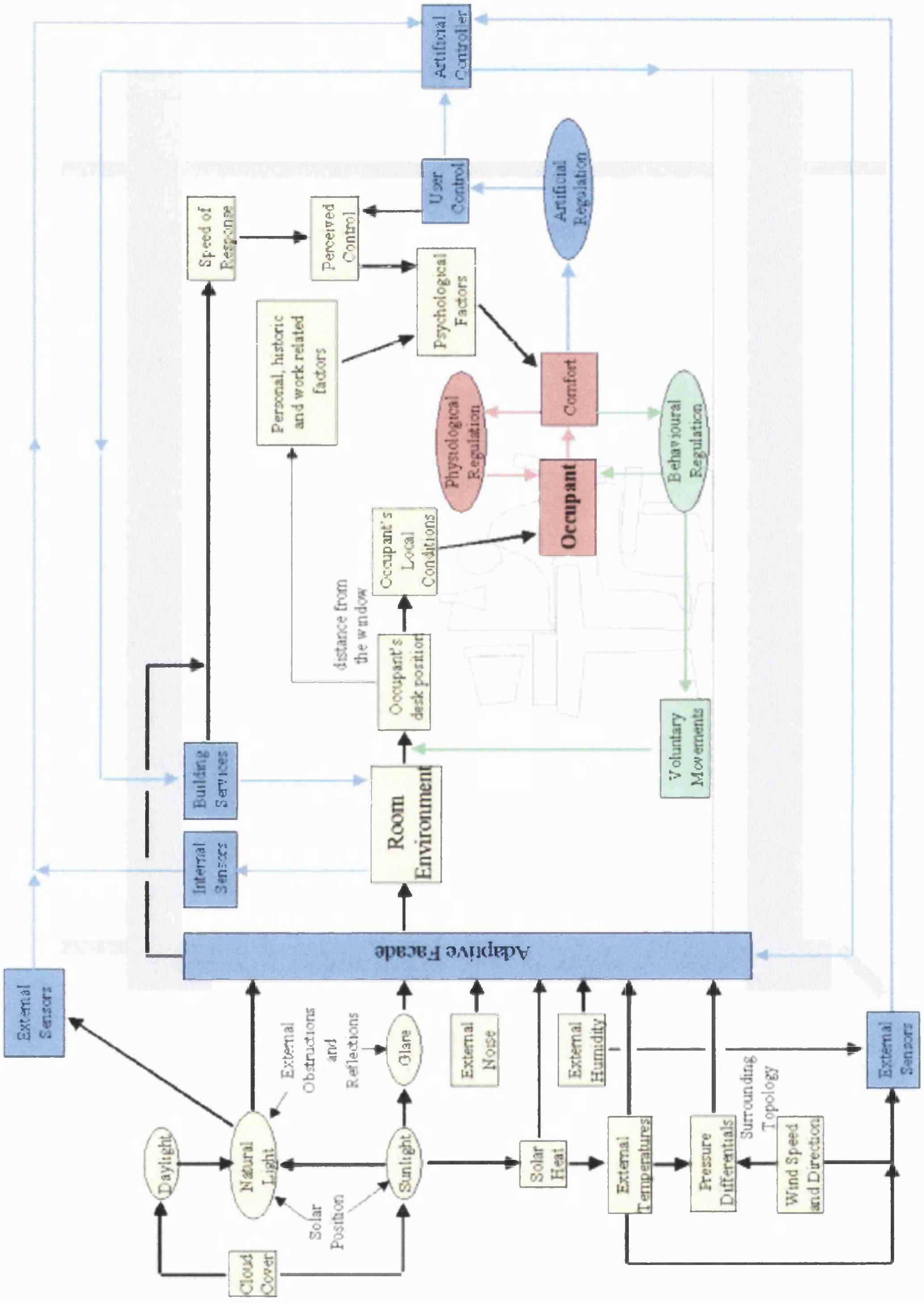


Figure 2.1 Diagram showing the interactions between the variables that influence the control and use of blinds and other adjustable facade devices.

2.2 Comfort

2.2.1 The Use of Blinds to Maintain Comfort

The primary goal of most building systems should be the attainment of 'comfort'. The term 'comfort' has a broad definition. When related to the built environment it can be defined as a state of mind that expresses an indifference to the environment; in other words the absence of discomfort; where discomfort is alleviated by making various adjustments.[34] The mechanisms that people use to alleviate discomfort can be divided into three categories:

(i) *Physiological* - adapting to the environment, e.g. dilation of blood vessels, control of pupil size etc.;
(ii) *Behavioural* - moving to another environment or averting one's line of sight;
(iii) *Artificial* - changing the environment through man-made devices.

A blind is an artificial mechanism that, when adjusted, influences primarily the thermal and visual environment. Figure 2.2[35] and Figure 2.3 illustrate how all three mechanisms of alleviating discomfort influence an individual's thermal and visual comfort respectively.

By using the blind in this way, users are only likely to make the decision to adjust the blind after they have reached a crisis of discomfort, and they are unlikely to adjust the blinds in advance of feeling discomfort. Indeed, blind use studies seem to suggest that occupants utilise their blinds to avoid worsening conditions of discomfort and very rarely utilise them to optimise their environments. [14] [15] [17] [33]

---

[34] Humphreys M.A., 1997, "An adaptive approach to thermal comfort criteria", in Naturally Ventilated Buildings for the Senses, the Economy and Society, Ed. D. Clements-Croome, E&FN Spon, pp129-137.

[35] The human body element of the thermal comfort diagram is adapted from Figure 1.4 in Clark R.P. and Edholm O.G., 1985, "Man and his Thermal Environment", Edward Arnold Publishers, London.

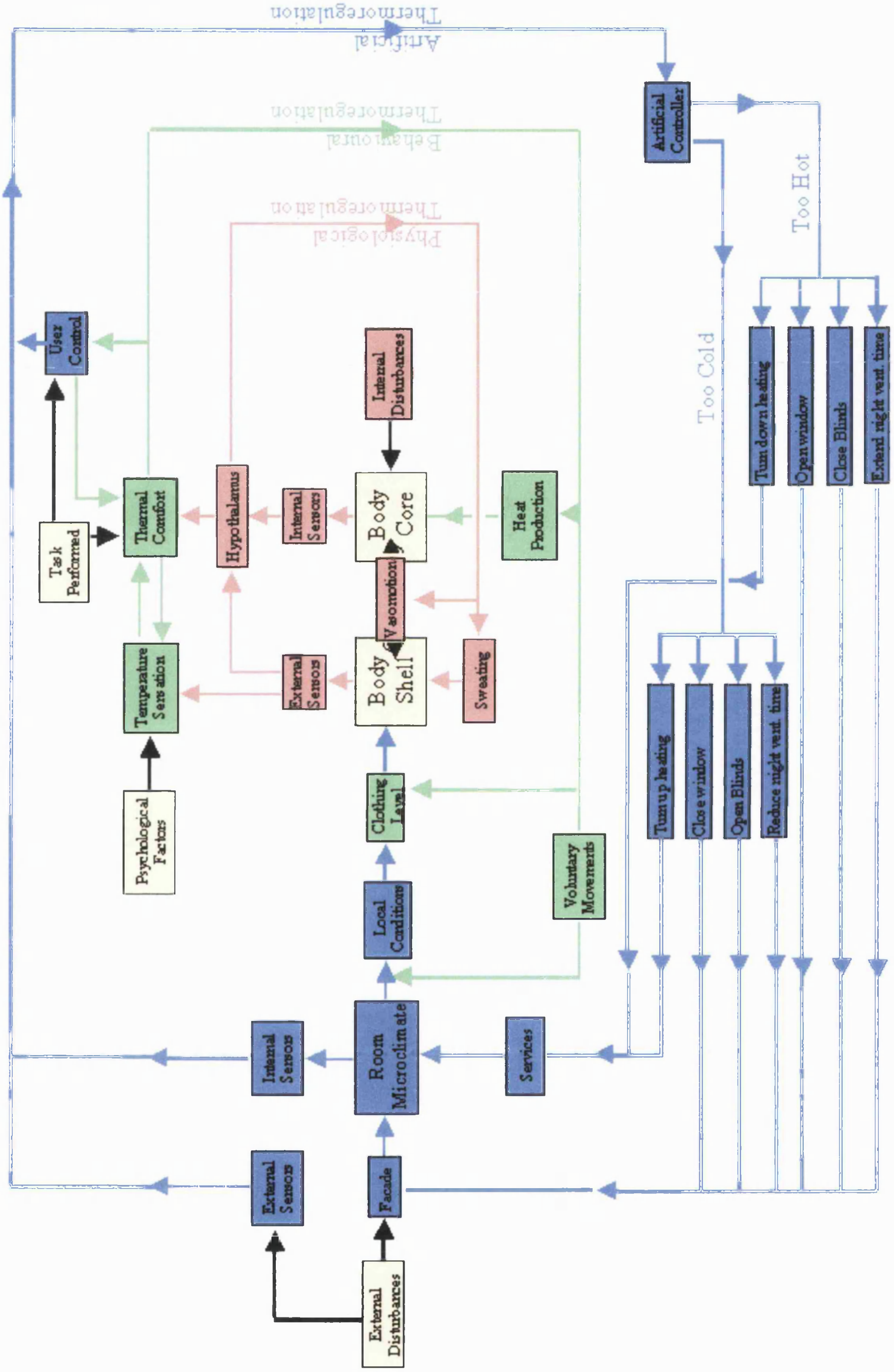


Figure 2.2: Diagram showing the physiological, behavioural and artificial mechanisms of adjustment used in thermo-regulatory responses.

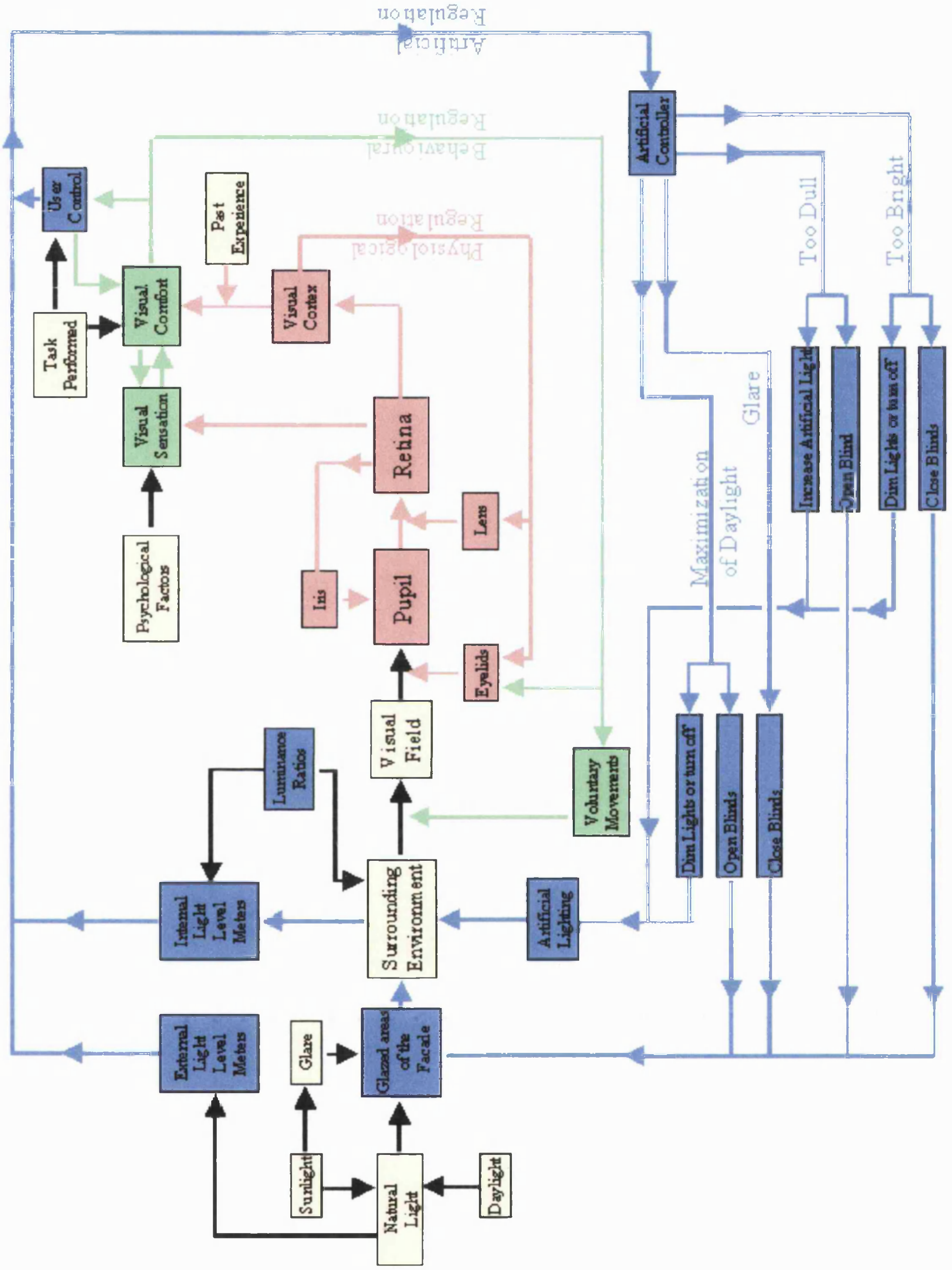


Figure 2.3: Diagram showing the physiological, behavioural and artificial mechanisms of adjustment used in photo-regulatory responses.

2.2.2 Environmental and Physical Factors

It has been shown that an individual adjusts a blind in order to alleviate discomfort caused by glare and direct solar heat gain originating from the glazed elements of the facade. However, identifying the exact environmental and physical conditions that cause discomfort and lead to a particular response is a difficult task. Users seem to adjust their blinds in response to random external events and other factors, such as the brightness of the sky and the contrast in the room, which are largely influenced by the climatic and contextual variables discussed later.

One way to quantify comfort is to use environmental comfort theory to make an assessment of the influence of certain physical factors.[36] [37] However, although these formula represent good engineering models for the human physiology, they are somewhat limited when used to simulate an individual's perception of comfort in a real world situation, especially when considering the visual environment (see Appendix II).[38] [39] [40] The visual perception of a space is dependent on sunlight patterns, sky views, the way surfaces of that space are lit, (and hence the contextual elements such as the proportions of the room and the relationship of the window to the space) and their relative brightness.[41] [42] In addition, the settings and environments in which office workers feel most comfortable are as varied as their individual physiologies and psychologies.

---

[36] Simmonds P., 1993, "Thermal comfort and optimal energy use", ASHRAE Transactions, Vol.99, pp10371048.

[37] Yang K.H. and Su C.H., 1997, "An approach to building energy savings using the PMV index", *Building and Environment*, Vol.32, No.1, pp25-30.

[38] Davies A.D.M. and Davies M.G., 1995, "The adaptive model of thermal comfort", *Building Services Research and Technology*, Vol.16, No.1, pp51-53.

[39] Proshansky H.M., 1972, "Methodology in environmental psychology; problems and issues", *Human Factors*, 14, pp451-460.

[40] Russel J.A. and Ward L.M., 1982, "Environmental psychology", *Ann. Rev. Psychol.*, 33, pp651-688.

[41] Clemo A. and Thompson M., 1996, "The evaluation of daylit interiors", CIBSE/ASHRAE Joint National Conference 1996, Vol.II, Harrogate, 29 Sept - 1 Oct, pp397-402.

[42] Loe D.L., Mansfield K.P. and Rowlands E., 1994, "Appearance of lit environment and its relevance in lighting design", Lighting Research and Technology, Vol.26, No.3, pp119-133.

2.2.3 Psychological and Social Factors

Blind use studies have shown that the way in which people control their blinds varies greatly from person to person.[33] Some people prefer to sit in direct sunlight and some people do not. Such variability can be attributed to the fact that many psychological and social factors influence whether an individual is comfortable or not. These include organisational and managerial factors, such as stress, as well as personal and social factors, such as perception of control. In addition, personal preferences are often influenced by past experience and a number of adaptive mechanisms. Each person, organisation, task and climate represents a change in psychological variables, thus making an individual's perception of comfort very difficult to predict.

A questionnaire carried out by Inoue et al, as part of their studies into blind use, asked occupants to identify the reasons why they opened their blinds.[17] Along with the more expected answers such as:

"because the room got darker"

were more sublime answers such as:

"So as to recreate oneself" and *"So as to get outside atmosphere"*

An occupancy survey by Williams et al highlighted some of the social problems associated with the control of blinds in open plan offices by revealing that a large proportion of occupants in the buildings studied did not adjust blinds at the request of their colleagues.[43]

---

[43] Williams R.N., Boothby W.B. and Kirby L., 1997, "Modern Ventilation Techniques - The indoor environment and occupant perception", Proceeding of the CIBSE National Conference, Vol. I, Alexandra Palace, 5-7 Oct 1997.

Another important factor in psychological comfort is a view out of a glazed part of the facade.[44] Rubin et al considered the important affect of 'view out' on the way building occupants controlled blinds, highlighting the fact that a view from a window is influenced by contextual issues such as the occupants desk position within a room and their proximity to the window.[14] The study also identified the equal importance of 'view in' or visual intrusion, which is affected by the proximity of other buildings and the window's relationship to public spaces and pathways.

Lindsay and Littlefair expanded on this theme by acknowledging the effect of sill height on views out and thus the operation of blinds.[33] Their study also recognised that users on the lower floors of one of the buildings studied seemed to operate their blinds less than those on the upper floors.

Daylight utilisation within buildings and views out of buildings are provided for the benefit of occupants. Therefore a daylight control device, such as a venetian blind, should respond to their visual and perceptual needs. As these needs are so variable and difficult to anticipate, we must allow the occupant the luxury of being able to make adjustments and this point has led to a broad agreement among researchers that providing individual control of the local environment can enhance individual comfort. [26 45 46 47 48 49]

A possible explanation for this fact can be found by referring back to Figure 2.2 and Figure 2.3. Both Figures highlight the importance of individual user control for linking the automatic control loop to the occupant's own thermo-regulatory and

---

[44] Markus T.A., 1967, "The significance of sunshine and view for office workers", in Sunlight in Buildings, ed. Hopkinson R.G., Boeurcentrum International, Rotterdam.

[45] Haigh D., 1981, "User response in environment control", in The Architecture of Energy (Eds. D Hawkes and J Owers), Construction Press, London.

[46] Preller L. Et al., 1990, Indoor Air '90, Fifth International Conference on Indoor Air Quality and Climate, pp227-230.

[47] Bordass W.T., Bromley A.K.R. and Leaman A.J., 1995, "Comfort, control and energy efficiency in offices", BRE Information Paper, IP3/95.

[48] Slater A., 1995, "Occupant use of lighting controls in offices", *CIBSE Journal*, Vol.17, No.8, pp43.

[49] Bauman F., Arens E., 1996, "Task/ambient conditioning systems: Engineering and application guidelines", Center for Environmental Design Research, University of California, Berkeley, CA.

photo-regulatory responses. In the office environment, the majority of behavioural responses are limited by dress codes, desk positioning or task, therefore when discomfort arises the only option is often the user over-ride. If this link were to be broken, the occupant would be reliant on the designer supplying conditions within the comfort band provided by their own physiological responses, which varies from person to person.[50] [51] As a result, the perception of control is also an important factor in influencing comfort and the way in which people control blinds.

2.3 Climate

2.3.1 Fluctuations in Weather

The term climate is used here not only to characterise the prevailing weather conditions of a particular site, but also to describe variations in weather conditions and its effect on the microclimates created from the influence of certain contextual elements. Indeed when considering the luminous environment and its effect on blind control, we can see that the individual occupant is very sensitive to minute by minute fluctuations in weather and it is the impact of these short-term variations that account for a large proportion of blind adjustments. Broad seasonal and diurnal differences are important when considering temperatures, but these differences can largely be accommodated by adjustments to clothing and mechanical systems.[52] [53] [54]

---

[50] Humphreys M.A., 1993, "Field studies and climate chamber experiments in thermal comfort research", in Thermal Comfort: Past, Present and Future, BRE 264, Ed. N.A. Oseland and M.A. Humphreys, BRE, Garston 9-10 June 1993, pp52-72.

[51] Heijs W., 1993, "The dependent caviable in thermal comfort research some psychological considerations", in Thermal Comfort: Past, Present and Future, BRE 264, Ed. N.A. Oseland and M.A. Humphreys, BRE, Garston 9-10 June 1993, pp40-51.

[52] Newsham_G.R., 1997, "Clothing as a thermal comfort moderator and the effect on energy consumption", *Energy and Buildings*, Vol.26, pp283-291.

[53] Parsons K.C. et al, 1997, "A climatic chamber study into the validity of Fanger's PMV/PPD thermal comfort index for subjects wearing different levels of clothing insulation", CIBSE National Conference 1997, Alexandra Palace, 5-7 Oct, Vol.I, pp193-205.

[54] De Dear R.J., 1993, "Outdoor climatic influences on indoor thermal comfort requirements", in Thermal Comfort: Past, Present and Future, BRE 264, Ed. N.A. Oseland and M.A. Humphreys, BRE, Garston 9-10 June 1993, pp52-72.

2.3.2 Sky Conditions

For the design of a blind control strategy the main diversity in the external climate is the variability of solar radiation due to changes in cloud cover. Although such environmental stimulus is often welcomed and valued by the users, it is also potentially inconvenient and it is then that they will utilise their option of control. Therefore, an automated blind control system should be able to adapt to constantly changing daylight conditions to create desirable internal visual environment.

In the UK, daylight designers are encouraged to use the C.I.E. Overcast Sky Model as their standard reference sky. However, the variability of the actual outdoor daylight conditions due to cloud cover, time of day, season and turbidity is enormous. Therefore in reality this standard condition is very rarely encountered, as cloud cover is rarely thick enough to give such uniformity in the luminance of the sky.[55] In fact the daylight source varies in two ways. Under diffuse conditions it extends across a large solid angle and has a variable luminance distribution depending on the type of cloud cover.[56] Under direct sun conditions, the sun's rays shine through all or some of the window at a highly variable incident angle depending on the solar position. On such occasions the average luminance of a sunlit window can be as high as 20,000 cd/m^2, enough to create glare and far above the level needed for sight.

All of the blind use studies available to date support the hypothesis that window blinds are used to block the climatic element of solar radiation in the form of heat gain and glare. However, a few of the studies disagree on the affect of climatic variation. This fact is largely due to the nature of each of the buildings studied and the methodologies utilised in those studies.

Initial investigations by Rubin et al found that occupants had preferred window blind positions that were relatively independent of seasonal and climatic variations.[14] Rea

---

[55] Rutten A.J.F., 1994, "Sky Luminance Research Imperative for Adequate Control of Temporary Supplementary artificial Lighting Installations", *Building and Environment*, Vol.29, No.1, pp10-111.

[56] Harrison A.W., 1991, "Directional sky luminance versus cloud cover and solar position", *Solar Energy*, Vol.46, No.1, pp13-19.

challenged this view, after his study revealed that blind use was linked to solar radiation and the degree of cloud cover.[16]

The more in depth study by Inoue et al in Japan revealed that a non linear relationship existed between the amount of solar radiation that enters through a window and the rate of blind use.[17] The study also showed that the solar irradiance threshold value that stimulated blind use fell in a wide band and could on occasions be quite low.

2.3.3 Internal Microclimate

The relationship between the direct sunlight and the distribution of interior illuminance within a space is an important factor in affecting the way people perceive their visual environment.

The relative contributions of direct sunlight and diffuse daylight in interior lighting depend on the latitude and longitude of the site and on the local climate and topology with respect to the distributed sky. The total amount of light admitted into the space is the product of the incident illumination, aperture size, glazing properties, blind position, room surface materials and other light loss factors.

With vertical glazing elements, diffuse light from the sky falls off rapidly as one moves away from the window, usually reaching a level of less than a 1% Daylight Factor at roughly 3-5m from the window. Depending on the factors mentioned above, the illuminance in the area adjacent to the window may exceed 20% of exterior values or twenty times higher than values at 5m from the window. When direct sun penetrates the space, the gradients of interior illuminance across the space can be even higher. These light patterns may create objectionable effects in terms of balance and contrast. However, the exact nature of this contrast will be largely dependent on the design of the window wall and the nature of the artificial lighting and surface materials within the space.

2.3.4 Air Temperatures

In addition to lighting gradients, temperature gradients near the window will be dependent on the design of the window wall. High temperature gradients can affect the way people positioned near those windows control their blinds.

Although there is little direct evidence to link blind use to rising internal and external air temperatures, Lindsay and Littlefair's study showed that in a building that is prone to overheating, occupants may be cautious of letting the sun into their room.[33] Such a scenario can result in the blinds being kept closed for the majority of the hot summer months.

2.4 Context

2.4.1 Window Orientation

All of the blind use studies mentioned so far support the hypothesis that blind use is related to window orientation by showing that peak occlusions (the fraction of glazed area covered by blinds) were witnessed when the sun was on the facade.[14 16 17 33]

In addition, the studies by Lindsay and Littlefair, and Inoue et al both identified the link between blind use, the amount of sunshine and the position of the sun in relation to the facade.[33 17] Inoue et al showed that the amount the incident solar radiation penetrated into the space affected the way people controlled their blinds. Lindsay and Littlefair demonstrated that the angle of incidence of the sun was a major factor in prompting blind use.

2.4.2 Location of Surrounding Buildings and Vegetation

The geometrical relationship between the occupant's location within a room, the window and the surrounding obstructions can influence a user's interaction with their blind system.[33] The shading effect of trees, which can vary seasonally, and other larger obstructions, such as adjacent buildings, can provide relief from direct solar penetrations during certain times of the year, and as a result lead to occupants opening their blinds more. Whereas solar glare due to reflections from adjacent buildings or the ground can cause considerable discomfort, affect daylight penetration into the room and lead to occupants closing the blinds. [57]

2.4.3 The Use of the Internal Space

Glare can be experienced in two forms. Firstly, when the sun or a bright sky encroaches within the field of view (discomfort glare) and secondly when a reflection of the sun or a bright sky intrudes upon the area of the visual task reducing the contrast of the task itself (disability glare). Identifying the external sky conditions that result in these forms of glare for an individual controlling a blind is highly dependent on the way that individual has his task area positioned and viewing directions orientated.

In addition, an individual's reaction to glare, such as veiling reflections, is task dependent and subject to personal preference. On occasions occupants may prefer to accept the contrast across a task in order to appreciate the high brightness produced by the sun.[44] This is more likely to occur with paper based or desk based work. On other occasions, such as when reading VDU screens, occupants may decide to block the sun's intrusion on to the task area.[58]

The importance of occupant desk position was highlighted in a study by Vine et al where occupants were asked to position their blinds at what they considered to be the optimum viewing angle.[59] The preferred slat angles varied greatly between occupants due to differences in the geometrical relationship between each occupant and their blinds and how much of the sky could be seen.

---

[57] Tregenza P.R., 1995, "Mean daylight illuminances in rooms facing sunlit streets", *Building and Environment*, Vol.30, No.1, pp83-89.

[58] Anon, 1995, "Daylighting design for display screen equipment", Building Research Establishment, IP10/95, Jan 1 1995.

[59] Vine E. et al, 1998, "Office worker response to an automated venetian blind and electric lighting system a pilot study", *Energy and Buildings*, Vol.28, pp205-218.

When a building is designed, the tasks to be undertaken in a space are often unknown. Also within any group of people there are large interpersonal differences in clothing, activities and preferences. The nature of tasks and work groups, the partitioning of workspace and the occupants themselves all may change several times within the lifetime of the building services and the facade. A control system should have the flexibility to deal with these variables.

2.4.4 The Overall Building Design

The overall design of a building can also influence the way people control blinds. The differences between an individual's visual perceptions of spaces within narrow plan buildings and deep plan buildings can be significant, especially when considering open plan offices. The type of glass used in the facade can also be a factor. Whether a designer specifies spectrally reflective glass, high absorbing glass or clear glass may result in variations in blind use.

The poor design of other elements of the building can also have a great effect. We have already mentioned the example used by Lindsay and Littlefair for a building that is prone to overheating when we discussed the effect of air temperatures.[33] Other examples could include erroneous user responses resulting from ineffective air conditioning, heating or lighting systems. Indeed, installed systems can vary substantially from a design ideal and differences between control algorithms, product design (sensors and electronics), sensor or actuator placement, installation and calibration techniques, and operational problems can all affect the occupant's perception of their environment.

Poor design of the blind itself can influence the way people interact with it. The use of manual controls that are intrinsically difficult to operate[45] or the positioning of the blind so that it is conflicting with other elements in the building, for example cutting off ventilation, can both be telling factors.[60]

---

[60] Bordass W., Cohen R., Standeven M. and Leaman A., 1999, "Assessment of Building Performance in Use: 2, Technical Review of the PROBE Buildings", Third Draft, www.usablebuildings.co.uk, 15 Jun 1999.

2.5 Chapter Summary

The findings of this chapter are key for the reader to understand the methodological approach taken for the rest of the study. So far the study has identified the fact that comfort on an individual user and contextual scale is complex and difficult to deal with and that data is not available to analyse it successfully.

Venetian blinds are provided in buildings to offer the occupants protection against the negative factors associated with the transparent glazed elements of the building skin, such as overheating, glare and intrusion. This chapter highlights this by stating that blinds are a significant artificial means of alleviating discomfort and uses Figures 2.1-2.3 to illustrate how this mechanism fits within an individual user's feedback loops.

However, along with the detailed study reviewed in Appendix II, it has also been demonstrated that there is more to comfort than physiological factors and that transparent elements of the building skin can also offer a number of positive benefits for people in buildings. These include views out, visual contact with the outside world, sunshine and warmth, daylight and a sense of spaciousness. Therefore, aside from energy saving qualities, the alteration of window blinds can dramatically influence the mood and character of an interior, and as a result, the way each individual perceives the space.

After considering all of these facts, it is safe to say that few environmental control technologies have such an immediate impact on the quality of the internal environment and the physical and psychological wellbeing of its occupants. As a result, it is vital that we identify the factors that influence an individual's control actions and use them to understand their decision-making processes and their expectations of an automated system.

To start off the process, this chapter has used the limited data currently available to propose three categories of factors that influence the way individuals interact with blinds, and has shown that the relationships between them are complex and contain a high degree of uncertainty. A successful solution to the problem of automating blinds should attempt to resolve these complexities and variabilities, which are inherently:

(i) *Dynamic* - many parameters change over time and at different rates;
(ii) *Non-linear* - some parameters exhibit different types of behaviour in different regions due to a choice of internal and external conditions;
(iii) *Stochastic* - some parameters are subject to large unpredictable/chaotic environmental disturbances;
(iv) *Multi-dimensional* - many different mechanisms interact in a complex manner;
(v) *Unmeasurable* - some variables are difficult to measure, have unknown relationships, or are difficult or expensive to evaluate in real time, i.e. occupant satisfaction, psychological factors and future cloud cover.

By taking into account a large majority of these variables, the process becomes less amenable to direct mathematical modelling based on physical laws. Indeed, the blind use studies reviewed in this chapter all identify the fact that there is a large variation in the way people alter blinds. Each individual arrives at a preferred set of responses to a series of environmental variations as a result of their individual weighting of the positive factors (e.g. views out and light in) and the negative factors (e.g. glare in, view in) for their particular context. Consequently, researchers now believe that individual comfort and satisfaction can only be attained universally by providing individual control of the local environment.[47] [49] [61] But is this provision enough?

The challenge seems to lie in integrating the individual users' priorities with energy efficient control, to produce a smooth, transparent, comfortable, energy efficient control strategy. The next step was to try to understand the factors considered within current blind control techniques, which are reviewed in the next chapter.

---

[61] Kroner W.M. and Stark-Martin J.A., 1994, "Environmentally responsive workstations and office-worker productivity", *ASHRAE Transactions*, Vol.100, Part 2, pp750-755.

Chapter 3: Automated Blind Control Techniques

3.1 Introduction

By reviewing manufacturers' product literature, research papers and automated facade building case studies [see Appendix III], it was shown that the current state of the art for blind control algorithms could be divided into three categories, representing three levels of complexity:

(i) *Threshold controllers*
(ii) *Sun blocking controllers*
(iii) *Mode and scene controllers*

The systems examined ranged from commercially available products to installations that had been implemented either on an individual building scale or through research. Controllers tended to utilise simple stand-alone open-loop control algorithms that operated to reduce instantaneous building loads. However, a few manufacturers and researchers had developed more advanced systems that could be integrated with the building's lighting system.

Each type of strategy has a different effect on the overall performance of the building, in terms of energy consumption and user comfort. All are marketed on the basis of providing energy efficient solutions and most provide a degree of individual control to enhance worker comfort and productivity. This Chapter reviews the nature of automated blind control by examining each of the levels of complexity out-lined above, and attempts to explain why automated blind control systems are not well received by occupants.

3.2 Threshold Controllers

3.2.1 A Brief Description of Operation

The threshold controller, the simplest of the three types of controller, is common amongst many commercial systems available today. Its operation relies on the utilisation of a solar illuminance or irradiance sensor readings, to lower or raise the blinds after a pre-set threshold value is passed [see Figure 3.1].

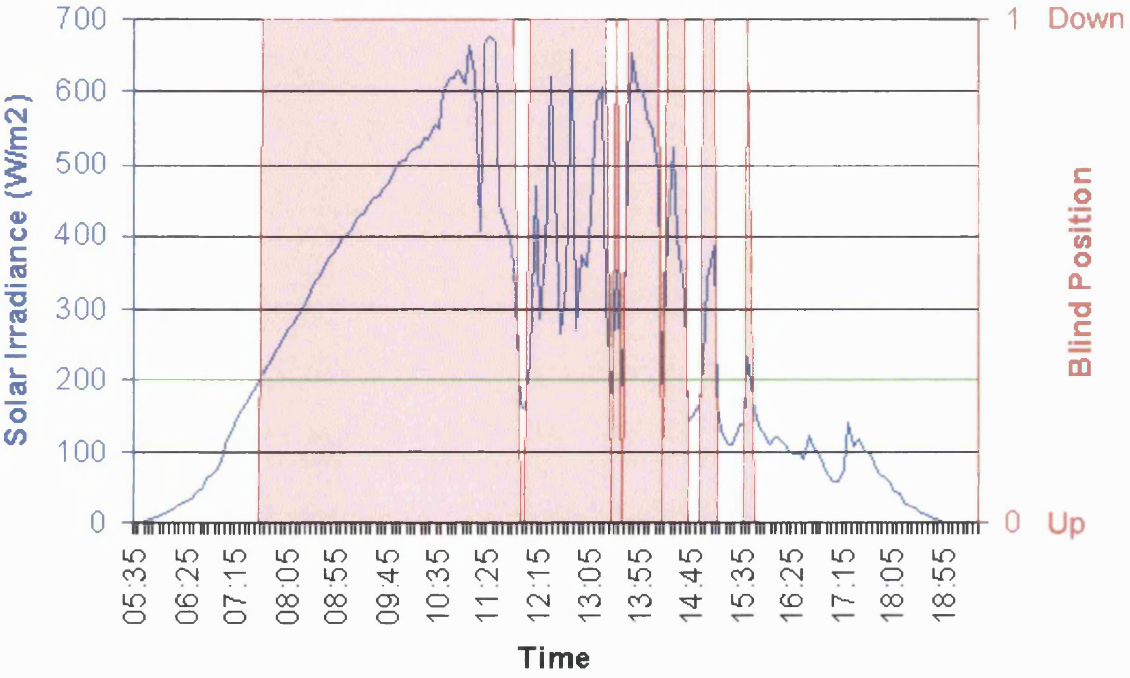


Figure 3.1: Graph showing how the vertical blind position varies with a threshold controller, operating with a threshold of 200 W/m^2 and without any time delays

Once lowered the blinds are tilted to a default position, usually 45 degrees from horizontal (45° blind angle), which is thought to optimise the solar protection provided by the blind throughout the year whilst still providing a certain degree of view to the outside.[62] [63] [64] [65] Default threshold levels tend to be set by the

---

[62] Paulme C., 1999, "Functional Profile: BIU-2DC Lon2", Somfy AEG Technical Literature, Cluses, France.

[63] Anon, 1999, "Huppe external blinds: Environmental control, exterior views, adjustable slats and modern design", Huppe Form technical catalogue, Oldenburg, Germany.

[64] Aleo F. and Sciuto S., 1993, "Smart System for the combined control of natural and artificial lighting", Proceedings of the International Symposium on Energy Efficient Buildings, Leinfelden-Echterdingen, Germany, March 9-11, 1993.

manufacturers to around 24,000 lux or 150 W/m^2, but these can be altered at any stage if they are determined to be ineffective.[62]

3.2.2 Time Delay

As threshold control is essentially an open loop control methodology, it is usually implemented with a series of time delays that help prevent the blind from reacting incessantly to variations in daylight level around the threshold, due to cloud motion. A short time delay of about 30 seconds is used when the sensor reading rises above the threshold level (sun coming out from behind a cloud). This is to account for any short burst of sunshine whilst still providing a quick enough response to the threat of glare. A longer time delay, of about 10 minutes, is used for when the sensor reading falls below the threshold level (sun going behind a cloud). This longer delay provides the system with a certain amount of stability on days with mixed sky conditions. Both of these time delays can also be adjusted if their default settings are found to be ineffective.

3.2.3 Sensors

The type, position and number of sun sensors used as inputs to the control algorithm can vary from system to system. Some low cost systems often rely on just a single horizontal illuminance or irradiance reading to control all the blinds in the building.[17] Other systems provide some orientational information to the controller by using a specified number of simple vertical sun sensors positioned on each facade. In some cases, global and diffuse sensors on the roof are used in conjuncture with a few simple sensors on the facade to provide a better representation of the direct and sky component of the natural light.

The studies of Inoue et al recognised the threshold controller as the primary type of controller in Japan at the time. However, their investigations into the way in which people control blinds identified the importance of the geometric relationship between

---

[65] Barnard N., 1997, " Case Study 6: Blind Control - Simulation Analysis", DETR PIT Report on Controls Design by Simulation, Oscar Faber Applied Research, St Albans, Dec 1997.

the occupant's position in the room, the window and the solar position. In order to take this geometric factor into account, the researchers suggested a modification to the threshold technique that utilised a localised sensor for each window that could be moved to a location suitable for determining the solar cut off angle required by the occupant. Understandably, this solution was found to add a large amount of cost to the system, as it required a large number of sensors and was reasonably impractical in terms of commissioning and maintenance. As a result, the authors proposed the use of a simple solar blocking algorithm based on a timer approach, reviewed in section 3.3.3, which offered the blind controller improved functionality and the building additional energy savings.

3.2.4 The Energy Savings from Dynamic Operation

Over the last ten years, a series of extensive studies on the application of daylight control in buildings have been undertaken by the Lawrence Berkley Laboratories [LBL] based at the University of California. Their research, which has mainly focused on the integration of venetian blinds and lighting systems, was taken largely from an environmental and technological viewpoint.

As part of their work, the researchers demonstrated the energy savings that could be made by improving the operation of simple threshold controllers. The studies showed that peak cooling load reductions of 6-15% were attained by dynamically altering the slat angle compared to a 45° static blind, and 18-32% compared to a static horizontal blind.[66] Their investigations also demonstrated that daily lighting energy reductions of 19 to 52% (45° blind angle) and -14 to 11% (0° blind angle) were achieved throughout the year. These findings highlight the benefits of adjusting the blind slat angle as the external conditions vary. The next two levels of complexity both utilise this extra functionality.

---

[66] Lee E.S, DiBartolomeo D.L. and Selkowitz S.E., 1998, "Thermal and daylighting performance of an automated venetian blind and lighting system in a full scale private office", *Energy and Buildings*, Vol.29, pp47-63.

3.3 Sun Blocking Controllers

3.3.1 A Brief Description of the System

Sun blocking controllers operate in a similar fashion to the threshold controllers but in addition adjust their slat angle in relation to the position of the sun. They achieve this through the use of either solar positioning algorithms or sensor techniques.

3.3.2 Solar Positioning Algorithms

The position of the sun in the sky is quantified by the use of two angles, solar altitude and azimuth. Both of these angles can be calculated, for any time of year, by knowing the latitude and longitude of a site. From these angles we can determine the angle the sun makes on the vertical plane perpendicular to the window plane (wall solar altitude angle) [see Figure 3.2]. A solar positioning algorithm blind controller uses this angle to adjust the blind slat angle to block solar penetration and maximise view.

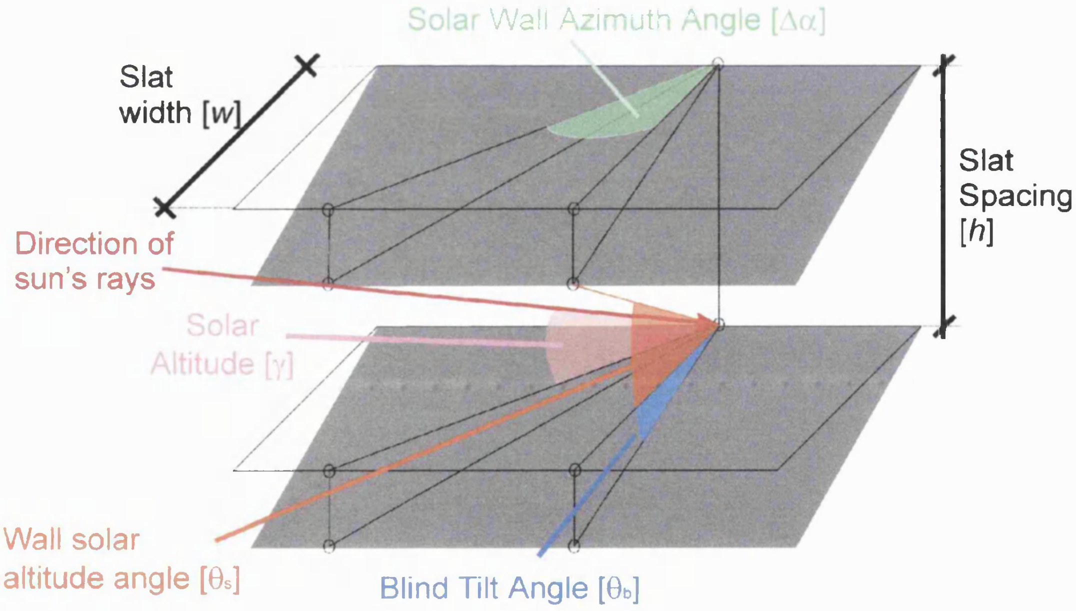


Figure 3.2: Diagram showing the angles used to calculate the vertical solar incident angle perpendicular to the window/blind plane

A variety of methods are available for carrying out the calculation of solar altitude and azimuth, and some of these are reviewed in Appendix IV where an assessment is made on their accuracy and ability to be applied to micro-controllers.

3.3.3 The Pre-Determined Solar Positioning Timer Approach

The simplest type of sun blocking controller found during these studies resulted from the work of Inoue et al in 1988.[17] The researchers calculated the position of the sun in relation to the facade of their treated building for various times of day at various times of year. This knowledge was then transferred to the blind controller so that it could adjust the blind tilt depending on a real time clock reading. However, this type of algorithm required a large amount of site specific design and programming work to be carried out at the design stage, and thus was not regarded as a viable option from a practical and commercial standpoint.

3.3.4 The Real Time Solar Positioning Algorithm Approach

As building control technologies have developed, the ability of the controller to be able to carry out complex variable dependent calculations on line, such as solar positioning algorithms, has been enhanced. Such blind controllers use these algorithms to adjust the blind tilt to the required blocking angle and reassess their decision after a pre-defined time period, usually 10 minutes.[4] [63] [64] [67]

However, these on-line control algorithms also require a certain amount of on-site commissioning, so that inputs such as latitude, orientation and time zone can be entered for each installation. This process was considered to be impractical by some researchers, who decided to investigate different methods of adjusting the slat angle in response to a change in solar position.[68] [69]

---

[67] Galanta A., Batturi E. and Viadana R., 1996, "A smart control strategy for shading devices to improve the thermal and visual comfort", Proceedings of the 4th European Conference on Architecture, Berlin, Germany, 26-29 March 1996, pp358-361.

[68] Luecke G.R. and Slaughter J., 1995, "Design, development and testing of an automated window shade controller", Transactions of the ASME, Journal of Solar Energy Engineering, Vol.117, No.4, pp326-332, Nov 1995.

[69] Lee E.S. et al, 1998, "Integrated Performance of an Automated Venetian Blind/Electric Lighting System in a Full-Scale Private Office", Proceedings of the ASHRAE/DOE/BTECC Conference,

3.3.5 The Closed Loop Sensor Control Approach

Luecke and Slaughter developed a closed loop blind controller that utilised a sensor just behind the blind to optimise the blind tilt position.[68] This sun blocking approach was able to provide two modes of operation, a winter strategy and a summer strategy. The winter strategy modulated the blind tilt angle to maximise the reading on the internal photo-sensor thus maximising the amount of solar heat entering the space. The summer strategy modulated the blind tilt angle to minimise the reading on the internal photo-sensor, thus minimising the solar heat entering the space. The controller also had a default horizontal tilt position for cloudy conditions. An example of a similar style of controller being utilised in practice can be found by referring to the case study on the Occidental Chemical Building in Appendix III.

LBL used computer simulation modelling to compare the annual energy savings achieved by a closed loop sensor system that could cater for changing seasonal needs, to those achieved by a system that just blocked sunlight. The results showed that adding complexity to the system in this way did not result in additional energy benefits.[8] These findings were attributed to the fact that the complex system investigated only considered a single performance criteria (i.e. the reading on the sensor), and therefore concluded that to achieve adequate energy savings, complex controllers should utilise multiple performance criteria, such as those to be reviewed in Section 3.4.

Rheault and Bilgen also demonstrated the use of closed loop control algorithms to control blinds.[70] In their case the blind was used to obtain an optimum floating temperature from internal and external temperature sensor readings, and was considered more for use within passive solar collectors.

Thermal Performance of the Exterior Envelopes of Buildings VII, Clearwater Beach, Florida, December 7-11, 1998.

[70] Rheault S. and Bilgen E., 1987, "Heat transfer optimisation of an automated blind window system", ASES 12th national Passive Solar Conference, Portland OR. pp122-128.

3.3.6 The Open Loop Sensor Control Approach

Another way to adjust the blinds tilt angle to take into account the sun's position in the sky is by using a sensor to measure it. The greatest proponents for this approach were LBL. As part of their investigations, researchers at LBL developed a sun angle sensor that determined the solar position relative to the window plane, and then controlled the angle of the venetian blinds to solely block the sun.[69]

The sensor was a twin diode array that yielded solar altitude angles for the window-facing hemisphere, rather than the more accurate solar wall altitude angle. This meant that the blind position always over-compensated for solar penetration, but the LBL team felt that this was acceptable as they considered the incorporation of the solar wall altitude angle on a global scale would result in high costs.[71]

Although initial tests showed that the sensor performed well in terms of positioning the blind to block the sun, later tests revealed that the sensor measured less accurately for solar altitudes greater than 45 degrees (to within 10-17%). Therefore it was decided the sensor required further work to be made commercially viable.[71]

A different form of open loop sensor control can also be found in some complex lighting systems that utilise sky scanners with a number of illuminance sensors at different orientations to measure the luminous distribution of the sky. These sensors are sometimes used to calculate the position of the sun by interpolating data from all the sensors about the relative brightness of different parts of the sky. This method provides a reasonable level of accuracy for occasions when glare might occur. [72]

---

[71] DiBartolomeo D.L. et al, 1996, "Developing a dynamic envelope/lighting control system with field measurements", Journal of Illuminating Engineering Society, Vol. 26, No. 1, pp146-164.

[72] Anon, 1999, "Daylight Dynamics", Luxmate Technical Literature, Luxmate, London

3.4 Mode and Scene Controllers

3.4.1 A Brief Description of Operation

The more sophisticated mode and scene controllers are not as common as the other two controller types. They can use data from a variety of sensors, such as internal light sensors, temperature sensors, occupancy sensors etc., to decide between a series of modes, each utilising different blind control algorithms. [65] [73]

3.4.2 Modes

Operational modes can be divided into three types:

- *Light Transmitting* - positions the blind to ensure that an adequate amount of daylight is transmitted into the space without causing glare;
- *Light Directing* - positions the blind to maximise the daylight penetration into the space by using the blind to reflect daylight towards the ceiling;
- *Sun Blocking* - as with the sun blocking controller category, positions the blind to ensure that the sun's rays are not allowed to penetrate through the blind.

Different modes are utilised in different seasons to improve the annual performance of the system.

3.4.3 Choosing Between Modes

The decision about which mode to take at any one time is made on the basis of how each is predicted to perform in terms of:

- reducing the heating and cooling loads;
- maximising daylight contribution;
- maximising views;

---

[73] Kim J.J. and Jones J., 1993, "A conceptual framework for dynamic control of daylighting and electric lighting systems", Conference Record of the 1993 IEEE Industry Applications Society, Ontario, Canada, 2-8 Oct 1993, pp2358-2364.

- minimising glare.

The mode is selected on the basis of a trade off between all of these performance criteria, which each have associated weightings. The mode that is predicted to perform best in the current situation is selected. This assessment is often made on an hourly basis.

3.4.4 Maximising Daylight and Sun Blocking

For any sun position in the window-facing hemisphere, two critical blind tilt angles are available that just cut off the sun's rays [see Figure 3.3]. Therefore any of the tilt angles outside the range of the two cut off angles also result in solar blocking. This gives the control system a certain amount of flexibility when choosing between blind tilt angles that block the sun and blind tilt angles that maximise daylight.

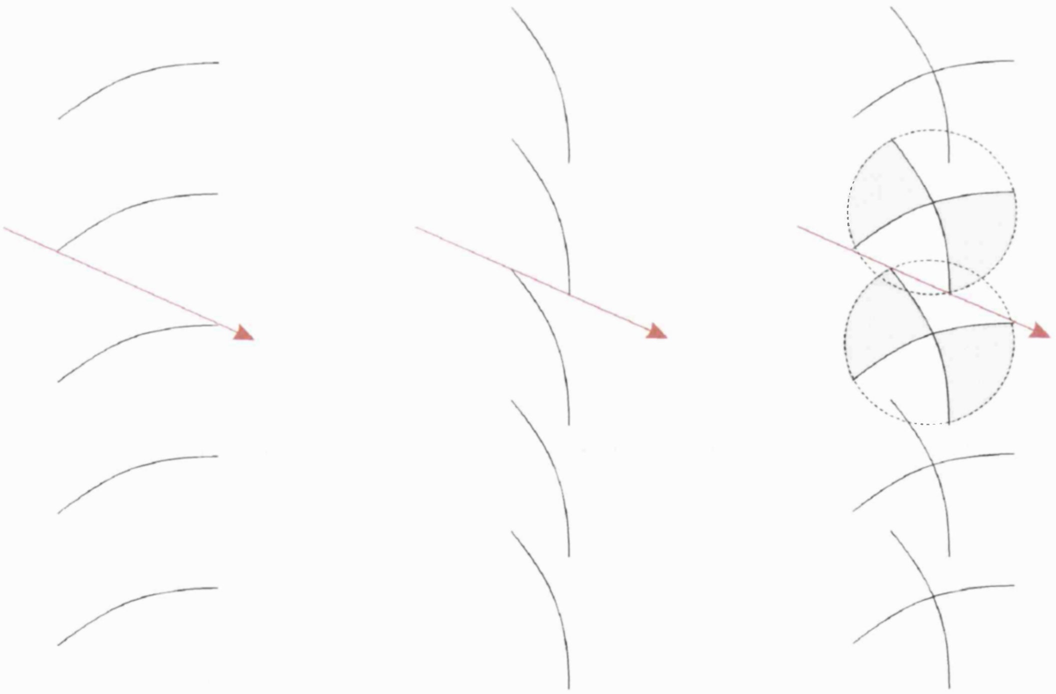


Figure 3.3: Diagram showing the two cut off tilt angles for a particular solar wall altitude angle. The shaded area illustrates all of the tilt angles that would permit solar penetration.

Through computer simulation modelling, the researchers at LBL showed that a control strategy that blocked the sun and optimised the daylight contribution into the room, resulted in better energy savings than a threshold strategy that simply blocks the sun and maximised the view. They also showed that by controlling mid-pane venetian blinds to optimise the daylight illuminance within a room, annual energy savings of 16-26% and reductions in peak demand of 17-24% were possible for their cooling dominated climate.[12]

However, when LBL tried to implement this strategy in practice, they found it was extremely difficult to achieve these predicted energy benefits. One reason was that, instead of having continuous blind movement, they found their blind could only be controlled to move in 15-degree increments to block the sun and maximise one of the control mode objectives.

Another reason was that the thermal and optical complexity of the blind made the closed loop control, necessary to implement such a strategy successfully, extremely difficult to fine tune. Relationship of daylight illuminance to blind angle is non-linear, and thus blind movement is restricted and control optimisation made more complex. Also the need to prevent distractions from continuous blind movement caused by large variations in external illuminance, meant that a time delay needed to be added. This also had a great effect on the system's efficiency.

In the end they showed that such a system, when used for light redirecting and optimising, can result in 4:1 gradient of light across the room and the occupant having a direct view of sky through the blinds. These factors provide problems in terms of achieving adequate lighting quality and visual comfort. They concluded that limiting slat movement to angles below horizontal would increase views to outside, daylight uniformity throughout the space and thus occupant comfort. Nevertheless, this was at the expense of the energy savings.[69 71]

Although some work had been done by LBL to investigate user response to the systems they developed, these studies were limited as they merely involved 14 people being placed in three separate control environments for an hour each. The studies found that the occupants regarded the systems as performing well but the researchers themselves recognised the limitations of the study.[59]

3.4.5 Scene Control

A few lighting control companies have developed products that store and recreate a series of architectural/theatrical lighting scenes by co-ordinating the daylight control from the blinds with the artificial lighting system.[73] This provides a good way of allowing the occupants to program a series of their preferred blind and light settings into the control system so that they can be easily recalled at the user's convenience.

Scene controllers largely operate in an open-loop fashion and therefore require a large amount of internal and external sensor information and complex sky scanners are often employed to provide the required information about the current sky conditions (see Figure 3.4).

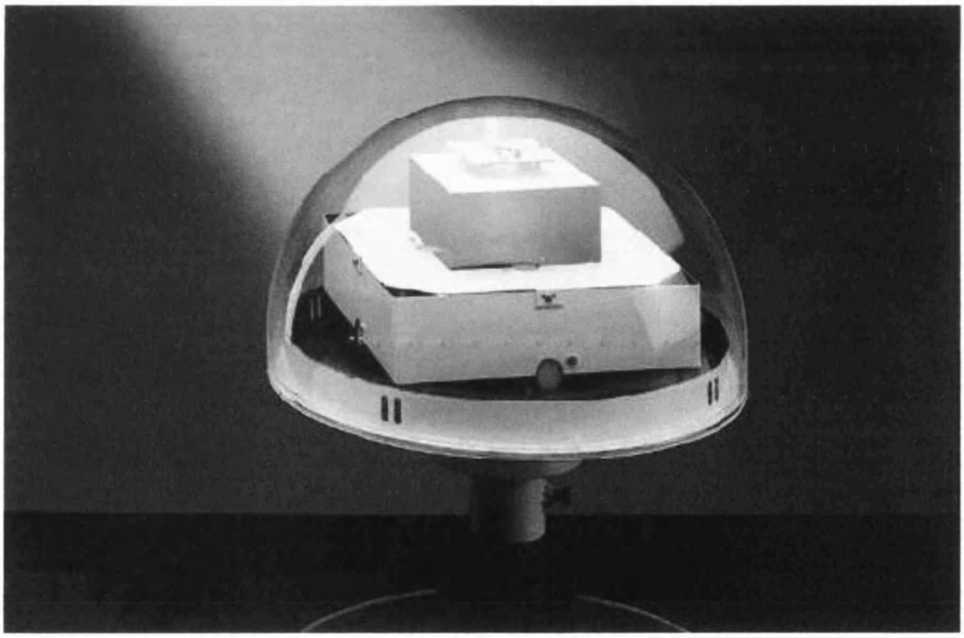


Figure 3.4: A sky scanner that is used in one lighting controls manufacturer's scene control system. The scanner has four horizontal and four vertical illuminance sensors, one of each for each compass orientation. The scanner also has a global irradiance sensor located at the top.

Chapter Four will go on to demonstrate that the incorporation of such complexity within automated blind systems to enhance user satisfaction can lead to revenge effects.[74] These include specialised and costly commissioning and management processes, and systems that are difficult for users to understand and which do not meet individual expectations.

3.5 Chapter Summary

This chapter has developed and proposed a new categorisation system for emphasising the levels of functional complexity inherent within current systems. The work was supported by both the work in Appendix III and the lessons learnt from the practical system implementation, described later in Chapters Five and Six. By undertaking this work we now have a better understanding of current and proposed automated blind control techniques and the factors they incorporate within their control algorithms. We can now use this information to compare those factors to the factors previously identified in Chapter Two, as being important in an individual user's decision making process, in order to gain an insight into the shortcomings of current system methodologies. This comparison is described in Chapter Four, which begins by showing how significant improvements in meeting users' needs are possible by allowing the individual user to interact with the system through a simple interface.

---

[74] 'Revenge effect' term coined by Edward Tenner to describe negative side effects of a technological solution to a problem. Tenner E, 1996, "Why things bite back: Predicting the problems of progress", Fourth Estate Ltd., London.

Chapter 4: Meeting Individual Expectations with Automated Blind Control

4.1 Introduction

Chapter Three reviewed the current state of the art of automated blind control and revealed that systems often rely on only a few simple physical variables to drive their decision-making processes. Such systems are unable to deal with the complex nature of occupant satisfaction and the many factors that influence the way in which individuals control blinds. In an attempt to overcome this problem and attain universal user satisfaction, many control system designers now provide the occupant with a means to exercise some form of control. This provision enables the user to alleviate any discomfort not dealt with by the system, thus adding the important psychological benefit of the 'perception of control'.

However, recent post occupancy studies have identified that even when provided with a user over-ride, occupants are often still unsatisfied with the operation of automated blind systems.[32] One explanation for this phenomenon might be that because a venetian blind has a major impact on the thermal, visual and psychological well-being of a space, the provision of occupant over-ride alone is often not enough if the automated control strategy itself does not meet the user's expectations.

This chapter reviews these issues by studying the implementation of individual control and by bringing together the work outlined in Chapter Two and Chapter Three in order to compare what is expected by the occupant and what is provided by the control system. By doing this we can start to understand why people become frustrated with blind control systems and identify possible options for improvement.

4.2 Providing Individual Control

4.2.1 Balancing the Advantages and Disadvantages of Providing Individual Control

Chapter Two showed us that providing a user over-ride facility in situations where automatic control affects occupants performing tasks is essential for ensuring user satisfaction. Indeed the complexity of the human decision making process means that an automatic control system, no matter how advanced, cannot accurately pre-empt every user adjustment.

Unfortunately, certain disadvantages are associated with providing individual user control. Firstly, when occupants exercise control, often to alleviate discomfort, they alter their environment to suit their individual preferences and as a result may reduce energy savings. Secondly, the provision of individual control can add capital and operating costs to the system.

In theory, these disadvantages are easily outweighed by the primary benefits of individual control, namely increased user satisfaction and productivity. However to achieve this balance in practice, it is extremely important to ensure that the users both understand and are able to operate the system, and that the automatic control algorithms utilised reduce the need for occupant interaction by providing an adequate and appropriate amount of control.

4.2.2 Providing an Adequate User Interface

If occupant over-ride is to be successful it is vital that the designer makes it clear to the occupant what controls they have and how they should use them. If the users do not understand the operation of the controls, the interface may distract the users from their work and may not always result in the desired response.

Problems tend to occur when increasingly complex systems, such as mode/scene controllers, offer extra functionality to the user, often when it is not required.

Therefore when designing such increasingly elaborate systems, any additional complexity should be locked into the control code itself and not imposed on the users, thus freeing them from the need to continually issue commands. Indeed for a venetian blind controller a simple up-down switch, with adequate fine-tuning capabilities, is often all that the user requires to control both the blind tilt and lift.

Such simplicity has advantages for both the user and the designer. For the user it is easy to understand and convenient to use. For the designer it is efficient and inexpensive, as it does not require a great deal of ergonomic design to be successful.

4.2.3 The Importance of Adequate Automatic Control

The most common method for providing individual control within an automated system is to provide acceptable energy efficient automatic control for a high proportion of the time, and allow the occupants to over-ride the system whenever they need to avoid discomfort.[47] When the occupant is given this form of personal control, they are provided with three different forms of individual control:[75]

(i) *Decisional control* - the opportunity to make various adjustments;
(ii) *Cognitive control* - the perception of control, or the way in which an event is interpreted and appraised;
(iii) *Behavioural control* - making an adjustment to avoid a threatening event, such as overheating.

Environmental psychologists Veitch and Gifford identified that the availability of decisional control (e.g. thermostats, window blinds etc.) and the perception of control, both contribute to user satisfaction. However, the exercise of control (behavioural control) often reduces occupant satisfaction.[27] Therefore, just as providing no means to over-ride an automated system can lead to user dissatisfaction, the constant need to intervene with a system can also lead to dissatisfaction.

This problem is compounded with the issue of how long after the user has made an adjustment does the automated control system take back overall control? People often over-ride systems when they are not comfortable and get frustrated if their actions to alter blinds to maintain comfort are thwarted. If a system is causing annoyance, it is common for the facilities manager to reduce occupant complaints by increasing the time delay on over-riding occupant adjustments, so that the automatic system only adjusts occasionally. However, this modification can severely undermine the overall energy efficiency of the system and often negates the reasons for installing such a system in the first place.

Similarly, inappropriate automated interventions can also be exasperating. If a system does not respond according to the occupant's expectations (for instance if the blind lowers when the window is in shade), then the occupant will soon become dissatisfied with that system.

The blind has a large influence on the occupant's visual perception of the internal environment. It is therefore insufficient to design an automated blind control system primarily to achieve energy objectives, restrain it within empirically derived comfort bands, and rely on the user to make adjustments when necessary. If one is to ensure user acceptance, one must also ensure that an effective user interface is available and most importantly, that the control system's operation meets the user's expectations. It is the inability of modern day control systems to satisfy these criteria that has led to them not being accepted by users.

---

[75] Averill J.R., 1973, "Personal control over aversive stimuli and its relationship to stress", *Psychological Bulletin*, Vol.80, 286-303.

4.3 The Shortcomings of Current Blind Control Techniques

4.3.1 Lack of Understanding

In general, automated blind systems have not fulfilled their promise as energy efficient devices that can also enhance occupant comfort and performance. Part of this problem can be traced back to a lack of adequate information in the industry about the way in which people control blinds. Therefore the key focus for improvement, and an additional aim for this work, should be the development of a better understanding of what individual users really want, and do not want, from blind control systems and how they need to use them. This could lead to better briefing, enhanced design, and improved integration with other systems, such as lighting systems.

4.3.2 What are Users' Expectations of an Automated Blind System?

Although it is dangerous to generalise about individual users' needs and expectations, it is safe to say that a user expects a control system's response to be similar to their own response. After all, an automated building control system should be designed to respond to their needs.

Figure 4.1 illustrates how the subconscious human decision making process and the automatic blind's decision making process run side by side in real time. If the control system's actions are approximately equal to the occupant's desired action when a change in environmental conditions occurs, then the occupant will be satisfied with that action and it will meet his/her expectations.

If we think of the human response in terms of the feedback control mechanisms illustrated in Figures 2.1-2.3, we can say that the user's expectations of the way the control system should respond are linked to what they themselves are sensing and what they believe the control system should be sensing. The designers of control systems also have expectations about what they believe occupants need and these are usually reflected in variables included within a system's design parameters.

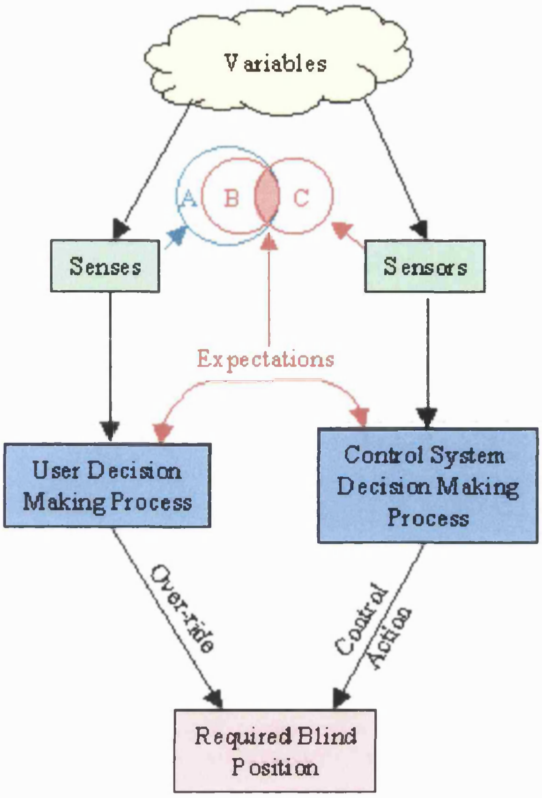


Figure 4.1: Diagram showing the relationship between a user's decision making processes and an automated blind's decision making process.

The Venn diagram within Figure 4.1 illustrates this. Set A represents all the variables sensed by the user. Set C represents all the factors sensed by the control system and thus its expectations of the user.

Chapter Two highlighted the complex nature of the occupant's decision to adjust their blind. Some influences, such as psychological factors, are virtually impossible for the artificial system to ascertain and although these factors are important, it is safe to say that they should not come into effect when the user considers the operation of the automatic blind system itself. Indeed, the user should not expect the system to respond to their every whim, because, as we have seen in Chapter One, some individuals enjoy the ability to be able to alter the blind to 'recreate oneself' and such a possibility provides real psychological benefits. Therefore Set B on the Venn diagram represents the factors that the user expects the control system to respond to (the users expectations), with subconscious psychological factors falling outside B within A.

Based on the comments of individual users in the existing user studies reviewed in Chapter One and Two, it seems save to assume that the factors that the user does expect the system to deal with, are those they regard as the simple environmental factors, which are associated with the way they believe the system is controlled. For example:

- the sun is shining on their desk: the blind should be down;
- the sun is no longer on their window: the blind should be up;
- the external environment is producing glare: the blind should be down.

These perceptions are strongly related to the individual environmental conditions that the user experiences in their own context. Therefore, it is their specific contextual conditions that are important to them and these conditions are influence by factors such as the orientation of their desks, the reflectance of pictures on the walls and so on.

It has been proposed that user frustration with automated blinds is linked to the user's expectations, which is in turn linked to their own ability to appraise the situation and their understanding of what factors the artificial system is taking into account. Therefore for the control system to be successful Set C must largely be within Set B. In other words the control system must meet an individual user's expectations, more often than not, to avoid repetitive occupant over-rides.

4.3.3 How Well do Current Control Systems meet User Expectations?

To examine how well current automated blind control techniques meet user expectations, we can compare factors being taken into account in the individual's decision making process with the factors being taken into account in the control strategies.

Table 4.1 compares the ability of each of the levels of complexity in current blind control techniques, outlined in Chapter Three, to deal with the factors outlined in Chapter Two. The ratings have been devised from the experience gained by the author in the test room study, to be reviewed in Chapter Five, and through discussions with the manufacturers and studies into the workings of their devices.

	Threshold	Sun Blocking	Mode/Scene
Comfort			
Physical Environmental Variables	●	●●	●●●
Individual User Preferences	●	●	●●
View Out	●	●●●	●●●
Usability	●●●●	●●●●	●●
Manageability	●●●●	●●●	●
Climate			
Fluctuations in Weather	●●	●●	●●●
Luminous Distribution of the Sky	●●	●●	●●●
Context			
Solar Geometry	●	●●●	●●●
Internal Finishes	●	●	●●
External Obstructions	●	●	●●●
Use of Space	●	●	●
Individual Building Design	●	●	●●

Key

●	= very poor
●●	= poor
●●●	= average
●●●●	= good
●●●●●	= very good

Table 4.1 Table rating the ability of each of the levels of complexity of automatic blind control to deal with the factors that influence the way occupants themselves control blinds.

As a large variation in the nature of controllers exists, a good practice example from each category has been chosen for the rating process.

- Threshold controller system: - tilts blind to 45° once lowered; utilises a sun sensor on each facade, time delays and a ten minute control interval.
- Sun-blocking controller: - algorithm driven; tilts slats to block the sun and maximise view; incorporates time delays, a ten-minute control interval and sensors on each facade.
- Mode/scene controller: - linked with illuminance, temperature and occupancy sensors; can be commissioned to take into account obstructions; allows occupants to select lighting scenes; utilises a complex sky scanner.

Each system is rated as if it had no occupant over-ride. In other words, the rating refers to their automatic mode. Where algorithms have the ability to take into account occupant preferences, such as with the mode/scene controller, the effect of this has been included when making a judgement on their performance.

The review identified that the shortcomings of blind controllers can largely be attributed to the fact that they are either:

- *too simple* - such as threshold and sun blocking controllers, because they do not account for the diversity of factors that affect occupant blind control, or
- *too complex* - such as mode or scene controllers, because they are time consuming to design, manage and install, and are therefore inflexible and often difficult for the users to understand.

4.3.4 Shortcomings of Simple Threshold and Sun Blocking Controllers

Overall, these first two controller types have simplifications, which although make installation and maintenance easier and their cost lower, also reduce the system's ability to provide user satisfaction. By relying on only a few physical variables to make their control decisions, they are unable to deal with a multitude of physical factors that influence the human decision making process. Indeed one or two solar illuminance sensors for each facade can not account for the variations in lighting conditions experienced from office to office and from desk to desk.

Whilst solar blocking and threshold control is necessary, it is not sufficient for providing visually pleasing and glare free environments. Glare can occur from a bright overcast sky with illuminances well below default thresholds and is dependent on a multitude of contextual factors. These blind control systems do not account for glare in this way as they simply operate to lighting levels.

4.3.5 Shortcomings of More Complex Mode/Scene Controllers

Mode/Scene controllers on the other hand have the ability to take into account many more variables, both physical and contextual. However, the controllers are limited by the fact that these additional factors often have to be entered into the system manually at the design/commissioning stage, for each blind, in each building. This methodology makes the overall system complex and difficult to use and manage. More importantly, this added complexity does not seem to have enhanced the system's ability to meet user expectations, because the systems largely rely on comfort indices that are unable to give a good overall impression of the conditions experienced by the user across the whole space. Daylight is constantly changing with solar position and sky conditions and ones complete experience of the daylit environment cannot always be reduced to a few measurable terms.

4.3.6 The Balance Between Complexity and Usability

The temptation is to think that by adding complexity to a control system, we can improve its response to the environment in which it operates. While this is true to a certain extent, adding too much complexity often results in additional problems. It should also be remembered that no matter how complex a system becomes it will never be able to accurately pre-empt every user adjustment.

Researchers at Carnegie Mellon University in Pittsburgh have taken the idea of 'added complexity makes a better system' to an extreme.[76] In an attempt to improve the energy performance and comfort of control systems, they have proposed the use of multi-aspect virtual models of buildings ‘to supplement and enrich the informational repertoire of building control systems’.[76] These models are essentially electronic versions of the buildings themselves, with all the contextual factors, such as reflectances of wall paints and the nature of external obstructions, programmed into their algorithms. Real time inputs to the model are provided by a large number of sensors that provide information on climatic variations. The researchers suggest that such models would be able to move backwards in time to analyse the building's past performance and move forward in time to predict the building's future performance. This ability would then be used to aid the control system when choosing between a variety of modes.

In order to demonstrate their proposal, the researchers gave an example of how such a system might be utilised to aid the control of shading louvres, used on the exterior of their test building, by incorporating a variety of performance criteria, such as the:

- annual building energy need;
- light distribution uniformity;
- accumulative deviation of average illuminance from a target value; and
- average cumulative comfort indices for the space.

They concluded that these performance criteria could be formulated to simultaneously address economical and ecological considerations whilst providing thermally and visually comfortable conditions in buildings, thus reducing the need for occupant interaction.

---

[76] Mahdavi A., Chang S. and Pal V., 1999, " Model-based integration of contextual forces into advanced building control systems", Proceedings of "Intelligent and Responsive Buildings", Brugge, 29-30 March 1999, CIB Working Commission W098.

Theoretically control strategies should be designed to meet such performance criteria. However resolving multiple, possibly conflicting, criteria can lead to an involved and cumbersome process of rule-making and determining weight factors, that will in most cases result in some of the criteria being only partially satisfied. These chosen criteria will affect the energy performance and if incorrect, comfort. The difficulty of resolving conflicts between performance criteria is due principally to the different 'value' perceived by various end users and owners. As a result, alterations to the system are often required to meet users' needs as they arrive and due to the complex nature of the system these alterations are often difficult, time-consuming and costly to implement.

Although this example implements the contextual factors already identified as being important for meeting users' expectations, through a highly complex building model, it fails to provide effective control because a number of practical considerations that were overlooked during its theoretical development make it inflexible to alterations.

Research by others has also highlighted some of the practical shortcomings of complex daylight and shading systems. During his investigations into installed daylighting and lighting systems, Slater found that inappropriate or unrealistic assumptions were often made in lighting design about occupant behaviour and their visual requirements.[77] He also found that:

- few installations were operated as designed;
- some control technologies were more difficult to design, specify and commission than might have been expected and adverse user reactions were common;
- attempts to utilise blinds to reflect daylight in deep open plan offices often resulted in glare, particularly when display screens were poorly orientated; and

---

[77] Slater A., 1995, "Occupant Use of Lighting Controls: A review of current practice, problems and how to avoid them", Proceedings of the CIBSE National Conference 1995, Vol. I, Eastbourne, 1-3 Oct 1995.

- interfaces between control systems, occupants and managers often failed to follow ergonomic principles, by not being the of right kind, in the right place or straightforward to use and understand.

Similar findings were also identified during LBL's research into automated blind control techniques and both studies attributed the inadequate performance of current systems to such practical constraints as above.[59] [69] [71] Therefore if we are to improve blind control techniques, we must develop systems that avoid these problems by finding the right balance between system complexity and usability.

4.4 Chapter Summary

This chapter has compared users' needs and expectations to the automatic operation of existing and proposed blind control systems using: firstly the knowledge gained in the literature survey reviewed in the last three chapters; secondly the author's experiential judgements as a designer and installer of products; and thirdly anecdotal evidence on issues like usability and manageability.

An answer to the question, 'why don't people like automated blinds?' is proposed. It states that current systems are either too simple, including too few variables to deal with the intricacies of individual users' needs and expectations, or in cases where designers have attempted to include more of these variables, too complex to be practical in real world situations. This proposed answer is an important first step in determining a solution to the problem of improving blind control strategies. It suggests that the solution must be practical, balancing the issues of complexity and usability, whilst still catering for the individual needs of individual users.

Therefore the hypothesis for the thesis has become:

Can existing control strategies be developed to incorporate factors that have been identified as being important to an 'individual' user, without the need for a complex management intensive structure?

The next step to solving the problem of improving blind control systems should be to gather knowledge to enable the hypothesis to be investigated fully. This involves gaining more detailed knowledge of:

- the factors that affect the way individuals control blinds;
- the controls technology available to deal with manageability issues; and
- how the two can be married together.

Two methods of knowledge acquisition tend to be used in control system development: experimental data or expert knowledge. It could be argued that to proceed, the first step should be to carry out a broad detailed experimental investigation to identify the factors that affect the way a number of individual users interact with blinds, as no detailed information is currently available. This approach was considered, but was discounted for the following reasons:

The complexity and variability of individual user needs and contexts, outlined in Chapter Two, suggests that traditional experimental methods would require a large number of individuals to be tested. Although the data from such a study could be used to recognise patterns it could not be averaged to obtain a smaller data set, as in this study we are interested in the data from each individual in their individual context. In fact, the results of the surveys carried out by others and reviewed in Chapter Two would be equally valid to this approach. Infact, obtaining a detailed data set for each individual user is the most important aspect of this work. But as the number of times the individual interacts with the blind may be small, the data set should be long to provide the author with enough data to recognise the patterns. Therefore the lengthy time requirements associated with the gathering of broad experiment data to meet the aims and scope of the problem, as identified, were considered to lie well outside the timescale of this study.

The shortcomings of current systems can largely be attributed to the fact that to date product developers and researchers have been thinking about the problem of blind control from either a lighting or a controls perspective. Whilst developing the problem solving methodology for the next stage of the study it was considered important to try to overcome these failings by continuing to think of the system as a whole and utilise integrated environmental design to solve the interdisciplinary technological problems that are often missed with component-orientated research. In other words, it became clear that the author should develop detailed expert knowledge of each of the important aspects of the system design, such as the controls technology, energy considerations, but more importantly, individual user responses and experiences. It was considered that the best way to achieve this, within the remaining timescale of the project, was to build a state of the art system and to experience it as an individual user. Therefore a full-scale test room facility was proposed to explore the issues discussed.

The objectives of test room study were:

- gain knowledge of control technologies that cater for manageability and usability concerns whilst also supporting the idea of controlling systems to suit individual users;
- to gain more detailed knowledge of the intricate workings of automated blind systems, software and hardware; and
- to gain individual experience of the system as an individual so as to identify important factors and put them in a particular individual context.

Meeting these objectives was seen as an important part of the process of gaining enough knowledge to propose a solution to the problem of automating blinds by exploring the hypothesis outlined above.

In order to keep the work focussed on these objectives and to use the time most effectively, the definition of the target solution was narrowed. The complex factors outlined in Chapter Two were simplified by assuming that the building as a whole was well designed and the occupant is situated within a single occupancy space (such as a cellular office where only one user interacts with the blind). This decision reduced the number of operational control schemes that needed to be investigated and simplified relationship between the individual and the system to provide the necessary starting point for the development of a solution to the problem. Details of the test room set-up are outlined in the next Chapter.

Chapter 5: The Test Room Study

5.1 Introduction

5.1.1 The Aims of Setting Up the Test Room Study

The test room study implemented as part of this research differs from other test room studies already undertaken in this field, in that its goal is not to use a large number of tightly calibrated sensors to make a scientific evaluation of the environmental performance of automated blind systems. Instead it uses existing off the shelf technology to build a prototype of an automated blind control system that adopts a similar number and type of sensors to those applied in real, economically viable and commercially available systems. The main objective was to investigate a series of practical issues already identified as being important to the design of automated blind control strategies, in order that the thesis could be developed to aid the evolution of current systems and meet realistic near-term goals.

In addition to investigating these practical issues, the study also provided the opportunity to place some of the lessons learnt during the initial paper based research within a particular context. This process was used to enhance the author's understanding of the problems associated with the automation of venetian blinds, through both exposure and observation, and relate them to a state of the art controller developed during the study.

The practical issues considered by the test room study were divided into the following categories:

- *Manageability*
- *Interoperability*
- *Usability*
- *Sensitivity*

5.1.2 Manageability

The installation, commissioning and management of any control system should be simple and effective as well as having the flexibility to deal with the diverse nature of organisational behaviour and changing occupant needs. For example, a system should be able to cope with changes in: workstation location and partitioning; working groups; task and use of space; working hours; as well as the various types of users that may need to interact with it: occupants, cleaners, security staff and over-time workers.

This part of the study considers the implications of a dynamically controlled integrated system with regard to the selection and specification of an appropriate supporting building controls infrastructure. The proposal to implement adaptive control at an individual occupant level could add a certain degree of complexity to a control system when it is considered on a global building scale. It should be ensured that this added functional complexity does not interfere with the management of the system and therefore we must find a framework that supports both.

5.1.3 Interoperability

The need for effective communication between multi-vendor control systems is paramount to the realisation of effective automated building systems. For example, in order to reduce the need for duplication of components, a blind controller may need to use data broadcast by the heating system's temperature sensors and/or the lighting system's photo-sensors. By doing so, one has the opportunity to provide control systems with greater functionality at a reduced price. This part of the work looks at ways of achieving seamless integration of lighting and blind control systems along with a few elements of other systems.

5.1.4 Usability

The start of this chapter highlighted the fact that it is important to ensure that the interface between the occupant and the control system is user friendly. This is especially true when proposing to use this interface as a means of providing feedback to an adaptive control system. Therefore it is vital that we identify a user feedback system that offers an acceptable level of performance for use in our new framework. It must:

- be simple and intuitive to use and not complicated by unnecessary features;
- be easily accessible to the user;
- provide a rapid response; and
- provide adequate fine-tuning capabilities

5.1.5 Sensitivity

Another aspect of meeting users' expectations is ensuring that the sensitivity of the control system to the changes in input variables, such as the external light level, is appropriate for the degree of control required. The blind should not distract the users by being too sensitive to variations and should make adjustments that are imperceptible to the user, initiating gradual rather than step changes. We must be aware of the fine-tuning capabilities of commercially available automated blinds and their performance in terms of defined control strategy goals in real-time under variable sun and sky conditions.

5.2 Field Bus Systems

5.2.1 The Technological Solution to the Practical Problem

For the proposed control framework to satisfy the practical criteria set for it, it must be capable of being incorporated within a Building Management System (BMS). At the time this study began, there was a drive by many practitioners to make building control systems evolve from closed proprietary black box systems to more open systems with flexible physical media and interoperable/interchangeable components, named field bus systems. A description of the difference between closed propriety control and field bus control can be found in Appendix V.

Interoperability in each type of field bus system is achieved through the use of standardised communication protocols. By ensuring that all devices installed on a network use such a protocol, or language, a global building-wide blind control system can talk to other automated building systems such as security, fire and lighting systems or can be linked to computational processes such as energy use diagnostics.

The advantages given for this kind of interoperability over the proprietary nature of traditional systems include:

- *Lower cost* - due to the need for less wiring;
- *Easier to commission, manage and maintain* – one operator can look after many systems as they look and feel the same;
- *More flexible and adaptable* - no proprietary manufacturers locked into system and the system's logical connections can be changed at any time without physically changing the wiring;
- *Interoperable plug and play functionality* – data sharing simple between HVAC, lighting, fire and security systems, offering possible safety benefits;
- *More information* – the data transparency of the network allows unprecedented levels of information to be obtained about a process in real time;
- *Better fault diagnostics* – a modular approach means that faults can be quickly identified and traced to specific locations.

Field bus systems facilitate decentralised control and intelligence and, as a result, are able to accommodate non-hierarchical control of multiple building sub-systems whilst supporting centralised monitoring and diagnostics. In addition to simply changing set points, these intelligent nodes are capable of self-diagnostics, calibration, validation, linearisation, interoperable messaging and the ability to share resources. With the dramatic increased functionality of the microprocessor there is plenty of untapped potential to make automated blinds easier to use, diagnose, monitor, integrate and operate.

The proposal for automated blind development, outlined in the last chapter, requires a good modular integrated building control system to allow it to function as intended. The blind control might need to be triggered by an occupant's needs within a room, or on a building wide scale in response to another part of the system. After close examination it was decided that 'field bus technology' could offer all of the characteristics we require to implement our ideas effectively. However, a decision had to be made on which of the many different types of field bus systems to use.

5.2.2 Types of Field Bus System

Several field bus protocols are currently competing for the building automation market. These include:

- *LonWorks*
- *European Installation Bus (EIB)*
- *BATIBUS*
- *PROFIBUS*
- *WORLDFIP - The Factory Instrumentation Protocol (FIP)*
- *FND*
- *ASHRAE's BACnet*

The availability of all these systems and their individual communications protocols has produced a somewhat complex marketplace, with industry unwilling to commit to a specific bus before the emergence of a clear market leader. The situation has been compounded by the standardisation process that is been unable to define a clear standard. Therefore before choosing a protocol for use in this research, a list of functional requirements was drawn up and each protocol was examined for its ability to meet these requirements (see Appendix V).

The decision making process identified LonWorks as the field bus system that demonstrated the most attributes required to develop the proposed framework. This choice has since been reinforced by that fact that most observers now seem to regard LonWorks as the *de facto* standard for the UK and Scandinavia.[78] [79] [80] [81]

The decision to use LonWorks had the additional benefit of allowing the author to accommodate the test room study within a larger research project, also undertaken by the author on behalf of the Centre for Window and Cladding Technology (CWCT), that had received significant funding from the Department of Environment Transport and Regions. The aim of the project was to investigate and demonstrate the possibility of integrating automated facade components, such as venetian blinds and window openings, with building services systems using the LonWorks technology.

5.2.3 LonWorks

A LonWorks system consists of a network of micro-controllers that communicate in a peer to peer fashion over a variety of different media, including twisted pair, power line and radio frequencies. The heart of the system is the Neuron chip, a specialised chip that incorporates three 8-bit microprocessors, one for handling application code and two for handling network communication.[82] This chip offers the distributed intelligence needed to implement learning on a node by node basis.

The robust characteristics of LonWorks applications can be attributed to the standard communications protocol LonTalk, which is much like Ethernet but optimised for small data packets. The two network communications chips in each Neuron have embedded firmware necessary for running the protocol and communicate with other devices on the network through network variables. These network variable are either

---

[78] Wilmhurst P. and Hamblin J., 1999, "Openness – A customer driven evolution" Proceedings of the CIBSE National Conference 1999, Harrogate, pp128-135.

[79] Aldridge A., 1993, "Trends in control system development", Proceedings of the Conference of 'Intelligent buildings today and in the future', UCE, 7th October 1993, pp21-29.

[80] Faithfull M., 1997, "Can we talk?", *CIBSE Journal Building Services Supplement on BMS controls*, November 1997, pp10-11.

[81] Hobson S., 1998, "Who needs integrated building controls?", *M&E Design*, April 1998, pp30-32.

[82] LonWorks Technology Device Data, Motorola, 1997

selected from a comprehensive list of Standard Network Variable Types (SNVTs) published by an interoperability association named LonMark, or are custom structures defined by a particular device manufacturer.[83] [84] The device manufacturer can also add their own custom application code to the third microprocessor in the chip by using software that is an event driven variant of ANSI C known as Neuron C.[85]

Together the three microprocessors allow the network to be capable of control on a number of levels. At a low level there is enough control capability available on a Neuron to deal with low level IO, dealing with sensors, actuators and running control loops. At a middle level the network can be made into an integrated system that co-ordinates low level nodes to give good system performance. At a high level, network management and front-end co-ordination can be achieved through comprehensive Application Programming Interface (API) facilities and the LonWorks Network Services (LNS) Architecture, which applies client-server architectures to control networking.[86]

The interconnection of network variables is done in a binding process by a network management tool at installation. This is a soft-wiring process that uses software to make logical connections in the system and downloads this information to be stored in the EEPROM memory of each Neuron chip. This procedure is possible through component based software design and the LonWorks Component Architecture (LCA) which enabled links to be made between the control system and the high level operating systems, such as Windows 95 and NT, translating IO to IT. A more detailed review of LonWorks can be found in Appendix VI.

---

[83] LonMark Application Layer Interoperability Guidelines, Version 3.0, LonMark Interoperability Association, Palo Alto, CA, 1996.

[84] The SNVT Master List and Programmers Guide, May 1997, Echelon Corporation, 4015 Miranda Avenue, Palo Alto, California.

[85] Neuron C Reference Guide, Document No. 29350, Echelon Corporation, 4015 Miranda Avenue, Palo Alto, California.

[86] Lund J., 1996, "The LonWorks Network Services (LNS) Architecture Strategic Overview – Client Server Control", Document No. 39305 Rev1, Echelon Corporation, 4015 Miranda Avenue, Palo Alto, California.

5.3 The Test Room

5.3.1 The Test Room Layout and Location

A test space similar in size to a single occupant cellular office space, 2m wide by 3.8m deep and 2.4m high, was instrumented and configured to incorporate two automated vents, an automated internal venetian blind and daylight linked lighting system. The room was located in the offices of the CWCT, a research unit based at the University of Bath in the UK, (Latitude 51° 22' N, Longitude 2° 21' W) [see Figure 5.1 and Figure 5.3]. The window wall was orientated 40 degrees East of due South and was the room was on the second floor.

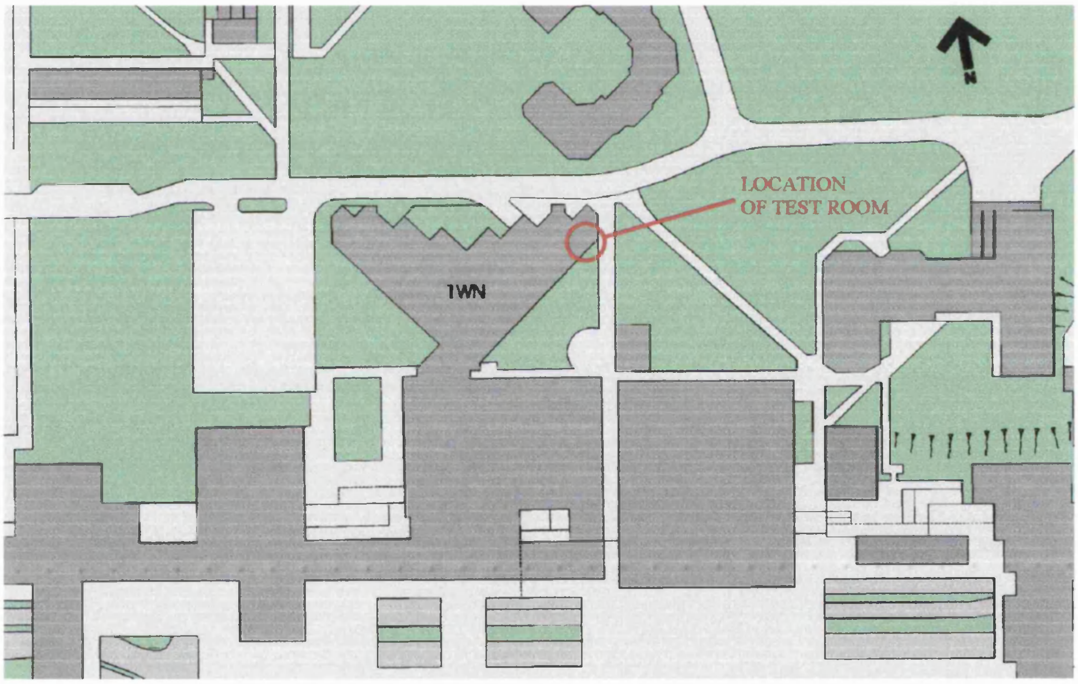


Figure 5.1: Diagram of room's location on campus

The test room was formed from part of a much larger office at the end of a building that accommodated windows in many orientations. Therefore it was vital that the test room was provided with effective partitioning to ensure that its environment was largely determined by alterations to its own window and not by alterations to windows in the adjoining office.

The partitioning was achieved by using a series of large floor to ceiling height bookshelves that were already in the room, and full of books, to provide a certain degree of thermal mass between the test room and the rest of the room.

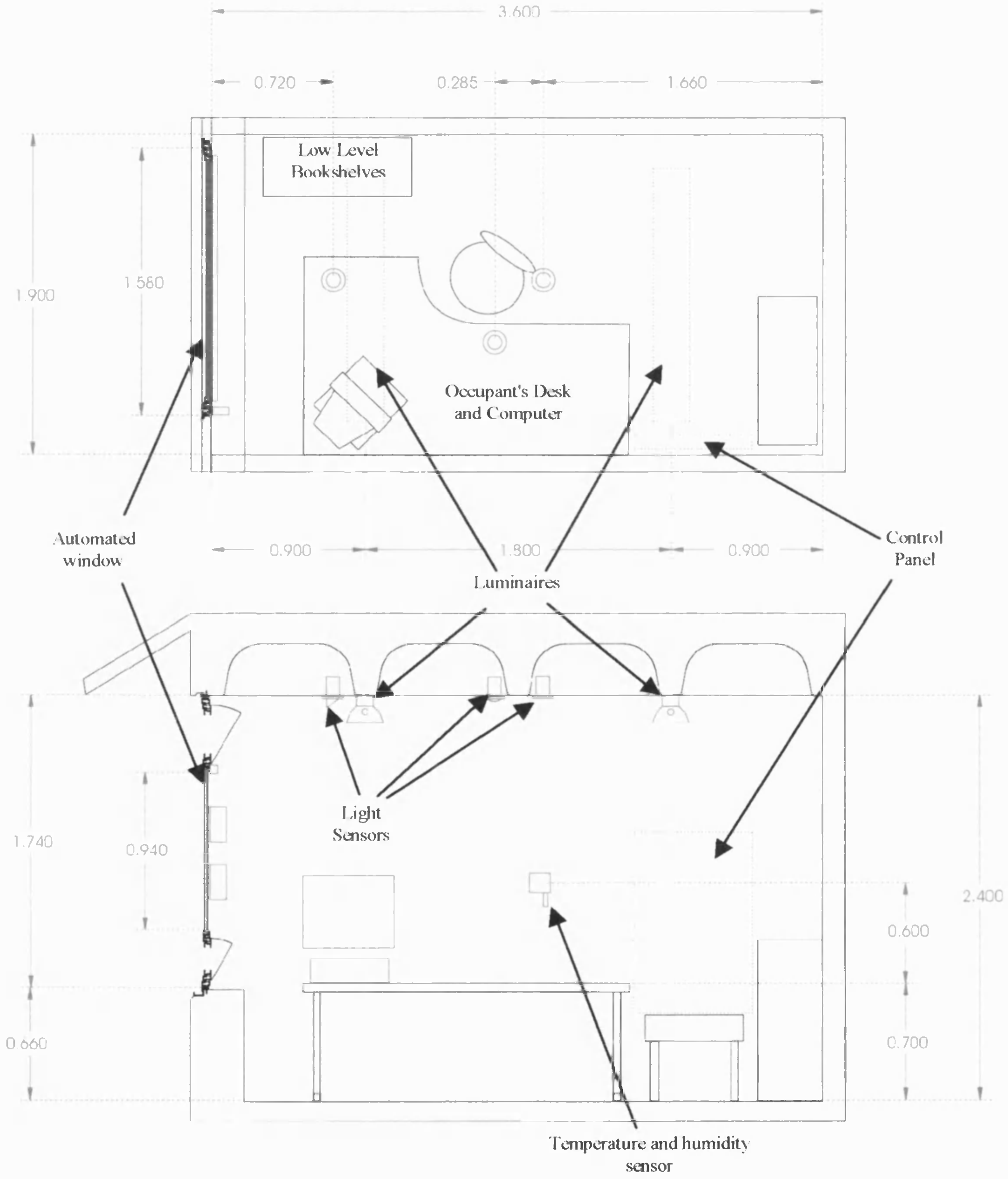


Figure 5.2: Plan and section of the room showing location of various hardware items

Off-white thermally lined curtains were used to finish off the dark backs of the bookcases, which formed two sides of the internal surfaces of the room. This ensured that any gaps in the bookcases were closed off to air movement and light and that the optical properties of the internal surfaces of the test cell were more typical of the surfaces usually found in offices [see Figure 5.3].

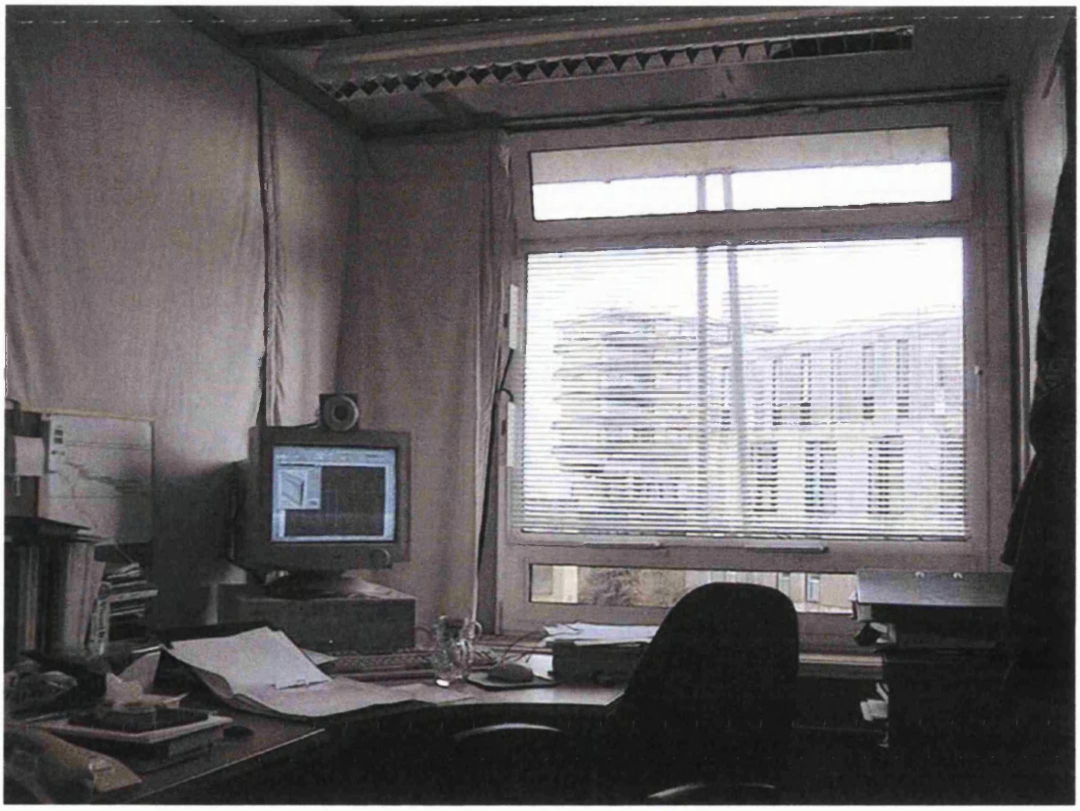


Figure 5.3: A picture of the test room, showing the desk location, the new window, one of the new lights and the computer used for monitoring and control.

The test room blended a variety of 'off-the-shelf' products, such as a motorised venetian blind, a controllable lighting system, a series of photo-sensors and other sensor monitoring equipment, which were selected for their interoperable functioanlity and suitability for creating an 'intelligent office space'. All of the devices were hardwired to a twisted pair communications bus, which formed the LonWorks control network [see Figure 5.4].

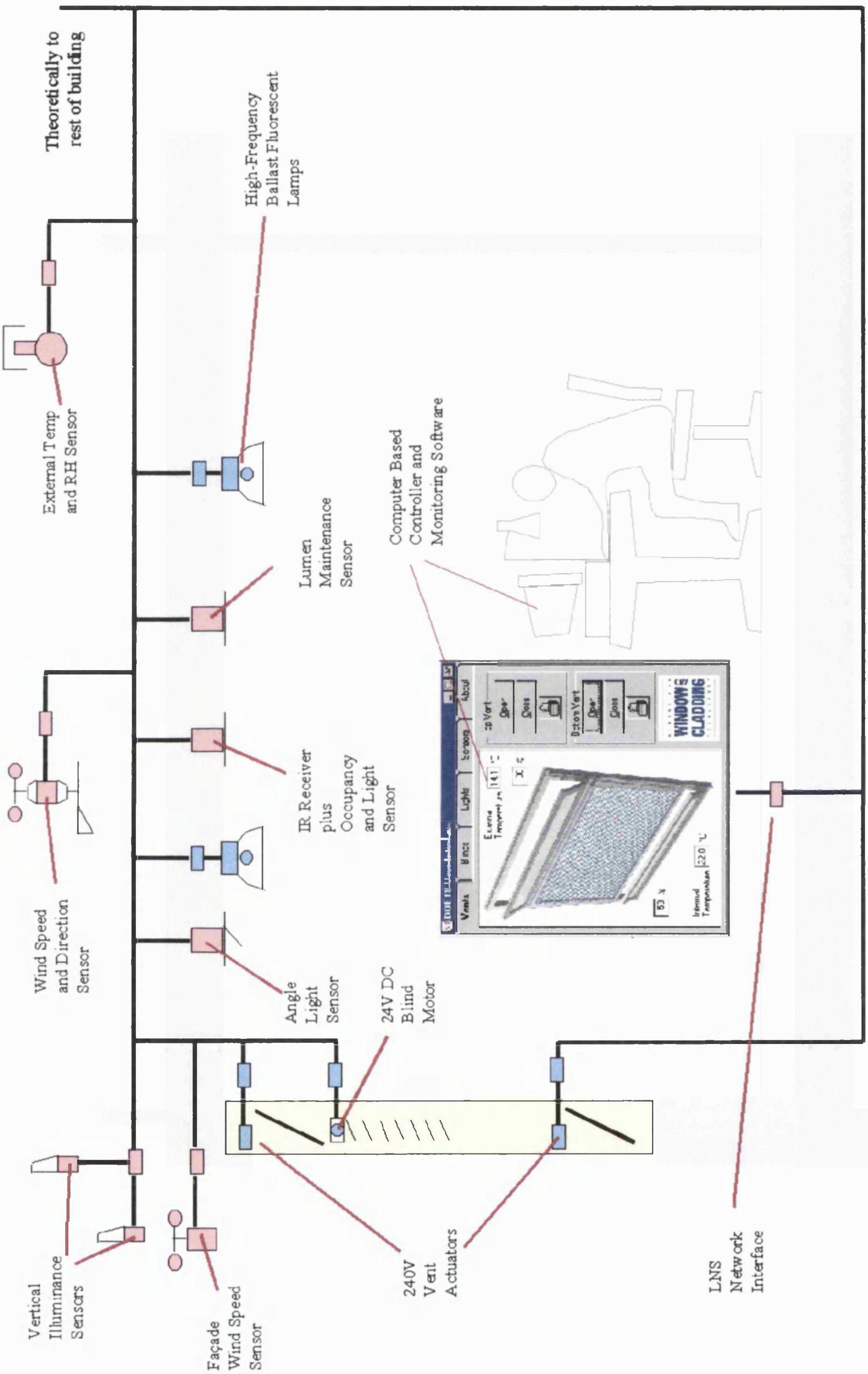


Figure 5.4: A schematic illustrating the location of each device and how they were connected on the LonWorks network using a bus topology.

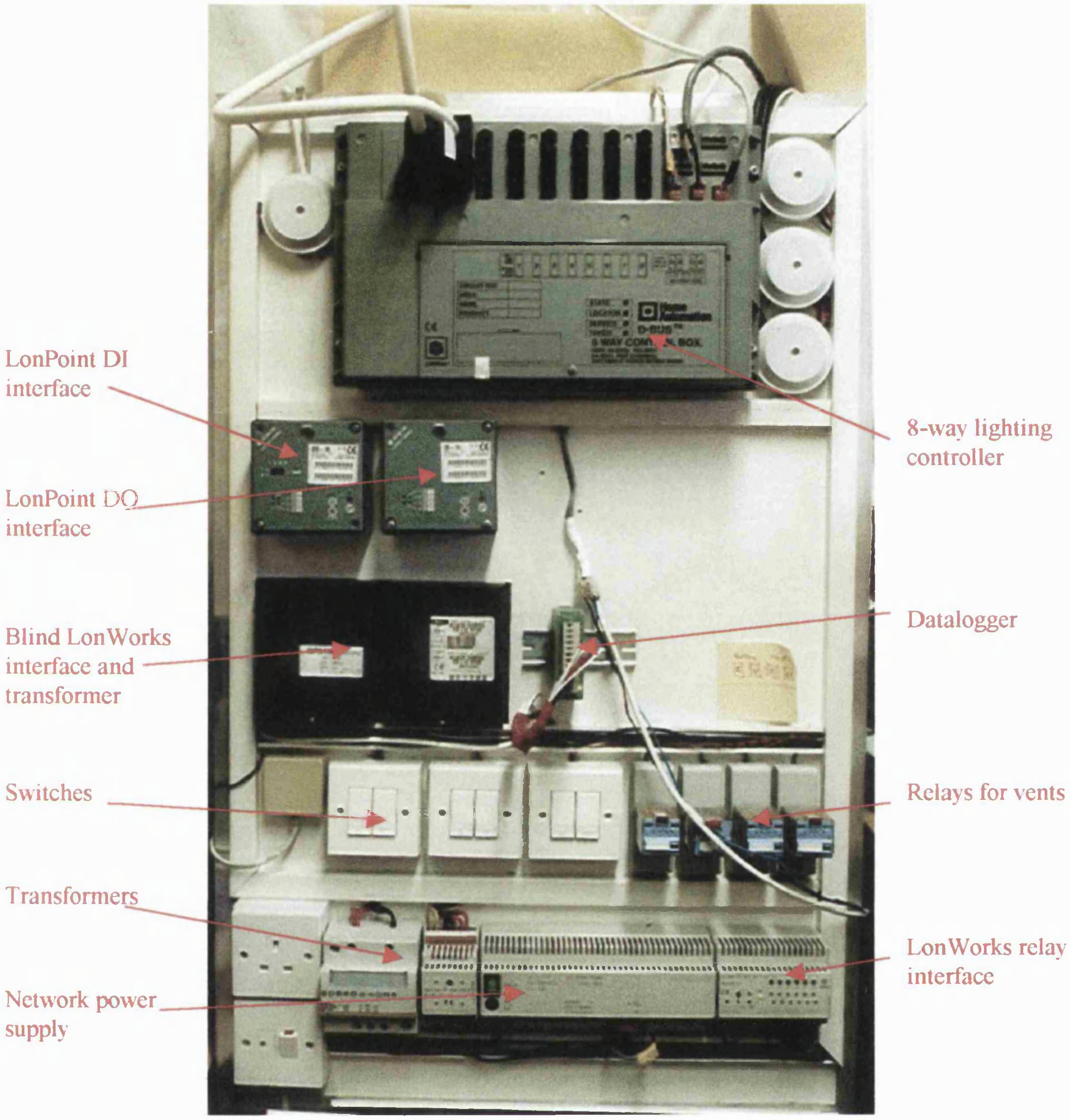


Figure 5.5: A picture of the control panel

The majority of the control network wiring and interfacing was located on a specially built control panel, situated next to the main desk in the room [see Figure 5.5]. The panel accommodated all of the LonWorks network devices that were needed in addition to the sensors and actuators themselves. These devices included:

- Power supply for the LonWorks network;
- LonWorks interfaces for some of the LonWorks compatible actuators;
- LonWorks interfaces and relays for the non-LonWorks compatible devices;

- Transformers;
- Manual over-ride switches for the vents and the blinds;
- Datalogger; and
- Link to the LonTalk PCLTA Network Card in the PC.

Although measuring 630mm wide by 1000mm the high panel was quite large, many of the devices mounted on it were only included for demonstration purposes and others were only commercially available with a large number of outputs. On a large-scale application, the control panel for each cellular office would be about a sixth of the size shown here and would be hidden from sight.

In addition to the manual switches included on the control panel, a hand-held Infra-Red controller and a computer based Visual Basic controller were also used for occupant over-ride. These devices will be discussed in more detail in the next two chapters.

Figure 5.6: A photograph of the test room weather station located on the roof.

Internal sensors included an internal temperature and humidity sensor, an occupancy sensor, a variety of different light sensors. A weather station was constructed on the flat roof directly above the test room [see Figure 5.6]. The weather station comprised of sensors for the measurement of wind speed and direction, vertical facade illuminance, external temperature and relative humidity.

More details of the test cell and the CWCT project can be found in Appendix VII.

5.4 The Hardware

5.4.1 The Automated Window

To incorporate the hardware necessary to carry out the work, the existing aluminium sash window in the office had to be replaced with a new thermally broken aluminium window system that could incorporate automated vents [see Figure 5.7].

Figure 5.7: A photograph showing the new window in place.

The window was approximately 1.7m high by 1.4m wide and consisted of three double glazed window lights; the upper and lower of which were bottom hung inward opening vents, measuring 350mm and 250mm in height respectively [see Figure 5.8]. For the purposes of the CWCT study, both of these opening vents were motorised using two separate 240V AC electric motors. Each motor actuated a scissor-stay opener via a drive rod and all the mechanisms were fixed to the window frame and finished in the same off white powder coating as the frame itself.

Solar protection was provided in the form of a mini venetian blind and was only needed to cover the large central glazed section of the window. This was because for the majority of the year the top light was protected from the sun by a large overhang just above the window and the bottom light was too low to solar penetration to affect the working plane. This arrangement ensured that the installed blind did not obstruct the opening and closing of the automated vents.

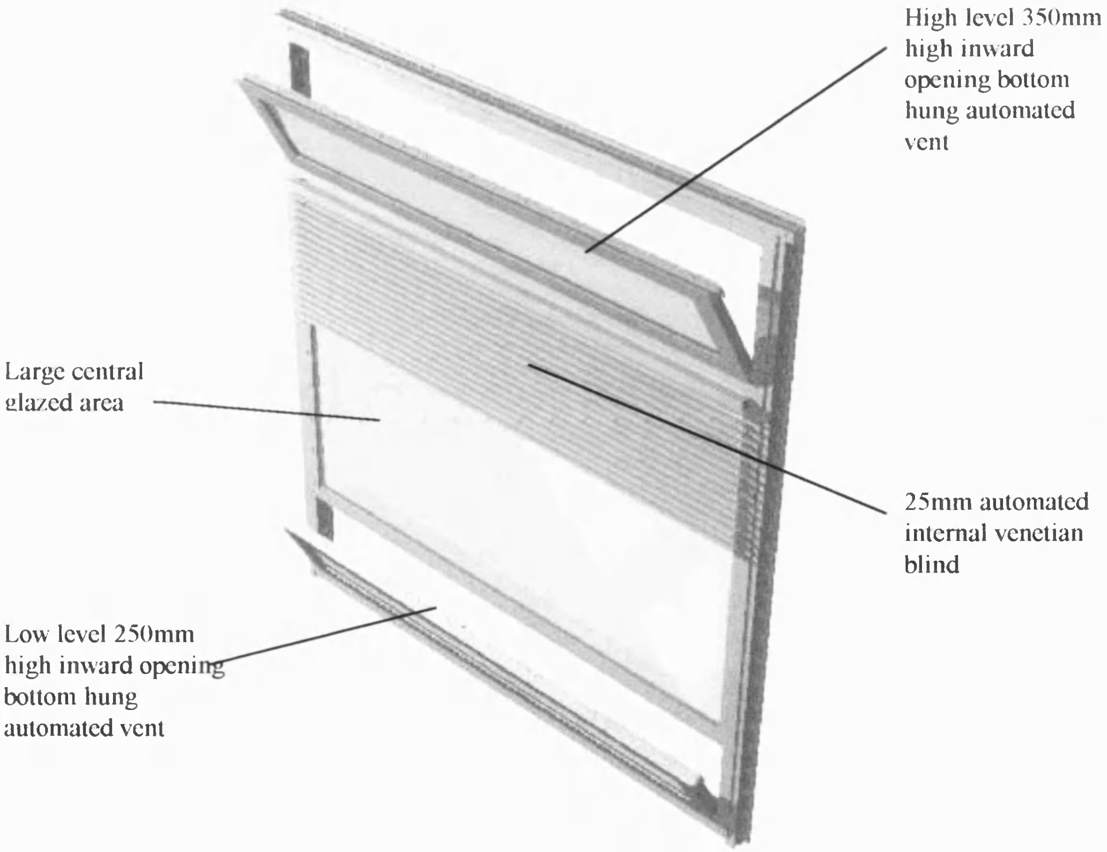


Figure 5.8: An image showing the configuration of the test room window

5.4.2 The Automated Blind Hardware

The automated blind system incorporated within the test cell comprised of a standard off-the-shelf motorised venetian blind, commercially available photo-sensors and sun sensors, and a LonWorks compatible control interface.

The blind itself was a slightly curved semi-specular white aluminium mini venetian blind with a 25mm slat width, a reflectivity of approximately 80% and a slat spacing to slat width ratio of 4:5. The blind was approximately 1m in height and was fixed to the inside of the window using a bracket designed to allow the blind to be removed easily for maintenance and cleaning.

When choosing the automated components for the blind, it was noted that a large number of the commercially available mini-blind systems, and the blinds used in the studies undertaken by others, did not provide any lift motion. However, for this study, which was carried out in a climate with a predominately overcast sky condition, it was thought that a system that incorporated both the lift and the tilt motion would be required if we were to try to cater for the majority of users' needs. Therefore a system was chosen that offered this dual functionality.

The blind head rail housed a small 24V DC tubular motor, which activated both the blind lift mode and the blind tilt mode [see Figure 5.9]. It achieved this through the use of a specially designed cord pick up mechanism that used the friction of the blind cords to move the blind tilt when the motor moved. This mechanism will be described in more detail in the next chapter.

The LonWorks interface, provided by the manufactuer and located on the control panel, held the default control algorithm within its Neuron chip (see Figure 5.10). It was programmed to follow a very simple threshold control algorithm that lowered the blind and positioned it at 45 degrees after a threshold in external illuminance was passed.

Figure 5.9: A picture of the 25mm blind used in the test room showing the small tubular motor located in the blind head-rail.

Figure 5.10: Photograph of LonWorks Blind Controller and Transformer

Two sun sensors were placed on the roof just above the window in a vertical position and were orientated perpendicular to the window facade [see Figure 5.11]. It was felt that these two sensors would provide a reasonable representation of the sky conditions experienced by the window. The input to the blind controller was determined by the sensors' LonWorks interface, which broadcast the higher reading from both of the sensors across the network to take into account over-shading [see Figure 5.12].

Figure 5.11: The simple vertical solar illuminance sensor supplied by the blind control manufacturer.

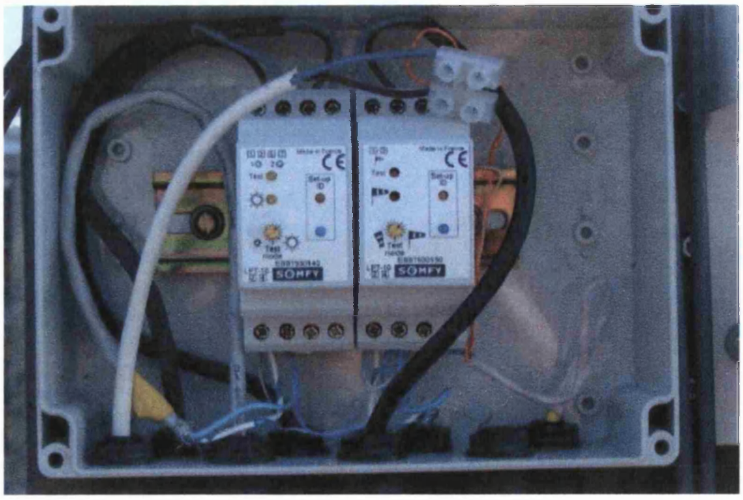


Figure 5.12: The LonWorks interface for the blind control manufacturer's sun sensors and wind speed sensors.

5.4.3 Lighting Hardware

The test room was fitted with a complete 'off-the-shelf' lighting system. The specification for the selection of that system was that it should be capable of:

- integration into a LonWorks network;
- switching on in response to occupant switching;
- switching off in response to changes in occupancy or occupant switching; and
- responding to changes in daylight levels and blind actions to minimise electric lighting output whilst maintaining a specified minimum working plane illuminance (say 500 lux).

The system selected included:

- two one-lamp fluorescent luminaires supported on the ribs of the coffered ceiling each incorporating a dimmable high frequency electronic ballast [see Figure 5.13];

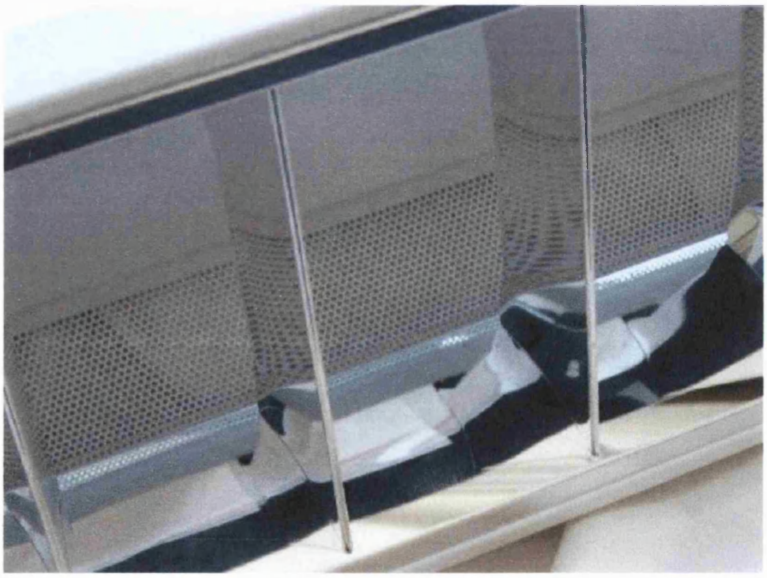


Figure 5.13: The luminaires used in the test room. They included mirrored reflectors to limit glare and a perforated casing to allow a proportion of light to be directed upwards.

- a lighting controller that processed the signals received from the photo-sensors to adjust the control voltage across the electronic ballasts [see Figure 5.14];

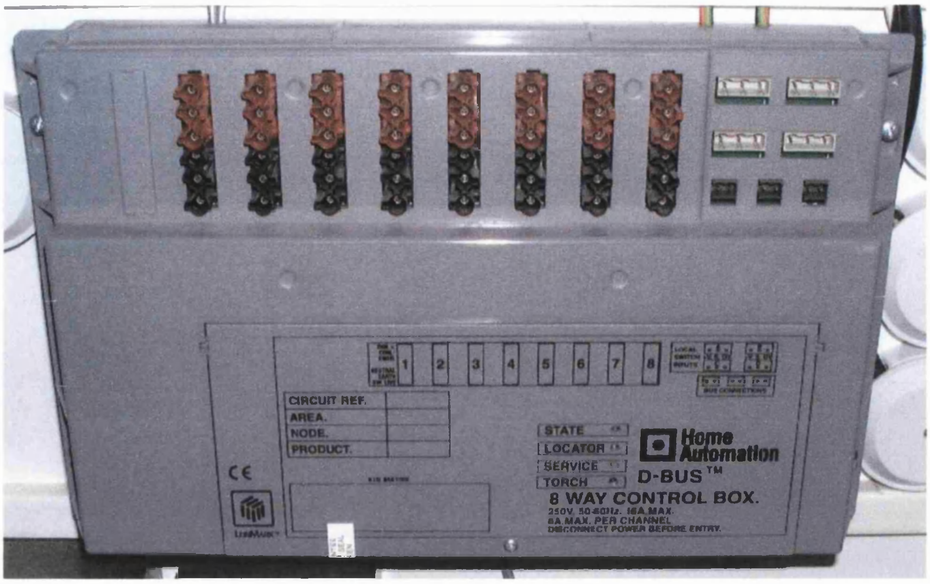


Figure 5.14: The lighting control box used to modulate the voltage to the high frequency ballasts. A standard 8-way box was used despite only needing two sockets.

- a series of photo-sensors that simply measured the amount of daylight entering the room and whether the lighting levels were above the design threshold; and
- an occupancy sensor - for 'lights off' occupancy switching.

The key elements in the system were the three sensors that were mounted on the ceiling of the test room; one of which was shielded, the other two unshielded.

The shielded sensor was an open loop sensor positioned just behind the blind with its sensing element facing the window. The purpose of this sensor was to provide the system with a representation of the amount of daylight coming into the room from the window. The photocell in the sensor generated an electrical signal proportional to the amount of light sensed. This signal was then processed within the sensor's LonWorks Neuron Chip and then broadcast across the network as a light level network variable. No feedback was available to this sensor as it was largely shielded from the electric light.

Figure 5.15: The shield window sensor, or angle sensor. Unfortunately the shielding was not totally successful as the hole in the casing for the service LED let light through to the sensing element. This affected the control of the lights so as a result a piece of paper was fitted to shield the sensor from the direct light from the lamp.

The signal from the shielded sensor was sent to one of the unshielded sensors, used to measure the illuminance on the working plane. This sensor incorporated a closed loop control algorithm that performed a number of tasks. Firstly it used the reading from its own photocell to control the light output of the lamp furthest away from the window by sending control signals to the lamp controller box. Secondly it used the shielded sensor's reading to modulate the light output of the lamp nearest the window. The algorithm linked both of these functions to produce the correct illuminance on the working plane and used a configurable relationship to control the lamp outputs according to the depth of daylight penetration into the room. This meant that the lamp nearer the window was always dimmer than the lamp further away from the window during the daylight hours. This sensor, which also had the ability to perform lumen maintenance functions, was referred to as the 'Lumen Maintenance Sensor'.

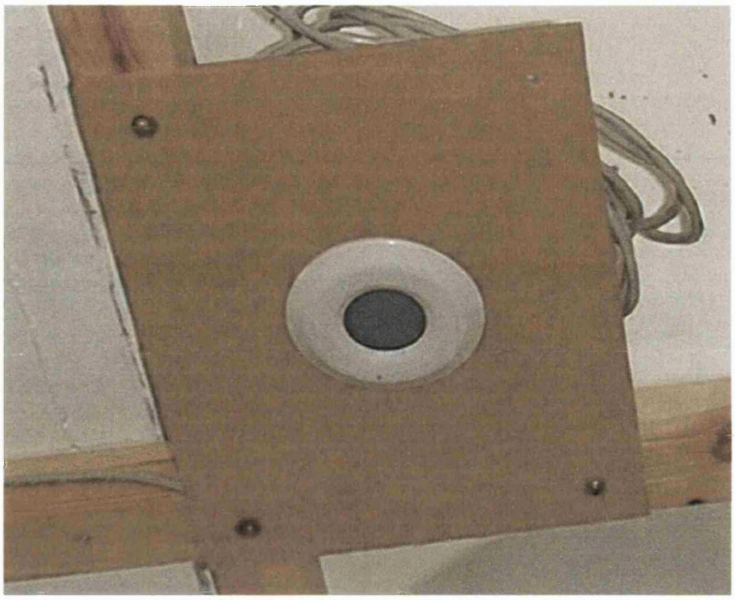


Figure 5.16: Photo of the Lumen Maintenance Sensor

The third sensor, also an unshielded sensor, was called the 3-in-1 sensor because it performed three functions (see Figure 5.17):

(i) occupancy sensor – for occupancy switching;

(ii) infra-red receiver - picked up signals from the hand-held IR controller; and

(iii) light sensor - used in this case primarily for monitoring purposes and not for control.

Figure 5.17: The 3 in 1 sensor with its Fresnel lens for improved occupancy sensing.

5.5 The Software

5.5.1 Virtual Control

In addition to its flexible network management functionality, one of the primary reasons for choosing LonWorks as a controls framework was its ability to link seamlessly with the computer software environment. This fact meant that:

- the test room control system could be installed and adjusted, using windows computer based software; and
- much of the control logic supplied with the equipment by the manufacturers could be by-passed by writing a Visual Basic program that was able to control much of the system.

Therefore the computer located in the test room not only provided a tool for the occupant to perform their everyday computing tasks, it also provided an interface with the LonWorks network for the purposes of network management, monitoring and control.

Although implementing the control algorithms on the PC using Visual Basic went against the decentralised idea of LonWorks, it gave the author the ability to develop control strategies on-line and facilitate the fine-tuning of the control system response, set points and control logic. In practice, updating the application code in the Neuron chips of the LonWorks devices would mean developing the Neuron C code, debugging it and then down loading it to the chip. This is an expensive and time-consuming process, and was outside the budget of this study, as it required a large initial investment in the equipment to download the programs to the LonWorks chips.

Through the use of dynamic data exchange, the visual basic program had the ability to process inputs from sensors (in the form of sensor object network variables on the control network) according to various control algorithms and produce appropriately scaled outputs to drive actuators (such as the blind motor using various actuator network variables). Therefore it was possible to create a virtual PC based LonWorks network, on top of the real LonWorks network, that had separate function blocks for each device and used network variables to communicate between those blocks.

This gave the author the opportunity to develop the blind control logic from first principles, and the flexibility to implement and test a variety of control strategies in software without the need to construct physical electronic circuits. This process is described in Chapter Seven.

The software used in this project to provide this functionality included:

- *LonMaker for Windows*
- *LNS DDE Server*
- *Visual Basic*
- *LNS plug-ins for certain items of hardware*

LonMaker provided the network management tool for commissioning and monitoring the network. The LNS DDE server provided the ability for the windows environment, i.e. Visual Basic and Excel, to communicate with the control network. Visual Basic provided the ability to write programs for controlling the window and design computer user interfaces. The LNS data logger plug-in software combined with Excel to collect data on the test cell's performance.

5.5.2 LonMaker for Windows

LonMaker for Windows is a network management tool that provides the ability to bind and manage nodes on a LonWorks network. It uses the Visio drawing package as a graphical interface, enabling a simple 'user friendly' drag and drop functional block diagram to be used [see Figure 5.18].

The LonMaker tool functions are as follows:

- *Network Design Tool* – allows network design without being connected to a network (off-site) or whist being connected to a network (on-site);

- *Network Installation Tool* – allows a network to be rapidly installed using a series of plug and play features that reduce commissioning time. The LonMaker browser provides complete access to all network variables and configuration properties;
- *Network Documentation Tool* – creates a Visio drawing in parallel with the network design and installation process, which can be used to represent the installed network;
- *Network Maintenance Tool* – allows devices, routers, channels, subsystems and connections to be easily added, tested, removed, modified or replaced to support system maintenance. The tool also gives the operator the ability to enable, disable, wink, reset, test and log devices as well as the ability to monitor network variables and messages in real time.

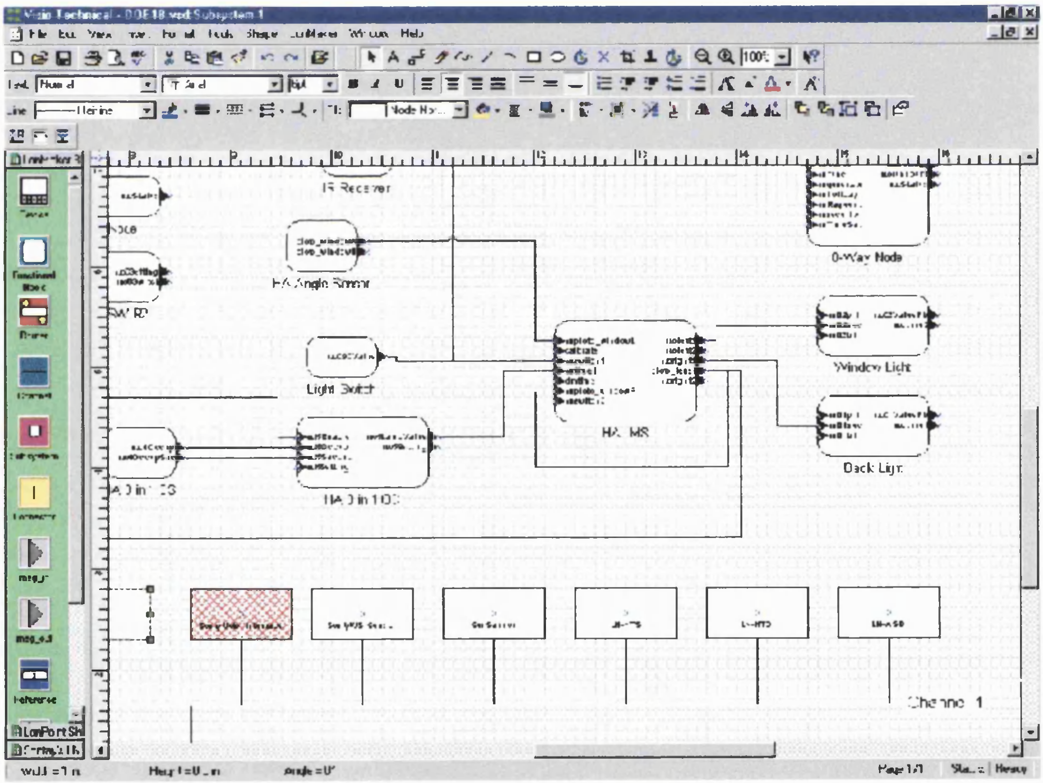
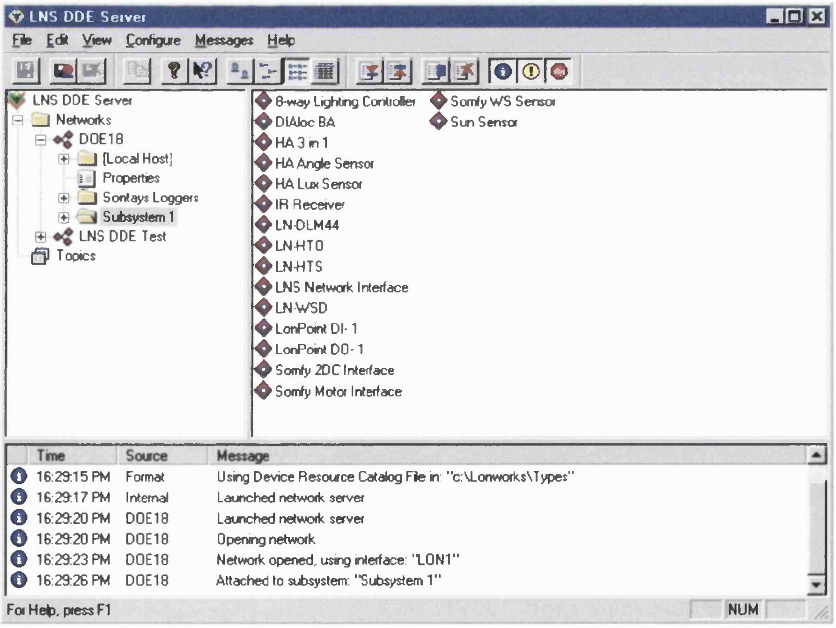


Figure 5.18: Screenshot of the LonMaker for Windows Software, within its Visio environment.

5.5.3 The LNS DDE Server

Dynamic Data Exchange (DDE) defines a standard way for Windows applications to share information with one another. The LNS DDE Server provides a LonWorks network with the ability to integrate seamlessly with standard Windows programs such as Excel, Visual Basic, Access etc.

When applications share information with each other using DDE, they are said to be having a DDE conversation. Each conversation has a well defined beginning, middle and end. To begin a conversation, one application (the client or destination application) asks another application (the server) to open a communication channel (different to a LonWorks channel). Once the conversation is established, the client can send and receive data from the server on the DDE channel. For example, an Excel spreadsheet may ask the LNS DDE server for the current temperature outside for an automatic calculation procedure. This is very similar to the way the Neuron chips talk to one another. Links can be set up easily by copying a variable in the LNS DDE Server User Interface and then pasting it in the target application (see Figure 5.19).



```
Time         Source    Message
16:29:15 PM  Format    Using Device Resource Catalog File in: "c:\Lonworks\Types"
16:29:17 PM  Internal  Launched network server
16:29:20 PM  DOE18     Launched network server
16:29:20 PM  DOE18     Opening network
16:29:23 PM  DOE18     Network opened, using interface: "LON1"
16:29:26 PM  DOE18     Attached to subsystem: "Subsystem 1"
```

Figure 5.19: A screenshot of the LNS DDE server (similar in format to the Windows Explorer)

The LNS DDE Server can be used as a driver for many popular Human Machine Interface (HMI) and Supervisory Control and Data Acquisition (SCADA) packages. In this project Visual Basic is used with the LNS DDE Server to enable a Visual Basic program to run the application code for the window.

5.5.4 Visual Basic

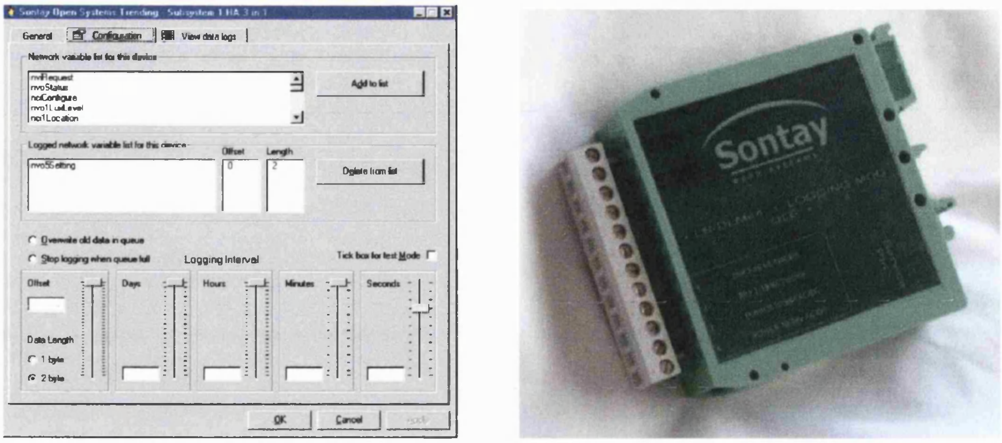
Visual Basic is an object orientated program language for designing Graphical User Interfaces. It provides the user with the ability to create programs within standard Windows forms that utilise components such as buttons and text boxes, simply by dragging and dropping those components from a standard template file. As with any other programming language, Visual Basic can also be used to control program flow with a series of 'IF...THEN' rules and program timers.

In this project, Visual Basic was used with the LNS DDE Server to create a program that could control the window blinds and vents from the computer. The program developed also allowed the occupant within the space to control the automated window through the computer and view data from the system in real time (see Appendix VII).

5.5.5 The LNS Plug-ins

A LonMaker LNS Plug-in for a data logger combined with Excel to provide the ability to collect data on the test cell's performance (see Figure 5.20). The plug-in had two main functions, firstly the configuration of time intervals and the selection of the measured variables, and secondly the management and viewing of the downloading process. The downloaded format allowed the data to be filtered and managed in either a database program, such as Access, or a spreadsheet program, such as Excel.

Another LNS plug-in incorporated within the test cell was the configuration and Logging tool for the Sontay LN-HTS internal humidity and temperature sensor. This plug-in allows the user to adjust the configuration properties using a simple graphical interface. It also displays current network variables and plots a real-time graph of temperature and humidity readings.



Sontay Open Systems Trending - Subsystem 1 HA 3 in 1

General | Configuration | View data logs

Network variable list for this device:
- nviRequest
- nvoStatus
- nciConfigure
- nvo1LuxLevel
- nci1Location

Logged network variable list for this device:

Variable	Offset	Length
nvo55etting	0	2

○ Overwrite old data in queue
○ Stop logging when queue full

Tick box for test Mode ☐

Logging Interval: Offset, Days, Hours, Minutes, Seconds

Data Length
○ 1 byte
● 2 byte

Figure 5.20: Left – Screenshot from the Data-logger LNS Plug-in, Right – A picture of the Data-logger

5.6 Study Methodology

The initial installation and configuration stage of the test room study commenced in January 1999 and was spread over a period of four months. This was slightly later and longer than initially planned due to delays with obtaining equipment from the hardware suppliers, many of whom were giving their components free of charge for the research and had other commercial business to which they had to give priority. However, during this time, the author was able to become familiar with the workings of the installed hardware, the LonWorks technology and the software needed for the latter stages of the project.

Once commissioned, the author used the test room as an office for a further nine months until the end of January 2000. This enabled further fine-tuning and experiments to be carried out with the system operating in the background and data was collected on the external variables to which the system was responding, the internal variables that it affected and other information pertaining to the status of the blind and window. This data was sampled every minute and placed into Excel charts for analysis.

Throughout the period of the study a series of blind control strategies were implemented and tested. At first, the simple control strategy supplied with the blind controller was used. Then gradually, over the year, a prototype system was developed by refining the control system algorithms and hardware configurations using the recorded data, the real time data display and the author's own subjective responses. As part of that process, additional system parameters were added in an attempt to provide some of the extra functionality identified in Chapter Three and to find the right balance between system complexity and usability.

Further investigations focussed on practical issues related to blind control, such as how the control was affected by the technological constraints of the blind mechanism and the sensitivity of the sensors. Observations were also made on the factors leading the need to over-ride the control system due to solar glare and how they relate to some of the lessons learnt in Chapter Two.

5.7 Chapter Summary

The test room was set up successfully and provided a sound base for the range of studies outlined at the end of Chapter Four. The set-up objectives of the study were to:

- examine how the design of the dynamic envelope and lighting system affects its integration within the whole building infrastructure (practical issues);
- demonstrate the potential of new, emerging technologies by constructing a working prototype of an automated venetian blind and provide direct links between research and construction implementation;
- evaluate how well current venetian blind systems satisfy defined control strategy goals in real-time under variable sun and sky conditions;
- observe at first hand factors that influence blind control;

The study successfully achieved all these goals. The next four chapters cover the complete findings of the study.

Chapter 6: The Initial Findings of the Test Room Study

6.1 Introduction

This chapter, reviews some of the lessons learnt from the initial process of setting up and running the test cell. In particular, it deals with the findings related to each of the practical issues identified at the beginning of Chapter Five as influencing the development of blind control algorithms, namely: manageability, interoperability, usability and sensitivity.

The test room study attempted to tackle many of these issues by implementing the open field bus structure of LonWorks as the backbone of the control framework. This chapter assesses the performance of both the LonWorks technology and the installed hardware, whilst also reviewing the structure and limitations of the proprietary blind control system installed.

Chapters 7 and 9 go on to describe the iterative process used to extend this work and implement improved blind control algorithms on the existing system, whilst Chapter 8 covers the individual user experience gained during the whole study. All four of these Chapters describe the knowledge acquisition process that was key to the goals and methodology proposed.

6.2 Manageability

This part of the study provided a starting point for understanding the importance of the balance between complexity and usability within control system design and helped ensure that the control framework that was being proposed struck that balance.

If a building control device is to be practical, it must be capable of being managed effectively both during the installation and commissioning processes and once installed. Although the test room system was not large enough in scale to allow a thorough investigation into building control system manageability, it did provide a significant insight into the procedures used by, and the requirements made on, building control equipment and their application to a single control device. Indeed the technology and procedures used in the study were identical to the technology and procedures that would be used on large installations and therefore a noteworthy amount of knowledge was gained by simply studying individual device functionality.

6.2.1 Installation

The installation of the control components was undertaken using the LonMaker software (described in Chapter 5). Although there were certain commercial restraints associated with this software (discussed in Appendix VI), its installation procedures utilised simple drag and drop functionality and a series of installation wizards, which were shown to be simple, flexible, fast and effective to use. A detailed description of the installation procedure can be found in Appendix VI.

The applications in each device were divided into one or more functional blocks, each containing different network variables and configuration properties. Following the installation of the devices, the control system functionality needed to be set up by connecting, or binding, the network variables of these functional blocks together to create one control task (See Figure 6.1).

The binding process, which was also fairly straightforward, simply involved dragging a connection shape from the stencil until one end sat over the first network variable requiring connection, the unconnected end was then dragged to a second network variable of the same type and the connection was made. Once connected in software the LonMaker tool then instructed the Neuron chips in both devices to talk to one another and the devices became connected in hardware.

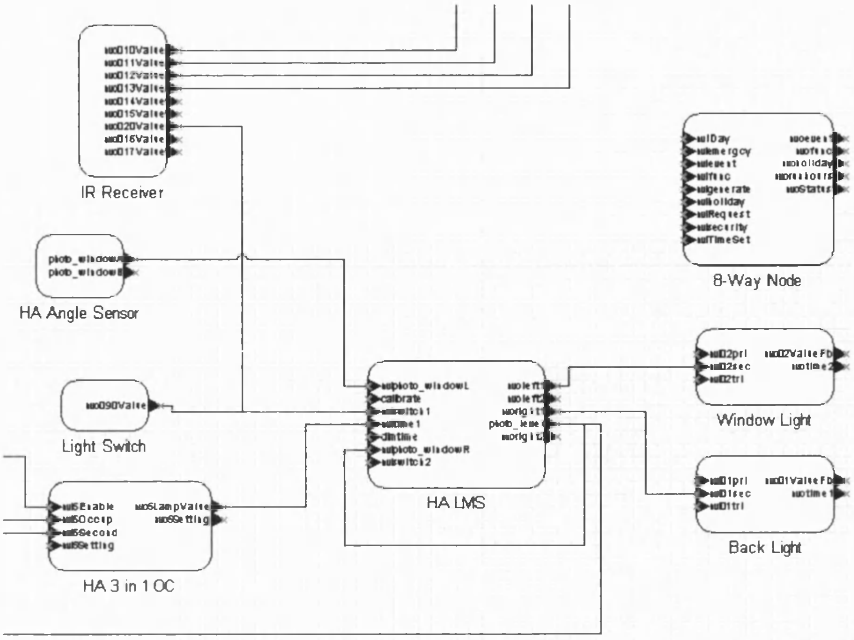


Figure 6.1: Screenshot showing functional blocks from the test cell bound by logical connections.

The real strength of the LonWorks LNS system seemed to be its ability to allow the installer to design the network without being connected to it. Once the network was attached, a simple commissioning step could associate the LonMaker device shapes with the physical devices and load the pre-defined configuration properties into the devices.

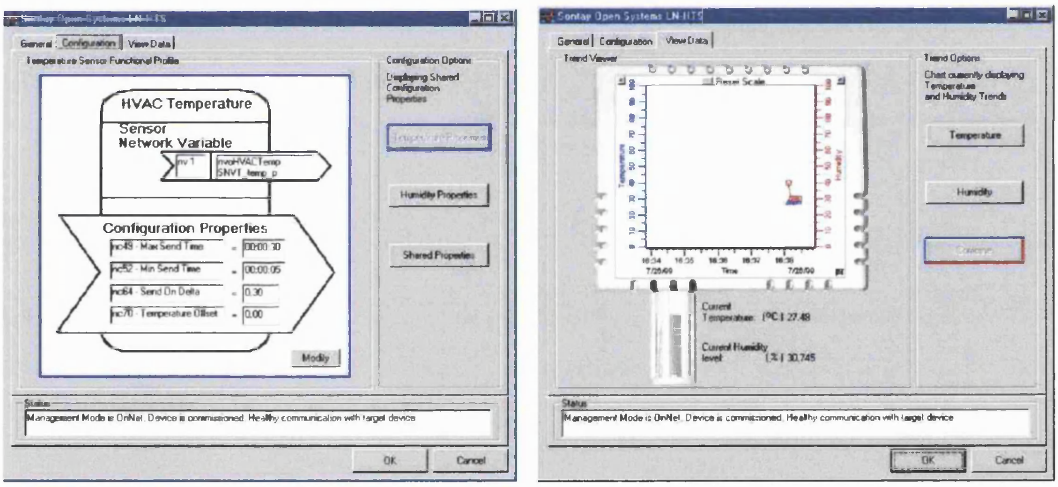
In addition, the flexible nature of the technology also allows the manufacturer to meet individual specifications for particular projects by either customising the configuration properties in each device before shipping them to site, or by providing the installer with the bespoke application files on disk to download during the installation.

The advantage of all these methods is that, since most of the time consuming actions, e.g. problem solving, data entry and processing is done off-site, the network commissioning on-site, is very quick, easy, and error free. This effectively provides a 'plug and play' installation procedure that can greatly reduce on-site costs and has beneficial implications for the development of the control algorithms that incorporate a few additional global configuration properties, such as solar positioning algorithms.

6.2.2 Commissioning

Despite the ideal concept of pre-commissioned ‘plug and play’ technology, most systems require some form of commissioning to be undertaken before they are handed over to a client fit for purpose. This process involves the fine-tuning of configuration properties to suit the specific device context and the adjustment of network variables to ensure the devices are functioning satisfactorily.

In a LonWorks system, a LNS ActiveX Plug-in can be used to simplify this commissioning process. Examples of plug-ins used in the study include the Sontay Datalogger plug-in, described in the last chapter, and the Sontay LN-HTS Internal Temperature and Humidity Sensor plug-in, shown in Figure 6.2. Both were 'easy to use' configuration interfaces with simple 'click and go' Visual Basic controls that were also supplied with the ability to provide real time trending of device data that could be used to ensure the devices were functioning correctly.



nc49 - Max Send Time = 00:00:30
nc52 - Min Send Time = 00:00:05
nc64 - Send On Delta = 0.30
nc70 - Temperature Offset = 0.00

Current Temperature: [°C] 27.48
Current Humidity level: [%] 30.745

Management Mode is OnNet. Device is commissioned. Healthy communication with target device

Figure 6.2: The Sontay Open Systems LN-HTS plug-in; left – the configuration window; right - the data trending window.

6.2.3 Network Management

The LonMaker software also provided a simple and user-friendly means of managing the control network and an effective means of diagnosing problems on the network or monitoring the systems performance. The network management functions included the ability to:

- add new devices;
- move and replace existing devices;
- replace device application code;
- test, disable (to isolate network problems), enable, reset and wink devices;
- obtain status summaries;
- generate reports of network problems;
- monitor devices in real time by using LNS text boxes, monitoring connections or the LonMaker Browser;
- visually display network errors using colour coding of devices to allow the network manager to quickly identify any node errors and communications errors etc (see Figure 6.3); and
- back up the system database and retrieve information if things go wrong.

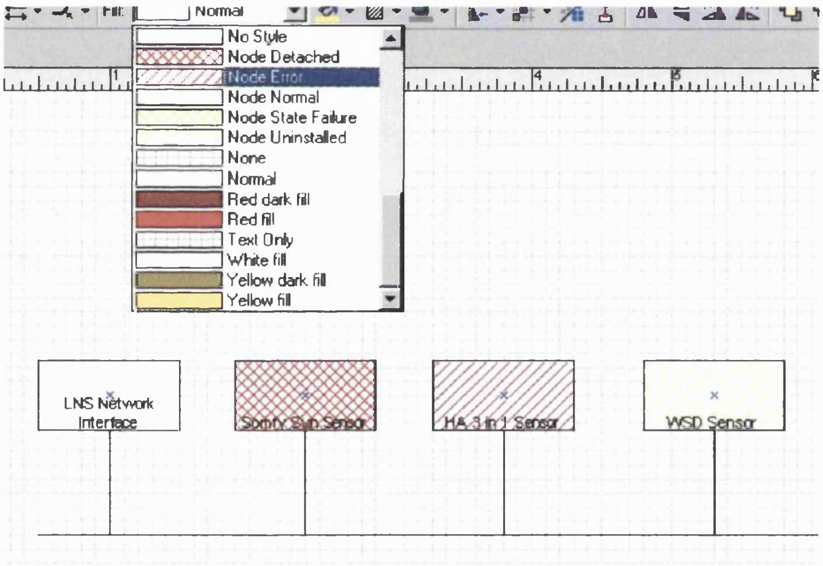


Figure 6.3: Screenshot showing colour coding of network errors on LonMaker visual interface

All of the management functionality discussed in this section was built into the LonWorks protocol embedded within the Neuron chip. Therefore to obtain this functionality, LonWorks compatible blind controllers do not require any special programming from the product manufacturer as long as they implement the LonWorks protocol in the form of the chip.

In terms of the future development of automated blind systems, the primary advantage of identifying a system with such advanced management features is that strategies that take into account a few additional easily pre-programmed variables can be considered without jeopardising the balance between complexity and usability. However, despite these advantages it is still clear that the highly complex systems proposed by others and reviewed in Chapter Four, would still require large amounts of site specific design and commissioning time and therefore are not suitable.

6.3 Interoperability

One of the primary reasons for implementing a field bus technology, such as LonWorks, within this study was the promise, made by many, of an open control network structure that supports multi-vendor interoperability. In the LonWorks system, this interoperability is achieved through the use of Standard Network Variable Types (SNVTs) and Standard Configuration Property Types (SCPTs) defined by a LonWorks user group named The LonMark Interoperability Association. This group also defines a series of standard functional profiles that describe which SNVTs and SCPTs should be applied to a list of common devices. These standard data types, along with any manufacturer specific data types, determine the content and structure of the data being broadcast across the network. By ensuring that each device uses the same data type, interoperability can be assured.

In the test cell, LonWorks devices from a number of different manufacturers were successfully integrated. This integration also extended to non-LonWorks compatible devices through the use of LonWorks DI and DO (Digital Input and Output) modules. For example the automated vent actuators were added to the LonWorks network using a DO module and a Relay module. The DO module translated the digital signal created by the vent switches to a series of SNVT_switch signals. These signals were then broadcast over the LonWorks network and received by the relay module, which switched relays to control the vent actuators. The SNVT_switch output from the LonWorks IR controller was also used to switch these vents.

The demonstration of this interoperable technology meant that the proposed control framework would be able to include network variable information from a variety of devices, such as occupancy sensors, internal light sensors and temperature sensors, that already exist within high specification buildings, to gather additional information about the system's environment, if required.

6.4 Usability

So far this chapter has demonstrated that, in terms of manageability, the LonWorks networking technology was relatively simple and user friendly. In this section, we consider the performance of the links between the everyday user and the control system via the user interface or over-ride. This link is vital to the success of a system that responds to individual user's needs. Initially two types of user interface hardware were installed in the test room, which were manual switches and a hand-held Infra-Red controller.

6.4.1 The Manual Switches

The manual switches provided a basic, but effective, form of user control and override where the degree of component movement was proportional to the period of button activation. The simplicity of the interface made it reasonably self-explanatory and user friendly as long as each switch was labelled clearly and correctly and the system provided both an appropriate degree of control and a rapid response to any occupant request.

The one disadvantage of the switches was that the occupant had to leave their desk to activate them. In a true application, it may be possible to locate the switches nearer to the user but this will be dependent on having some appropriate wall or partition space, which is often not available.

6.4.2 The Hand-held Infra-Red Controller

The hand-held IR controller, shown in Figure 6.4, provided the occupant with the option of overriding the system from anywhere within the room, thus offering more flexibility than the manual switches. The controller used in this study had 8 IR channels that could each send either a switch or a scene setting network variable signal. This meant that a maximum of four devices could be controlled with open/close or on/off signals, or one or more devices could be controlled with up to eight pre-set scenes.

Figure 6.4: The Infra-Red Controller

When tested in switch mode, the controller allowed control of each vent separately, the blinds and the lights. Despite experiencing noticeable time delays when switching, the system performed adequately when controlling the raising and lowering of the blind. However, the system was not satisfactory when attempting to fine-tune the tilt angle of the blind, which required a fast response to provide the occupant with visual feedback.

When tested in scene mode, the IR controller controlled both the blinds and the lights with four scenes each. The use of this mode did not seem to offer many advantages to the user, apart from being able to position the slats of the blind accurately at pre-set angles and made the control interface slightly inflexible.

Other disadvantages of the IR controller included the fact that the controller was prone to being lost or removed from the correct control location and the fact that it required batteries, which need to be recharged.

6.5 Sensitivity of The Hardware

This part of the study allowed the author to experience first hand some of the capabilities and practical limitations of hardware typically found in the marketplace. It concentrated on the sensor technology available to provide sensory input to a system, the lighting technology that the system must interact with and complement, and the blind movement hardware itself.

6.5.1 The Sensors

The photocell sensors supplied with the automated blind and dimmable lighting systems were reasonably simple in their design and manufacture. This meant that the overall operation of photo-sensors was largely dependent on the distribution, spectral composition and intensity of light within the field of view and the adjustment setting on the commissioning controls. Therefore great care was taken when positioning and calibrating the sensors to ensure the system would perform satisfactorily.

The Sun Sensors

The photodiode in the simple, low cost sun sensor was housed within a white plastic covering that diffused the daylight radiated onto it. This cover eliminated a certain degree of light distribution and directivity and providing an overall illuminance level for the sensor-facing hemisphere. To compensate for this, both the sun sensors were positioned vertically, facing perpendicular to the window, to provide the system with an idea of the solar illuminance on the window. The two sensors were positioned far enough apart on the roof to ensure that if the one sensor was in shade the other could still provide an accurate representation of the sky.

The sensors were limited to a range of 3,000 to 55,000 lux (see Figure 6.5), but values outside this range did not seem to affect the system's ability to control the blind's raise and lower functions. This is because illuminance values above approximately 24,000 lux were considered to characterise a bright sky and generally values below 5,000 lux were considered to characterise a dull sky. Indeed the sensors provided a reasonably good representation of the illuminance on the vertical window plane in real time and their performance was deemed adequate for the open loop control of the blinds. Therefore it was shown that expensive scientifically calibrated sensors, often used in research, were not necessary to meet the control objectives of the simple threshold controllers, as first thought.

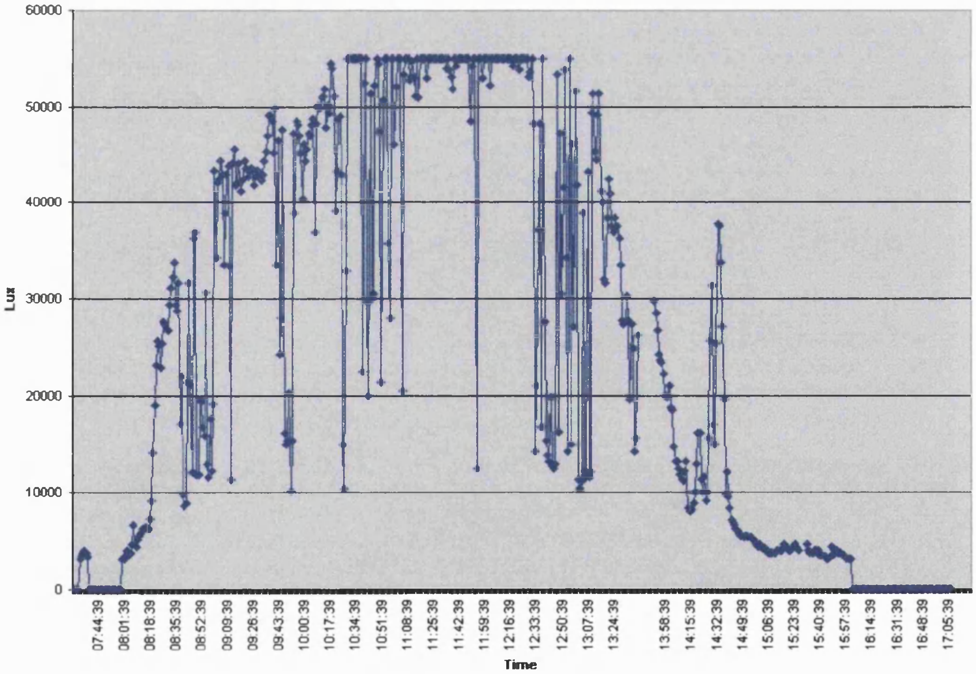


Figure 6.5: Graph showing an example of the illuminance readings provided by the solar illuminance sensor. This data was logged at 1 minute intervals on the 9/11/99.

The Internal Light Sensors

The positioning of the internal lumen maintenance sensor that controlled the daylight linked lighting system was a slightly more difficult task. The closed loop lighting control strategy was highly dependent on the magnitude of the signals produced by the sensors, and studies by others had demonstrated that the incorrect positioning of these sensors had a great effect on the energy efficiency of the overall system.[87] [88]

After consulting the lighting control manufacturer it was decided to follow guidance outlined in a LBL study, which identified the best results achieved by daylight linked control in a small office were witnessed when a closed loop sensor is located just above the occupant's desk.[89]

Once the internal light sensors' placement had been determined, adjustments were made to each sensor's configuration properties, using the LonMaker software, to ensure that they provided a good representation of the illuminance on the working plane directly below them. Unfortunately, the lighting control manufacturer was unable to provide any clear instructions on how this calibration procedure should be undertaken, therefore guidance again had to be taken from a LBL research paper.[90] It was discovered that the daylight conditions in which a sensor was calibrated greatly affected the accuracy of that sensor in varied conditions. Therefore calibration was achieved by comparing the reading given out by each sensor to the illuminance measured at the working plane by a handheld photometer during clear sky conditions. Then this measurement was fine-tuned by taking supplementary measurements during a variety of sky conditions until an acceptable level of accuracy was achieved.

---

[87] Bierman A. and Conway K., 2000, "Characterising daylight photo-sensor system performance to help overcome market barriers", *Journal of the Illuminating Engineering Society*, Winter 2000, pp 101-115.

[88] Mistrick R. et al., 2000, "A comparison of photosensor controlled electronic dimming systems in a small office", *Journal of the Illuminating Engineering Society*, Winter 2000, pp66-80.

[89] Mistrick R. and Thongtipaya J., 2000, "Analysis of daylight photocell placement and view in a small office", *Journal of the Illuminating Engineering Society*, Summer 1997, pp150-160.

[90] Lee E.S., DiBartolomeo D.L. and Selkowitz S.E., 1999, "The effect of venetian blinds on daylight photoelectric control performance", *Journal of the Illuminating Engineering Society*, Vol. 28, No.1 1999.

Despite these teething problems, the sensors were eventually calibrated to provide a satisfactory level of performance as can be seen from the data in Figure 6.6. However, the difficulties experienced helped illustrate the dangers of relying on internal sensors to control daylighting and lighting systems in a closed loop manner.

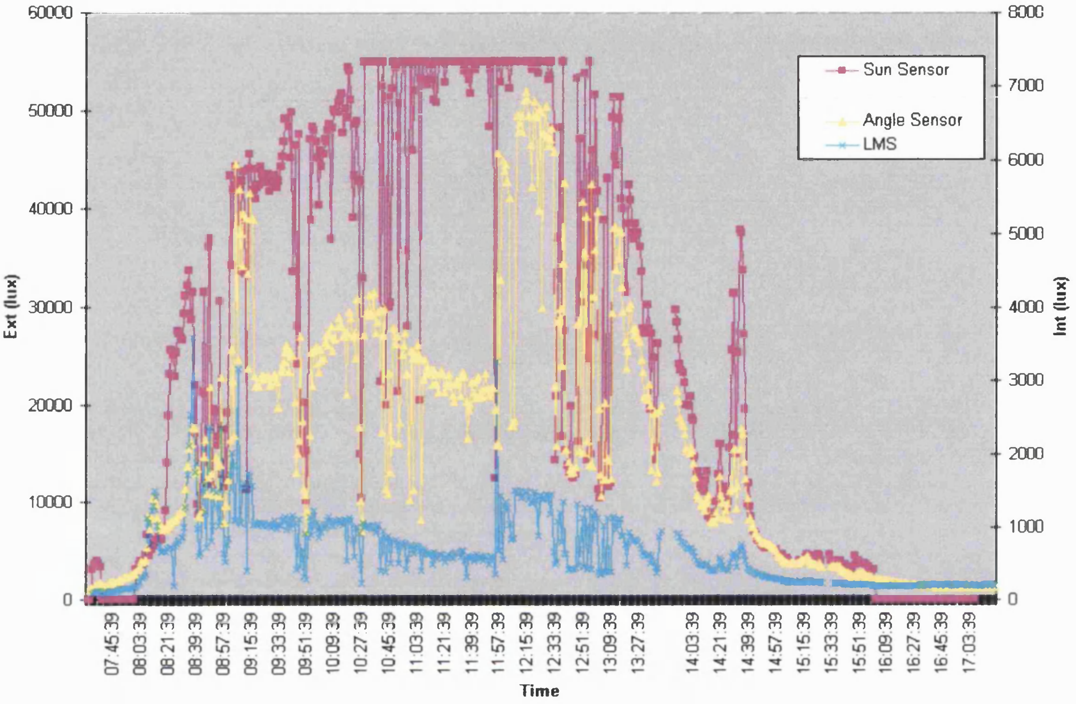


Figure 6.6: Graph showing how the illuminance readings from the internal illuminance sensors (right hand scale) and external sun sensors (left hand scale) on the 9/11/99. The dip in internal illuminance readings is due to the blinds coming down.

The Occupancy Sensor

Due to the size of the test room, it was difficult to position the occupancy sensing element of the 3 in 1 sensor in a position that avoided casual tripping by passers by. To aid this, a curtain was placed over the test room entrance, which could be closed during the commissioning and testing stages or when the space was unoccupied.

During the study it soon became clear that the effective use of occupancy sensing for the purposes of control and on-off switching relied on a balancing act between device sensitivity and off-delay time. The sensitivity had to be adjusted to ensure that relatively small movements by occupants resulted in an occupancy signal, whilst movement triggered by gusts of wind were ignored. By reducing the sensitivity slightly and lengthening the off-delay time, this balance could be achieved without compromising the performance of the system.

The off-time delay should not be set too high, as this could result in a large reduction of energy savings, but should be set high enough to allow the lights to remain on if the occupant just popped out of the room for a moment. The factory default off-delay time was ten minutes and this seemed to perform adequately when used to control the light output (see Figure 6.7).

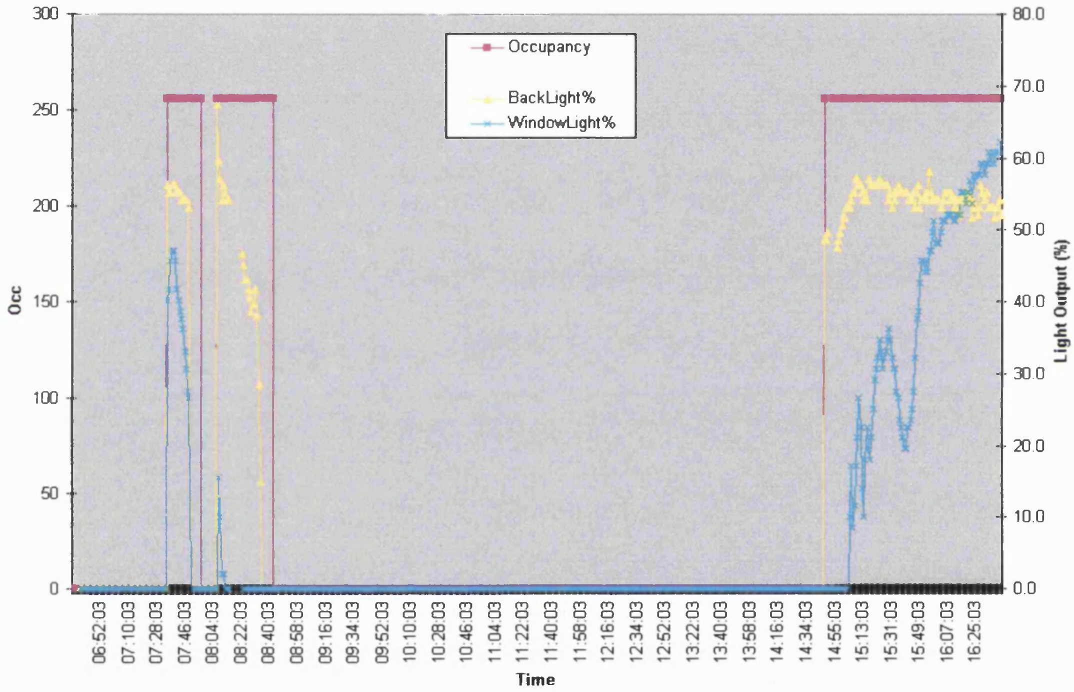


Figure 6.7: Graph showing an example of the readings from the occupancy sensor and the two light outputs. This shows how the lights were switched to respond to the occupancy sensor.

The Temperature and Humidity Sensors

The commissioning of both the internal and external temperature and humidity sensors was straightforward as they were not subjected to such large and rapid variations in sensor readings, and provided accurate and effective measurements (see Figure 6.8). The flat nature of the internal temperature readings taken also helped emphasis the point that there was no need to use internal temperatures to control blinds on a room by room basis unless it was part of a building wide energy strategy.

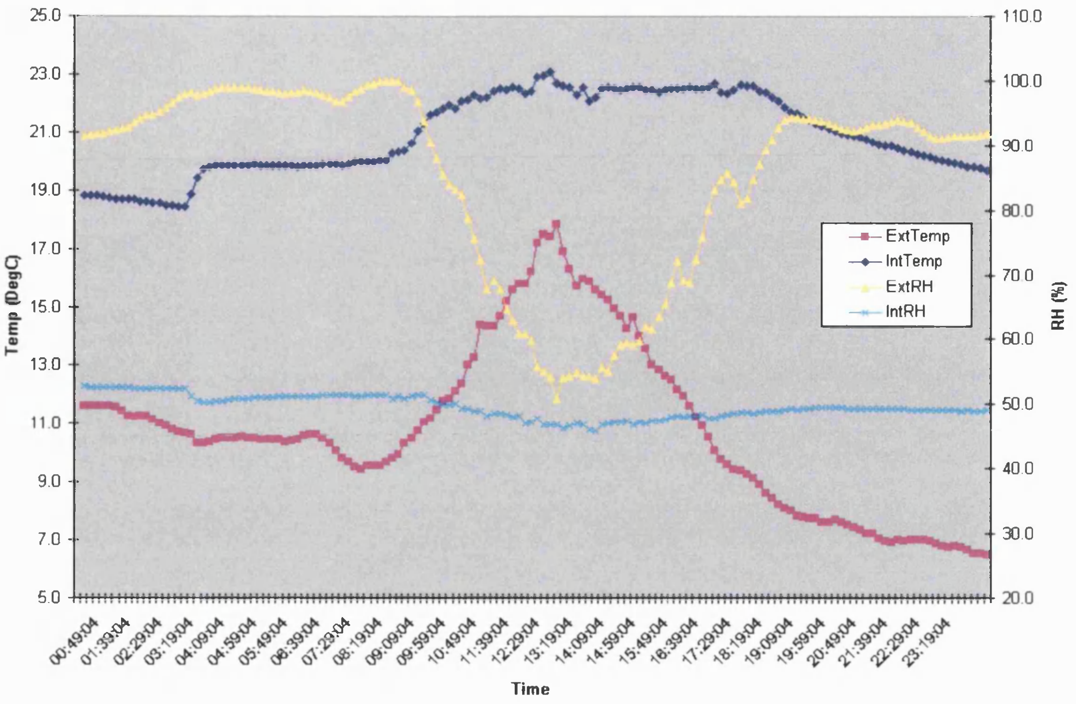


Figure 6.8: Graph showing data from the temperature and humidity sensors on the 9/11/99.

6.5.2 Lighting Control System

Part of the test room study was to consider the interface between the envelope and lighting systems to their sensitivity and range of performance. The study found that the implementation of an integrated automated venetian blind and lighting system presents many technical challenges to both the designer and the installer.

These challenges were highlighted in LBL's attempts to implement fully integrated blind and lighting systems, which used closed loop control to maximise the daylight output and minimise the electrical load. They found that such systems were very difficult to implement due to the problem already highlighted, i.e. the fact that the correlation between the photo-sensor signal and the measured daylight work-plane illuminance varied with blind angle. They also found that trying to redirect sunlight into spaces often resulted in glare, and as a result, meant the system provided little overall benefit.

In light of this work, a decision was made in this study to limit the blind and lighting integration to an open loop blind control system and a simple top-up lighting system. This decision was important as it would simplify the relationship between the two systems and, as a result, the blind control framework developed in the next few chapters.

The operational objective of the lighting system was to maintain a relatively constant minimum illuminance at the working plane, regardless of the blind's position, without annoying the occupant with constant switching. The ability of the lighting system to meet this objective was firstly examined by measuring the signals produced by the two light level controllers and comparing them to the external illuminance (see Figure 6.9). The comparison showed that electric lighting power output was inversely proportional to the daylight levels. As the internal light levels fell below the design set-point, and therefore that daylight linking was achieved (see Figure 6.10).

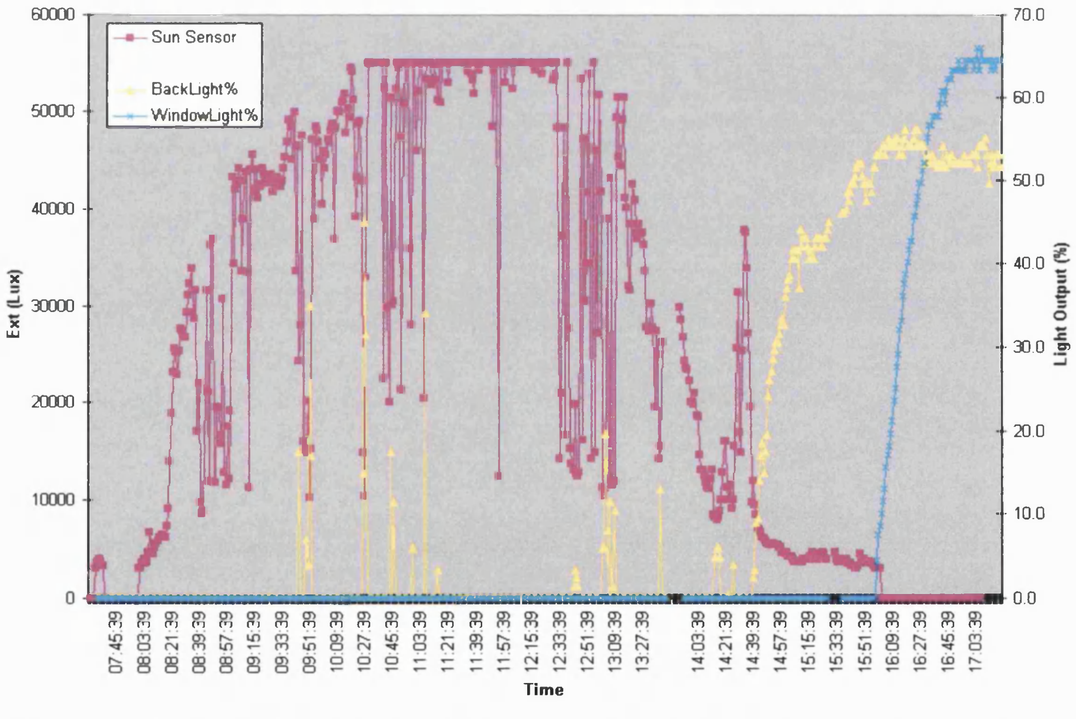


Figure 6.9: Graph illustrating the daylight linking capability of the lighting system by showing the output of the lights compared to the external illuminance. Notice that the light furthest away from the window comes on first. The blips during the day are caused by the movement of the blind.

Figure 6.10: Picture showing the rear light brighter than the window light for daylight linking.

When the daylight levels within the room rose above the design illuminance threshold, the high frequency ballasts in the luminaires were rated to produce a 10% minimum light output for a user-specified number of minutes before turning the lamps off. This delay implemented by the lighting controller was incorporated to reduce annoying on/off cycling of the lights during variable daylight conditions. The default off-delay was 10 minutes and this value seemed to perform adequately.

The test room study showed that the technology was available to allow the lighting and the blind systems to be integrated seamlessly. However, certain shortcomings were apparent in the sensors and lighting control system that were implemented. These included product design, installation, calibration and operational problems, such as noticeable stepped dimming and difficult commissioning procedures.

After witnessing the advantages of developing the Active X plug-ins to improve on-site commissioning techniques earlier in the chapter, it was thought that by simply encouraging lighting control manufacturers to utilise this technology, many of these problems could be eliminated. However, although this will undoubted help the current situation, when considering the development of improved blind control techniques, a number of problems would still be caused by the simplistic design of some core daylight and lighting control system components, such as the sensors. Therefore any blind control algorithms developed should avoid making too much demand on integrating with such systems if they are to avoid unnecessary complexity. After all, a user does not control a blind to maximise daylight, they control it to alleviate glare and discomfort, and so while an occupant is in the room the blind control system should try to operate in a similar manner, whilst still incorporating some daylight enhancing features.

6.5.3 The Automated Blind Hardware

The automated blind control device installed in the test room had a number of limitations associated with its adjustment mechanism which had an impact on the way in which the proposed framework could be developed. In order to tilt the blind the end of the pick up system, which collected the lift cord, had a rotating groove that created enough friction with the tilt cord to alter blind tilt angle when the blind was between the end limits of its tilting movement (see Figure 6.11). Once the blind had reached the end limits of tilting motion, the blind simply raised or lowered depending on the motor direction.

This meant that any vertical movement of the blind required the blind slats to close before the blind could be raised or lowered. Such movements were very distracting as they cut off the daylight coming through the window and affected the output of the lighting system.

Figure 6.11: The blind head-rail cord pick-up mechanism. Through the large opening one can see the central lift cord wound around the cord collector and through the smaller opening one can see the tilt cord sitting in its groove.

In order to provide greater control and fine tuning of the two functions of the pick up system, the blind system controller managed the voltage across the motor to provide two motor speeds. At the beginning of each movement, when the tilting action was expected, the motor ran at a slow speed, and then after a certain period, when a raising or lowering motion was more likely, the motor ran at a much higher speed. This functionality was implemented on the Neuron chip and could be configured through the network management tool on the computer to suit the size of blind (see Appendix VII).

Unfortunately despite this electronic wizardry, the nature of the cord-pick up mechanism and the fact that there was no control feedback within the system meant that there were unavoidable errors when trying to position the blinds accurately due to cord slippage in the rotating friction groove. This meant that there were certain restrictions to the blind positioning algorithm implemented, as small variable errors would escalate to large errors over a period of time. This will be discussed in greater detail in Section 6.6.

Despite its limitations, the LonWorks compatible blind control mechanism installed in the test cell clearly demonstrated the application of distributed control techniques on blind systems. However, as the next section will describe, distributed techniques alone are not enough to improve the acceptance of control systems if they do not incorporate satisfactory control techniques.

6.6 Sensitivity of the Control Software

6.6.1 Distributed Control System Design

It has been shown that manageable distributed peer-to-peer technology is available to develop usable and adaptable systems. However, the implementation of a distributed control approach requires a paradigm shift in the way that control algorithms are developed.

A common feature of most field bus systems is their small bandwidth, which means that they cannot interchange large amounts of data and commands to make their decisions or negotiate with other agents. Therefore an ideal distributed architecture is driven solely by physical data, not commands, required to make decisions at each node being sent over the network.

Two complimentary approaches are used to achieve this end; decomposition and abstraction.

- *Decomposition* – the systematic breakdown of a complex system into smaller simpler parts. At each level of decomposition there is an appropriate level of functional description.
- *Abstraction* - the allocation of implementation detail to the level most appropriate, thereby simplifying the system and the objects contained within it.

The following rules should be considered when partitioning the tasks within a distributed system. A process must be:

- logically simple and closely related to the function it performs;
- contribute to the logical simplification of the overall system;
- minimise runtime inter process communication.

Using the LonWorks controller supplied with the blind motor as a starting point for the development of new blind control techniques provided the opportunity to investigate these issues by studying its framework and ensuring that any further developments conformed to decentralised control system design methodologies.

6.6.2 The Characterisation of the Functionality

The LonWorks compatible blind controller, used in the test cell study, was supplied with a simple threshold control algorithm embedded within its Neuron Chip. As described earlier, the operation of the algorithm could be altered through the LonMaker network management tool by using its device configuration properties. These configuration properties and the overall device functional profile for the controller are discussed in more detail in Appendix VII. However, for the purposes of this thesis, the core functionality of the algorithm has been broken down into the following components, some of which are concerned with improving the energy efficiency of the system and some of which are concerned with its acceptability to users:

- The assessment of the sky condition;
- The evaluation of the blind position;
- The movement of the blind to block unwanted direct solar radiation and glare;
- The movement of the blind to maximise daylight;
- The occupant over-ride facility; and
- The control time interval.

6.6.3 The Assessment of the Sky Condition

The controller's assessment of the sky condition compared the highest external solar illuminance reading from the two external solar illuminance sensors, to a pre-set threshold value that could be adjusted using the configuration properties (see nviLightThres in Appendix VII). If the external illuminance was above the threshold value, the sun was considered 'to be out' and the blind was lowered, if below, the sun was considered 'to be in' and the blind was raised.

During the spring/summer months that the controller was initially tested, the default threshold value of 24,000 lux appeared to provide a good indication of whether the sun was in or out. Unfortunately, as the control algorithms were developed it became apparent that this value was less successful at other times of year. This discovery will be discussed in more detail later in the Chapter 8.

6.6.4 The Evaluation of the Blind Position

With no means of feedback included on the blind positioning mechanism, timers within the Neuron chip were required to keep track of the status of the blind position and tilt (see nvoSBlindStatus in Appendix VII). The position value had a maximum value of 0% for fully lowered and a minimum value 100% for fully raised. The tilt position went from 0 degrees closed with blinds facing outwards, through 90 degrees horizontal, to 180 degrees blinds facing inwards. Both these values were updated and displayed immediately after any movement.

To combat the errors in the blind positioning mechanism discussed earlier, the manufacturer had developed a blind positioning algorithm that reset itself, by closing the blind, before adjusting its tilt in response to a scene function. Like the vertical movement, this action was annoying because each time the blind adjusted its tilt angle the room was plunged into darkness, triggering a reaction in the artificial lighting system.

The transition of the test room blind from fully lowered to fully raised took approximately 20 seconds and the time for the blind to go from closed facing inwards to fully closed facing outwards was approximately 5.5 seconds. Both these timings could be entered into the control algorithm as configuration properties (see nciMotorPrmtr and nciLocalPrmtr in Appendix VII) so that the blind positioning algorithm and the twin speed control could be adjusted for all shapes and sizes of blind.

It was decided that the operation of the blind control mechanism as supplied was unacceptable for this study, as the majority of the larger blinds used commercially were available with a separate tilt motor that eliminated these problems. Chapter 8 will review the processes undertaken to improve the blind positioning algorithm along with other elements of the blind control system. A detailed analysis of the errors incurred by the blind system and their impact on the blind control algorithms developed in the next chapter can be found in Appendix VIII.

6.6.5 The Movement of the Blind to Block Solar Radiation and Glare

When the external illuminance rose above its threshold value, the blind lowered fully and set itself to a default blind angle. The pre-configured blind angle was 45 degrees, which represented the optimum fixed angle for solar blocking and daylight admittance throughout the year in a northern European climate. This value could be altered to any value through the configuration properties (see nviLightSetting in Appendix VII).

The algorithm also incorporated a time delay to ensure that the blind only came down after the threshold had been passed for a certain period of time. Again this variable was configurable through the configuration properties (see nciLightPrmtr in Appendix VII) and was included to avoid any unnecessary oscillations of the blind in partly cloudy conditions. The default time period was 180 seconds, but throughout the study 30 seconds seemed to perform adequately.

6.6.6 The Movement of the Blind to Maximise Daylight

To maximise daylight use during overcast conditions, the blind was raised when the risk of solar penetration and glare was thought to have passed. This was determined when the external illuminance had passed beneath the pre-set threshold for a pre-determined period of time. Again this time delay was incorporated to ensure that the blind did not oscillate unnecessarily during partly cloudy conditions and could be adjusted within the configuration properties (see nciLightPrmtr in Appendix VII). The default for this delay was 900 seconds (15 minutes).

6.6.7 The Occupant Over-ride Facility

As discussed ealier, two switches, located on the control panel, were hardwired to the blind controller to provide the occupant with the ability to control either the blind up or the blind down relays. Pressing a switch once would activate a relay and pressing it again could deactivate it.

The priority level and priority duration of this occupant over-ride could be set using the device configuration properties (see nciLocalPrmtr in Appendix VII). The priority could either be a one-shot priority that provided over-ride without a priority duration, or it could be linked to an occupancy sensor to provide priority until the occupant left the room. This setting was disabled as it prevented the author from observing the automatic control operating and because it undermined the energy goals of the system (see nviOccNonOcc in Appendix VII).

The facility was also available to pre-set a number of scenes in the form of blind positions and tilt angles for user control purposes (see nviSceneConfigMotor, nviSceneLocal and nviSceneZone in Appendix VII). This method of switching the device was not as flexible as the hard-wired manual switches, but did provide a more satisfactory method of controlling the blind with the IR hand-held controller.

6.6.8 The Control Time Interval

The control algorithm was essentially an event driven algorithm, therefore the control time interval was determined by the instant the external solar illuminance passed through the threshold value and the timers associated with those movements. Additional timers were supplied that were configured to poll the sensors at predefined intervals to ensure that they were still connected, but many of these were disabled as a default to limit network traffic.

A flow diagram illustrating the operation of the existing controller as a whole can be found in Figure 6.12 with a simplified version in Figure 6.13.

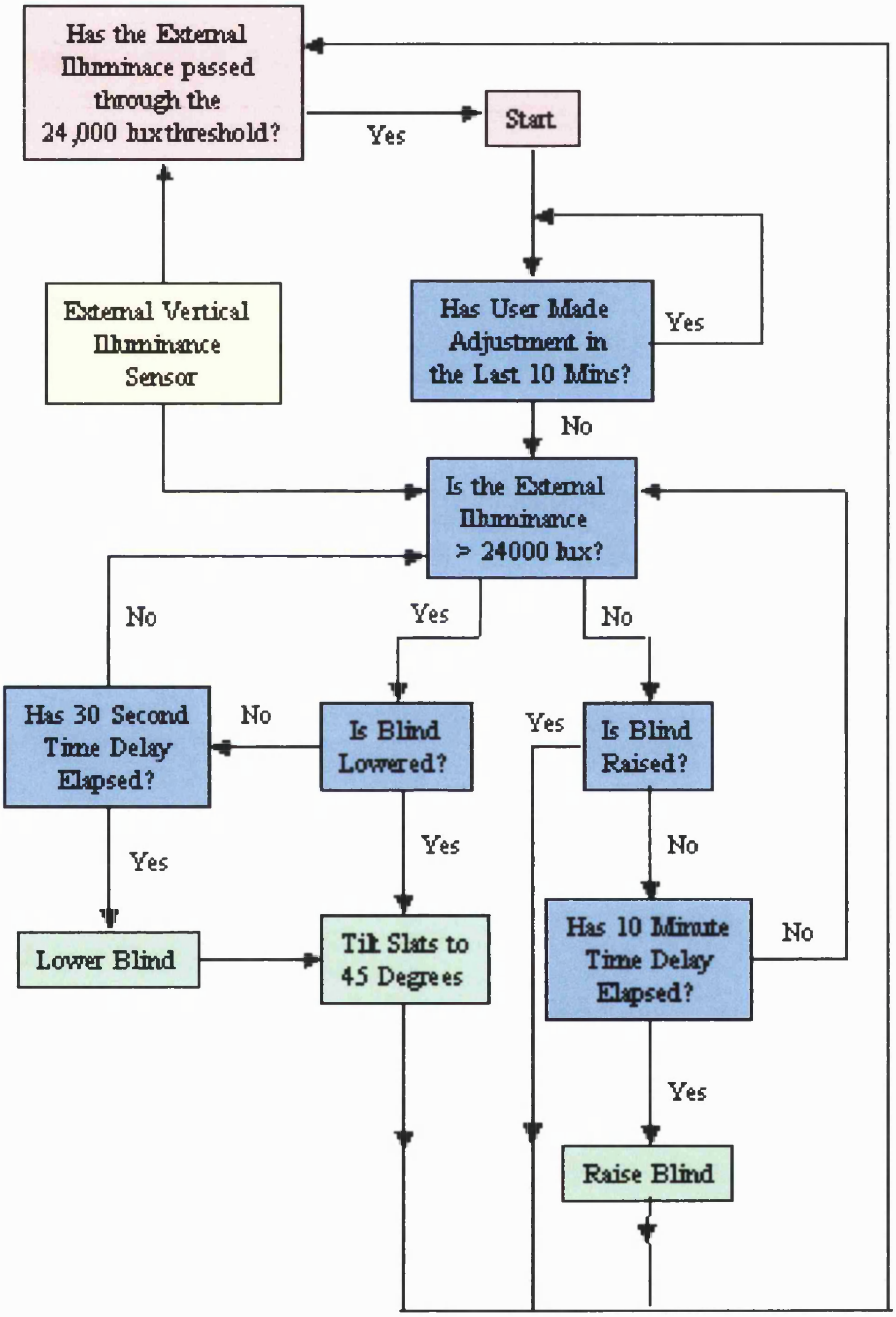


Figure 6.12: A flow diagram showing the operation of the existing controller

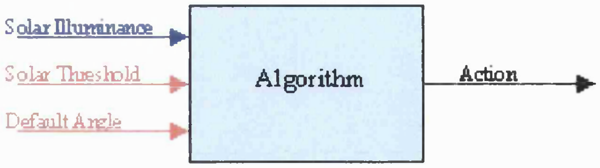


Figure 6.13: A simple diagram illustrating the existing controller's core inputs and outputs. Network Variable Inputs are shown in Blue and Network Configuration Properties are shown in Red.

6.6.9 The Shortcomings of the Existing Algorithm

After experiencing the system in action for a few months and considering the lessons learnt earlier in the study, it was concluded that the threshold controller supplied with the automated blind system would be unable to meet users' expectations and energy targets for a number of reasons:

Firstly, the fixed solar blocking angle used by the controller reduced the daylight penetration into the space during the summer months, in addition to limiting the view through the blind. This was substantiated by the LBL studies reviewed in Section 3.2.4 which showed that dynamic blind tilt angle control could give lighting savings of 18-32% compared to a partially closed static blind.

Secondly, the singularity of the blind action implemented by the manufacturer's controller, coupled with the time delays added to restrict blind oscillation, resulted in the event driven control algorithm reacting to only a tiny proportion of the possible inputs. In addition to this, a number of opt-out network variables, such as the occupant over-ride timer, were included within the control algorithms to limit the automated functionality further. Both these factors meant that the blind would often remain in the same position for most of the day.

This minimalist control approach may have been a reaction by the manufacturer to the kind of occupant criticism of automated blind systems described at the end of Chapter One, and their knowledge of the limitations of the blind movement mechanism described in this chapter. However, an automated controller that moves infrequently and responds largely to user over-rides, does not provide many more benefits, in terms of energy savings and comfort, than a less expensive manually operated blind and therefore should not be deemed satisfactory, merely because it does not disturb its users.

6.7 Chapter Summary

Implementation of a manageable control infrastructure and a user-centred control algorithm alone do not guarantee the satisfaction of control strategy goals. Resolving issues, such as how often to activate the system under rapidly changing sky conditions, or how much precision is necessary to control blind tilt angle, are equally important when considering a system that must balance energy targets with individual user's needs.

Installing the hardware within the test room allowed the author to gain hands on experience and expert knowledge of the sensitivity and limitations of the typical hardware and control strategies available. The hardware selection for the study was based on functionality (typical of commercially available systems), cost and availability. A decision was made to try and mimic a practical application by minimising the number of sensors used, thus reducing the overall cost and simplifying the installation, but with the drawback of not being able to collect comprehensive data sets. This was successful as the sensors chosen provided a suitable level of control, and gathered sufficient data and information to document the overall performance of the system. Examples of these data sets have been shown and the data itself is included on the attached CD-ROM.

The sensors were found to have some weaknesses that could inhibit their application to the more complex control strategies proposed by others and discussed in Chapters Three and Four, e.g. the difficulties in commissioning and fine-tuning light sensors for closed loop control. These difficulties justified the decision to limit the algorithm's initial inputs to only a couple of simple sensors. However, it is acknowledged that this decision could put more onus on the control algorithm.

Although the blind mechanism used also demonstrated some limitations, it should be noted that larger slatted blinds, which are incorporated in the majority of automated buildings and are the subject of the occupancy studies reviewed in Chapter One, do not have the limitation of the system reviewed in this work. This is because these types of blinds incorporate two motors to control blind tilt and lift separately. Therefore nothing can be read into the blind's movement limitations as the problems encountered resulted from the adoption of a relatively new and unique automated blind mechanism for controlling the smaller blind chosen for the window. In hindsight, a system that could move vertically, with its slats in the horizontal position, and then adjust its slats accurately would have been preferable for assessing the current 'state of the art' of blind control mechanisms. Despite these shortcomings, Chapter Seven and Appendix VIII discuss how it is possible to incorporate blind movement errors within a system without sacrificing performance and acceptability.

Setting up the study also demonstrated that the LonWorks control infrastructure was simple to use and manage, as it was similar in its workings to standard IP systems that have now become the norm in most offices and households. It was shown that 'plug and play' technology could reduce the need for commissioning in slightly more complex systems, as blinds could have their configuration properties set before they arrive on site.

To supplement these findings, the study reviewed the off-the-shelf products used in order to understand the building wide issues of local (space by space) and global (whole building) control, and enable the author to gain knowledge of how to design an appropriate building control framework that could be flexible, easily re-configured and cost effective. It was shown that local level control loops, necessary to implement control on an individual occupant basis, were possible whilst still having a device integrated within a large global system with whole building diagnostics, monitoring and control functionality.

Although it could be argued that these experiential findings were limited by the fact that the author's own subjective responses were unlikely to be representative of the global population, it can be assumed that they are reasonably representative of typical control installation engineers and system integrators who are often from a similar technical background.

By studying the Neuron processor and LonWorks technology in detail (see Appendices V and VI), it became apparent that there was a great deal of untapped potential to further improve blind control systems, and integrate them within a sophisticated building wide control system. It was found that the architecture of field bus systems differs from the client-server architecture of traditional control systems by the fact that each node within the system incorporates a low cost micro-controller that has the ability to collect inputs and process them on their own, whilst co-operating with other nodes to achieve a greater goal. The use of local processing and enhanced communications protocols in this way can improve equipment performance, but more importantly, can provide a wealth of information about the operation of the equipment previously unavailable. In addition to simply changing set points and interoperable messaging, many of these intelligent nodes were capable of self-diagnosis, calibration, validation, linearisation, and the ability to share resources. Therefore it was proposed to take this field bus technology, which was currently being used simply as a network management solution, and incorporate it as the backbone to a new control framework for blinds.

The last two chapters have described how the simple threshold control algorithm, supplied with the blind controller, was installed and evaluated to determine its upper and lower boundaries of performance. This chapter, along with Appendix VII, has categorised the functionality of the controller to aid the identification of development opportunities. The existing algorithm was essentially designed to provide energy efficient operation with some occupant over-ride facility if necessary. In light of these initial test room findings, it was decided that a suitable starting point for the investigations was to try to develop the simple control strategy provided with the blind motor to a standard that meets current best practice for automated blind operation.

Therefore the algorithm would be slowly modified and developed over time, using observations from the system's performance and the information gathered from LBL's research into energy efficient blind control, to incorporate some of the features of the more advanced algorithms described in Chapter Three. This procedure made it possible to draw comparisons between the incremental performance improvements of any developed strategies relative to the simpler strategy and evaluate whether complex integrated strategies were warranted within the practical constraints outlined in this chapter.

Chapter 7: Implementing a Current Best Practice Control System

7.1 Introduction

A substantial amount of research has already been undertaken by others on the energy aspects of blinds and it was this research that has driven the development of automatic control to date. The objective of this part of the study was not to repeat this work, but to use it to assist in the definition of a new automated blind control framework that best satisfies both common energy objectives and the individual users' needs identified earlier in the study.

Whilst, for comfort reasons, it is necessary to integrate the user more tightly into the control system, for ecological and economic reasons, the automated system is still required to provide improvements in energy performance compared to a manually operated system.

This chapter is devoted to defining an automatic mode of operation that is not too complex, but is both flexible and adaptable enough to be developed to incorporate any individual user factors indentified, whilst still providing an adequate level of performance. It describes the iterative process undertaken to develop and implement an improved energy efficient control algorithm for the motorised blind used in the test room, and illustrates the practical bugs encountered and the solutions put forward.

The resulting blind control algorithm represents a best practice example of what was achievable by applying published knowledge in the field to the blind mechanism used in the study. Once up and running the algorithm was used to identify how similar energy saving strategies could be adapted to suit certain user specific constraints and consider the alterations to the system that needed to be made.

7.2 Defining the Functionality of a Best Practice Control System

7.2.1 Increasing the System's Complexity

Chapter Three characterised blind control systems into three layers of complexity. The existing blind control algorithm, supplied by the blind manufacturer, represented the lower layer, the threshold controller, and the two higher layers were sun blocking algorithms and mode controllers. A decision had to be made about which type, if any, would offer the most energy efficient and desirable solution for the proposed system.

The review of these three types of systems in Chapter Three and Chapter Four showed that although, in theory, mode controllers offer the highest levels of functionality and the biggest energy savings, in practice their set up procedures are complex and their overall performance is often lower than expected.

Indeed the extensive research carried out by LBL on systems that chose between different energy saving modes, such as light transmitting and light directing modes, showed them to be reasonably ineffective.[8 66 69] One reason for this is because the net system efficiency of a daylighting system, in a similar fashion to occupants needs, varied with the window size and type, the room geometry, various optical parameters and with the intensity and distribution of skylight and sunlight. A blind's effect on the cooling load and the daylight contribution to the space cannot be accurately predicted, and therefore it becomes very difficult to balance and optimise it for the conflicting purposes of daylight and solar control, as well as complexity and usability.

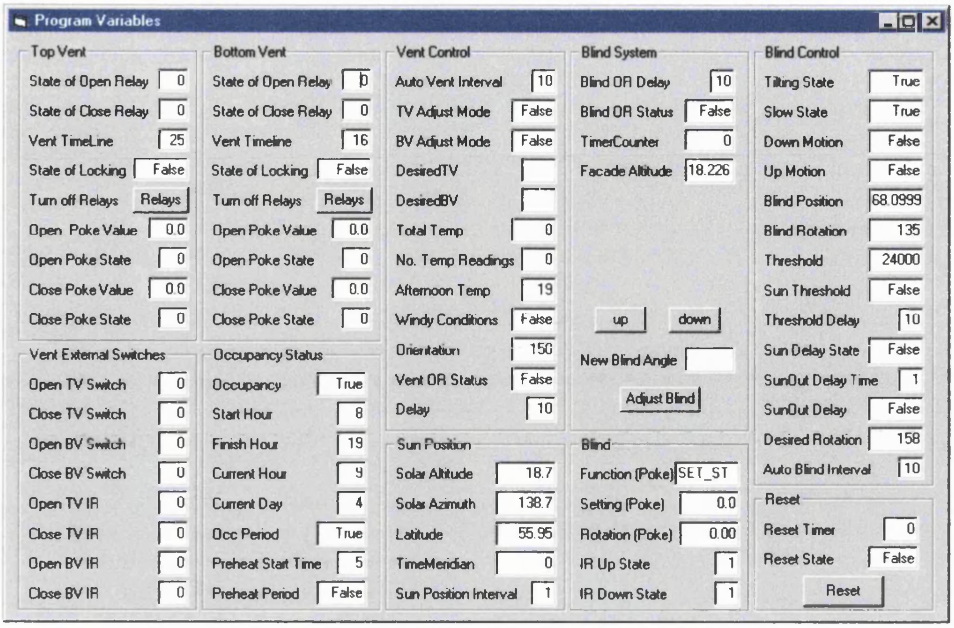
Another reason that some mode controllers were ineffective was because closed loop control was difficult to achieve in practice. The difficulties associated with closed loop control were partially experienced during the commissioning of the daylight top-up lighting system within the test room (reviewed in Chapter Six). Had the blind system also been included within the feedback algorithms, the complexity of the process would have increased to the extent that even user-friendly commissioning devices would not have eased the task.

For this reason, it was thought that the large amounts of data required for highly complex controllers, that effectively model the building and choose between modes based on predicted energy performance, could still not be accommodated within the control framework in a cost effective manner. In view of these findings, it was considered more prudent to use simpler open loop sun blocking control strategies based on instantaneous, measured data as a precursor to the use of the adaptive control algorithms proposed with its less ambitious form of mode control.

7.2.2 Setting up a Test Bed Control System

In order to develop the existing control algorithm, its basic elements were reproduced on the test room computer using a Visual Basic program. The program was divided into a series of functional blocks, similar in structure to the LonWorks control algorithm described in Appendix VII. Software variables were used to mimic network variables and configuration properties by forming links within and between the program's functional blocks, and through the use of the DDE Server, between the program and the control hardware devices. By removing all of the logical connections to the blind controller's processor chip in this way, the existing algorithm was effectively disabled and the blind controller itself became an actuator responding to switching signals from the Visual Basic program.

The program's variables were displayed on a hidden form (Form2) that could be viewed by pressing the F1 button while the program was running. A picture of the form is shown in Figure 7.1. The form allowed the author to monitor the program's variables and alter the program's configuration properties for debugging and experimental purposes. This process is now reviewed. For ease of comparison, the improvements made to the existing algorithm during the iterative design process are reviewed using the same components of functionality used to describe the existing algorithm in Chapter Six.



Top Vent
State of Open Relay 0
State of Close Relay 0
Vent TimeLine 25
State of Locking False
Turn off Relays
Open Poke Value 0.0
Open Poke State 0
Close Poke Value 0.0
Close Poke State 0

Vent External Switches
Open TV Switch 0
Close TV Switch 0
Open BV Switch 0
Close BV Switch 0
Open TV IR 0
Close TV IR 0
Open BV IR 0
Close BV IR 0

Bottom Vent
State of Open Relay 0
State of Close Relay 0
Vent Timeline 16
State of Locking False
Turn off Relays
Open Poke Value 0.0
Open Poke State 0
Close Poke Value 0.0
Close Poke State 0

Occupancy Status
Occupancy True
Start Hour 8
Finish Hour 19
Current Hour 9
Current Day 4
Occ Period True
Preheat Start Time 5
Preheat Period False

Vent Control
Auto Vent Interval 10
TV Adjust Mode False
BV Adjust Mode False
DesiredTV
DesiredBV
Total Temp 0
No. Temp Readings 0
Afternoon Temp 19
Windy Conditions False
Orientation 150
Vent OR Status False
Delay 10

Sun Position
Solar Altitude 18.7
Solar Azimuth 138.7
Latitude 55.95
TimeMeridian 0
Sun Position Interval 1

Blind System
Blind OR Delay 10
Blind OR Status False
TimerCounter 0
Facade Altitude 18.226
New Blind Angle

Blind
Function (Poke) SET_ST
Setting (Poke) 0.0
Rotation (Poke) 0.00
IR Up State 1
IR Down State 1

Blind Control
Tilting State True
Slow State True
Down Motion False
Up Motion False
Blind Position 68.0999
Blind Rotation 135
Threshold 24000
Sun Threshold False
Threshold Delay 10
Sun Delay State False
SunOut Delay Time 1
SunOut Delay False
Desired Rotation 158
Auto Blind Interval 10

Reset
Reset Timer 0
Reset State False

Figure 7.1: The commissioning and monitoring window of the Visual Basic program (Form2).

7.2.3 Assessment of the Sky Condition

To improve the blind control algorithm's assessment of the sky condition and enable it to tilt the blind to block the sun, a 'where is the sun?' question was added to the 'is the sun out?' question. Chapter Three reviewed a number of ways of answering this question within a blind control strategy and identifies two main options: a solar positioning algorithms and a open loop solar positioning sensor.

Solar positioning algorithms provide a controller with the sun's position in terms of altitude and azimuth (see Appendix IV). The implementation of these algorithms requires access to a timer to provide the true solar time, and the latitude and longitude of the site to be entered as configuration properties for each individual blind. By adding the orientation of the window facade to these inputs, it is also possible for the algorithm to determine where the sun is in relation to the facade.

LBL's research concluded that this process was extremely time consuming, costly and therefore non-viable, and encouraged the use of individual open loop window sensors. However, the findings outlined in Chapter Six demonstrated that the installation and commissioning processes used within our chosen control structure, LonWorks, were simple, quick and effective. In addition, it was shown that the technology would allow both the longitude and latitude configuration properties, which were unlikely to change during the design, to be pre-installed prior to reaching site. This meant that adding a few global variables and one individual variable to a device's application code would have very little effect on the device's manageability and that as a result, solar positioning algorithms could be implemented on both a local and global scale without adding too much cost to the system.

The process of choosing a suitable algorithm, that took into account all the errors in the system, is described in Appendix VIII and the Visual Basic code implemented for this algorithm can be found in Appendix IX. The solar positioning algorithm module of the program provided the blind control module with an altitude and azimuth angle. The blind control module then converted these variables into a solar wall altitude angle and used this to control the blind tilt angle (see Appendix IV). This changed the blind controller assessment of the sky condition from a one dimensional decision, using just solar illuminance, to a two dimensional decision using both the sky illuminance and the solar wall altitude angle.

7.2.4 The Evaluation of the Blind Position

As discussed in the Chapter Six, the existing blind positioning algorithm proved to be inadequate for the purpose of minute blind tilt angle adjustments. Therefore a blind positioning algorithm was developed that did not require the blind to close each time a small adjustment in the blind tilt angle was required. The algorithm used a half second timer within the Visual Basic Program to keep track of the blind's tilt angle and lift position and utilised similar configuration properties to the existing controller to commission the algorithm for the size of the blind used. In order to deal with the inaccuracies in the blind mechanism identified in Appendix VIII, the algorithm reset itself twice a day during periods when the room was unoccupied. The final code for this algorithm can be found in Appendix IX.

The algorithm denoted the vertical blind position as 0% for a fully raised blind and 100% for a fully lowered blind. In terms of tilt angle, a value of 90 degrees denoted a horizontal blind position and the values of 0 and 180 degrees denoted the downward and upward fully closed positions respectively. For example, a position value of 50% and a tilt angle value of 135 degrees denoted a blind that was half down with its blind blades tilting 45 degrees upwards (light redirecting).

In order to simplify the automatic blind control code, the blind tilt angle was controlled in 22.5 degree increments. This size of movement was determined to be visually acceptable and studies by LBL had demonstrated that the discrete positioning of the blind in this way did not have a significant effect on the energy saving performance of a automated blind system compared to a system with continuous movement.[71]

7.2.5 The Movement of the Blind to Block Solar Radiation and Glare

As with the existing controller, the blinds lowered when the external illuminance rose above a pre-defined threshold for a short pre-defined time delay. The blind control module of the code used the solar wall altitude angle, determined by the solar positioning algorithm, to calculate the discrete blind tilt angle required to cut off/block direct sun, whilst maximising the view or openness of the blind.

The geometric equation that related the required blind tilt angle to the position of the sun, in terms of the solar wall altitude angle, was determined by the blind slat spacing [h] to blind slat width [w] ratio. The Venetian blinds used in the study had a slat spacing to slat width ratio of 1:1.25. However, in order to accommodate the small inaccuracies witnessed in the blind positioning mechanism, it was decided to use a simplified version of the equation, with a ratio of 1:1 (see Appendix VIII). This decision simplified the relationship so that the desired blind angle was twice the wall solar altitude angle. The Visual Basic code for this relationship can be found in Appendix IX.

In addition to this basic functionality, the control algorithms used the solar positioning information to raise the blinds when the sun was not on the facade. Using the internal and external temperature readings for controlling the blind during the occupied period were considered but the data collected showed that temperature in the test room was not variable enough to warrant this approach (see Figure 6.8).

7.2.6 The Movement of the Blind to Maximise Daylight

Chapter Three highlighted how tilting the blind slats to reflect light deeper into a room was difficult to achieve in practice and often unpopular with users due to discomfort and disability glare caused by the reflected sunlight from the slats. In addition, Chapter Two demonstrated that views and connection with the outside world are extremely important to user comfort and that strategies which unnecessarily reduce desirable views in order to control daylight admittance should be avoided.

With these facts in mind, a decision was made not to use the upward blind angles within the automated blind operation and to restrict the movement to five possible tilt angles between 0 degrees to 90 degrees. This meant that an additional rule had to be added to the program so that the solar blocking algorithm was not executed when the solar incident angle was above 45 degrees. On occasions when the wall solar altitude angle rose above this threshold the slats would remain in the horizontal position.

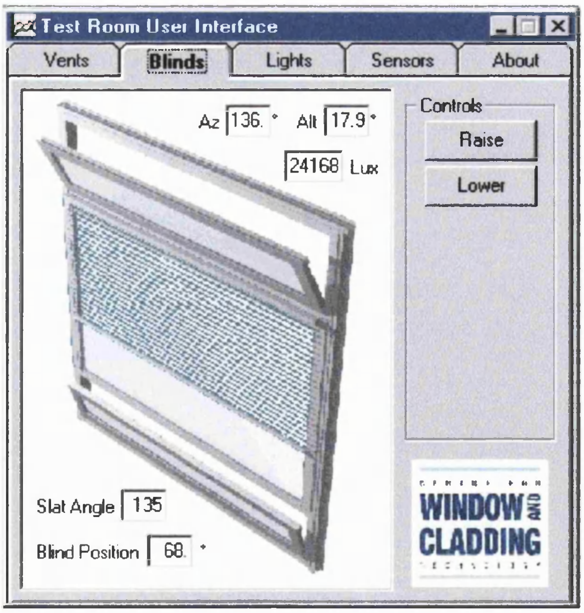
As part of the development process studies were undertaken on the effect of altering the movement delay time, used in the existing algorithm and also implemented within the improved algorithms, on the system performance. Lengthening the delay time has the advantages of increasing the longevity of the motorised system, diminishing the attention drawn to it by the occupants, and dampening blind oscillations in partly cloudy conditions. However, it has the disadvantage of reducing the system's ability to maximise the daylight levels within the room. Shortening the time delay can result in more energy efficient operation but implementing too short a delay can often lead to annoying blind oscillations. Therefore a compromise between the two had to be found.

LBL concluded from their studies that attempts to decrease blind motion during partly cloudy conditions did affect the energy performance of the system, but not significantly. Therefore the time delay should be determined by what can be tolerated by the user.

On a series of partly cloudy days near the start of the test study period, the blind was tested using a number of different time delays, ranging from 5 minutes to the existing controller's default of 15 minutes. From the author's own subjective observations a time delay of around 10 minutes was deemed to provide an adequate performance and this value was used for the rest of the study period where it continued to perform well.

7.2.7 The Occupant Override Facility

The user over-ride facility for the system was enhanced through the development of the Visual Basic Program interface. The interface was developed to incorporate familiar user over-ride features embedded in common 'Windows' components so that non-technical users would be able to interact with the system at a low level. It comprised of five tabs each providing different functions to the user. These included the vent control tab, the blind control tab, the lighting control tab, the sensor reading tab and the about tab (see Figure 7.2).

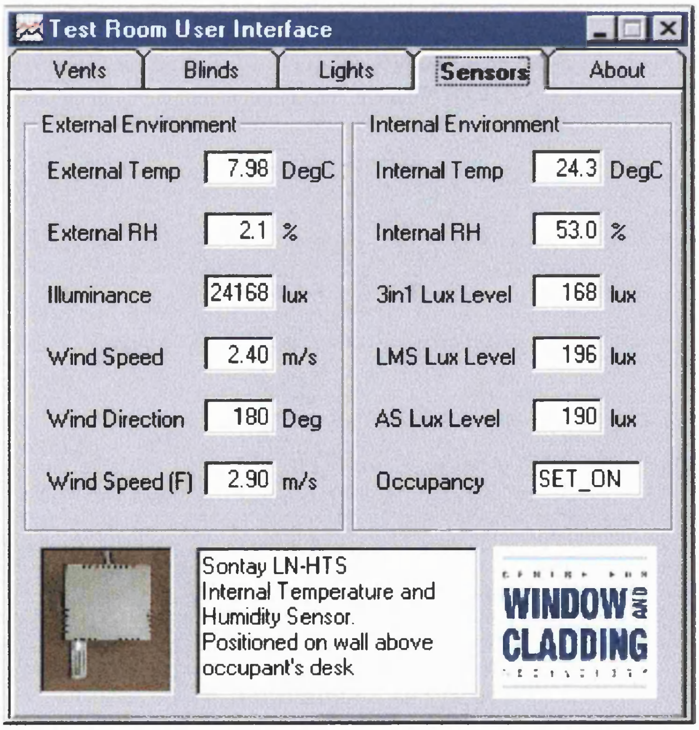


Az 136. ° Alt 17.9 °
24168 Lux
Slat Angle 135
Blind Position 68. °

Figure 7.2: The test room user interface showing the blind control tab

The blind control tab allowed the occupant to raise, lower and tilt the blind by clicking a few buttons. The raise and lower buttons would activate their respective relays on a mouse click and deactivate the relays on a second mouse click. Rules had to be applied to the program to ensure that both the raise and lower relays were not activated at the same time. When a blind was in the process of raising, lowering or tilting, the vertical position and tilt angle would be calculated and displayed in text boxes on the interface every half-second to provide the user with feedback on his or her actions. In addition, the image on the display would adjust to provide visual feedback of both the blind position and tilt.

The external illuminance and the solar altitude and azimuth were also displayed on this tab to allow the more discerning occupant to recognise patterns for themselves. Further information related to the blind control was displayed on the sensor-reading tab (see Figure 7.3). The other tabs are discussed in Appendix VII.



External Environment

External Temp 7.98 DegC
External RH 2.1 %
Illuminance 24168 lux
Wind Speed 2.40 m/s
Wind Direction 180 Deg
Wind Speed (F) 2.90 m/s

Internal Environment

Internal Temp 24.3 DegC
Internal RH 53.0 %
3in1 Lux Level 168 lux
LMS Lux Level 196 lux
AS Lux Level 190 lux
Occupancy SET_ON

Sontay LN-HTS
Internal Temperature and
Humidity Sensor.
Positioned on wall above
occupant's desk

Figure 7.3: The test room user interface showing the sensor reading display tab

The developed blind control strategy respected and recognised user over-rides from the user interfaces, namely the manual switches, the IR controller and the Visual Basic Controller. Once a user had made an adjustment on any of the user interfaces, the Visual Basic program started a timer that ran for a predetermined period. Whilst this timer was running no automatic adjustments were allowed to occur so as to avoid the user becoming frustrated with the system.

In addition, the user over-ride priority could be configured in the Monitoring and Configuration window of the Visual Basic program and was set to ten minutes. This level of functionality was not available with the existing controller.

7.2.8 The Control Time Interval.

As with the existing controller, the blind control algorithm developed was essentially an open loop controller with no feedback within its control loops from the environment that it influenced. This meant that if the control logic was to adjust the blind tilt angle in response to the movement of the sun, internal timers were required to enable the control system to re-assess its actions at pre-defined time intervals. Therefore unless the external illuminance passed through its threshold level or the occupant had just made an adjustment, the control loops described in this section and outlined in Appendix IX were repeated every ten minutes. This time control interval could be set in the configuration and monitoring window of the Visual Basic program and performed well, as the blind often did not move for an hour or two in stable conditions.

A flow diagram illustrating the control loops of the control algorithms developed can be found in Figure 7.5. The simplified version in Figure 7.4 below illustrates the core functionality and primary inputs used by the algorithm developed and can be used to demonstrate the rise in complexity of the algorithm from a one-dimensional control decision to a two-dimensional control decision, when compared to Figure 6.13.

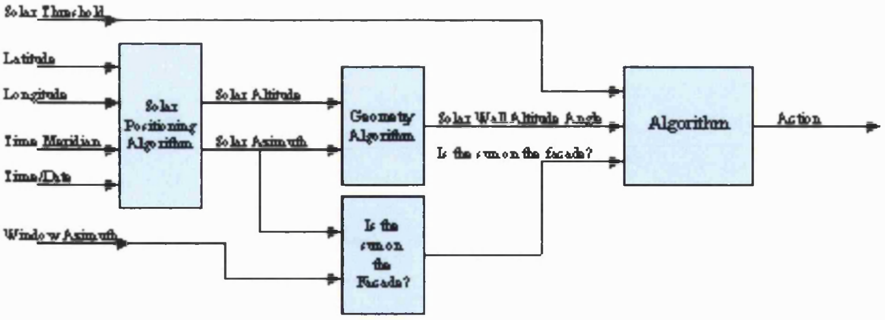


Figure 7.4: Core inputs and outputs of the modified control algorithm

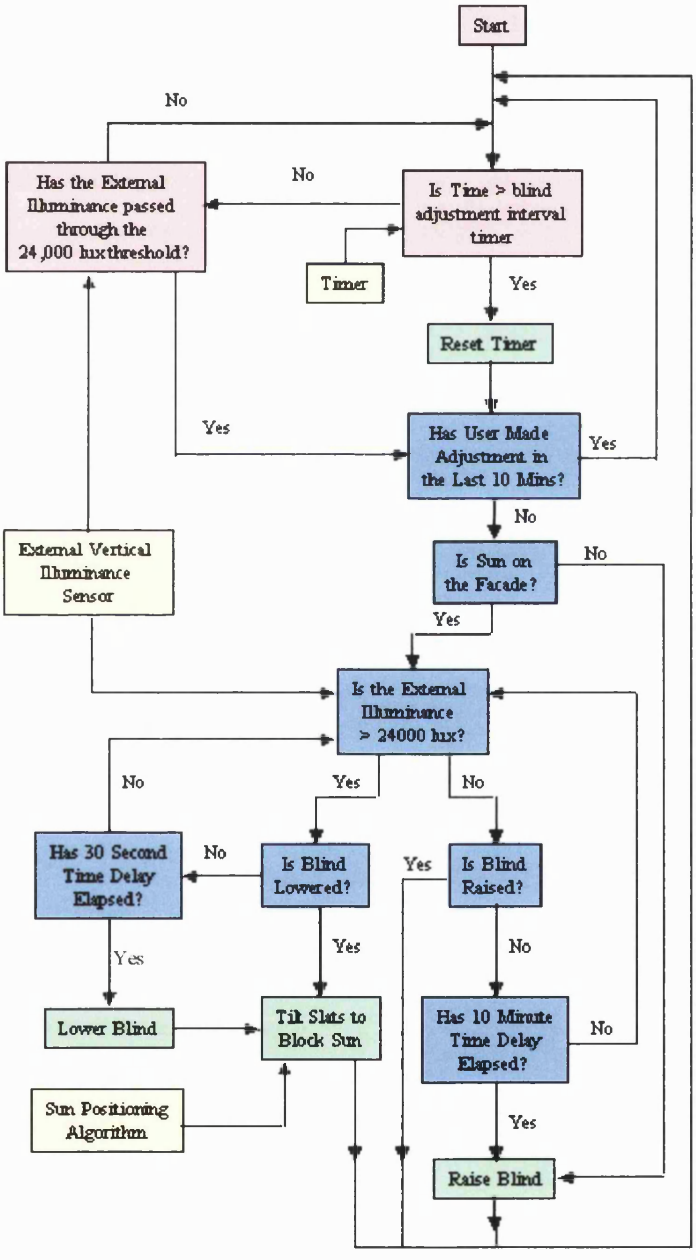


Figure 7.5: A flow diagram outlining the blind control algorithm developed using current best practice techniques.

7.3 Chapter Summary

The implementation of the control algorithm using a Visual Basic program on the test room PC enabled the easy modification and development of the installed control system. A single simple strategy was implemented and complexity was added to it gradually using first hand observations and the findings of the literature study. This provided the opportunity to use the experience gained from the intial studies to evaluate, develop and experiment with the implementation of a variety of different blind control techniques, by allowing practical bugs to be experienced, identified and resolved first hand. The control strategy developed represented, what others considered to be, the current state of the art of energy efficient and user-friendly blind control techniques.

The blind control strategy calculated the position of the sun at regular short intervals and used this information, and the reading from the solar illuminance sensor, to decide whether to lower or adjust the tilt of the blinds to block direct solar penetration into the space. The success of this strategy can largely be attributed to the detailed study of the options available for control, reviewed in Appendices IV and VIII. The blinds would then be raised when the sky conditions allowed.

The system offered priority to user over-rides through a variety of interfaces, which each provided the user with a quick and effective response. A demonstration prototype of a multimedia based user interface was successfully developed and shown to be easy to understand, identify and use, and provide adequate feedback and fine-tuning capabilities.

Developing the resulting control system expanded the overall knowledge base gained from the initial set-up study and provided a more advanced system for the author to experience. Chapter Eight reviews the author's experiential observations, which provide enough information to enable the author to identify and access the feasibility of adding complexity to the system to cater for an individual user's needs.

Chapter 8: Individual User Experience and Knowledge Gained from the Test Room Study

8.1 Introduction

8.1.1 Assessing the Performance of the System

The control algorithm developed in the last chapter, represents a 'state of the art' solution defined by implementing the lessons learnt throughout the study to find a compromise between system energy efficiency, control performance, human comfort and cost.

Once up and running, the author assessed the system from an individual user's perspective for a period of 6 months, from July to December 1999. The system was deemed to perform well during most sunlit conditions and met its primary control objective, i.e. block sun when external illuminance above 24,000 lux. The blind angle tracked the solar position, blocking direct solar penetration and maximising views throughout the day. This operation tied in well with the lighting system, which reacted to the changing levels of daylight entering through the blind. The direct sunlight only penetrated through the blinds on occasions when the sun had come out from behind a cloud and the blind would not be lowered until the 30-second time delay had passed, but this was tolerable for the author as the blind reacted quickly after the time delay had expired.

Unfortunately, on occasions it was observed that although the blind system did meet its basic design objectives, it was not able to meet fundamental user's objectives, namely the reduction of glare and maximisation of views, and on these occasions the system had to be over-ridden. This chapter reviews and discusses those observations.

8.2 User Observations

8.2.1 Recording User Interaction as a Measure of the User Experience

Throughout the test room study, notes were taken on the cause of, and sensor readings at, each instance the blind had to be over-ridden. The purpose of these notes was to help the author to discover patterns in user over-rides, if they existed, and aid the development of an improved control framework. The following day these results were written onto the day's solar illuminance graph downloaded from the data logger. An example of a day's readings and notes can be found in Figure 8.1.

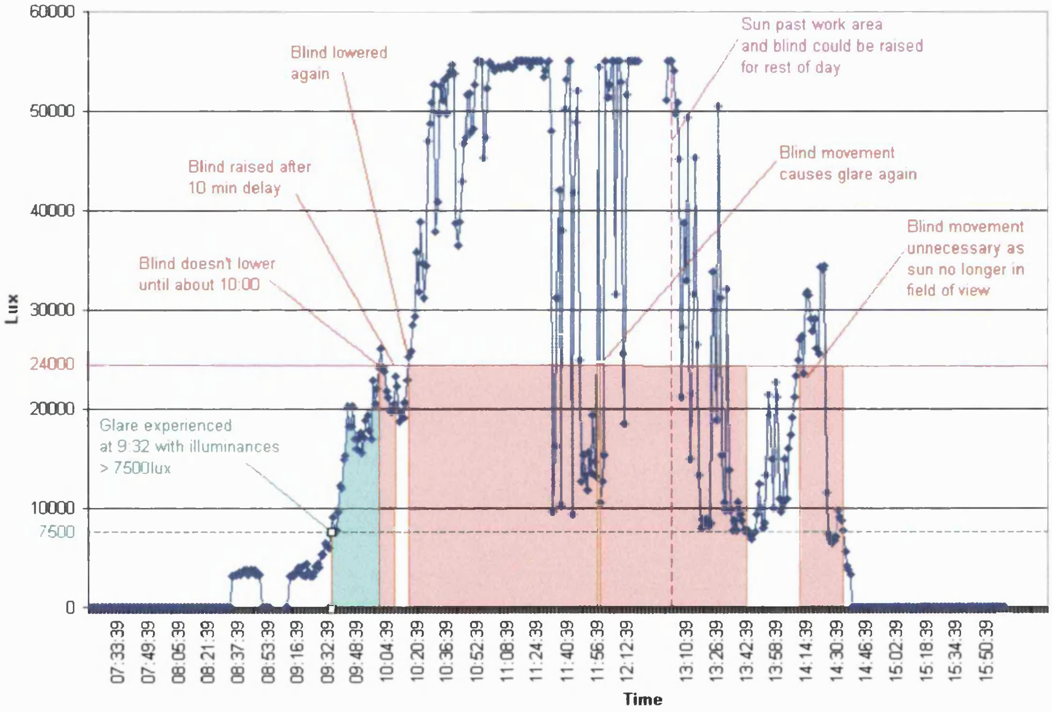


Figure 8.1: Graph showing the solar illuminance readings on the 25/11/99 and the notes taken on the blind's performance. The green shaded areas are when glare occurs and the red shaded areas are when the blind is lowered.

The notes and readings taken throughout the test period are summarised in Table 8.1 and Table 8.2 and the full data for those dates can be found on the attached CD.

Date	Time	Altitude	Azimuth	Illuminance	Notes
28/10/99	09:47	12.0	147.1	10,000	Glare due to thick clouds diffusing sunlight
	09:49-10:46	23.9	162.0	7,782	Still Glary. Glare stays until illuminace rises to 24,681 lux at 10:46 and blind comes down
03/11/99	08:33	10.7	131.1	17,000	Glare
	09:08	14.6	138.8	8,894	Glare until sky gets to 24,000 lux at 09:10
04/11/99	09:42	17.5	146.8	20,000	Brief moment of glare as sky rises to just above 20,000 lux from just below 10,000 lux for a couple of minutes
09/11/99	08:49	10.9	135.4	13,118	Glare through thin clouds, can make out shape of sun
	08:50-08:55	11.5	136.7	11,784	Illuminance fluctuates from 11,000 to 22,000, glare experienced throughout. 08:55 blinds lowered
	13:39	18.0	206.6	26,237	Blinds raised as sun not on work area. Blinds open for rest of day
10/11/99	10:10	17.9	154.0	10,000	Overcast morning with bright patches above library building.
11/11/99	08:15	6.4	128.4	0	Glare through clouds next to Norwood House, sensor must be in shadow cast by Norwood
12/11/99	08:45	9.7	134.8	8,004	Glare through thin clouds
	08:47-08:48	9.7	134.8	8,225	Illuminance fluctuates between 8,204 and 12,007, glare experienced throughout.
	08:48-08:52	10.4	136.3	13,341	Illuminace rises to 40,000, above threshold but kept blinds up, brief moments of glary 8,689 lux and 13,000 lux, Glary above threshold for rest of morning
	12:30	20.5	189.2	24,000	No glare

Table 8.1: User observations from the test room study (Part I)

Date	Time	Altitude	Azimuth	Illuminance	Notes
12/11/99 (cont)	12:31	20.5	189.2	55,100	Glare with sun on computer screen, but not on me
	12:35	20.5	189.2	33,000	No Glare due to lower illuminance and oblique angle of sun on screen
	14:08	15.0	213.1	46,694	Return from lunch, sun no longer a problem, so blind could be raised for rest of afternoon
17/11/99	10:47	18.2	163.2	8,356	Glare, had to lower blinds to perform tasks.
18/11/99	09:23-09:54	12.1-14.7	143.6-150.6	7,782	Glare until sky gets to 24,000 lux threshold at 09:54
	15:15	7.0	226.8	24,678	Overcast afternoon, <5,000 lux, sun comes out and illuminance rises to above threshold, blind comes down, but it is not needed as sun is not effecting task area
19/11/99	08:37	7.1	133.7	5,558	Slight glare as sun starts to rise above Norwood House, not disabilitating, OK until you look at it.
22/11/99	10:12-10:20	15.0-15.5	154.9-156.7	7,200	Glare through thick clouds rising to just below 10,000 lux before falling
23/11/99	10:02-10:11	14.1-14.7	152.6-154.6	7,000	Glare as sun rises from 7,000 lux to just above 20,000 lux, until finally rising above threshold.
24/11/99	11:49	18.1	178.2	23,000	Glare as illuminance rises to threshold
25/11/99	09:31-10:06	11.2	145.57	7,337	Large glare source in sky just above library as illuminance rises to 24,000 lux
	10:07-10:20	14.0-14.9	153.7-156.7	20,000	Blind lowers but after ten minutes raises because illuminance is below threshold, but still glary
30/11/99	10:30-10:31	14.49	158.89	8,000	Temporary glare as illuminance rises above 8,000 lux for a couple of minutes

Table 8.2: User observations from the test room study (Part II)

8.2.2 The Difficulty with Identifying Glare

The notes outlined in Table 8.1 and Table 8.2 illustrate that one of the primary observations of the study was that at certain times of the year, when the sun was low and the sky was overcast, slightly hazy, foggy or partly cloudy, illuminances well below the controller's 24,000 lux threshold caused disability glare. Figure 8.1 contains data from one such occurrence and illustrates how glare was experienced at measured illuminances below 7,500 lux.

The visualisation of the data shown in the graph in Figure 8.1 can be enhanced by creating the Waldram diagram shown in Figure 8.2, which illustrates the users' view through the window from a seated position at the desk (see plan in the top left hand corner). This diagram was created by using scaled plans of the site and a theodolite positioned in the author's seated position within the room.

The diagram shows how the window view is mainly obstructed by a four-storey library building approximate 30m away, an eight-storey student residential building approximately 100m away (Norwood House) and the external overhang shown in the section in Figure 5.2. The diagram also illustrates the sun path in the window facing hemisphere and how in November the sun is blocked by the buildings opposite for about an hour, before rising out above the library building. This is also reflected in the graph in Figure 8.1 by the drop in illuminance at around 08:50.

Figure 8.2 summarises the entire data set gathered by showing the position of the sun and the sky illuminances when similar glare observations were made. We can see from this, that on all but one of the occasions that glare was experienced below the threshold value, the sun was within the author's field of view. On the one occasion that it wasn't, the sun was just behind a building and the fuzzy scattering of light by the clouds was still strong enough to cause bright patches around the buildings and in the author's field of view.

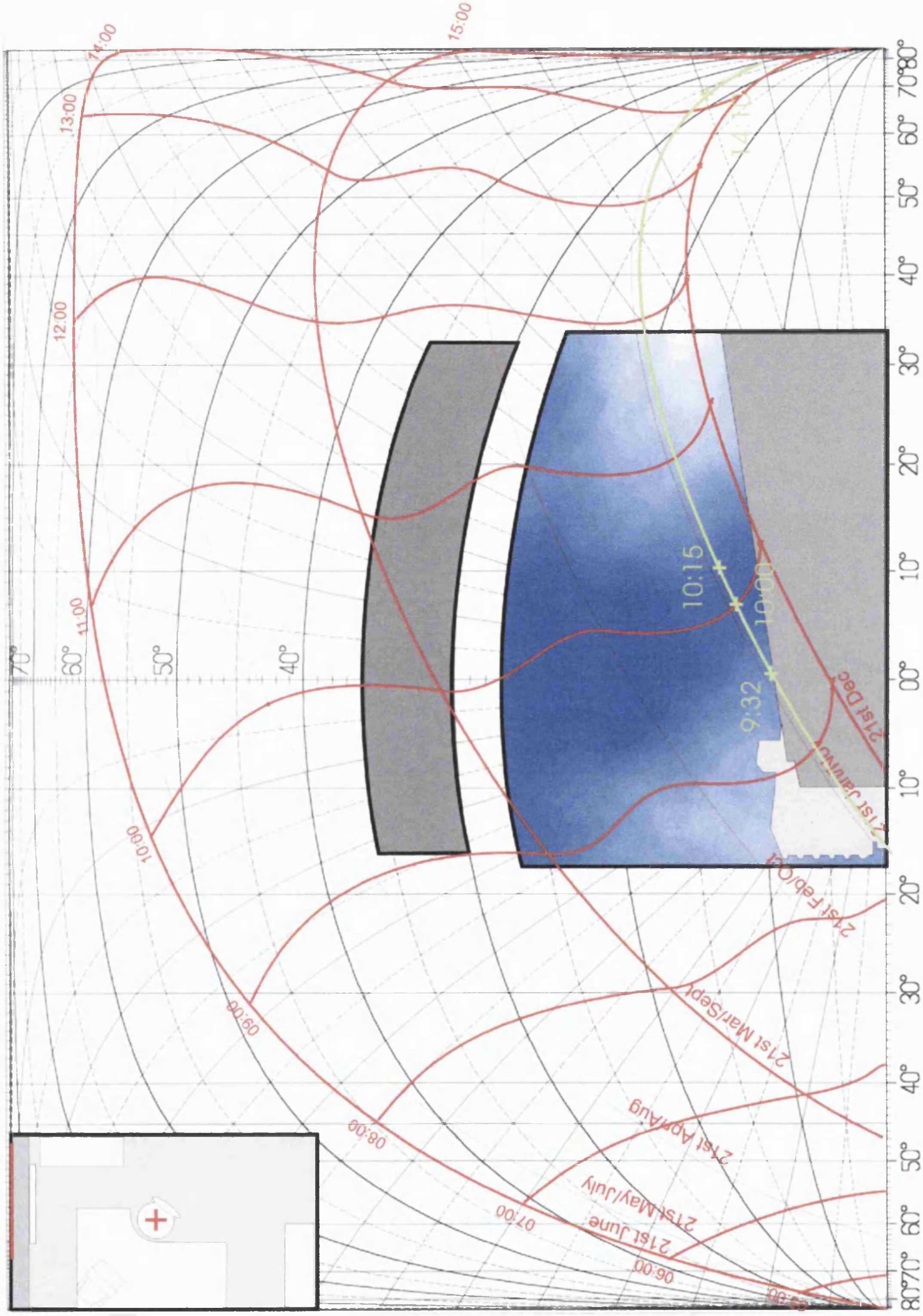


Figure 8.2: Waldram diagram showing view out of the window from the seat within the test cell and the sun path associated with the window facing hemisphere.

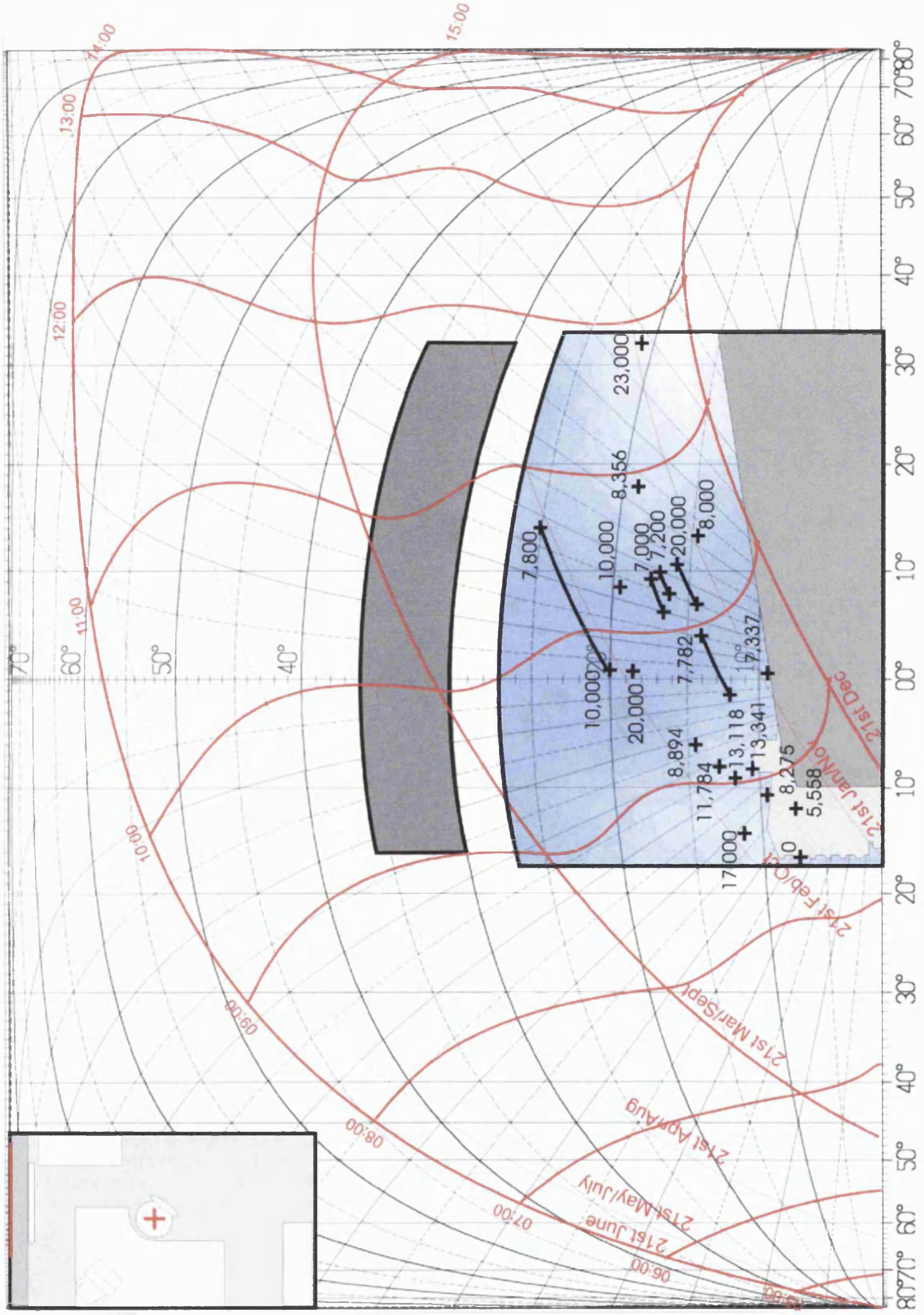


Figure 8.3: Waldram diagram showing view out of the window with the positions of the sun and the sky illuminances on the occasions when glare was observed.

A further example of the glare occurring in the author's field of view can not be denoted on a Waldram Diagram as it was observed when the sun was not in the window facing hemisphere. On more than one occasion, when the sun was out and in a certain zone of the sky at around 5pm during the equinox period, the geometry of the site caused an image of the sun to be reflected off the glazed elements of one of the buildings opposite. This caused severe disability glare in sub threshold illuminance conditions and prompted the author to lower the blinds (see Figure 8.4).

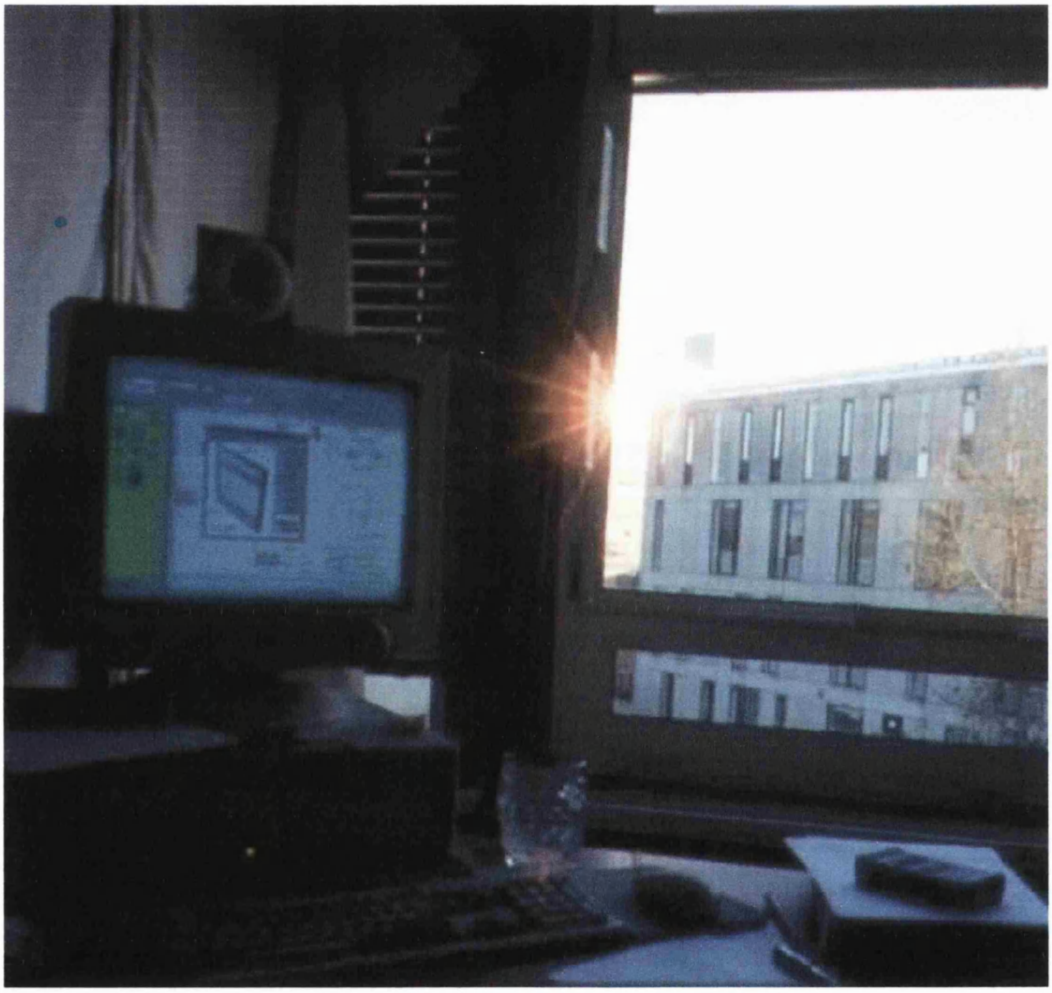


Figure 8.4: Photograph showing one of the occasions when the sun was not shining on the facade but glare reflected from an opposite building caused discomfort.

These observations start to illustrate the fact that the context of the surrounding buildings can have a large effect on conditions that determine glare.

8.2.3 The Importance of Contextual Factors

Another observations that was made was that the sun only really presented a problem when it shone in and around the task area or in an area that might reflect glare onto the user's computer screen (see Figure 8.5). In fact during the cold winter months allowing the sun to enter other non-task related areas of the room was regarded as quite pleasant due to its warming effect. This point is reflected on the graph in Figure 8.1, where the blinds could have been raised after 13:00 because the sun had passed the task area field of view and thus avoided, what the occupant considered to be a series of unnecessary blind movements.

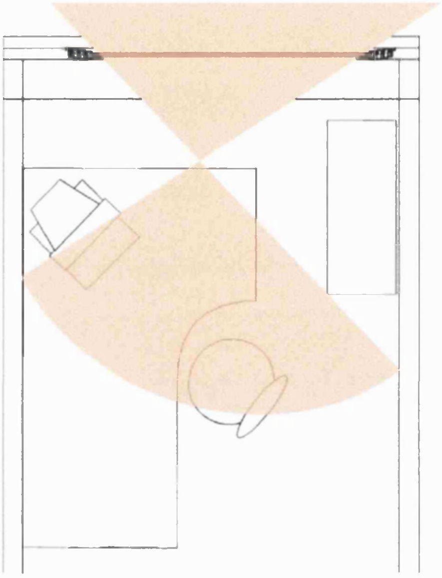


Figure 8.5: Diagram showing the angle of sunlight coming into the room that caused glare. In the cold winter months, the sun could have been allowed to penetrate through the blind to the areas outside the coloured zone.

Figure 8.6 summarises all the user observations on this contextual issue. It should be noted that this trend was more difficult to detect than the glare because the author, like any other user, was not looking to optimise the environmental conditions within the space, but simply to interact with the blind when glare prevented him from carrying out a task. In fact, the author's initial attention was drawn to the pattern by distracting blind movement, for what seemed to be no reason at all, when the external illuminance rose above the threshold level.

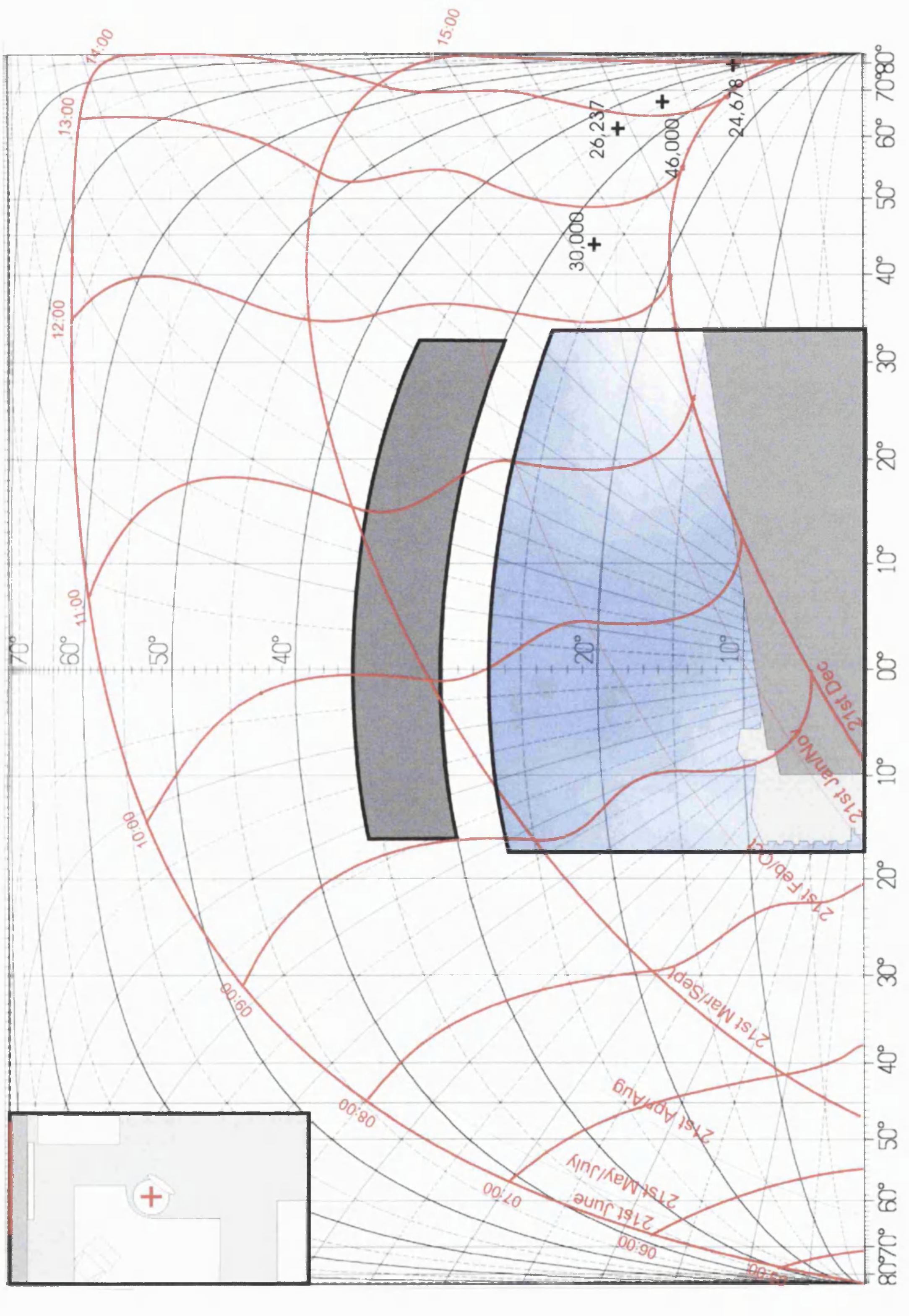


Figure 8.6: Waldram diagram showing view out of the window with the positions of the sun and the sky illuminances on the occasions when the sky illuminance was above the threshold but no glare was observed.

8.2.4 The Importance of the Occupant's Position in the Room

So far the findings have shown that the relationship between an individual occupant's expectations and the position of the sun is largely a contextual one, determined by the orientation of the window and its relation to the individual user and any external obstructions, such as buildings and overhangs. However, the occupant's position and orientation within the room also has a large effect on this relationship. To illustrate this, Figure 8.7 and Figure 8.8 show how the Waldram diagram changes with different desk locations and orientations within the room and how different zones of the sky, at different times of the year, can influence the direct glare incident on an occupant from a bright sun or sky.

8.3 Chapter Summary

By sitting and working in the test room throughout the duration of the study the author's understanding of the human factors affecting blind control, discussed in Chapter Two, were enhanced. The test room demonstrated that a simple sun tracking threshold controller provided adequate control when the user was not subjected to the possibility of direct solar exposure (July to Sept). However, under certain conditions in the winter months, a number of instances of direct glare at sky illuminances below 10,000 lux were recorded.

Previous studies had looked at the nature of the actions that users took in the domain of automated blind control and highlighted the conditions that they seemed to complain about most. The review in Chapters Two and Four showed that blind use is strongly related to the relationship between the sun's rhythm and the geometry, and use of a room. It follows a pattern depending on the position of the sun, the time of day and the sky condition. The findings of the test room study not only confirmed those from previous studies but also extended them by gathering enough detailed first hand knowledge of the findings to enable the author to begin to derive a solution to the problem.

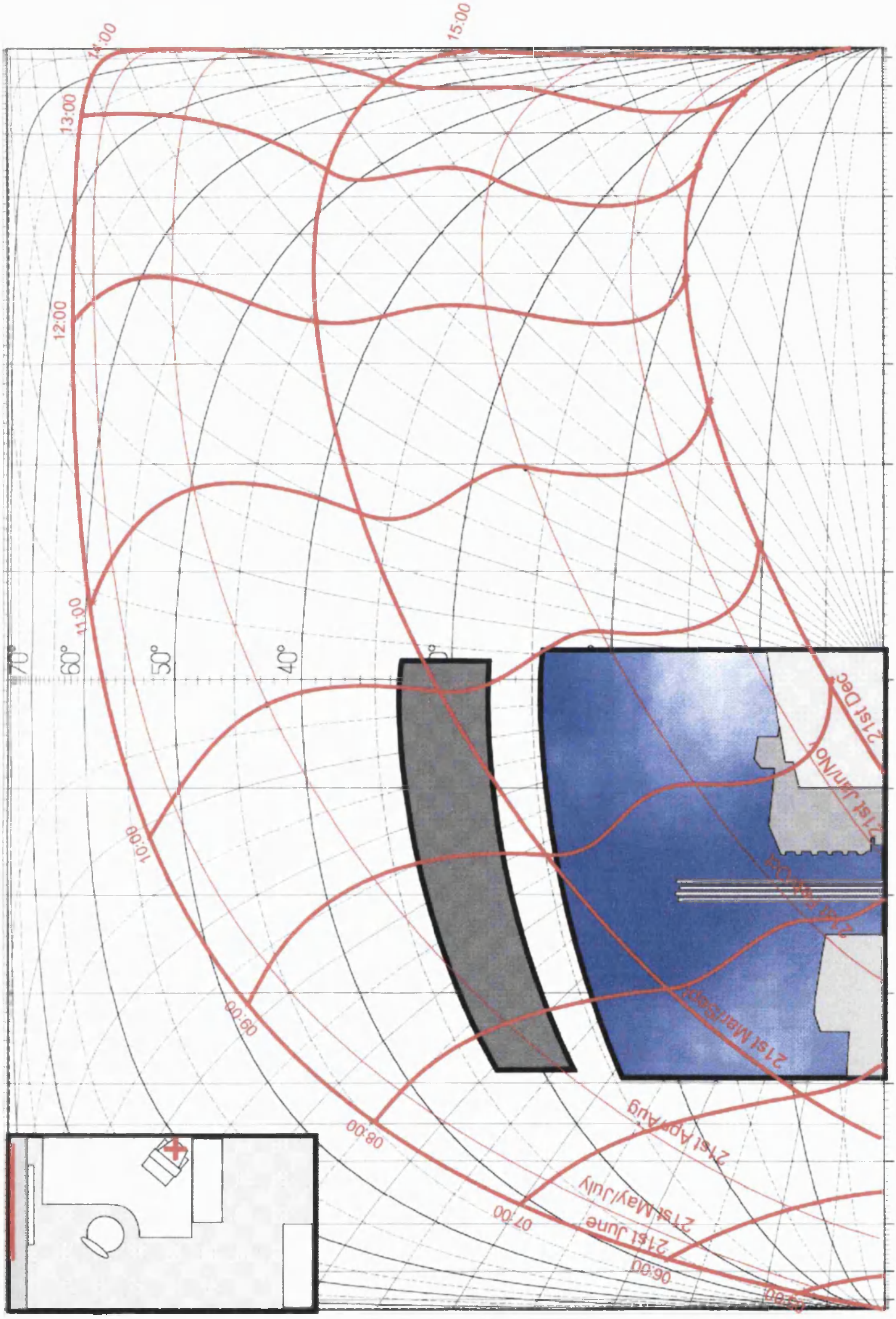


Figure 8.7: Waldram diagram showing the possible reflection from the computer screen if the desk was orientated differently within the test cell. Notice that the sun would affect the occupant at different times of day at different times of year.

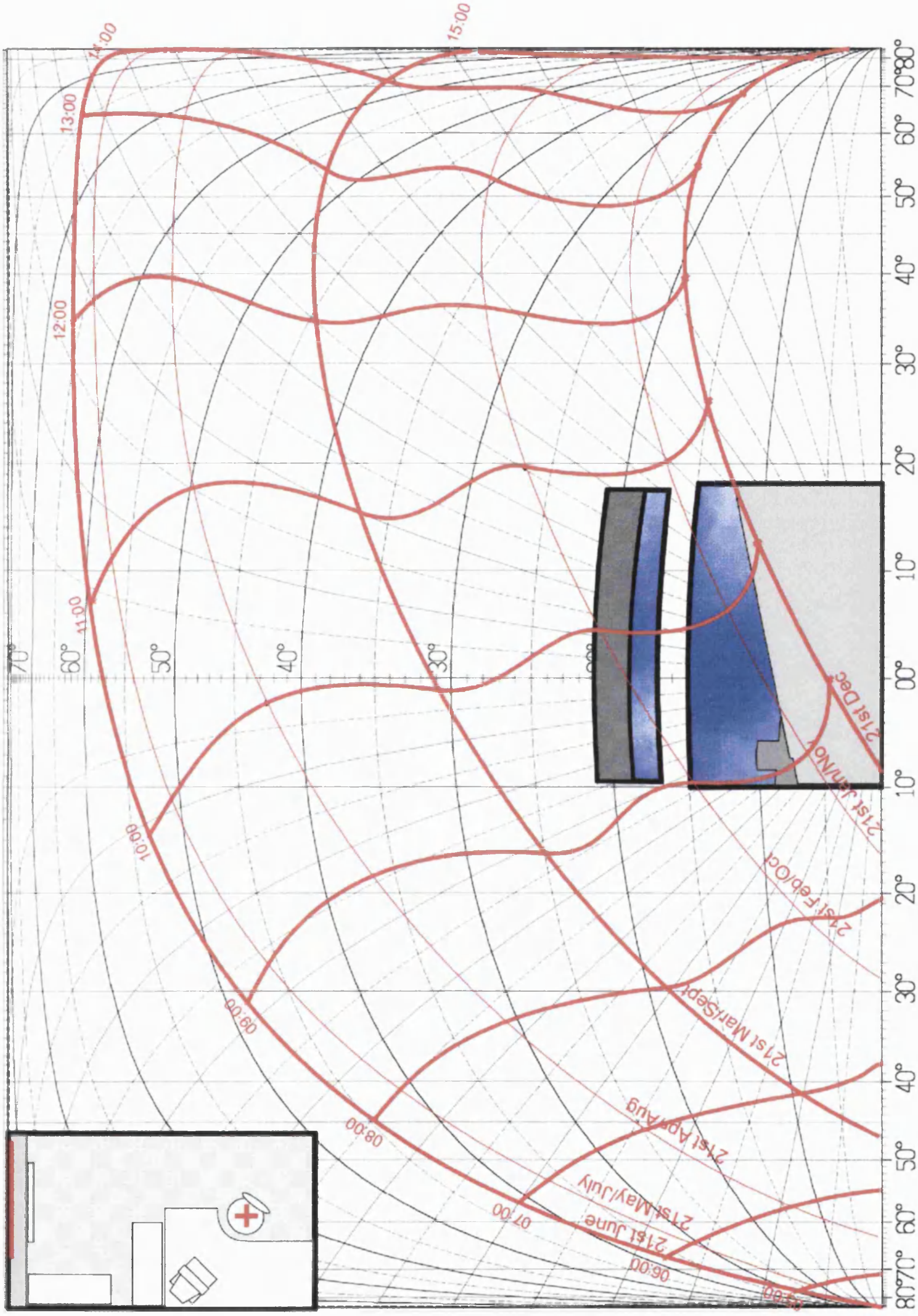


Figure 8.8: Waldram diagram showing the view out of the window if the desk was moved to the back of the test cell. Again the sun would only affect the occupant during the late morning between the equinoxes and the winter solstice.

One simple option for resolving the problem of glare caused by the low sun being scattered by the thin clouds, might be to lower the threshold of the controller to 7,500 lux. This approach would seriously undermine the energy efficiency of the system for the rest of the year when the 24,000 lux threshold performed well, and it would be necessary for the user to change the threshold throughout the year - a strategy that is unlikely to be undertaken successfully.

Indeed glare, the primary reason for a person to interact with a blind, is multidimensional and cannot be established from a few physical variables, such as solar illuminance, without knowledge of desk orientation and location, the nature of occupant tasks and the internal optical conditions. Therefore the important next step to improving the way automated blinds cater for the needs of individual users was to try to incorporate these factors within a new control framework that did not rely on complex and management intensive data input.

People are good at judging whether they need the blinds down to protect against glare or solar penetration. People are not so good at putting the blinds back up once the threat has past. A sophisticated control system that allows customisation of control criteria is needed, if multiple criteria such as energy and comfort are to be accommodated. But does this additional sophistication need to be as complex as some would suggest? Or can a more flexible and straightforward approach be taken that might offer us adequate performance at a fraction of the cost, by being well designed and implemented in software?

The detailed knowledge and data gained in this experiential and experimental stage of the work allowed the author to start to form an idea of how such a framework could be implemented in practice.

Chapter 9: A Simple Framework for an Adaptive Blind Controller

9.1 Introduction

The problem of controlling automated blinds to respond to variations between different users in different contexts requires an appropriate framework. The test room study demonstrated that decentralised field bus technology offers a sound base for this framework. Its advantages include:

(i) reduced control system complexity, due to control algorithms being local to each blind and user;
(ii) increased system usability, manageability and flexibility due to better integration with IT systems, and advanced network management functionality, although highly complex systems that require large amounts of site specific data were still considered to be too onerous for practical use; and
(iii) interoperable communications protocols, allowing information from other systems' devices, such as occupancy sensors, to be incorporated within the control strategy.

The test room also highlighted certain hardware issues that needed to be incorporated:

(i) sensor technologies were adequate for open loop control but not closed loop control;
(ii) user interfaces were simple but intuitive;
(iii) blind movement mechanisms were limited in their capabilities.

This chapter uses the knowledge and experience gained throughout the study to propose a simple framework that is able to meet the aims and objectives of the study and aid the development of improved blind control strategies in the future.

9.2 Starting to Define the Framework

9.2.1 Aims of the Framework

The design goals of the new framework are to:

- accommodate a control algorithm that minimises energy consumption but is adaptable to maximise comfort and reduce user interaction over time.
- keep the cost and complexity of the installation to a minimum, by using currently available technology, integrating it with other systems and reducing design and commissioning times; and
- present a simple user interface.

9.2.2 Meeting the Needs of Individual Users

The venetian blind has a huge influence on an occupant's perception of the internal environment. The factors affecting the way in which people control blinds will vary with location and climate, orientation, building type, interior spatial design, exterior obstructions and shading, as well as the needs determined by an individual's visual tasks. These factors are often interrelated in complex ways and are strongly time dependent given the variability of climate effects.

If designers are going to improve the way in which automated venetian blinds operate, they need to evolve beyond the idea that fixed controls strategies are designed, installed and commissioned to keep measured variables within the required tolerances. They must begin to recognise the context in which controls are being used and the fact that users often want to alter the targets that systems are asked to achieve to suit their own individual circumstances and needs. Therefore, the next step in automated blind control development should be to use new techniques to produce flexible control strategies that can cater for the individual environmental requirements of each occupant. However, such systems should not be complicated by the need for large amounts of manual data processing and systems management that enable them to obtain feedback about occupant preferences.

To solve this design conundrum, it is proposed to utilise the need for a provision of individual control to obtain information about the conditions that cause discomfort. This information may then be used by the system to adjust its energy saving control mappings accordingly. By doing this, the system is effectively using the occupant as an additional sensor and thus enhancing the integration between itself and the user.

9.2.3 Meeting the Challenge with a Flexible Control System

The proposed approach to the problem is essentially an open loop controller, with local feedback and fine tuning capabilities at each individual blind, as a means of tying an individual user more closely into a control strategy and thus providing a controller with a better chance of meeting their expectations. This concept is vaguely similar to the way in which Thermostatic Radiator Valves work on a compensated circuit. The compensator is the open loop controller that adjusts the heating flow temperature to the radiator circuit depending on the external weather conditions. The thermostatic radiator valves then fine-tune the adjustments. The only difference is that all the control is done in the Blind's local processor, and whether it wants to follow an open loop response is dependent on the localised closed loop response from the occupant.

Studies on occupant behaviour suggest that allowing the user to modify the default setting of an automatic controller to their preferred setting will often result in a decrease in the energy efficiency of the system.[91] However, similar studies also show that without this means of adjusting a control system's functionality to meet their individual needs, occupants get frustrated with automated systems. Therefore the design challenge lies in integrating the user's priorities with energy efficient control. Where there is a conflict, a new energy strategy must be developed to accommodate the user's personal preferences by finding the right balance between the provision of appropriate conditions (acceptable for most of the time) and the scope for intervention. The need for too much intervention is annoying, but the system should not take over or get in the way.

---

[91] Kempton W., Feuermann D. and McGarity A., 1992, "'I always turn it on super': - user decisions about when and how to operate room air conditioning units", *Energy and Buildings*, Vol.18, 177-191.

9.2.4 The Implementation of Occupancy Switching

One user-friendly way of balancing these needs and saving energy is through occupancy switching. Occupancy patterns can have a significant influence on energy consumption, due to the reduced need for energy when a building or room is unoccupied. People may occupy their workspace all day or for part of the day, their occupancy pattern may be regular or irregular, they may be constantly in and out of their room or they may stay in their room for long periods of time. The control system must be capable of deciding when to switch between strategies or modes depending on occupancy status, because the system does not need to cater for individual needs when the user is not in the room. These modes and their relationship to one another form the basic starting framework of the proposed system.

The test room study was used to demonstrate that the occupancy sensor used for a lighting control system could be used to provide information to the blind control system and was an effective means of altering a controller's strategy or mode when an occupant leaves or enters a room.

To investigate this, two basic modes of operation were implemented:

- *unoccupied mode* – used information from the temperature sensors to raise or lower the blind, depending on whether it was the heating or cooling season, during unoccupied hours (see Appendix IX); and
- *occupied mode* – controlled the blind to balance energy efficiency with occupant comfort for the majority of the time (described in Chapter Seven).

The sensor's time delay allowed the occupant to leave the room for short periods (ten minutes) without the system adjusting the control strategy. If the occupant left the room for a longer period, then the system could change strategies to a low energy strategy.

9.2.5 Modes of Operation

Four modes of operation, related to occupancy status and occupant interaction, can be defined to help the controller adapt to individual needs, whilst still optimising the energy performance of the system. These are:

(i) *unoccupied mode* – when the building is not within the predefined occupancy period (after everyone has gone home), the controller operates the blinds to minimise energy consumption but with no consideration for maximising daylight;

(ii) *energy saving background mode* - for when the occupant has been out of the room for a certain period of time or has not yet arrived in the building, this only occurs during normal occupancy hours (i.e. between 08:00 and 18:00 for a typical office). This controls the blind to provide energy efficient operation with some consideration of daylight to help prevent the occupant from switching on the electric lighting when they return;

(iii) *occupied mode* - for when the occupant is in the room, or has been in the recent past, and the system is incorporating any preferences learnt. In this state, the controller determines a blind position that is based originally on the energy strategy but adapted, or fine tuned, in a way that is most likely to be accepted by the user with respect to the history of interactions learnt from the occupant to date; and

(iv) *manual mode* - when the user interacts with the controller to dictate a blind position, the controller applies this state, with immediate effect, for a default time period (i.e. 10 minutes). After the controller has applied the request it adjusts an adaptive mapping between the input and output variables.

9.2.6 The Interaction Between the Modes

Figure 9.1 shows how the energy saving background mode, the occupied mode and the manual mode interact to produce an adaptive controller that learns occupant preferences whilst still performing in an energy efficient manner.

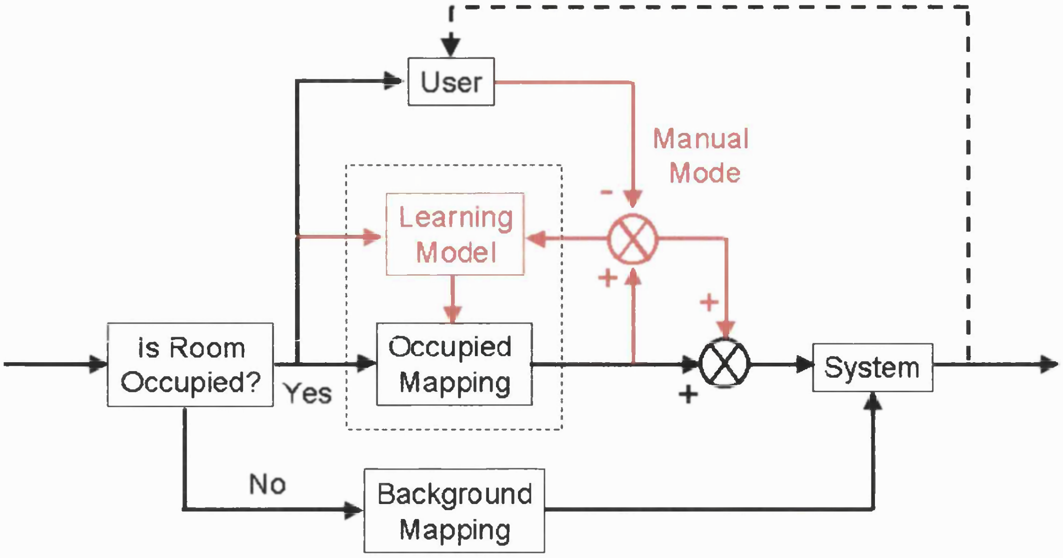


Figure 9.1: Figure showing the interactions between the energy saving background mapping mode, the occupied mode and the manual mode, during the occupancy period.

Any control system can be regarded as a means of mapping a series of inputs to a series of outputs, as demonstrated in Figure 6.13 and Figure 7.4. Therefore the difference between various systems is essentially the number and type of inputs and outputs used, and the nature of these mappings. It is assumed that the blind positions chosen by the user reflect their preferred blind position for a specific set of circumstances (or input variables) and therefore in essence the controller is trying to establish a temporal pattern of desired blind positions. Each blind position given by the occupant is recorded along with the solar altitude and azimuth and external illuminance at the time the interaction took place. The sum of these interactions forms a data set used to generate a schedule that corresponds to the instructions given to the controller. Adaptations to the control mappings are undertaken by some form of adaptive or learning mechanism (discussed in Chapter 10).

The next stage of the work was to define each mode of operation and how they were related to each other by investigating the nature of the input variables that should or could be used to enhance the control system's understanding of the environment to which it is exposed and the structure of the mappings. One of the most important aspects of this was the nature of the human interface and how it related to the learning mechanism that adapted the occupied mode.

9.3 The Manual Mode

9.3.1 Defining the Functionality of the Manual Mode

People require information to gain knowledge and make decisions. Today, information is often gained by sitting down at a personal computer to organise and collate the data through software programs. This information is then shared with others through an IT network.

The test room demonstrated that the LonWorks technology provided various means of supplying the wealth of information on the LonWorks network to the business level network, and by using dynamic data exchange, it was possible to create a Visual Basic user interface that allowed the occupant to make adjustments to the window components and lights from the computer. The opportunities that such an interface offers the designer are endless, so it was decided to undertake a study on how such an interface could best be used within the proposed framework.

9.3.2 The User Interface as an Inquisitor?

In addition to improving the usability of control systems, a user interface has the potential to enhance a system's ability to learn from occupant interaction by helping it understand why occupants make certain adjustments. For example, if an occupant wishes to close a blind, is it because they are hot, they are being subjected to glare or because they wish to have visual privacy? The environmental parameters that distinguish each of these causes can often be complex or difficult to evaluate; therefore using a control interface could provide a means of deciphering between which factors are the cause of which type of response.

A brief study into different user-interface techniques identified three possible approaches:

- *the traditional approach* -to simply rely on traditional control commands, such as 'open blinds' and 'close blinds' (a 'what you see is what you get' approach);
- *the environmental adjustment approach* - to replace the dialogue incorporated within the user interface, such as 'open window' or 'close blind', with an alternative decision making dialogue, such as increase air flow, reduce temperature and reduce natural light. The control system would then make a control decision depending on which action was the most energy efficient;
- *the questionnaire approach* - to ask the occupant a series of questions about why they are making an adjustment when they request one, before the system would make the adjustment.

Although all three of these approaches were considered valid, the second and third options were discounted for this study. It was thought that both of these approaches would fundamentally change the way people interact with blinds and for them to be developed successfully more detailed data on individual user interactions would be required to enable the system to cater for the irregularities of human behaviour. Without this data, these techniques would probably cause more frustration than most existing systems because they would be unlikely to react in a way that met users' expectations. In addition, the third option was thought to increase the complexity of the user interface and thus increased the likelihood of annoying the user, and did not meet the framework objectives.

Therefore, it was decided to rely on the simple, tried and tested up down button technique that controlled the blind tilt and lift. This way the user interface to the new framework could be of any type, manual switches, IR controller or even computer based controller. In addition, any added functional complexity could be taken out of the occupant's view, thus making the system more transparent and easy to understand.

The future of automated blind control interfaces lies not in developing means of increasing the representation of information, but rather in allowing increasing amounts of functionality to be under-represented. A well-designed, transparent interface should selectively reveal to people just enough information to help them accomplish their goals or to do their tasks.

The user interface should be seen as the part of the system that communicates and shapes the user's experience rather than a representation of underlying functions, therefore the ergonomic design of the interface was also considered important. Both the manual push buttons and the IR controller buttons used in the test room were simple ergonomically, and the Visual Basic interface was designed using similar buttons and a design language that would be familiar to most PC user's.

9.4 The Unoccupied Mode

9.4.1 Defining the Functionality of the Unoccupied Mode

The unoccupied mode proposed for the framework will be the same as the unoccupied mode implemented in the test room. The functionality of this mode is outlined in code in Appendix IX. A brief summary is to say that when the building was unoccupied, the operation of the blind was determined by whether the HVAC system was in a cooling or heating mode. If the zone was in cooling mode, the blinds would close during the day, and open at night. If the zone was in heating mode then the blinds open if the sun was on the window, and close if it was not.

9.5 The Energy Saving Background Mode

9.5.1 Defining the Functionality of the Energy Saving Background Mode

This mode will also be the starting point for the Occupied Mode and will utilise the best practice control algorithm implemented in Chapter Seven. This algorithm is thought to be the best energy efficient strategy available to meet the majority of the user's needs in most circumstances and therefore should reduce the number of times the adaptive system will have to be over-ridden. To enable a copy of the energy saving strategy to be adapted to meet individual user's needs in the occupied mode, some changes need to be made to the structure of the algorithm. These are discussed in the next section.

9.6 The Occupied Mode

9.6.1 Re-evaluating the Input Variables

We have seen in Chapter Four that an occupant is most likely to get frustrated with a system if it fails to deal with a consistent pattern of physical events that cause discomfort. The findings outlined in Chapter Eight reinforce the fact that these events are often determined by the three dimensional geometric relationship between the sun, external obstructions, the window and the user's position in the room. Current control techniques simplify this relationship to a vertical solar incident angle perpendicular to the plane of the wall (wall solar altitude angle), a two dimensional relationship that is unable to adapt to individual contextual conditions.

Figure 9.2 shows the zones of the sky (red and white zones) that correspond to the blind tilt angles in the best practice algorithm when the sun is in those zones and the illuminance is above 24,000 lux. The green, yellow and blue areas denote the areas of the sky that are likely to result in glare at lower illuminance levels for the three possible desk positions in the test room (see Figure 8.6, Figure 8.7 and Figure 8.8 respectively).

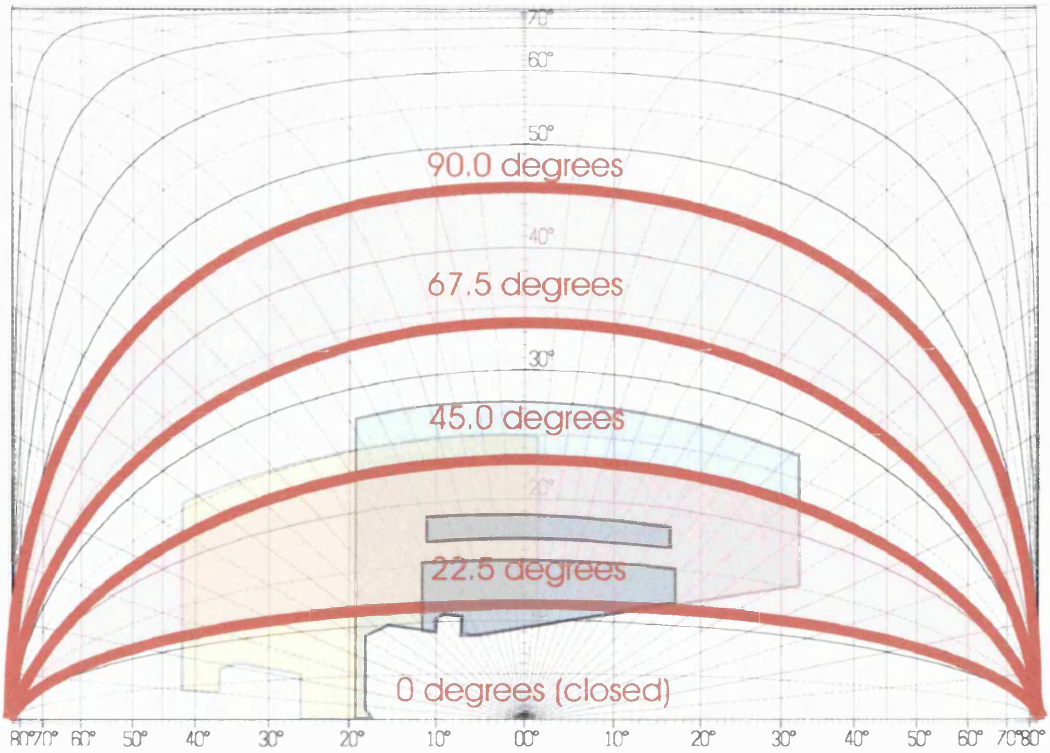


Figure 9.2: Waldram diagram showing how thresholds in the vertical wall altitude angle effect blind position (controlled using 22.5 Degree increments). The Figure demonstrates how the relationship used to position the blind is unable to take into account the relative azimuth of the sun.

9.6.2 Improving the Sky Assessment Procedures

Figure 9.2 illustrates that the 2 dimensional nature of current algorithms' control mappings mean that the sky is not partitioned enough for them to be able to decifer the sensitive areas for different contexts. If the framework is to be developed to recognise behavioural adjustments due to external climate and context, it must be able to cater for a three dimensional relationship between the occupant over-ride and the position of the sun. This can be achieved by dividing the sky into more zones. These zones can be mapped to control the position of the blind and these mapping can be adapted to respond to variables such as different desk positions, as demonstrated in Figure 9.2. Various methods for dividing the sky into zones were investigated.

9.6.3 The CIE Sky Zoning Method

One of the most common techniques used to divide the sky into zones is the CIE method, which divides the sky up into 145 zones in 12-degree bands of altitude for the purposes of sky scanning.[92] The zones are shown in the stereographic projection in Figure 9.3 and a routine that provides a zone number from the altitude and azimuth of a sky point can be found in Tregenza and Sharples (1993).[93]

Altitude of band centre (degrees)	*Number of zones in band*	*Azimuth increment (degrees)*	*Solid angle subtended by zone (steradians)*
6	30	12	0.0435
18	30	12	0.0416
30	24	15	0.0474
42	24	15	0.0407
54	18	20	0.0429
66	12	30	0.0445
78	6	60	0.0455
90	1	360	0.0344

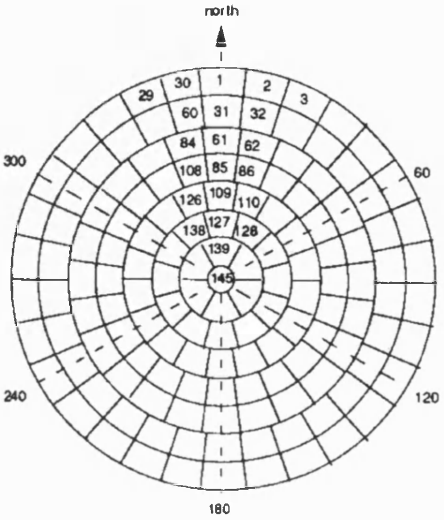


Figure 9.3: The CIE sky zoning system with a Table demonstrating how zones are allocated to different solar altitude.[93]

This method represented a tried and tested method of zoning the sky and other models are often based upon this basic model.[94] However, it was created to represent the luminous distribution of the sky and relies on complex mathematical formulae to determine the sky divisions based on constant solid angles and daylight contribution. These factors were not considered important to the development of the control methodology for horizontal shading elements, which are largely interested in the position of the sun in relation to the blind and simple formulae. Therefore a new technique was proposed that simply extended the sky assessment method outlined in Chapter Seven.

---

[92] Kendrick J.D., 1989, "Guide to recommended practice of daylight measurement", Commission Internationale de l'Eclairage [CIE], Vienna.

[93] Tregenza P. and Sharples S., 1993, "Daylight Algorithms", ETSU S 1350.

9.6.4 Creating a Sky Zoning Method for Blind Control

The best practice blind control algorithm developed in the test room study divided the sky in terms of the solar wall altitude angle to allow it to control the blind tilt to block the sun and maximise views for various solar positions (see Figure 9.2). Therefore it seemed logical to continue to use this angle, rather than the solar altitude angle used in the CIE method, to divide the sky for the purposes of blind control. The advantage of this approach is that the solar wall altitude angle is dependent on the orientation of the window and therefore each individual window can divide the sky up in relation to its own orientation, which is easier for an observer and designer to understand. Also, by using the vertical angle it would be possible to continue to map the zones to the required blind angles, and thus simplify the transfer of the energy efficient strategy to the new framework.

To deal with the scenario of reflected glare, the whole sky hemisphere was divided into zones rather than just the window-facing hemisphere (see Figure 9.4), as shown in the Waldram diagram of Figure 9.2.

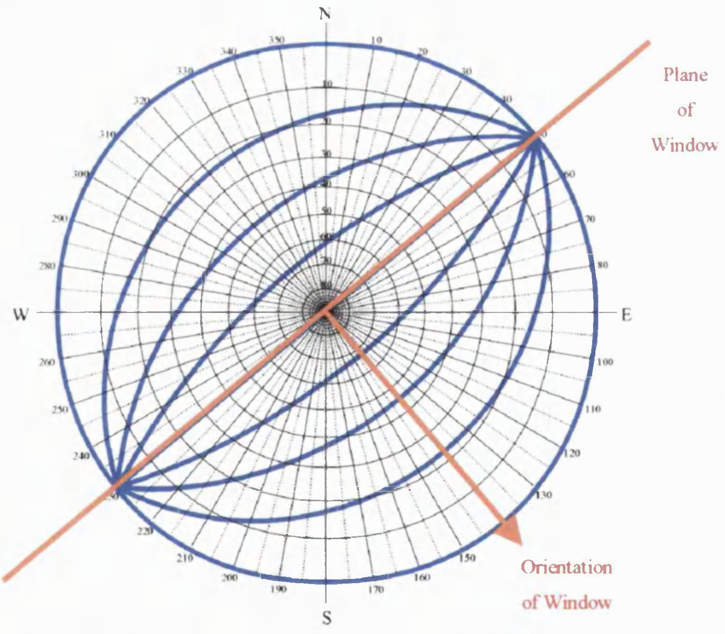


Figure 9.4: Stereographic projection showing the sky divided in terms of 22.5 degrees vertical solar wall altitude angles.

---

[94] Chain C., Dumortier D. and Fontoynont M., 1999, "A comprehensive model of luminance, correlated colour temperature and spectral distribution of skylight: Comparison with experimental data", *Solar Energy*, Vol.65, No.5, pp285-295.

To add the extra dimension necessary for the system to respond to geometric patterns, the solar wall altitude angle zones were sub-divided by a chosen relative azimuth angle. The sky zones could now be defined by referencing the solar wall altitude angle and the relative azimuth. For illustration purposes, Figure 9.5 divides the sky into 32 zones, and the window-facing hemisphere into 16 zones (see Figure 9.6).

Dividing the sky into this number of zones meant that the system could only control the tilt of blinds in the closed, 45 degree or horizontal positions. To allow the system to alter the blinds in the same incremental steps demonstrated in the test room study (i.e. 22.5 degree increments, see Figure 9.2), the sky would have to be divided up even more in terms of solar wall altitude angles. Figure 9.7 and Figure 9.8 shows how this can be done for solar wall altitude below 45 degrees (when the blind needed to be in a position other than horizontal to block the sun) in the window-facing hemisphere.

Figure 9.8 also illustrates that if we wish to pick out particular zones of the sky that may cause glare at illuminances under the typical 24,000 lux, then we would benefit from the sky being divided into yet more zones. Figure 9.9 and Figure 9.10 illustrate how this could be done by dividing the sky into 15 degree relative azimuth zones for solar wall altitude angles below 45 degrees in the window facing hemisphere.

If the occupant was very close to a largely glazed single story wall or situated in a room with a high level glazed area, then glare could occur in zones above this 45 degree angle. Therefore we may include these scenarios by continuing the relative azimuth dissection to zones above a 45 degree solar wall altitude angle (see Figure 9.11).

To illustrate the importance of the way the sky is divided, if we divide the zones in Figure 9.11 with a solar wall altitude angle below 45 degrees into zones with solar wall altitude angles of 11.25 degrees then the sky division begins to allow the glare zones identified in the test room to be defined reasonably accurately (see Figure 9.12).

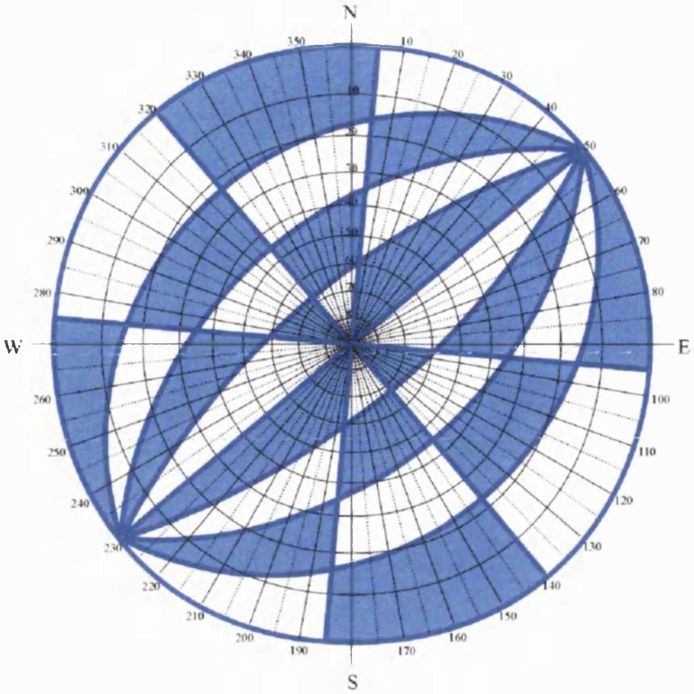


Figure 9.5: Stereographic protection showing 45 degree relative azimuth lines dissecting the vertical solar wall altitude lines to create sky zones that are related to the orientation of the plane of the window.

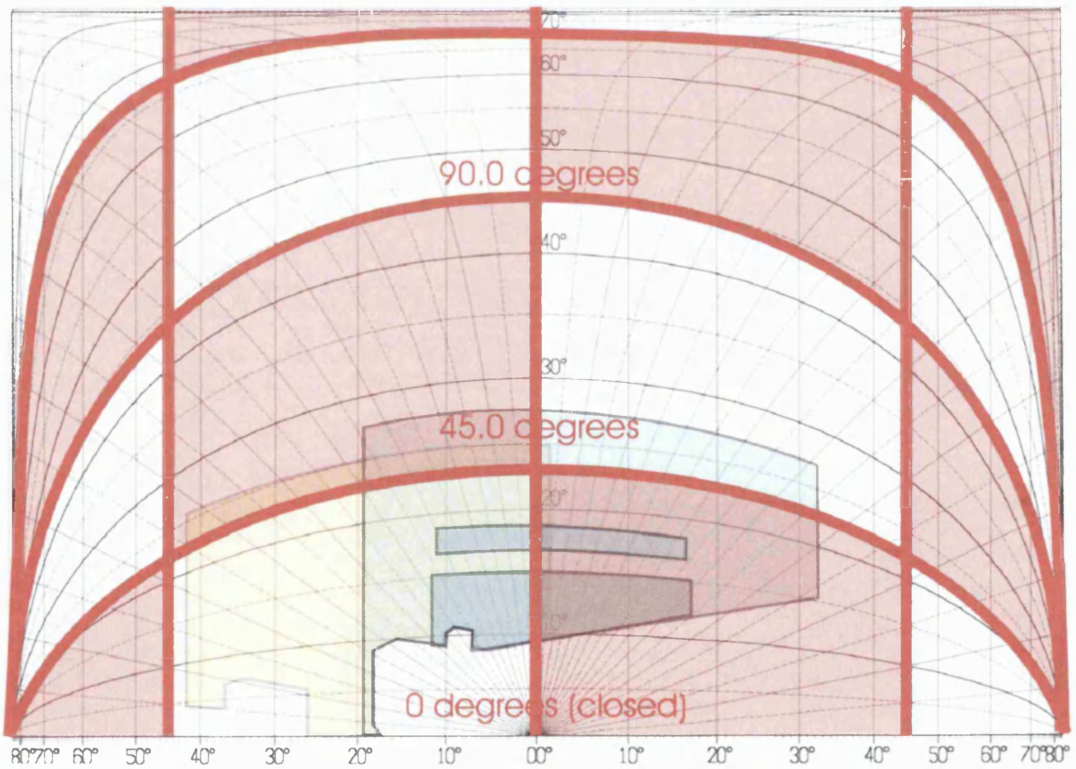


Figure 9.6: Waldram diagram showing the sky zones in the window facing hemisphere created using the sky zoning methods in Figure 9.5.

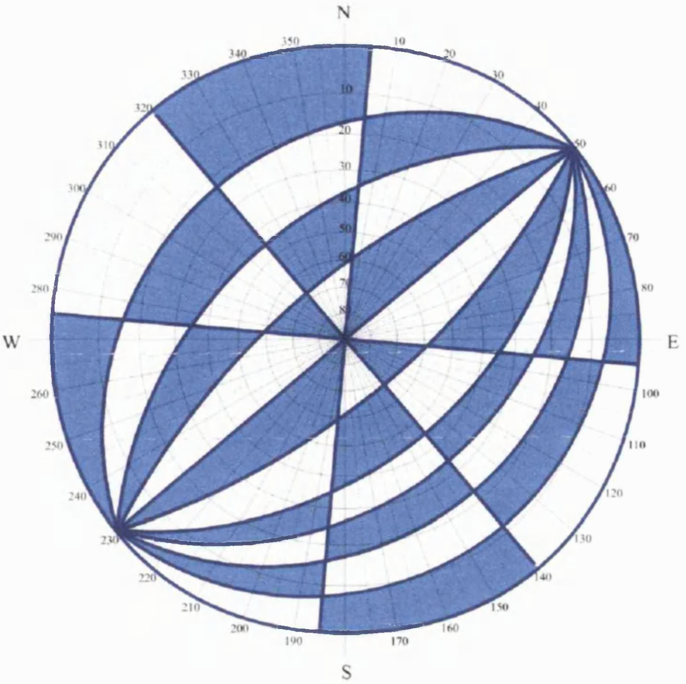


Figure 9.7: Stereographic protection showing 45 degree relative azimuth lines dissecting 11.25 degree vertical solar wall altitude lines for vertical solar wall altitudes below 45 degrees in the window facing hemisphere and 22.5 degree vertical solar wall altitude lines for the rest of the sky.

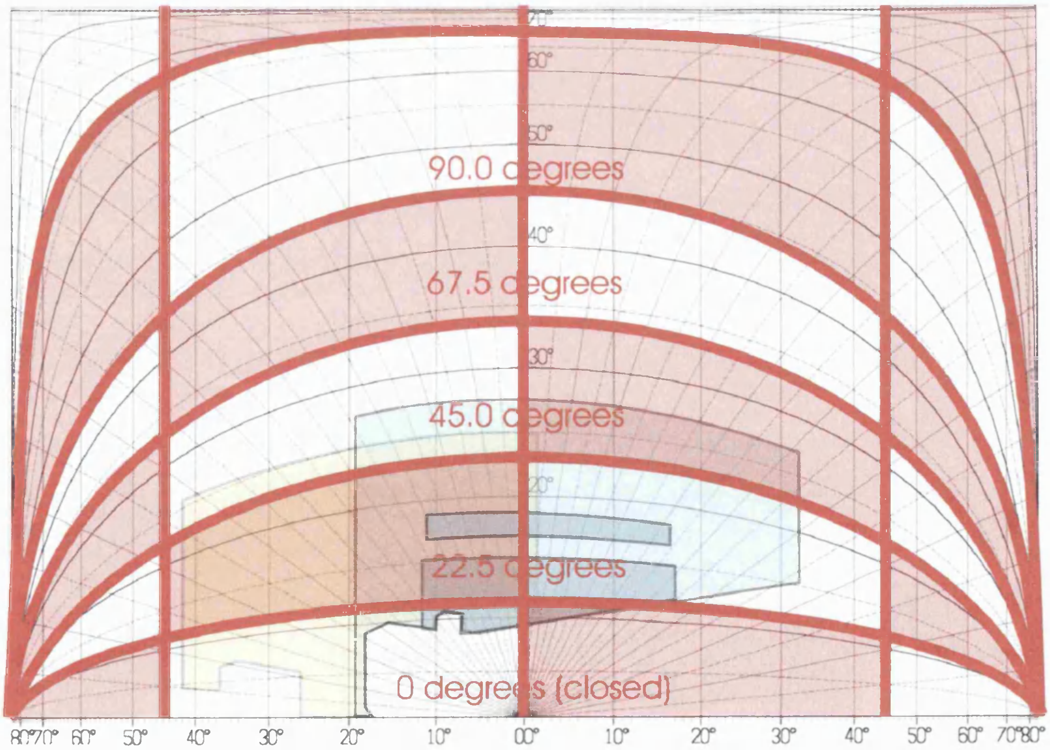


Figure 9.8: Waldram diagram showing the sky zones in the window-facing hemisphere created using the sky zoning methods in Figure 9.7.

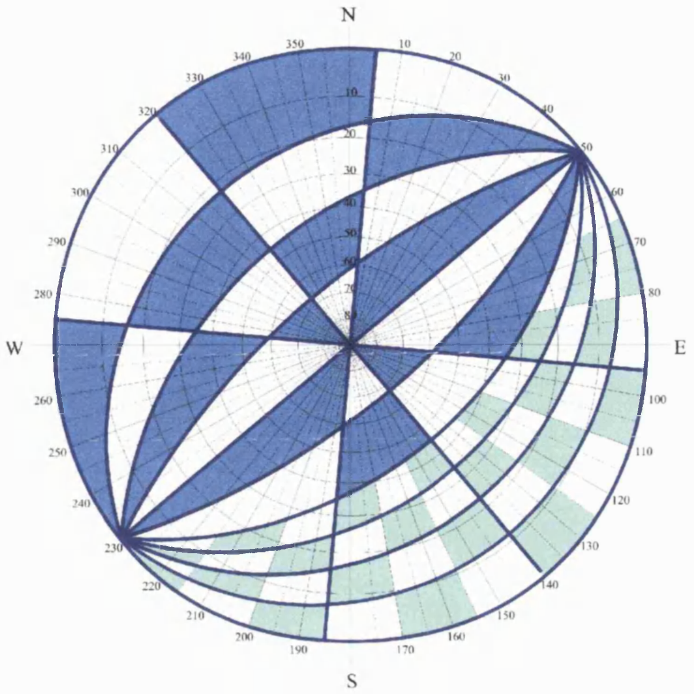


Figure 9.9: Stereographic protection showing 15 degree relative azimuth lines dissecting 11.25 degree vertical solar wall altitude lines for vertical solar wall altitudes below 45 degrees in the window facing hemisphere and 45 degree relative azimuth lines dissecting 22.5 degree vertical solar wall altitude lines for the rest of the sky

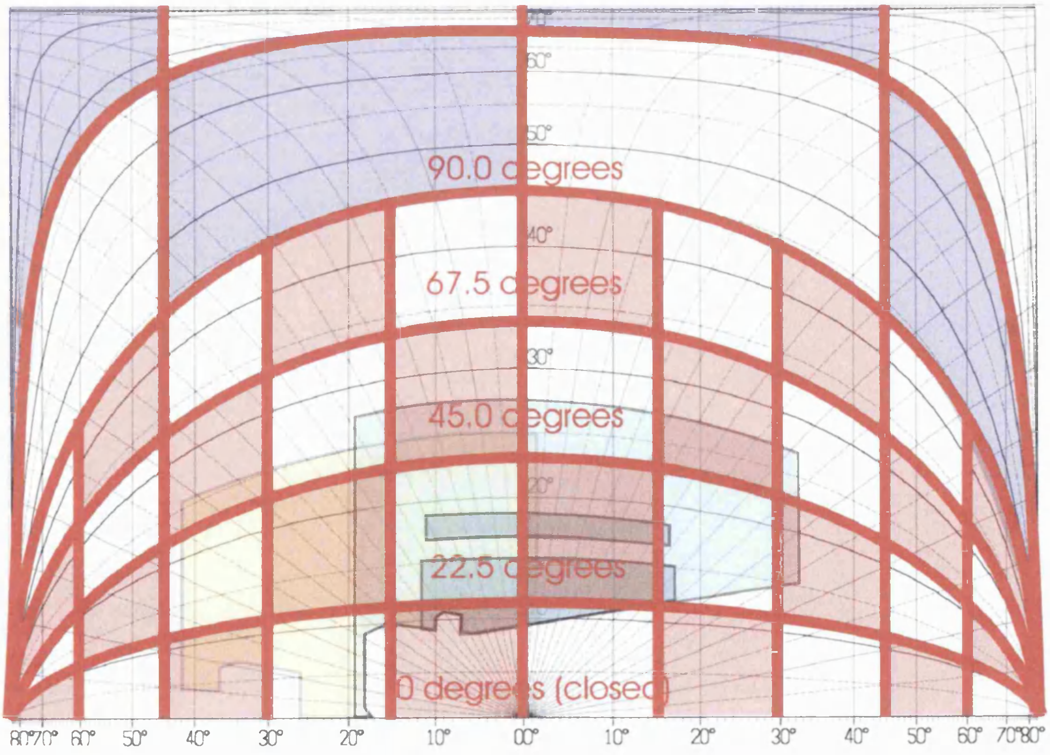


Figure 9.10: Waldram diagram showing the sky zones in the window-facing hemisphere created using the sky zoning methods in Figure 9.9.

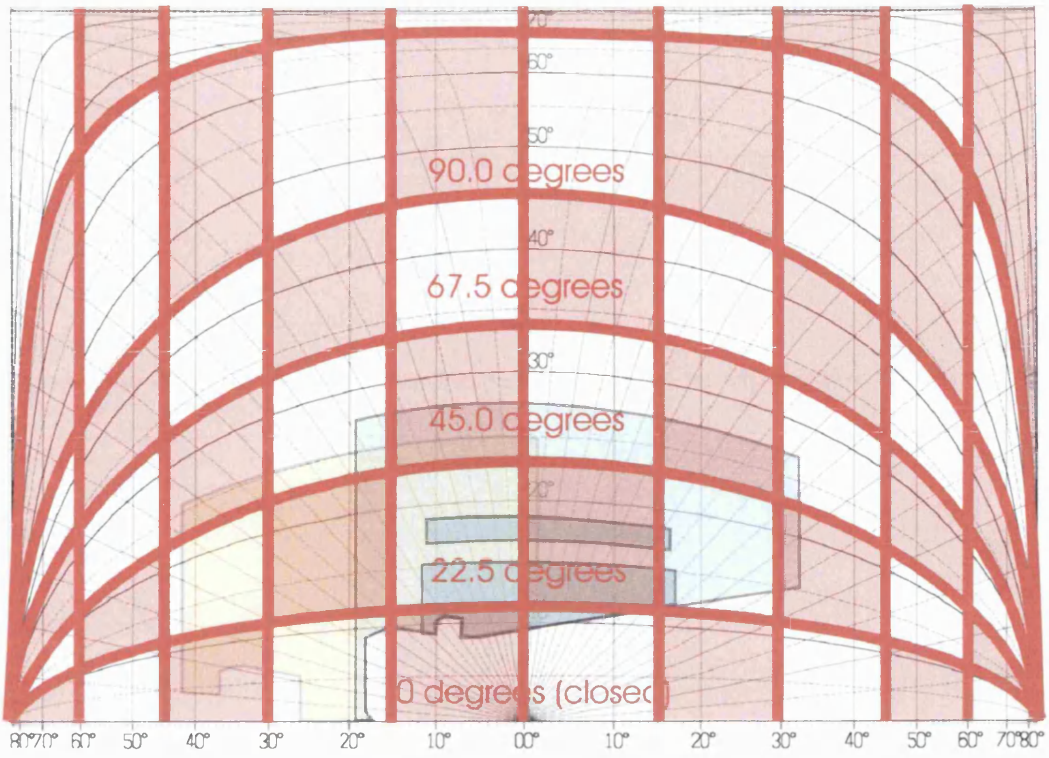


Figure 9.11: Waldram diagram showing the sky zones in the window-facing hemisphere created extending the sky zoning methods illustrated in Figure 9.10 to use a uniform 15 degree relative azimuth angle disection.

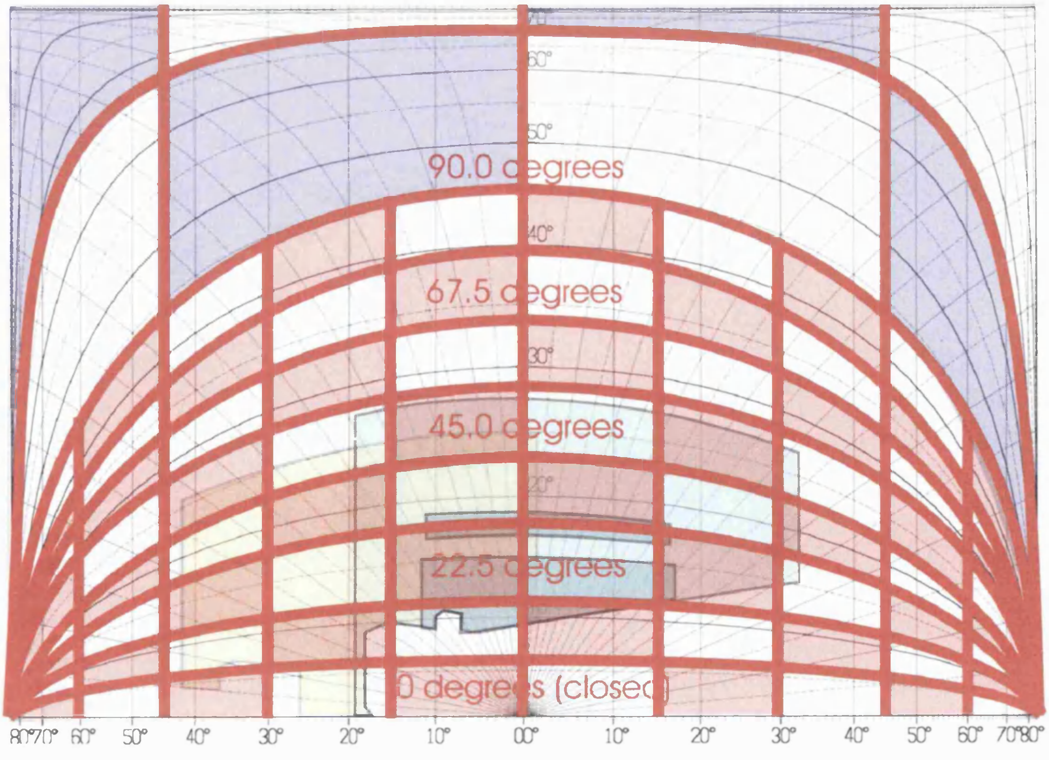


Figure 9.12: Figure 9.10 adapted for 11.25 Degree increments of Solar Wall Altitude Angle.

9.7 Chapter Summary

The experience gained in the test room set-up allowed the author to recognise a few key patterns of user frustration. This knowledge led to the proposal of a simple means of adapting an existing best practice control strategy to incorporate factors related to individual contexts, by dividing the sky into more distinct zones.

By proposing this methodology it would be possible to map a desired blind position to each individual sky zone and an external illuminance level. As a starting point, and for energy efficient operation, all of the zones within the same range of solar wall altitude angle have the same response, in a similar fashion to the best practice methodology outlined in the Chapter Seven. As the user interacts with the system, the desired blind angle or mapping for each zone changes to meet the user's requirements. The next chapter explores how these mappings might be achieved.

The number of zones necessary for allowing the system to recognise patterns requires more specific study. The test room allowed the author to experience a few contextual factors, but the number of sky zones necessary to provide effective blind control, for a wide range of users and contexts, could be much much larger. It has been shown that the more the sky is divided, the greater the system's ability to define particular problem zones in different contexts. Figure 9.13 shows how the whole sky could be divided by solar wall altitude angles of 11.25-degree increments and relative azimuths of 15-degree increments. However, it should also be noted that the more the sky is divided the more the user will need to interact with the system to teach it a satisfactory response in all of the problem zones. Therefore the total number of zones should be chosen in a detailed supplementary study, based on the likely system performance as well as the processor power of the controller.

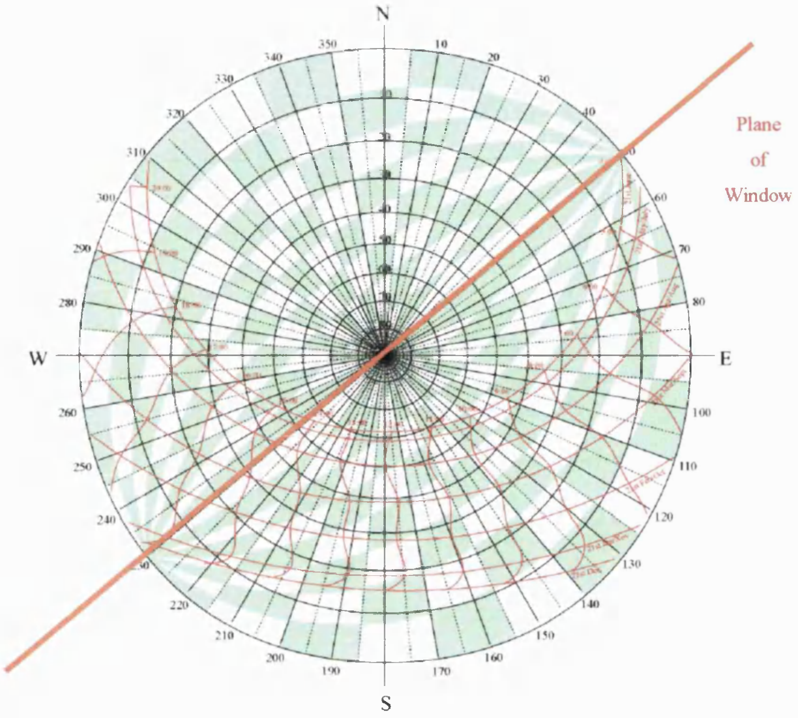


Figure 9.13: Stereographic protection showing 15 degree relative azimuth lines dissecting 11.25 degree vertical solar wall altitude lines throughout the sky hemisphere.

To accomplish learning on an individual occupant level, the system needs to be provided with some form of 'intelligence' to enable it to access and store individual user responses. This intelligence is essentially the interaction between the manual mode and the occupied mode. The next chapter rounds off the final details of the proposed solution to the problem by explaining how this might be done.

Chapter 10: The Use of Adaptive Algorithms

10.1 Introduction

The last Chapter proposed a new adaptive framework for automated blind control that incorporated important contextual factors, whilst not over-complicating the control set-up proceedures. This chapter builds on the base provided by the test room investigation by exploring how adaptive techniques could be incorporated within this framework, and using the Matlab control simulation software to demonstrate how the proposed system might work in practice.

10.2 Adaptive and Intelligent Algorithms

10.2.1 Adaptive or Intelligent Control

The interaction between the user and the automated blind system is complex and difficult to predict. Flexibility is one way of dealing with uncertainty in non-linear dynamic processes, but often unpredictable change can defeat flexible systems. Another approach is through the use of adaptive or intelligent modelling and control.

The field of intelligent control has been around for decades, but it has only recently found widespread application in the automotive, aerospace and manufacturing industries. Research into intelligent systems assimilates and integrates concepts and methodologies from a range of disciplines including artificial intelligence, neurophysiology, control theory, optimisation theory, and computer science.

Intelligent control does not claim to be artificial intelligence (AI) but a method that simply utilises basic AI elements in order to make better control systems. It is anticipated that by introducing learning elements into control systems, the systems will become more flexible and better able to deal with complex, real-world environments.

Users are good at dealing with complexity, but they are also error prone and sometimes lazy, and as a result are often unable to adjust a blind to optimise energy targets throughout the day. One aspect of trying to utilise adaptive/intelligent control within automated blind control is to allow a system to incorporate the creative, abstract and adaptive attributes of human interaction, while minimising the undesirable aspects such as unpredictability, inconsistency, fatigue, subjectivity and temporal instability.

10.2.2 Performance Characteristics of Adaptive or Intelligent Control Systems

Adaptive or intelligent controllers are generally self-organising and naturally able to cope with significant changes in a system and its environment while satisfying the control design requirements. To perform adequately, they should be able to function under significant process uncertainties by being robust enough to deal with unanticipated situations and errors. Indeed it is necessary to utilise a system that is able to respond approximately to input signals not contained in the training data, and produce similar outputs to similar inputs.

Most learning algorithms can be described according to their:

- *Accuracy* – ability to make the right decision. The situation after the decision should be better than before;
- *Convergence* –ability to find a solution to the problem; require minimal operator input and no detailed a priori knowledge of the process;
- *Robustness* – ability to stablise itself, i.e. that two devices do not interfere with each other in a way that makes their decisions oscillate or toggle (insensitive to noise and non-linearities);
- *Speed of response* –ability to respond within an acceptable time.

10.2.3 Types of Adaptive/Learning Architectures

There are many types of adaptive/learning architectures, all have different characteristics and are suited to different applications. This chapter attempts to identify the architecture that is most suited to being applied in future developments of the proposed system.

The three most popular learning algorithms are:

(i) knowledge based systems (or expert systems);
(ii) artificial neural networks (ANNs);
(iii) fuzzy logic.

These three main categories are not distinct and inter-relationships exist between all three, see Figure 10.1.

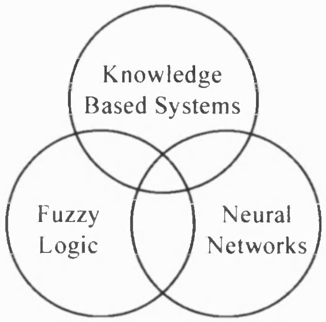


Figure 10.1: The relationship between the three main types of learning systems. [95]

Other methods do exist, such as genetic algorithms, which have been used in applications such as predicting natural light levels. [96] However, their application is limited and relatively unproven in the control field and therefore they will not be discussed in this work.

10.2.4 Knowledge Based Systems (expert systems)

A knowledge based control system is a flexible architecture for combining real-time algorithms and logic that can easily be implemented on a direct digital control (DDC) system. A block diagram of a typical expert system is shown in Figure 10.2.

---

[95] Brown M. and Harris C., 1994, "Neurofuzzy adaptive modelling and control", Prentice Hall International, Hemel Hempstead.

[96] Coley D.A. and Crabb J.A., 1997, "An artificial intelligence approach to the prediction of natural lighting levels", *Building and Environment*, Vol.32, No.2, pp81-85.

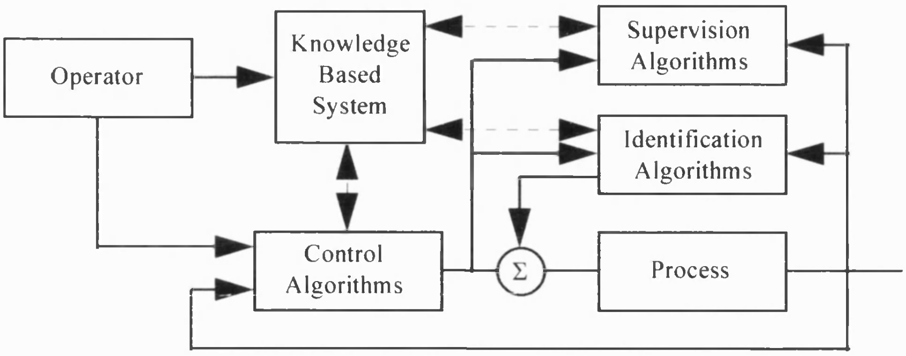


Figure 10.2: An example of an expert control system [97]

The system uses a collection of different algorithms for control, parameter estimation, diagnosis and supervision. A simple Proportional Integral (PI) control algorithm may be used when there is very little information about a process, and a more complicated algorithm, that attempts to optimise some performance index, may be used when information about a process has been obtained.

The knowledge based system tries to mimic the actions of an expert operator by using a previously assembled knowledge base; this consists of a collection of "IF.....THEN.....ELSE" statements, and inference procedures to decide which algorithm to use and when. This is done whilst also interacting with the true user or operator.

Examples of where expert systems have been applied within building control have shown them to outperform more traditional methods. Applications to date include:

(i) diagnosing building operational problems; [98] [99] [100]

(ii) analysing building energy consumption; [101] [102] [103]

---

[97] White D.A. and Sofge D.A., 1992, "Handbook of intelligent control: Neural, Fuzzy and Adaptive Approaches", Van Nostrand Reinhold, New York.

[98] Culp C.H., 1989, "Expert systems in preventive maintenance and diagnostics", *ASHRAE Journal*, Vol.31, No.8, pp24-27.

[99] Haberl J.S., Norford L.K. and Spadaro J.V., 1989, "Diagnosing building operational problems", *ASHRAE Journal*, Vol.31, No.6, pp20-30.

(iii) monitoring and tuning controls; [104] [105] and

(iv) design guidance.

10.2.5 Artificial Neural Networks

An artificial neural network (ANN) is a parallel processing dynamic system derived from a simplified model of the human brain. It consists of a large number of interconnecting and interacting nodes (or neurons), which by themselves are simple processing units but collectively form a powerful learning machine.

Signals are passed between nodes along weighted connections, whose weightings form the network's adjustable parameter. The arrangement of the network's nodes and connections defines its architecture and a number of possible variations exist. One popular arrangement is shown in Figure 10.3 where the nodes are arranged into layers and each node in one layer has connections only with nodes in the preceding layers. The output of each node is dependent on the weights of its inputs and its own firing function.

---

[100] Klima J., 1990, "An expert system to aid troubleshooting of operational problems in solar domestic hot water systems", *ASHRAE Transactions*, Vol.96, pt.1, pp1530-1538.

[101] Haberl J.S., Claridge D.E., 1987, "An expert systems for building energy consumption analysis: Prototype results", *ASHRAE Transactions*, Vol.93, pt.2, pp979-998.

[102] Haberl J.S. et al., 1988, "An expert system for a building energy consumption analysis: Applications at a University campus", *ASHRAE Transactions*, Vol.94, pt.1, pp1037-1062.

[103] Norford L.K., Allgeier A. and Spadaro J.V., 1990, "Improved energy information for a building operator; Exploring the possibilities of a quasi-real-time knowledge based system", *ASHRAE Transactions*, Vol.96, pt.1, pp1515-1523.

[104] Kaler G.M., 1990, "Embedded expert system development for monitoring packaged HVAC equipment", *ASHRAE Transactions*, Vol.96, pt.2, pp733-742.

[105] Kraus T.W. and Myron T.J., 1984, "Self-tuning PID controller uses pattern recognition approach", *Control Engineering*, pp106-111.

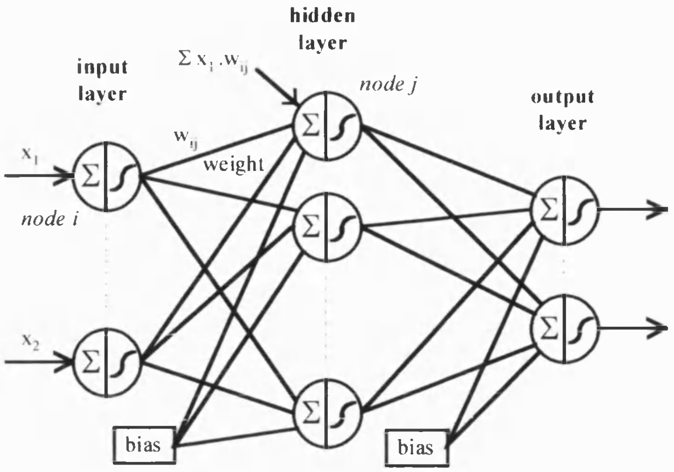


Figure 10.3: A feedforward multi-layer neural network where the lines between the circular nodes represent weighted links.

ANNs differ from expert systems in that they learn directly from example data instead of depending on pre-programmed rules derived from a knowledge base. In addition the information is stored as patterns of weights within the network architecture, not as a series of information bits that are easy to interpret, as used in standard computations. This fact means that the system's learning characteristics are partially opaque to the designer.

ANNs can be trained using a wide range of methods, one such method is backpropagation. [106] The weights are initially set as small random numbers and the objective of the training is to adjust the weights iteratively, so that the application of a set of inputs from a training data set produces the desired pre-determined set of outputs.

There are two types of training: supervised and unsupervised training. Supervised training requires the desired network output to be available so that the weights can be adapted to reduce the error. Unsupervised training organises the network structure based on the training inputs. For the on-line learning application proposed, the user

---

[106] Rumelhart D.E. and McClelland J.L., 1986, "Parallel distributed processing", MIT Press, Vols. 1 and 2.

over-ride would be used as a desired output in a supervised system that is initially trained on data.

The prime advantage of neural networks over other systems is their ability to generalise relationships as they show a strong level of robustness when faced with noisy inputs, such as erroneous user inputs.

Examples of where ANNs have already been applied within building control and have been shown to outperform more traditional control methods, include:

(i) analysing building energy consumption; [107] [108] [109]

(ii) monitoring and tuning controls. [110] [111] [112] [113]

10.2.6 Fuzzy Logic

Fuzzy Logic was first proposed in 1965 by Lofti Zadeh as a means of analysing by approximate reasoning. [114] Its vague nature provides a means for representing uncertainty as well as simulating the imprecision of human thought.

Fuzzy set theory extends classical set theory by allowing variables to take partial membership of a two or more sets. This enables a rule based system to have a smooth transition from one rule to another as opposed to the crisp on-off transition of a binary logic system.

---

[107] Kreider J.F. and Wang X.A., 1991, "Artificial neural network demonstration for automated generation of energy use predictors for commercial buildings", *ASHRAE Transactions*, Vol.97, pt.2, pp775-779.

[108] Curtiss P.S., Brandemuehl M.J. and Kreider J.F, 1994, "Energy management in central HVAC plants using neural networks", *ASHRAE Transactions*, Vol. 100, Pt.1.

[109] Dodier R.H. and Henze G.P, 1996, "Statistical analysis of neural networks as applied to building energy prediction", International Solar Energy Conference - ASME, pp495-505.

[110] Curtiss P.S., 1993, "Adaptive control of HVAC processes using predictive neural networks", *ASHRAE Transactions*, Vol.99, Pt.1, pp496-504.

[111] Hepworth S.J., Dexter A.L. and Willis S.T.P., 1994, "Neural network control of a non-linear heater battery", *Building Services Engineering Research and Technology*, Vol.15, No.3, pp119-129.

[112] Huang S.H. and Nelson R.M., 1994, "Delay time determination using an artificial neural network", *ASHRAE Transactions*, Vol.100, Pt.1, pp831-840.

[113] Curtiss P.S., Shavit G. and Kreider J.F, 1996, "Neural networks applied to buildings - A tutorial and case studies in prediction and adaptive control", *ASHRAE Transactions*, Vol. 102, Pt.1, pp1141-1146.

Fuzzy information is generally represented on a computer by a set of fuzzy rules which provide relationships between vague quantities. These relationships are typically linguistic production rules of an 'IF ....THEN' form and are characterised by membership functions (see Figure 10.4).

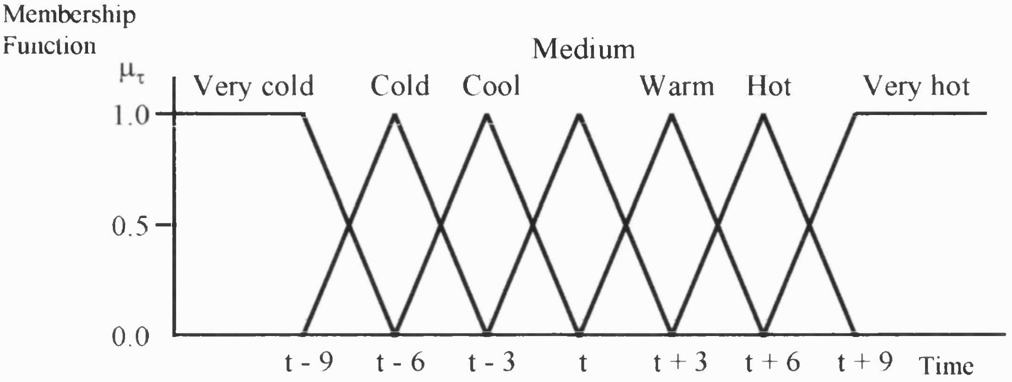


Figure 10.4: Diagram showing the possible membership functions for the linguistic terms that we use to describe temperature

The system uses these rules, their membership functions and an inference engine to make a control decision and adjust the adaptive machine, see Figure 10.5.

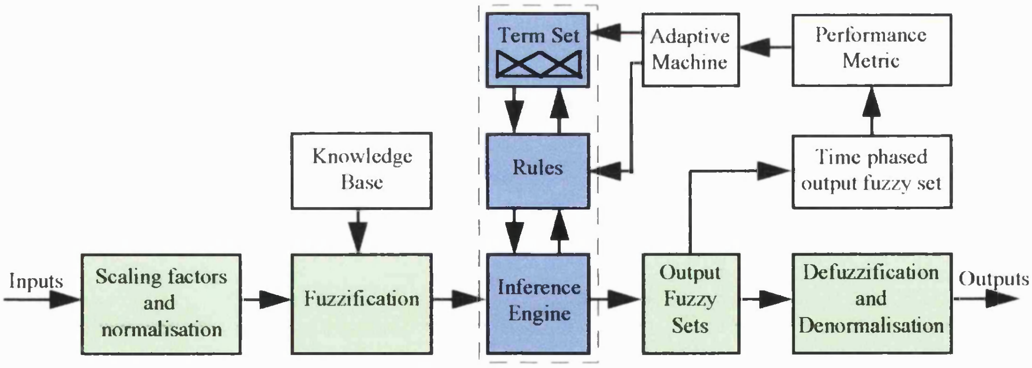


Figure 10.5: An example of a simple adaptive fuzzy logic control system

---

[114] Zadeh A.L., 1965, "Fuzzy sets", *Information control*, Vol.8, pp338-353.

Examples of where fuzzy logic has already been applied within building control and has been shown to outperform more traditional control methods, include:

(i) analysing building energy consumption; [115]

(ii) system control. [116] [117] [118] [119] [120] [121] [122] [123] [124]

However, in terms of using fuzzy logic as a means of controlling automated blinds, the most relevant work to date was work undertaken by Dounis in Athens, which demonstrated how fuzzy logic could be used to represent the fuzzy concept of occupant thermal and visual comfort and provide more flexible control systems. [125] [126] [127] [128] The work used computer simulation to incorporate various traditional physiological aspects of comfort control, such as Fanger's PMV, maintaining illuminance levels within acceptable limits and limiting glare indices.

---

[115] Shoureshi R., Brackney L. and DeRoo B., 1996, "Fuzzy based energy management system for large buildings", Proceedings of the ASME Dynamics Systems and Control Division, Vol.58, pp797-804.

[116] John R.W. and Dexter A.L., 1989, "Intelligent controls for building services", *Building Services Research and Technology*, Vol.10, No.4, pp131-141.

[117] Shoureshi R. and Rahmani K., 1989, "Intelligent control for building systems", Intelligent Control Systems, ed. Shoureshi R., ASME publication #H005491, NY, pp7-15.

[118] Huang S. and Nelson R.M., 1991, "A PID law combining fuzzy controller for HVAC applications", *ASHRAE Transactions*, Vol.97, pt.2, pp768-774.

[119] So A.T.P. et al., 1994, "Fuzzy air handling system controller", *Building Services Research and Technology*, Vol.15, No.2, pp95-105.

[120] Shoureshi R., Torcellini P. and Rahmani K., 1993, "Derivation and implementation of fuzzy optimal climate control", Proceedings of the American Control Conference, pp1860-1864.

[121] Rahmani K. and Shoureshi R., 1994, "Fuzzy based self-organising control for building systems", Proceedings of the American Control Conference, pp3060-3064.

[122] Arima M, Hara E.H. and Katzberg J.D, 1995, "A fuzzy logic and rough sets controller for HVAC Systems", IEEE WESCANEX '95 Proceedings, pp133-138.

[123] Kiff A. and Warwick K., 1996, "Distributed fuzzy logic for building management systems using local operating networks", IEE Colloquium (Digest), pp611-614.

[124] So. A.T.P., Chan W.L. and Tse W.L., 1997, "Self-learning fuzzy air handling system controller", *Building Services Research and Technology*, Vol.18, No.2, pp99-108.

[125] Dounis A.I. et al., 1994, "Thermal-comfort degradation by a visual comfort fuzzy-reasoning machine under natural ventilation", *Applied Energy*, Vol.48, pp115-130.

[126] Dounis A.I., Lefas C.C. and Argiriou A., 1995, "Knowledge-based versus classical control for solar-building designs", *Applied Energy*, Vol.50, pp281-292.

[127] Dounis A.I. et al., 1995, "Design of a fuzzy set environmental comfort system", *Energy and Buildings*, Vol.22, pp81-87.

[128] Dounis A.I., Manolakis D.E.. and Argiriou A., 1995, "A fuzzy rule-based approach to achieve visual comfort conditions", *International Journal of Systems Science*, Vol.26, No.7, pp1349-1361.

Although some facility for adapting the system's set points was included, the studies did not incorporate contextual factors into the algorithms or use occupant interactions as a means of adapting their system and therefore did not seem to offer much more than current systems. However, the work is useful for demonstrating the advantages and possible applications of fuzzy logic in environmental control systems.

10.2.7 Selecting an Architecture for the Proposed Framework

In order to develop an adaptive automated venetian blind system that meets the targets set for it, an appropriate adaptive control architecture needs to be selected using the relevant *a priori* knowledge set out in this thesis so far.

Knowledge acquisition lies at the heart of expert system development and can often continue indefinitely as more experience is gathered and the systems are improved. However, knowledge based systems require a structure that reflects the chronological order of making decisions. In blind control, the occupant's decision-making process is not clearly defined, therefore the application of Knowledge Based Systems alone is difficult.

Neural networks on the other hand, are able to cope with unknown and uncertain situations because they have the ability to generalise and demonstrate convergence and stability over a wide range of situations. However, the internal representations of ANNs are partially opaque, which makes it difficult for the designer to refine the system when it is learning and ensure correct operation during an on-line application. Another shortcoming is that they require large amounts of system specific data to allow the designer to select the correct network architecture and to train them to respond to real data adequately. As a result, neural network system design can more often than not be a process of trial and error, as good training results are no guarantee of good real system results when facilitating on-line learning.

Unlike neural networks, adaptive fuzzy algorithms provide some functional transparency through the state inter-relationships or dependencies incorporated within their underlying rule structure. This structure simplifies initialisation or validation processes for the designer.

After extensive reading on each technique, it was decided that a hierarchical and functional decomposition of the problem into subtasks was needed, using expert systems and fuzzy logic techniques, and that an architecture able to accommodate the complex behaviour that arises from the way an individual interacts with a blind, such as a neural network, should be adopted. However, this architecture should not be dependent on extensive data collection and trial and error commissioning.

The architecture chosen to meet these requirements, was an on-line NeuroFuzzy Adaptive Control architecture developed by Brown and Harris in their book 'NeuroFuzzy Adaptive Modelling and Control" published in 1994.[95]

10.3 NeuroFuzzy Control

Within a NeuroFuzzy system a knowledge base is incorporated so that the complexity of the system can be reduced to a set of linguistic rules rather than a series of complex mathematical algorithms. The use of fuzzy logic enables the system to make decisions where a degree of uncertainty is present and to enhance the user/control system interface by giving the control system a better chance at understanding the fuzzy concept of comfort. However what distinguishes this type of fuzzy system to any other is the fact that its fuzzy routines are incorporated in a lattice-based neural network architecture called an Associative Memory Network (AMN), thus providing the system will the ability to adopt a variety of learning algorithms if necessary, see Figure 10.6.

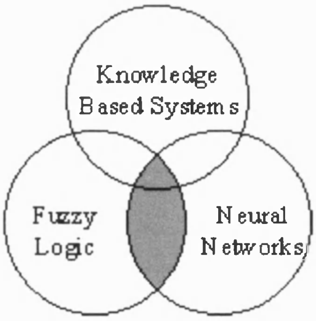


Figure 10.6: The area to be investigated within future work combines techniques and concepts from fuzzy, neural and knowledge based algorithms.[95]

10.3.1 Associative Memory Networks (AMNs)

AMNs are feedforward, supervised ANNs. They are universal approximate algorithms that can incorporate *a priori* knowledge into their structure. They are suitable for use with instantaneous training methods and are formed from the weighted sum of local bias functions, which are often defined over a small region of the input space.

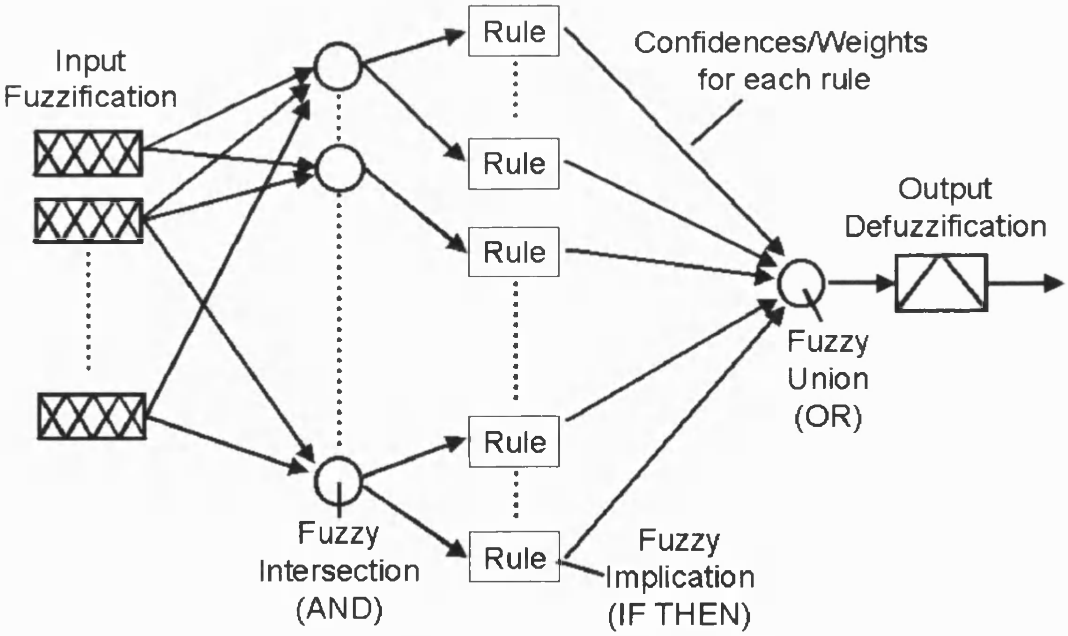


Figure 10.7: Diagram showing the typical architecture of an Associative Memory Network[95]

For these single layer AMNs, which are structured in a similar way to a Neural network, learning or adaptation occurs through weight or belief adjustment rather than structural change.

10.3.2 The Prime Advantages of AMNs

The main advantage of the AMN structure compared to other ANN structures is its ability to represent variables locally on the input space and use simple local linear mappings that produce a fast response. This way desired functions and information can be stored locally and generalisation can occur locally.

The advantage of this style of adaptive controller over any other style investigated is that they:

- are easy to train because they allow conventional linear learning to be applied to a non-linear control problem;
- enable *a priori* functional knowledge to be incorporated into the network structure;
- are robust to noise due to localised learning.

10.3.3 Fuzzy AMNs

A number of methods are available for representing the overlapping input and output mappings of an AMN. These include:

- Gaussian Radial Basis Functions;
- Cerebellar Model Articulation Controllers (CMAC);
- Basis B-spline networks;
- Fuzzy Logic controllers.

The advantage of using the fuzzy controllers over the other forms of AMNs is that although they are have similar training capabilities, their natural linguistic interpretation allows them to be functionally transparent to the designer and the commissioning engineer.

It might be thought that having a transparent structure may not be useful for systems that learn on-line, however it can be extremely useful for initialising the network using prior knowledge, as is required for the initial energy efficient mode described in the last Chapter, or if a system needed to be adapted or developed further. Indeed if an algorithm can learn, it can also forget, therefore it should be possible to verify whether the behaviour being stored is desirable.

10.4 Applying the Adaptive Architecture to the Framework

To demonstrate how a NeuroFuzzy Architecture could be applied to the framework proposed for the future development of automated blind control, a simple system model was created using the Matlab mathematical software. The purpose of the model was not to create a system that would be able to cope with the complexities of human behaviour, such a task would require many years of further research and development. The purpose was infact to reinforce the new approach being put forward, by proving that a methodology for implementation was available and thus providing a starting point for future work.

10.4.1 The Inputs

To simplify the illustration, the model divided the complete hemisphere of the sky into 32 zones using the simplified sky zoning technique described at the end of the last chapter (see Figure 9.5). Figure 10.8 to Figure 10.10 demonstrate how these zones were defined using two fuzzy logic sets representing Relative Azimuth and Solar Wall Incident Angle. The definition of the method used to intersect these sets, to produce a single zone, can be found in Appendix X. A solar positioning algorithm would calculate where the sun was as any time, in the same manner that it did in the energy efficient strategy but the two singleton inputs were converted to a fuzzy linguistic description of a sky zone.

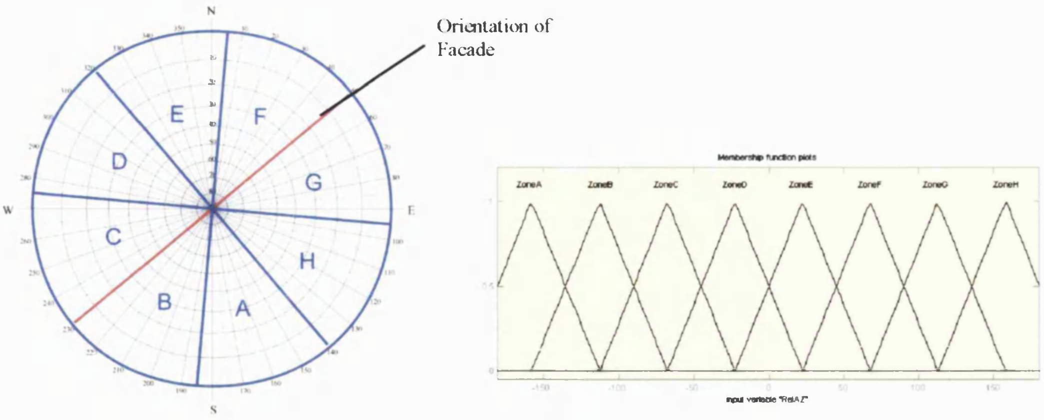


Figure 10.8: Input 1 – The Fuzzy Relative Azimuth Zones (RelAz)

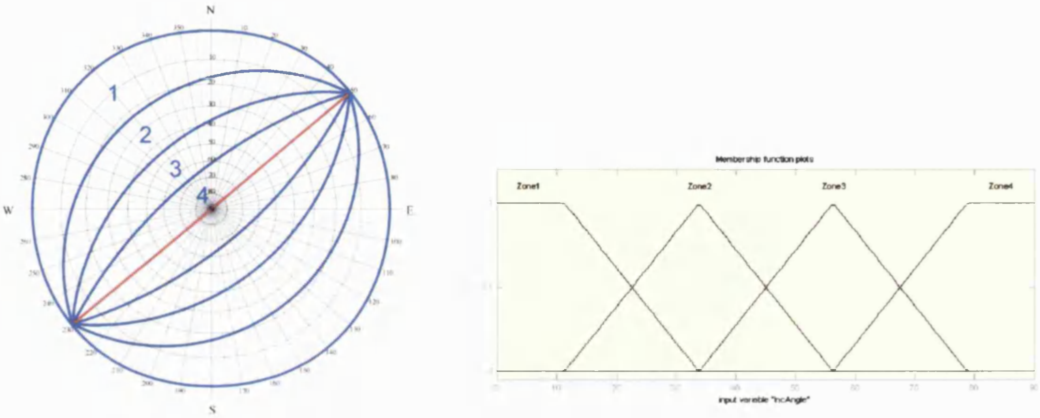


Figure 10.9: Input 2 The Fuzzy Solar Wall Altitude Zones (IncAngle)

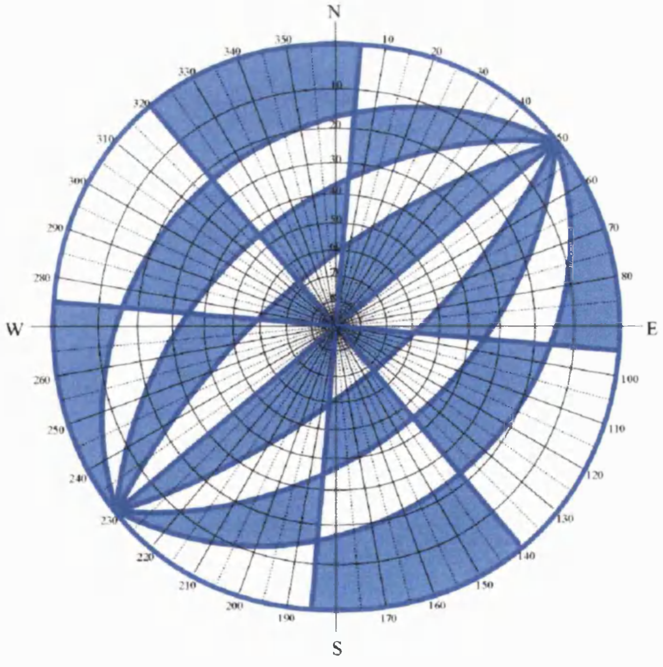


Figure 10.10: The sky zones

The use of fuzzy techniques in the framework can be justified by considering the fact that although, altitude and azimuth are traditionally crisp (non-fuzzy) values, when calculated with a solar positioning algorithm at an interval of every 3 minutes, they become slightly fuzzy in order to take into account the errors discussed in Appendix VIII. Also solar glare, which was shown to be related to the position of the sun, is not crisp. As seen in Chapter 8, the boundaries of troublesome areas of the sky can be fuzzy due to a number of atmospheric and contextual conditions. Therefore the advantage of using a fuzzy separation of the sky is that it will eventually allow a system to categorise a source of glare with membership of more than one sky zone, helping it deal with ambiguity and learn pattern in the input sets.

A third input set was then defined to represent the solar illuminance measured from the sun sensor. Again for simplicity the set was divided into only three membership function types: overcast, intermediate and clear (see Figure 10.11). These functions were derived intuitively from the basic observations made and the experience gained whilst interacting with the blind in the test room and monitoring the solar illuminance data. However, it is clear that before future developments of the framework can be made, a detailed study of the relationship between orientation, sky bightness and occupant interaction is required so that the sky can be classified into a larger number of brightness labels.

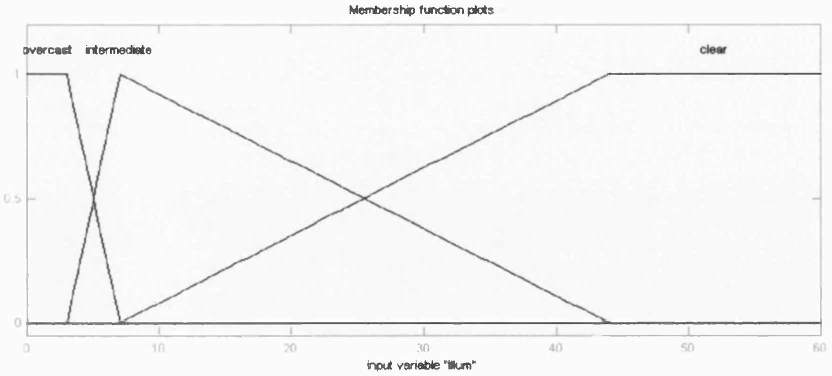


Figure 10.11: Input 3 – The Fuzzy Sky Illuminance Classification (Illum)

The two-dimensional fuzzy intersection of the sky zones, seen in Appendix X, was then extended to include the solar illuminance reading and a three-dimensional input space was created.

10.4.2 The Outputs

Two output sets were then defined. The first set being the vertical position of the blind: fully down or fully up (see Figure 10.12).

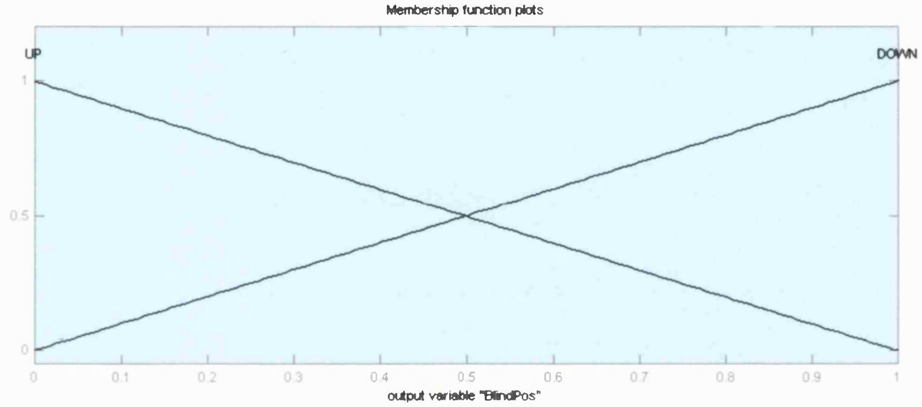


Figure 10.12: Output 1 – The Vertical Blind Position (BlindPos)

The second set was the tilt angle of the blind. Again for simplicity of presentation, this was modelled with only three membership function types: closed - 0 degree angle, 45 degree angle and open – 90 degree angle (see Figure 10.13).

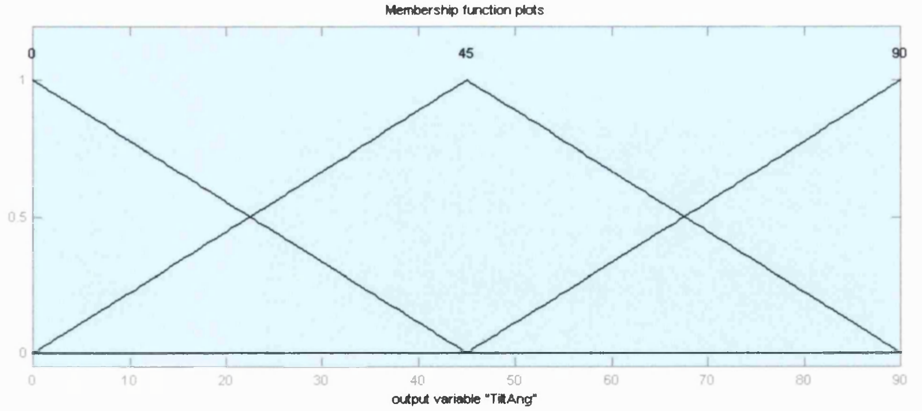


Figure 10.13: Output 2 – The Blind Tilt Angle (TiltAng)

10.4.3 The Inference Engine

The fuzzy AMN inference engine architecture is used to relate the fuzzy inputs to the fuzzy outputs (see Figure 10.14). This is done by utilising a series of fuzzy inference operators in parallel to create expert system rules that fire depending on the input space, and whether they have a confidence (conf) greater than zero.

As a starting point these rules were set up to provide the same functionality as the energy efficient strategy developed in the test room study. For example rules included:

- IF the RelAz Zone =A AND IncAngle Zone = 1 AND Illum=Clear
  THEN BlindPos = UP and TiltAng = 0 Conf=1
- IF the RelAz Zone =B AND IncAngle Zone = 1 AND Illum=Clear
  THEN BlindPos = UP and TiltAng = 0 Conf=1
- etc

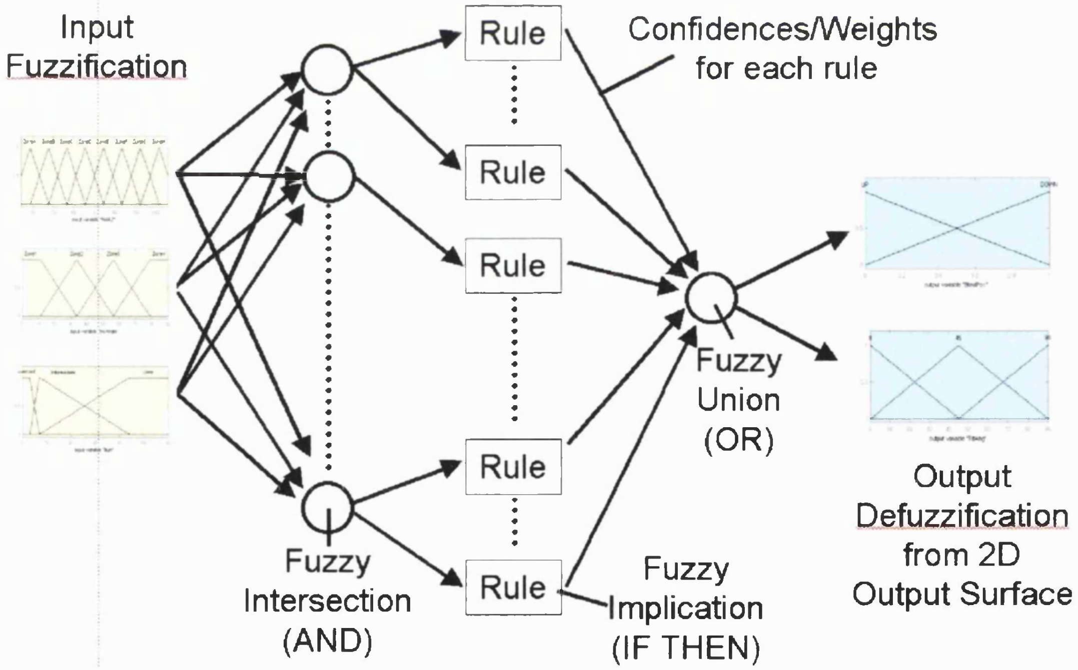


Figure 10.14: Diagram showing the basic form of the Associative Memory Network Inference Engine

Both Figure 10.15 and Figure 10.16 represent how the rules created in the inference engine mapped the inputs to the outputs to creat the energy efficient control strategy developed in Chapter Eight. We can see by looking at Figure 10.15 that the tilt angle output changes dramatically around the 24,000 lux sky illuminance threshold when the solar wall altitude angle is below 45 degrees. We can also see from Figure 10.16 that the vertical position of the blind is also affected by the sky illuminance threshold when the sun is on the window facade.

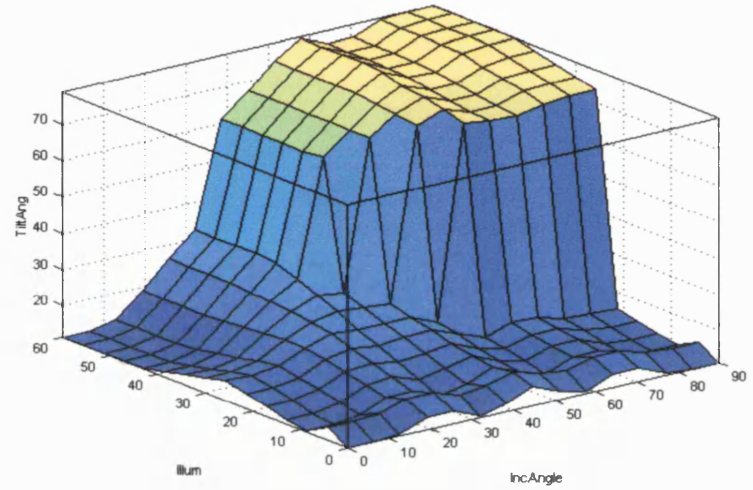


Figure 10.15: A Graph showing the relationship between the Solar Illuminance and the Solar Wall Altitude Angle Inputs and the Blind Tilt Angle Output

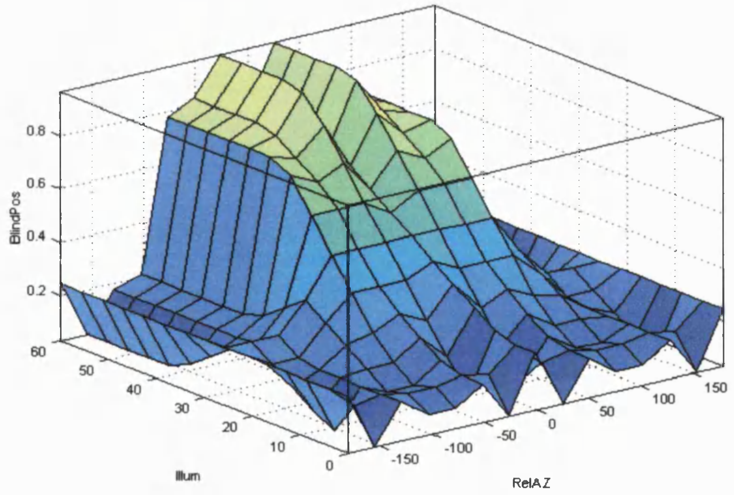


Figure 10.16: A Graph showing the relationship between the Solar Illuminance and Relative Azimuth Inputs and the Vertical Blind Position Output

This initial mapping was essentially an energy-saving two-dimensional controller described in Chapter Seven and Figure 9.1. One copy of this mapping should be kept for energy efficient mode operation and to provide the option of reseting the controller, the other should be adapted through user interaction.

10.4.4 The Adaptive Mechanism

In proposed system, a change in input conditions generates a request to test the rule base to determine whether an action is needed. If a user override occurs, then a request is sent to determine the overall condition of the system at the time of change and to adapt the weightings or confidences accordingly.

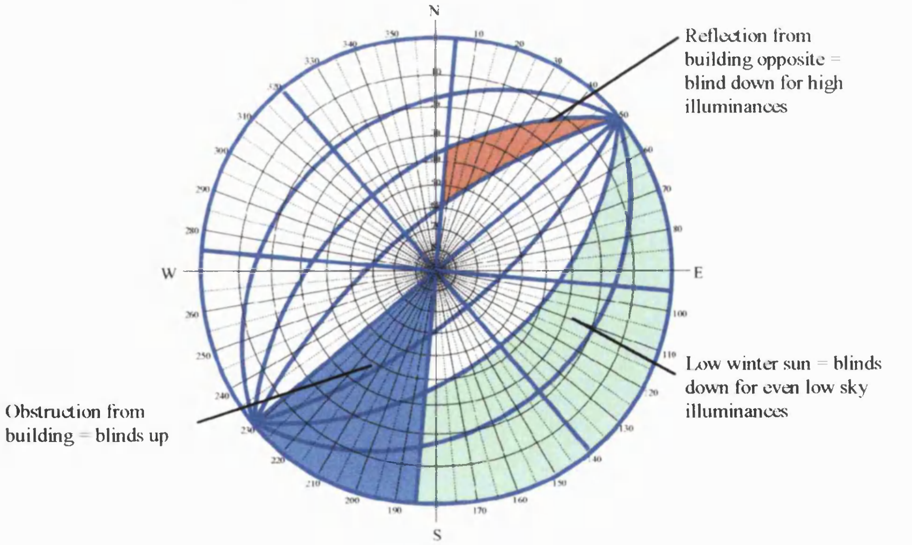
Adaptation within a NeuroFuzzy system is achieved by using an adapative mechanism to adjust the weights and confidences of each rule. The overall modelling abilities of a controller do not generally depend on a particular mechanism used, but are established theoretically by proving that an optimal system exists. Whether or not an adaptive procedure can modify the network's connections such that it approximates this optimal system, and its ability to be able to perform these operations in a reasonable time, depend on the network's architecture. It may be that a particular learning rule is appropriate for training a network because its convergence rate is slow or it may stop at a locally suboptimal solution. But once the architecture of the system is developed these learning algorithms can be adapted as more experience and data is gathered about the way in which users react to such systems.

According to Brown and Harris (1994), the NeuroFuzzy architecture selected, with its sparse internal representation and its linear set of adjustment parameters, is suited to on-line learning with simple instantaeous learning algorithms.[95] The authors also identified a number of traditional adaptive mechanisms available for use in NeuroFuzzy techniques, including, fuzzy interpretation and instantaneous gradient descent. However, despite being transparent to the designer and the process being easier, finding the right combination of rule and weight adjustment algorithms, still requires a certain amount of trial and error development, which fell outside the realms of this particular thesis.

10.4.5 Starting with a Simple Reinforcement Mechanism

The use of field bus technology within the proposed framework does limit the ability to implement learning agents that require large amounts of memory and computational power, such as conventional neural networks using back propagation and gradient descent learning rules. Therefore for the purposes of this work, a simple reinforcement system was used to illustrate the framework would be able to adjust the control output of the system over time using occupant interaction alone. The system simply punished rules which envoke a user override by reducing their rule confidence by 0.2, and rewarded rules, which had the user's chosen response by increasing those rules' confidences by 0.2. This meant that to make a permenant change to the system's response, a user had to excecute the same over-ride response under the same input conditions three times.

To illustrate this, a series of hypothetical user overrides were created as a response to certain climatic and contextual events (see Figure 10.17). These inputs were gradually fed into the system until the confidence of each effected rule had been changed five times. The effect these gradual changes had on the output surface that described the relationship between the sky illuminance, the relative azimuth and the vertical blind position can be seen in Figure 10.18.



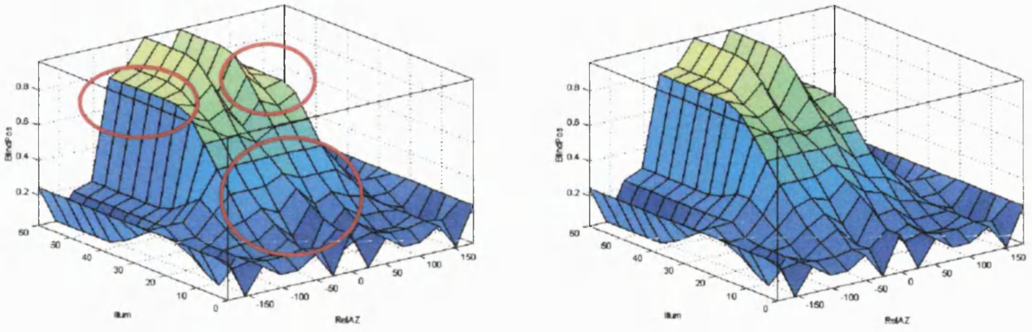
Reflection from building opposite = blind down for high illuminances

Low winter sun = blinds down for even low sky illuminances

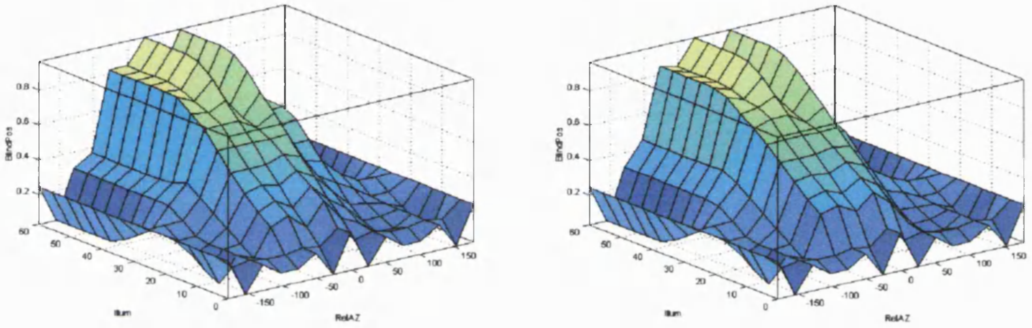
Obstruction from building = blinds up

Figure 10.17: The model user interactions given to the adaptive system ploted on a stereographic projection with the sky zones

(i) Initial Energy Efficient State (ii) Surface After One Set of Over-rides



(iii) Surface After Two Sets of Over-rides (iv) Surface After Three Sets of Over-rides



(v) Surface After Four Sets of Over-rides (vi) Surface After Five Sets of Over-rides

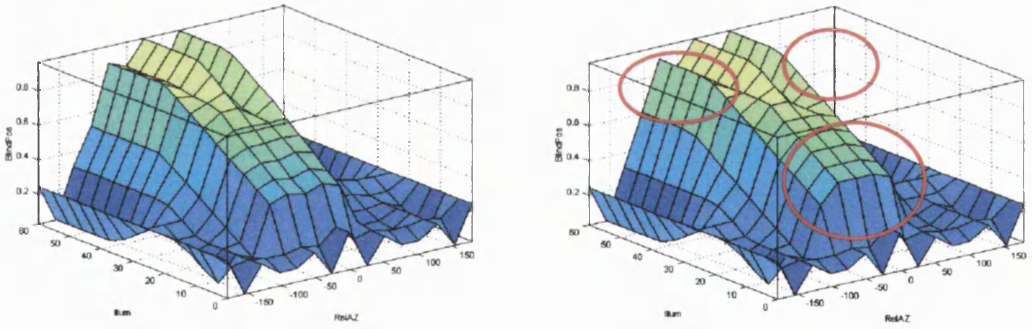


Figure 10.18: Figures showing a controller output surface before, during and after a period of adaptation, with the major changes highlighted in red.

10.5 Chapter Summary

The main reasons for choosing to use an adaptive or learning mechanism were:

- to take into account design information about the type of blind being used, which is often partially known, or unknown;
- to accommodate the time-varying operational environment, which are also only partially known and highly variable;
- to improve the performance of the blind over a wide range of operating conditions;
- to decrease, but not eliminate the required human intervention;
- to increase flexibility of the control system; and
- to reduce the initial design time, as well as installation and maintenance costs.

The methodology undertaken in this thesis has been to simplify any adaptive processes used by studying and designing the nature of the input space to an algorithm. The sky division technique outlined in Chapter Nine reduces the computational complexity of an adaptive algorithm by incorporating factors identified earlier in the study as being important to both a user's and a control system's interpretation of the environment.

Additional research into the accommodation of real-time adaptive control within the framework focused on adaptive algorithms that were linear in their adjustment connections. It was found that the use of non-linear models required more knowledge and data about the way in which people interact with the system, to enable the system to be trained. To date no such information is available in a readily assessable form. Therefore this type of knowledge acquisition is scope for future work.

The software model, discussed in this chapter, can be seen as the first step in this process. The inputs used in the example represent an ideal user, who provides similar overrides for certain scenarios. In reality, data from the user would be very noisy due to an individual sometimes reacting differently to similar inputs. It is hoped that future work will implement, test and fine-tune the techniques used here; by checking their validity in real world situations, and that developments will be made to the framework to ensure that it performs satisfactorily, in terms of accuracy, convergence, robustness and speed of response, whilst still maintaining its simplicity.

However, it should also be remembered that the problems of designing a network to learn on-line are greater than designing a network 'off-line', as many of the validation and verification proceedures must be performed autonomously. This can have an effect on computing resources and therefore the idea of spliting this functionality between nodes on a decentralised LonWorks network is attractive.

In order for an on-line adaptive mechanism to learn successfully on its own, a protective layer, or filter, is required to shield the learning element of the controller from poor data. For example, if the blind motor failed, or the controller lost communication with its sensors, the adaptive mechanism must be turned off. Such protection is simple to achieve through the LonWorks infrastructure. Each device is capable of signalling an error if communication to other important devices is lost. An interoperable internal light sensor can pick up motor failures by signalling that there is no change in internal illuminance levels during blind movement commands.

Another example of the type of protection the adaptive algorithm may require is an adaptation time. This would allow an occupant a certain period of time to adapt to the environment within a room they have just entered.[129] If an adjustment is made before the adaptation time has expired (say 5 mins), the necessary immediate response is given, but the user interface does not register the adjustment with the learning system.

The requirement for this adaptation time can be illustrated by considering an occupant who has just entered a daylit room from an artificially lit corridor. The contrast between a naturally lit room and a brightly artificially lit corridor is likely to make to occupant close the blinds.[41] However at other times, with the similar conditions and the occupant having been in the room for some time, having the blinds open may be satisfactory.

Many other filters could be added but a balance needs to be made between providing too much protection, where the controller only sees a small proportion of the data, and not enough, where the controller cannot follow any pattern. Again a great deal of research is needed in this area before any progress can be made. However, the framework that has been developed here easily allows for a protective jacket to be incorporated and for the complexity of this protective jacket to be developed alongside the learning algorithm.

The author recognises that validation studies utilising real spaces and blind systems are important when computer models are being applied, but such tests were not possible within the timescale of this research. It is hoped that the model developed will provide a starting base for further research into control system techniques.

---

[129] Fountain M. et al., 1994, "Comfort control for short-term occupancy", *Energy and Buildings*, pp1-13.

Chapter 11: Discussion

The aim of the research outlined in this thesis was to propose a solution the 'real-world' problem of how to improve automated blinds to better meet the needs of their users. This was an ambitious aim and required a direct and multidisciplinary problem solving approach to be taken if a solution was to be found within the timescale set aside for the study.

Firstly, the problem had to be defined and then the method of solution discovered. The definition relied on a detailed background literature study, which was used to catergorise user's needs and current blind control techniques. The key issues identified were that the factors affecting individual control were complex and often individual to a user and their context, and that current control systems were either too simple or too complicated to deal with this fact. This work led to the study's hypothesis:

Can existing control strategies be developed to incorporate factors that have been identified as being important to an 'individual' user, without the need for a complex management intensive structure?

Having defined the problem and carried out the initial background research, it became clear that the broad range of issues involved meant that a solution would only be found by exploring a number of different avenues, using a variety of techniques, some of which may fall outside the confines of academic studies.

For example, the originality of the proposal that automatic blind control systems should be designed to deal with individual users and their contexts in mind meant that there was little social science theory or practical based research available to inform empirical studies of the use of venetian blinds. Therefore, a detailed study into individual user control was required. At the same time, the proposal of a new control system meant that the author had to obtain a great deal of knowledge about the nature of automatic blind control systems, standard building management systems, and any recent technological developments.

It was decided that both aspects of this work could be investigated by undertaking an experiential study that enhanced the author's knowledge base on control systems by setting up a test room and providing the author with their own individual user experience. A key aspect of this was that the author experienced a standard, off-the-shelf system configured to work in a current 'state of the art' manner. Therefore only a few simple off-the-shelf sensors could be used, as would be the case in a real-world system, which limited the data collection opportunities associated with standard experimental set-ups.

A decision had to be made on whether to pursue the solution to the problem through this experiential problem solving approach or to carry out an experimental piece of research on one of the supporting areas, and not attempt to resolve the overall problem. It was decided that the methodology proposed would enable the author to investigate all aspects of the problem in the multi-disciplinary manner required and therefore the test room was set-up. In fact, the methodology itself, using the detailed information from an individual user to look at ways of fine-tuning an existing system is an important new proposal for building services control system development. Until now all systems have been designed using averaged user data and therefore have not catered for individual user needs. This research proposes and explores a new methodological approach.

The decision to gather experiential knowledge has been justified by the fact that the study was able to propose a solution that meets its initial aims. Namely that the proposed framework incorporated important contextual factors, previously unconsidered in all but the most complex and unmanageable control systems, without adding any complexity to the user interface, or the commissioning and maintance tasks. The best practice algorithm incorporated within the test room and its LonWorks framework was shown to be simple to operate and straightforward to commission and manage. The proposed framework uses exactly the same number of inputs and outputs as the best practice algorithm and therefore the same conclusion applies.

Indeed, the framework only incorporates the following items:

- a vertical external illuminance sensor orientated in the same direction as the window;
- a micro-controller with random access memory and a real time clock, that is capable of floating point arithmetic calculations, with time clock facilities and window orientation, latitude and longitude of the site global inputs;
- a simple user interface, similar to conventional systems; and
- an occupancy sensor.

These items allowed the system to take into account occupancy patterns, the illuminance of the sky facing the window, the position of the sun, the user's reactions to light quality and as a result contextual factors affecting an occupant's preferred visual conditions.

The difference is that the internal control mappings, which are hidden from the user, are more complex, in terms of multi-dimensionality, as they incorporate a new sky zoning technique, whilst still being transparent and easy to understand for when their internal operations need to be observed. Indeed, the maintenance staff will occassionally need to interact with the system when dealing with building churn (the relocation of occupants), and systems that have learnt incorrect patterns. When these situations occur the control mappings in the base energy state are used as a reset option for the learning part of the framework. However, good clear instructions to user's and maintenance staff alike on how the system is designed to operate should reduce any such occurances as it is felt that the self-learning aspect of the proposed system will eventually reduce long-term manageability issues.

In addition to these findings, the methodology was also justified by the fact that the author's main experiential findings concurred with those from other more experimental studies, therefore adding weight to the new detailed contextual basis for the proposed framework. The study also demonstrated some of the difficulties associated with proposing a framework that relied on data from individual users in individual contexts.

Throughout the six-month test period there were only 27 instances of the author having to over-ride the blind because of what was considered to be the system not living up to expectations. Had such a small amount of data been gathered from an non-expert subject, then it would probably have been more difficult to recognise the pattern that emerged. In the test room study a pattern was identified largely by using the author's detailed knowledge of the workings of the system and his personal experience of the contextual conditions in the room. This lead him to plot the data, the external context and the sun path on the Waldram diagrams to help define the pattern.

The other shortcoming of collecting data on individuals is that you are reliant on those individuals to collect the data and to understand the system they are assessing. If the researcher observes the users then it becomes difficult to know when the user is interacting with the system because they feel like a change of environment or because they are unhappy with the system's operation. Using an expert as the individual is a valid starting point for this work as it bypasses many of these issues, but it leaves the work reliant on experiential data, which should in time be backed up with data from other individuals, experts or not.

This study has tried to achieve a practical design solution centred on possible user responses. This thesis was considerably different to the theses of others on the subject, which generally tried to achieve a design solution that was idealised in academic discussions of good environmental design. The test room study undertaken as part of this research was an attempt to look at the problem from a new, but still broad, perspective. It allowed the author to observe, understand, visualise, predict, implement, evaluate and refine automated blind control strategies and the factors that influence a user's interaction with those strategies; and the lessons learnt during the study helped to form the framework for the knowledge acquisition and the control framework.

The elegant part of the proposed framework is that it is able to take into account the diversity of human interaction for different contexts, through its user centred occupied mode, and that the extent to which the energy saving background mode deviates from the occupied mode depends on the past history learnt from the occupants themselves. For example, if the user does not need to interact with the system a great deal, then the system functionality stays close to the original energy saving functionality. Whereas if the user continually interacts with the system, due to large areas of glass (say south facing) and therefore large areas of sky as potential glare sources, then the system will adapt more.

The key to transforming any control system design into a user-centred design is the development of a method for mapping human actions to control actions in a way that is intelligible to users, designers and engineers alike. This research has shown that the framework proposed is practically and theoretically possible and that it is more likely to meet certain individual users' needs, in certain contexts, than current systems. The question now is, how long will such a system take to adapt in real life conditions and once it does adapt will it be accurate and stable?

The next stage of the work must focus on the fine detail of the framework that allows it to recognise patterns in noisy data and adapt to a broad range of real operating conditions. For example, details like the number of sky zones, the range of illuminace values, and the adaptive mechanisms and filters used all need to be determined using more detailed studies. However, it should be remembered that the focus of such work would not be possible without the broad approach taken by this research.

Generally the investigations into adaptive/learning algorithms can proceed independently from the development of the model to which they are applied, after which the suitability of the algorithm for training a particular model can be assessed. However, if an on-line field bus based adaptive controller is to develop further, it is important to ensure that computational efficiency is maintained by employing simple learning rules with appropriately structured networks.

Future studies should test the framework in real life conditions and try to optimise the system's ability to deal with the large variations in inputs that it would be likely to encounter. It should be remembered that we do not want the system to pre-empt every user adjustment, but simply to adapt its control mappings and learn common patterns based on some predefined contextual issues in order to live up to individual expectations. However as the framework develops, the goal should be for each controller to be as unique as the individual that interacts with it, and the context that they are in.

The test room study demonstrated that additional sophistication within current automated blind control strategies are warranted, and proposed that a multi-criteria control strategy with operational flexibility may be invisible to the occupant and require no additional effort and cost, if well designed and implemented in software. However, the progression to this form of control needs to be taken one step at a time, if we are to ensure that control strategies meet the needs of their users. This thesis describes the beginning of evolution of current control system design, not a revolution in control system design.

Current technological and economic constraints mean that systems have to rely on a few sensor readings to reduce costs and simplify installation. It is likely that these limited number of data inputs to the proposed system may not be able to recognise complex patterns in user interaction. However, the causes of these complex patterns are not yet fully understood and therefore can only be incorporated into the framework once studied in detail.

Once this work has been undertaken, more information may come to light about the factors influencing the way people interact with blinds that hasn’t been included in this broad study, such as the internal and external temperatures in light weight buildings. The framework has been specifically designed to be flexible enough to accommodate any additional factors as are identified, thus facilitating a gradual evolution inherent with the gradual collection of individual user data.

The framework has been developed to allow the inputs to be changed or enhanced, or additions to be made, as more is learnt about how the users interact with the system. This staged approach ensures that the advancement of automated blind control strategies can be immediate and not hindered by the mass collection of complex information.

By proposing such a design methodology it is hoped that its implementation will lead to an improvement in the strategic design and thinking about all automated systems within buildings, and consequently an improvement in the comfort and energy efficiency of buildings.

Another clear constraint of the solution offered by this thesis is that it currently only considers single occupant rooms and that the context change between individual rooms and the open plan spaces can be great. In an open plan situation, the one-to-one relationship between the occupant and the various control devices, such as the window or blind, tends to vanish, making effective individual control more difficult. However, research by others has shown that multi-user democratic control is possible within open plan environmental control.[130] The author sees no reason why this proposition cannot be applied to the blind framework in the future, as long as building design respects the fact that buildings should not be too deep and that most interior space should be within 6m of a facade.

---

[130] Oseland N.A. et al., 1997, "A prototype system for democratic user control of zonal temperature in air conditioned offices", CIBSE National Conference 1997, Alexandra Palace, 5-7 Oct, Vol II, pp200-205.

Examples of other ways the work could be expanded include:

- a detailed study of patterns in user over-rides, and the generation inputs and rules that can be applied to the system to enable it to respond to these patterns without adding too much complexity. This study could incorporate the use of pre-programmed off-line training to improve our understanding of which inputs and interactions should be modelled to produce the most appropriate representation;
- the development of new user interface features to the Visual Basic controller, such as a preferred angle option that could be used for horizontal or seasonal adjustment;
- tying the adaptive mechanism in a blind device to separate adaptive mechanisms in other devices, such as an occupancy sensor that is able to learn occupancy patterns and adjusts its delay time accordingly, thus providing the blind controller with a better representation of the occupancy status of a room and allowing it to switch between modes more effectively;
- investigations into the definition of various grades of membership for each fuzzy category. For example, the number of sky zones or the classification of the sky illuminance. This work would require detailed analysis of solar irradiance data, sky type and cloud frequency; and
- the development of a means of adjusting the delay time of automated blind movement by classifying the sky conditions using solar radiation data recorded for the day so far (i.e. the frequency of the sun coming in and out of the clouds). If the sky was intermittent and the solar irradiance fluctuated greatly, the time delay would be long, if the sky was clear and the solar irradiance remained fairly constant the delay would be short. This delay time could then be reset on a daily basis.

Chapter 12: Conclusions

Until now the technological evolution of the automated blind, from a simple, manually controlled shading device to an energy saving 'intelligent facade' device, has largely neglected the needs of the user. This thesis suggests that a more demanding definition of 'intelligent control' is needed if users' needs are to be integrated within automatic blind control strategies and it proposes a new framework for a blind controller.

One of the key areas of practical control system development is knowledge acquisition. This thesis has addressed the need for improved automated blinds by forming a knowledge base from experience gained during a detailed literature search and a test room study. Having carried out this work a control system structure was developed that could satisfy the goals identified during the knowledge acquisition phase and accommodate the growth in the knowledge base expected in future studies.

The new framework utilises field bus technology to incorporate an adaptive controller that learns, on a blind by blind basis, from the behaviour of the occupants in their environment and adapts control mappings to meet each user's expectations. To provide the system with a firm base, the original control mappings that the occupant adapts are similar in operation to a best practice energy efficient strategy, also identified in the study, which may provide satisfactory conditions for the majority of the time. In addition, the system defaults to a fixed version of this energy efficient strategy when the occupant has left the room.

The key aspect of this integration of comfort criteria and energy criteria is the expansion of the two dimensional component, inherent in the sky assessment procedures of most energy saving strategies, to a three dimensional component that included a horizontal solar azimuth angle. By doing this, it was possible for an algorithm to divide the sky into a series of three dimensional fuzzy zones related to the orientation of the window facade, derive a set of fuzzy logic rules that controlled the tilt of the blind depending on the sun's location related to these zones and the external illuminance, and thus incorporate a number of key contextual factors identified as being important earlier in the study.

A linear adaptive control architecture was proposed and a means of adapting that architecture demonstrated by giving each one of the fuzzy rules a confidence, which was then adjusted using a simple adaptive algorithm. This means that instead of being programmed all at once, the system proposed could be programmed gradually as behaviour occurs and, over time, could reduce the number of interactions between the user and the blind, thus eliminating the need for lengthy and tedious commissioning and operating procedures through a complex user interface.

While this framework proposes an architecture that is fundamentally different to current systems in its approach, it retains the simple user interface required to maintain occupant satisfaction with the system and provides a straightforward means of enhancing a control system's understanding of a user's needs, without resorting to the complex performance related algorithms proposed by others. Therefore the proposal satisfies the thesis' primary aim, to find a framework that incorporates contextual factors whilst striking a suitable balance between complexity and usability. However, it is acknowledged that more validation and development work may be required before the framework can meet users' needs in real world environments.

The framework proposed in this thesis outlines a new approach to control system development that allows a system to incorporate users' needs, as those needs are identified. When developed fully the system will offer benefits to users for both environmental comfort and the whole life performance of the facade and will drive the continuing evolution of the 'intelligent facade'.

References

	Ref	Page
Aleo F. and Sciuto S., 1993, "Smart System for the combined control of natural and artificial lighting", Proceedings of the International Symposium on Energy Efficient Buildings, Leinfelden-Echterdingen, Germany, March 9-11, 1993.	64	29 33
Aldridge A., 1993, "Trends in control system development", Proceedings of the Conference of 'Intelligent buildings today and in the future', UCE, 7th October 1993, pp21-29.	79	62
Anon, 1995, "Daylighting design for display screen equipment", Building Research Establishment, IP10/95, Jan 1 1995.	58	24
Anon, 1995, prEN 12216, "Terminology and Definitions for Blinds and Shutters", 1995. British Standards Institute, London.	2	2
Anon, 1995 Neuron C Reference Guide, Document No. 29350, Echelon Corporation, 4015 Miranda Avenue, Palo Alto, California.	85	63
Anon, 1996 LonMark Application Layer Interoperability Guidelines, Version 3.0, LonMark Interoperability Association, Palo Alto, CA, 1996.	83	63
Anon, 1997 LonWorks Technology Device Data, Revision 4, Motorola Literature Distribution, Motorola, Colorado, USA, 1997	82	62
Anon, 1997, The SNVT Master List and Programmers Guide, May 1997, Echelon Corporation, 4015 Miranda Avenue, Palo Alto, California.	84	63
Anon, 1998, The British Blind & Shutter Association (BBSA) "Specifiers' Guide to Blinds and Shutters for 1998-1999",March 1998, BBSA, UK.	1	2
Anon, 1999, "Daylight Dynamics", Luxmate Technical Literature, Luxmate, London	72	35

Anon, 1999, "Huppe external blinds: Environmental control, exterior views, adjustable slats and modern design", Huppe Form technical catalogue, Oldenburg, Germany.	63	29 33
Arima M, Hara E.H. and Katzberg J.D, 1995, "A fuzzy logic and rough sets controller for HVAC Systems", IEEE WESCANEX '95 Proceedings, pp133-138.	122	169
Averill J.R., 1973, "Personal control over aversive stimuli and its relationship to stress", *Psychological Bulletin*, Vol.80, 286-303.	75	40
Barnard N., 1997, " Case Study 6: Blind Control - Simulation Analysis", DETR PIT Report on Controls Design by Simulation, Oscar Faber Applied Research, St Albans, Dec 1997.	65	30 36
Bauman F., Arens E., 1996, "Task/ambient conditioning systems: Engineering and application guidelines", Center for Environmental Design Research, University of California, Berkeley, CA.	49	19 27
Bierman A. and Conway K., 2000, "Characterising daylight photo-sensor system performance to help overcome market barriers", *Journal of the Illuminating Engineering Society*, Winter 2000, pp 101-115.	87	96
Bordass W., Bromley K. and Leaman A., 1993, "User and Occupant Controls in Office Buildings", Prepared for a conference entitled "Building Design, Technology and Occupant Well-being in Temperate Climates", Brussels, February 1993.	30	9 10
Bordass W.T., Bromley A.K.R. and Leaman A.J., 1995, "Comfort, control and energy efficiency in offices", BRE Information Paper, IP3/95.	47	19 27 43
Bordass W., Leaman A., 1997, "From feedback to strategy", Buildings in use '97: How buildings really work. Commonwealth Institute, London.	26	8 10 19
Bordass W., Cohen R., Standeven M. and Leaman A., 1999, "Assessment of Building Performance in Use: 2, Technical Review of the PROBE Buildings", Third Draft, www.usablebuildings.co.uk, 15 Jun 1999.	60	25

Brown M. and Harris C., 1994, "Neurofuzzy adaptive modelling and control", Prentice Hall International. Hemel Hempstead.	95	163 171 172 180
Chain C., Dumortier D. and Fontoynont M., 1999, "A comprehensive model of luminance, correlated colour temperature and spectral distribution of skylight: Comparison with experimental data", *Solar Energy*, Vol.65, No.5, pp285-295.	94	153
Cho S.H., Shin K.S. and Zaheer-Uddin M., 1995, "The effect of slat angle of windows with venetian blinds on heating and cooling loads of buildings in South Korea", *Energy*, Vol.20, No.12, pp1225-1236.	11	4
Clark R.P. and Edholm O.G., 1985, "Man and his Thermal Environment", Edward Arnold Publishers, London.	35	14
Clemo A. and Thompson M., 1996, "The evaluation of daylit interiors", CIBSE/ASHRAE Joint National Conference 1996, Vol.II, Harrogate, 29 Sept - 1 Oct, pp397-402.	41	17
Coley D.A. and Crabb J.A., 1997, "An artificial intelligence approach to the prediction of natural lighting levels", *Building and Environment*, Vol.32, No.2, pp81-85.	96	163
Collins B.L., 1979, "Window management: An overview", *ASHRAE Transactions*, Vol.85, pt.2, pp633-637.	15	4 13 14
Culp C.H., 1989, "Expert systems in preventive maintenance and diagnostics", *ASHRAE Journal*, Vol.31, No.8, pp24-27.	98	164
Curtiss P.S., 1993, "Adaptive control of HVAC processes using predictive neural networks", *ASHRAE Transactions*, Vol.99, Pt.1, pp496-504.	110	167
Curtiss P.S., Brandemuehl M.J. and Kreider J.F, 1994, "Energy management in central HVAC plants using neural networks", *ASHRAE Transactions*, Vol. 100, Pt.1.	108	167

Curtiss P.S., Shavit G. and Kreider J.F, 1996, "Neural networks applied to buildings - A tutorial and case studies in prediction and adaptive control", *ASHRAE Transactions*, Vol. 102, Pt.1, pp1141-1146.	113	167
Davies A.D.M. and Davies M.G., 1995, "The adaptive model of thermal comfort", *Building Services Research and Technology*, Vol.16, No.1, pp51-53.	38	17
De Dear R.J., 1993, "Outdoor climatic influences on indoor thermal comfort requirements", in Thermal Comfort: Past, Present and Future, BRE 264, Ed. N.A. Oseland and M.A. Humphreys, BRE, Garston 9-10 June 1993, pp52-72.	54	20
Deloe D., 1986, "Sensors light the way from security to energy savings", *Energy Management Technology*, Vol.10, No.1, pp48-49.	21	6
DiBartolomeo D.L. et al, 1996, "Developing a dynamic envelope/lighting control system with field measurements", Journal of Illuminating Engineering Society, Vol. 26, No. 1, pp146-164.	71	35 53 120
Dodier R.H. and Henze G.P, 1996, "Statistical analysis of neural networks as applied to building energy prediction", International Solar Energy Conference - ASME, pp495-505.	109	167
Dounis A.I. et al., 1994, "Thermal-comfort degradation by a visual comfort fuzzy-reasoning machine under natural ventilation", *Applied Energy*, Vol.48, pp115-130.	125	169
Dounis A.I., Lefas C.C. and Argiriou A., 1995, "Knowledge-based versus classical control for solar-building designs", *Applied Energy*, Vol.50, pp281-292.	126	169
Dounis A.I. et al., 1995, "Design of a fuzzy set environmental comfort system", *Energy and Buildings*, Vol.22, pp81-87.	127	169
Dounis A.I., Manolakis D.E.. and Argiriou A., 1995, "A fuzzy rule-based approach to achieve visual comfort conditions", *International Journal of Systems Science*, Vol.26, No.7, pp1349-1361.	128	169

Eames P.C. and Norton B., 1994, "A window blind reflector system for the deeper penetration of daylight into rooms without glare", International Journal of Ambient Energy, Vol.15, No.2, pp73-77.	13	4
Faithfull M., 1997, "Can we talk?", *CIBSE Journal Building Services Supplement on BMS controls*, November 1997, pp10-11.	80	62
Fisk D.J., 1981 "Thermal control of buildings", Applied Science, London.	7	3
Fountain M. et al., 1994, "Comfort control for short-term occupancy", *Energy and Buildings*, pp1-13.	129	185
Galanta A., Batturi E. and Viadana R., 1996, "A smart control strategy for shading devices to improve the thermal and visual comfort", Proceedings of the 4th European Conference on Architecture, Berlin, Germany, 26-29 March 1996, pp358-361.	67	33
Haberl J.S., Claridge D.E., 1987, "An expert systems for building energy consumption analysis: Prototype results", *ASHRAE Transactions,* Vol.93, pt.2, pp979-998.	101	165
Haberl J.S. et al., 1988, "An expert system for a building energy consumption analysis: Applications at a University campus", *ASHRAE Transactions,* Vol.94, pt.1, pp1037-1062.	102	165
Haberl J.S., Norford L.K. and Spadaro J.V., 1989, "Diagnosing building operational problems", *ASHRAE Journal*, Vol.31, No.6, pp20-30.	99	164
Haigh D., 1981, "User response in environment control", in The Architecture of Energy (Eds. D Hawkes and J Owers), Construction Press, London.	45	19 25
Harrison A.W., 1991, "Directional sky luminance versus cloud cover and solar position", *Solar Energy*, Vol.46, No.1, pp13-19.	56	21
Heijs W., 1993, "The dependent caviable in thermal comfort research some psychological considerations", in Thermal Comfort: Past, Present and Future, BRE 264, Ed. N.A. Oseland and M.A. Humphreys, BRE, Garston 9-10 June 1993, pp40-51.	51	20

Hepworth S.J., Dexter A.L. and Willis S.T.P., 1994, “Neural network control of a non-linear heater battery”, *Building Services Engineering Research and Technology*, Vol.15, No.3, pp119-129.	111	167
Hobson S., 1998, "Who needs integrated building controls?", *M&E Design*, April 1998, pp30-32.	81	62
Holz R. et al, 1997, "Effects of standard energy conserving measures on thermal comfort", *Building and Environment*, Vol.32, No., pp31-43.	25	8
Huang S.H. and Nelson R.M., 1994, “Delay time determination using an artificial neural network”, *ASHRAE Transactions*, Vol.100, Pt.1, pp831-840.	112	167
Huang S.H. and Nelson R.M., 1991, “A PID law combining fuzzy controller for HVAC applications”, *ASHRAE Transactions*, Vol.97, pt.2, pp768-774.	118	169
Humphreys M.A., 1993, "Field studies and climate chamber experiments in thermal comfort research", in Thermal Comfort: Past, Present and Future, BRE 264, Ed. N.A. Oseland and M.A. Humphreys, BRE, Garston 9-10 June 1993, pp52-72.	50	20
Humphreys M.A., 1997, "An adaptive approach to thermal comfort criteria", in Naturally Ventilated Buildings for the Senses, the Economy and Society, Ed. D. Clements-Croome, E&FN Spon, pp129-137.	34	14
Inoue T. et al., 1988, “The development of the optimal control system for window shading devices based on investigations in office buildings”, *ASHRAE Transaction*, Vol.94, pp1034-1049.	17	4 9 12 14 18 22 23 28 33
Jankowski W., 1993, “Occupancy sensors yield 25-75 percent energy savings”, *Facilities Design and Management*, Vol.12, No.1, p25.	22	6
John R.W. and Dexter A.L., 1989, “Intelligent controls for building services”, *Building Services Research and Technology*, Vol.10, No.4, pp131-141.	116	169

Jones R.H.L., 1980, “Solar radiation through windows - Theory and equations”, *Building Services Engineering Research & Technology*, 1, pp83-91.	6	3
Kaler G.M., 1990, “Embedded expert system development for monitoring packaged HVAC equipment”, *ASHRAE Transactions*, Vol.96, pt.2, pp733-742.	104	165
Kempton W., Feuermann D. and McGarity A., 1992, " 'I always turn it on super': - user decisions about when and how to operate room air conditioning units", *Energy and Buildings*, Vol.18, 177-191.	91	96
Kendrick J.D., 1989, "Guide to recommended practice of daylight measurement", Commission Internationale de l'Eclairage [CIE], Vienna.	92	152
Kiff A. and Warwick K., 1996, “Distributed fuzzy logic for building management systems using local operating networks”, IEE Colloquium (Digest), pp611-614.	123	169
Kim J.J. and Jones J., 1993, "A conceptual framework for dynamic control of daylighting and electric lighting systems", Conference Record of the 1993 IEEE Industry Applications Society, Ontario, Canada, 2-8 Oct 1993, pp2358-2364.	73	36 39
Klems J.H. and Warner J.L., 1997, “Solar Heat Gain Coefficient of complex fenestrations with a venetian blind for differing slat tilt angles”, *ASHRAE Transactions*, Vol.103, pt.1, pp1026-1034.	3	3
Klima J., 1990, “An expert system to aid troubleshooting of operational problems in solar domestic hot water systems”, *ASHRAE Transactions*, Vol.96, pt.1, pp1530-1538.	100	165
Kraus T.W. and Myron T.J., 1984, “Self-tuning PID controller uses pattern recognition approach”, *Control Engineering*, pp106-111.	105	165
Kreider J.F. and Wang X.A., 1991, “Artificial neural network demonstration for automated generation of energy use predictors for commercial buildings”, *ASHRAE Transactions*, Vol.97, pt.2, pp775-779.	107	167

Kroner W.M. and Stark-Martin J.A., 1994, "Environmentally responsive workstations and office-worker productivity", *ASHRAE Transactions*, Vol.100, Part 2, pp750-755.	61	27
Lee E.S. and Selkowitz S.E., 1995, "The design and evaluation of integrated envelope and lighting control strategies for commercial buildings", ASHRAE Transactions, Vol.101, Pt.1, pp326-342.	8	3 34
Lee E.S. et al, 1998, "Integrated Performance of an Automated Venetian Blind/Electric Lighting System in a Full-Scale Private Office", Proceedings of the ASHRAE/DOE/BTECC Conference, Thermal Performance of the Exterior Envelopes of Buildings VII, Clearwater Beach, Florida, December 7-11, 1998.	69	33 35 38 53
Lee E.S, DiBartolomeo D.L. and Selkowitz S.E., 1998, "Thermal and daylighting performance of an automated venetian blind and lighting system in a full scale private office", *Energy and Buildings*, Vol.29, pp47-63.	66	31
Lee E.S., DiBartolomeo D.L. and Selkowitz S.E., 1999, "The effect of venetian blinds on daylight photoelectric control performance", *Journal of the Illuminating Engineering Society*, Vol. 28, No.1 1999.	90	96
Lindsay C.R.T. and Littlefair P.J., 1992, "Occupant use of Venetian Blinds in Offices", PD 233/92, Building Research Establishment, Watford, October 1992.	33	12 14 18 19 23 25
Loe D.L., Mansfield K.P. and Rowlands E., 1994, "Appearance of lit environment and its relevance in lighting design", Lighting Research and Technology, Vol.26, No.3, pp119-133.	42	17
Luecke G.R. and Slaughter J., 1995, "Design, development and testing of an automated window shade controller", Transactions of the ASME, Journal of Solar Energy Engineering, Vol.117, No.4, pp326-332, Nov 1995.	68	33 34
Lund J., 1996, "The LonWorks Network Services (LNS) Architecture Strategic Overview – Client Server Control", Document No. 39305 Rev1, Echelon Corporation, 4015 Miranda Avenue, Palo Alto, California.	86	63

Mahdavi A. et al., 1995, “Effects of lighting, zoning and control strategies on energy use in commercial buildings”, *Journal of the Illuminating Engineering Society*, Winter 1995, pp25-35.	23	6
Mahdavi A., Chang S. and Pal V., 1999, " Model-based integration of contextual forces into advanced building control systems", Proceedings of "Intelligent and Responsive Buildings", Brugge, 29-30 March 1999, CIB Working Commission W098.	76	51
Markus T.A., 1967, “The significance of sunshine and view for office workers”, in Sunlight in Buildings, ed. Hopkinson R.G., Boeurcentrum International, Rotterdam.	44	19 24
Mills L.R. and McCluney W.R., 1993, “The benefits of using window shades”, *ASHRAE Journal*, Nov 1993, pp20-27.	9	3
Mistrick R. et al., 2000, "A comparison of photosensor controlled electronic dimming systems in a small office", *Journal of the Illuminating Engineering Society*, Winter 2000, pp66-80.	88	96
Mistrick R. and Thongtipaya J., 2000, "Analysis of daylight photocell placement and view in a small office", *Journal of the Illuminating Engineering Society*, Summer 1997, pp150-160.	89	96
Newsham G.R., 1994, “Manual control of window blinds and electric lighting: Implications for comfort and energy consumption”, *Indoor Environment*, 3, pp135-144.	10	3 4
Newsham_G.R., 1997, "Clothing as a thermal comfort moderator and the effect on energy consumption", *Energy and Buildings*, Vol.26, pp283-291.	52	20
Nicol J.F., 1966, “Radiation transmission characteristics of blind systems”, *Building Services Journal*, pp1-5.	5	3
Norford L.K., Allgeier A. and Spadaro J.V., 1990, “Improved energy information for a building operator; Exploring the possibilities of a quasi-real-time knowledge based system”, *ASHRAE Transactions*, Vol.96, pt.1, pp1515-1523.	103	165

Oseland N.A. et al., 1997, "A prototype system for democratic user control of zonal temperature in air conditioned offices", CIBSE National Conference 1997, Alexandra Palace, 5-7 Oct, Vol II, pp200-205.	130	192
Oseland N.A. and Williams A., 1997, "How best practice can improve productivity: The relationship between energy efficiency and staff productivity", Creating the Productive Workplace Conference, Workplace Comfort Forum, London.	29	8
Owens P.G.T., 1978, "The energy implications of comfort criteria: Visual aspects of comfort affecting energy consumption", Proceedings of Man, Environment and Buildings, Loughborough University, 28 Sept 1978, ppE1-E10.	18	4
Papamichael K. and Selkowitz S., 1986, "The luminous performance of vertical and horizontal slat-type shading devices", 1986 Illuminating Engineering Society of North America Conference, Boston, Massachusetts, pp232-282.	4	3 33
Papamichael K., Rubinstein F., Selkowitz S. and Ward G., 1986, "The integration of operable shading systems and electric lighting controls", Proceedings of the 1986 International Daylighting Conference, Nov 5-7, Long Beach CA, pp111-121.	19	4
Parsons K.C. et al, 1997, "A climatic chamber study into the validity of Fanger's PMV/PPD thermal comfort index for subjects wearing different levels of clothing insulation", CIBSE National Conference 1997, Alexandra Palace, 5-7 Oct, Vol.I, pp193-205.	53	20
Paulme C., 1999, "Functional Profile: BIU-2DC Lon2", Somfy AEG Technical Literature, Cluses, France.	62	27
Preller L. Et al., 1990, Indoor Air '90, Fifth International Conference on Indoor Air Quality and Climate, pp227-230.	46	19
Proshansky H.M., 1972, "Methodology in environmental psychology; problems and issues", *Human Factors*, 14, pp451-460	39	17

Rahmani K. and Shoureshi R., 1994, "Fuzzy based self-organising control for building systems", Proceedings of the American Control Conference, pp3060-3064.	121	169
Rea M.S., 1984, "Window blind occlusion: A pilot study", *Building and Environment*, Vol.19, No.2, pp133-137.	16	4 12 22 23
Rheault S. and Bilgen E., 1987, "Heat transfer optimisation of an automated blind window system", ASES 12th national Passive Solar Conference, Portland OR. pp122-128.	70	34
Richman E.E., Dittmer A.L., Keller J.M., 1996, "Field analysis of occupancy sensor operation: Parameters affecting lighting energy savings", *Journal of the Illuminating Engineering Society*, Winter 1996, pp83-92.	24	6
Rubin A.I., Collins B.L. and Tibbott R.L., 1978, "Window blinds as a potential energy saver: A case study," NBS Building Science Series No.112, Washington.	14	4 12 14 19 21 23
Rumelhart D.E. and McClelland J.L., 1986, "Parallel distributed processing", MIT Press, Vols. 1 and 2.	106	166
Russel J.A. and Ward L.M., 1982, "Environmental psychology", *Ann. Rev. Psychol.*, 33, pp651-688.	40	17
Rutten A.J.F., 1994, "Sky Luminance Research Imperative for Adequate Control of Temporary Supplementary artificial Lighting Installations", *Building and Environment*, Vol.29, No.1, pp10-111.	55	21
Selkowitz S.E. et al, 1994, "Envelope and lighting technology to reduce electric demand: Final Report - Phase II", Lawrence Berkeley Laboratory, University of Califoria, Berkeley, CA, January 14 1994.	12	4 38
Slater A., 1995, "Occupant use of lighting controls in offices", *CIBSE Journal*, Vol.17, No.8, pp43.	48	19

Slater A., 1995, 77 52
"Occupant Use of Lighting Controls: A review of current practice, problems and how to avoid them", Proceedings of the CIBSE National Conference 1995, Vol. I, Eastbourne, 1-3 Oct 1995.

Shavit G. and Wruck R., 1993, 20 4
"Energy conservation and control strategies for integrated lighting and HVAC systems", *ASHRAE Transactions*, Vol.99, pp785-789.

Shoureshi R. and Rahmani K., 1989, 117 169
"Intelligent control for building systems",
Intelligent Control Systems, ed. Shoureshi R., ASME publication #H005491, NY, pp7-15.

Shoureshi R., Torcellini P. and Rahmani K., 1993, 120 169
"Derivation and implementation of fuzzy optimal climate control", Proceedings of the American Control Conference, pp1860-1864.

Shoureshi R., Brackney L. and DeRoo B., 1996, 115 169
"Fuzzy based energy management system for large buildings", Proceedings of the ASME Dynamics Systems and Control Division, Vol.58, pp797-804.

Simmonds P., 1993, 36 17
"Thermal comfort and optimal energy use", ASHRAE Transactions, Vol.99, pp1037-1048.

So A.T.P. et al., 1994, 119 169
"Fuzzy air handling system controller",
Building Services Research and Technology, Vol.15, No.2, pp95-105

So. A.T.P., Chan W.L. and Tse W.L., 1997, 124 169
"Self-learning fuzzy air handling system controller",
Building Services Research and Technology, Vol.18, No.2, pp99-108.

Stevens S., 1998, 32 10 41
Occupants reactions to automated glass facades, Unpublished poster paper presented at EIBG Conf. Intelligent Buildings: Realising the Benefits, 6-8 October 1998 at BRE, Watford, UK.

Tenner E., 1996 74 40
"Why things Bite Back – Predicting the Problems of Progress"
Fourth Estate, London

Tregenza P.R. and Sharples S., 1993, "Daylight Algorithms", ETSU S 1350.	93	152
Tregenza P.R., 1995, "Mean daylight illuminances in rooms facing sunlit streets", *Building and Environment*, Vol.30, No.1, pp83-89.	57	24
Ure J.W. and Dunham S.J., 1997, "Lighting Control and Management Systems for Offices: Achieving User Satisfaction and Lower Operating Costs", Proceedings of the CIBSE National Conference 1997, Vol. II, Alexandra Palace, 5-7 Oct 1997.	31	9 10
Veitch J.A. and Gifford R., 1996, "Choice, perceived control, and performance decrements in the physical environment", *Journal of Environmental Psychology*, Vol.16, pp268-276.	27	8 43
Vine E. et al, 1998, "Office worker response to an automated venetian blind and electric lighting system a pilot study", *Energy and Buildings*, Vol.28, pp205-218.	59	24 39 53
White D.A. and Sofge D.A., 1992, "Handbook of intelligent control: Neural, Fuzzy and Adaptive Approaches", Van Nostrand Reinhold, New York.	97	164
Williams R.N., Boothby W.B. and Kirby L., 1997, "Modern Ventilation Techniques - The indoor environment and occupant perception", Proceeding of the CIBSE National Conference, Vol. I, Alexandra Palace, 5-7 Oct 1997.	43	18
Wilmhurst P. and Hamblin J., 1999, "Openness – A customer driven evolution" Proceedings of the CIBSE National Conference 1999, Harrogate, pp128-135.	78	62
Wyon D.P.,1993, "The economic benefits of a healthy indoor environment", Proceedings of Healthy Air '94, Italy, pp.405-416.	28	8
Yang K.H. and Su C.H., 1997, "An approach to building energy savings using the PMV index", *Building and Environment*, Vol.32, No.1, pp25-30.	37	17

Zadeh A.L., 1965,
"Fuzzy sets",
Information control, Vol.8, pp338-353.

Zadeh A.L., 1965	114	168

Bibliography

Abdou O.A. and Lorsch H.G., 1994,
“The impact of the building indoor environment on occupant productivity: Part III: Effects of Indoor Air Quality”,
ASHRAE Transactions, Vol.100, Part 2, pp902-913.

Aldridge A., 1993,
"Trends in control system development",
Proceedings of the Conference of 'Intelligent buildings today and in the future', UCE, 7th October 1993, pp21-29.

Amato I., 1992,
"Animating the material world",
Science, Vol.255, No. 5042, Jan 17 1992.

Anon, 1967,
“Evaluation of discomfort glare: the IES Glare Index System for artificial lighting installations”,
IES Technical Committee Tech. Rep. 10, Illuminating Engineering society, London.

Anon, 1975,
“Guide on interior lighting”,
Commission Internationale de l’Eclairage (CIE) Publication 29, Central Bureau of the CIE, Vienna.

Anon, 1984,
“Code for interior lighting”,
Chartered Institution of Building Services Engineers (CIBSE), London.

Anon, 1984,
"Recommended practice for the calculation of daylight availability",
IES Calculation Procedures Committee 1984, Journal of the Illuminating Engineering Society of North America, Vol.13 (4) pp 381-392

Anon, 1987,
"International lighting vocabulary",
Commission Internationale de l'Eclairage Comittee E-1.1, 1987, CIE Publication 17.4, Central Bureau of the CIE, Vienna.

Anon, 1992,
"The intelligent building in Europe,"
DEGW (London) and Teknibank (Milan) with the European Intelligent Building Group

Anon, 1992
"Wall technology - Volume A: Performance requirements",
CIRIA Special Publication 87, Construction Industry Research and Information Association (CIRIA), London.

Anon, 1992,
"Thermal environmental conditions for human occupancy",
ANSI/ASHRAE Standard 55-1992, ASHRAE.

Anon, 1995,
"Moderate Thermal Environments - Determination of PMV and PDD indices and specification of the conditions for thermal comfort",
ISO Standard 7730, International Standards Organisation, Geneva, Switzerland.

Anon, 1995,
"BACnet – A data communication protocol for building automation and control networks,"
ANSI/ASHRAE 135-1995, American Society of Heating, Refrigeration and Air Conditioning Engineers, Atlanta, GA

Anon, 1995,
"LonWorks Custom Node Development",
LonWorks Engineering Bulletin, January 1995, Echelon Corporation, 4015 Miranda Avenue, Palo Alto, California.

Anon, 1995,
"Introduction to the LonWorks Platform – An Overview of Principles and Practices",
Version 2, Echelon Corporation, 4015 Miranda Avenue, Palo Alto, California.

Anon, 1995,
"Neuron C Programmer's Guide",
Document No. 29300, Echelon Corporation, 4015 Miranda Avenue, Palo Alto, California.

Anon, 1995,
"Developing LNS Plug-ins using Visual Basic 5",
Document No. 39820, , Echelon Corporation, 4015 Miranda Avenue, Palo Alto, California.

Anon, 1996,
"Introduction to LonWorks Device and Network Design", 4-day training Course,
Version 4.1, Echelon Corporation, 4015 Miranda Avenue, Palo Alto, California.

Anon, 1996,
"LonMark Layers 1-6 Interoperability Guidelines",
Version 3.0, LonMark Interoperability Association, Palo Alto, CA, 1996.

Anon, 1997,
“LonMark Functional Profile: Real Time Keeper”,
Rev1, LonMark Interoperability Association, www.lonmark.org.

Anon, 1999,
"Fuzzy Logic Toolbox: For use with Matlab",
User's Guide, Version 2, The Mathworks Inc.

Anon, 1999,
"Fuzzy Logic Defuzzification Example",
Matlab Software, Version 5, The Mathworks Inc.

Anon, 2000
"Building Control Systems"
CIBSE Guide H, Chartered Institution of Building Services Engineers, London

Aronoff S. and Kaplan A., 1995
"Total Workplace Performance – Rethinking the Office Environment"
WDL Publications, Ottawa

Astrom K.J. and McAvoy T.J., 1992,
“Intelligent control: An overview and evaluation”,
in Handbook of Intelligent Control: Neural, Fuzzy and Adaptive Approaches, ed.
White D.A. and Softe D.A., Van Nostrand Reinhold, NY.

Baker N. And Standeven M., 1994,
“Comfort criteria for passively cooled buildings a PASCOOL task”,
Renewable Energy, Vol.5, Part II, pp977-984.

Baker N. And Standeven M., 1996,
"Thermal comfort in free running buildings",
Energy and Buildings, Vol. 23, pp175-182.

Beale R. and Jackson T., 1994
"Neural Computing – An Introduction"
Institue of Physics Publishing Limited, London

Bernecker C.A. et al, 1993,
"Task lighting in the open office: A visual comfort perspective",
Journal of the Illuminating Engineering Society, Winter 1993, pp18-25.

Beutell A.W., 1934,
"An analytical basis for a lighting code",
Journal of the Illuminating Engineering Society, Vol.27, pp5-.

Boone W., 1999,
"Implementing the latest technology in building automation: A project orientated methodology",
Proceedings of "Intelligent and Responsive Buildings" Brugge, 29-30 March 1999,
CIB Working Commission WO98, pp223-231.

Bordass W., Leaman A., 1997,
"Future buildings and their services",
Building Research and Information, Vol.25, No.4, pp190-195.

Burge S.A. et al., 1987,
"Sick Building Syndrome: A study of 4373 office workers",
Ann. Occ. Hygiene, Vol. 31, pp493-504.

Busch J., 1992,
"A tale of two populations: thermal comfort in air-conditioned and naturally ventilated offices in Thailand",
Energy and Buildings, Vol.18, pp235-249.

Butler D.L. and Biner P.M., 1987,
"Preferred lighting levels: Variability among settings, behaviours and individuals",
Environment and Behaviour, 6, pp695-721.

Butler D.L. and Biner P.M., 1989,
"Effects of setting on window preferences and factors associated with those preference",
Environment and Behaviour, 21, pp17-31.

Cena K.M., 1993,
"Thermal and non-thermal aspects of comfort surveys in homes and offices",
Thermal Comfort: Past, Present and Future, BRE 264, Edited by N.A. Oseland and M.A. Humphries, BRE, Garston, 9-10 June 1993.

Coen M.H., 1998,
"Design principles for intelligent environments",
Proceedings of the National Conference on Artificial Intelligence.

Coveney P. and Highfield R., 1995,
"Frontiers of Complexity: The search for order in the chaotic world",
Faber and Faber, London

Crowe N., 1997
"Nature and the Idea of the Man-Made World"
MIT Press, Massachusetts.

Davis M., 1981,
"A wall for all seasons",
RIBA Journal, Feb 1981.

de Dear R.J. and Fountain M.E, 1994,
"Field experiments on occupant comfort and office building thermal environments in a hot-humid climate",
ASHRAE Transactions, Vol. 100, No.2 pp457-475.

Donald I., 1994,
"Management and change in office environments",
The Journal of Environmental Psychology, Vol.14, pp21-30.

Dorgan C.E. et al., 1994,
"Productivity link to the indoor environment estimated relative to ASHRAE 62-1989",
Proceedings of Healthy Buildings '94, Budapest, pp461-472.

Dorgan C.E. and Dorgan P.E., 1997,
"Productivity link assessment to indoor air quality",
Creating the Productive Workplace Conference, Workplace Comfort Forum, London.

Drexler K.E., 1990,
"Engines of Creation: The coming era of nanotechnology",
Fourth Estate, London

Duffie J.A. and Beckman W.A., 1980,
"Solar engineering of thermal processes",
John Wiley & Sons, New York.

Einhorn H.D., 1979,
"Discomfort glare: a formula to bridge differences",
Lighting Research and Technology, Vol. 11, pp90-94.

Faithfull M., 1997,
"Can we talk?"
CIBSE Journal Building Services Supplement on BMS controls, November 1997, pp10-11

Fanger P.O., 1970,
"Thermal comfort analysis and applications in environmental engineering",
McGraw-Hill, New York.

Gagge A.P., Stolwijk J.A.J. and Nishi Y., 1971,
"An effective temperature scale based on a simple model of human physiological regulatory response",
ASHRAE Transactions, Vol. 77, No.1, pp247-262.

Goulding J.R., Lewis J.O. and Steemers T.C., 1986,
"Energy in architecture: The European passive solar handbook",
B.T.Batsford Limited, London, UK.

Grandguillaume J., Read R., Southwood W., 1994,
"The future of intelligent buildings",
The Arup Journal, No.4.

Harris J. and Wigginton M., 1998,
"A case study of the intelligent skin",
First Draft, Intelligent Buildings Research Programme, School of Architecture, University of Plymouth, August 1998.

Hebb D.E., 1949,
"Organisation of behaviour",
Wiley.

Heerwagen J.H. and Heerwagen D.R., 1986,
"Lighting and psychological comfort",
Lighting Design & Application, Vol.16, pp47-51.

Hobson S., 1998,
"Who needs integrated building controls?",
M&E Design, April 1998, pp30-32.

Humphreys M.A., 1976,
"Field studies of thermal comfort compared and applied",
Building Services Engineer, Vol. 44, pp5-27.

Humphreys M.A., 1993,
"Field studies and climate chamber experiments in thermal comfort research",
Proc. Conf. Thermal comfort: Past, present and future, Building Research Establishment.

Kelly K., 1994,
"Out of Control: The biology of machines",
Fourth Estate Limited, London.

Kerr P., 1995,
"Gridiron",
Chatto & Windus Ltd, London.

Knez I., 1995,
"Effects of indoor lighting on mood and cognition",
Journal of Environmental Psychology, Vol.15, pp39-51.

Kreider J.F. and Kreith F., 1981,
"Solar Energy Handbook",
McGraw-Hill, New York.

Kreider J.F. Hoogendoorn C.J. and Kreith F., 1987,
"Solar design: Components, systems, economics",
Hemisphere Publishing Corporation, New York.

Kroner W.M., 1997,
"An intelligent and responsive architecture",
Automation in Construction, Vol.200, pp1-13.

Leaman A. and Bordass W. 1997,
"Productivity in buildings: The killer variables",
Creating the Productive Workplace Conference, Workplace Comfort Forum, London.

Levermore G.J., 2000
"Building Energy Management Systems – Applications to Low-Energy HVAC and Natural Ventilation Control"
E & FN Spon, London.

Loftness V. et al, 1994,
"Beyond the open plan: New space planning concepts to support organisational, technical, and environmental change",
Center for Building Performance and Diagnostics, Carnegie Mellon University.

Lorsch H.G. and Abdou O.A., 1994,
"The impact of the building indoor environment on occupant productivity: Part I: Recent studies, measures and costs",
ASHRAE Transactions, Vol.100, Part 2, pp741-749.

Lorsch H.G. and Abdou O.A., 1994,
"The impact of the building indoor environment on occupant productivity: Part II: Effects of temperature",
ASHRAE Transactions, Vol.100, Part 2, pp895-901.

Lovelock J., 1988,
"The ages of Gaia: A biography of our living Earth,"
Oxford Paperbacks, Oxford University Press, Oxford

McIntyre D.A.,1982,
"Chamber studies - reductio ad absurdum?",
Energy and Buildings, Vol. 5, pp89-96.

Minsky M., 1986,
"Society of Mind",
New York, Simon and Schuster.

Minsky M.L., 1968,
"Matter, mind and models",
In "Semantic Information Processing" ed. Minsky M.L., MIT Press, Cambridge, Mass.

Muneer T., 1997,
"Solar Radiation & Daylight Models for the energy efficient design of buildings",
Architectural Press, Oxford UK.

Ne'eman E., 1974,
"Visual aspects of sunlighting in buildings",
Lighting Research and Technology, Vol.6, pp159-164.

Ne'eman E. and Hopkinson R.G., 1970,
"Critical minimum acceptable window size: A study of window design and provision of a view",
Lighting Research and Technology, Vol.2, pp17-27.

Penrose R., 1989,
"The Emperors New Mind",
Oxford University Press.

Paulme C., 1999,
"Functional Profile BIU-2DC Lon 2",
Version 2, Ref 157809, Somfy, Fr.

Pepler R.D. and Warner C.G., 1968,
"Temperature and learning: an experimental study,"
ASHRAE Transactions, Vol.74, Part 2, pp211-219.

Phillips E.M. and Pugh D.S., 1994,
"How to get a PhD – A Handbook for Students and Supervisors"
Second Edition, Open University Press, Buckingham.

Ross T.J., 1995,
"Fuzzy Logic with Engineering Applications"
McGraw-Hill, New York.

Scheerbart P., 1914,
"Glasarchiteckur",
translation published in Glass Architecture, Alpine Architecture, November Books Ltd, 1972.

Schendler B.R., 1990,
"Even your walls will have brains",
Fortune, Vol. 122, No. 15, p16, Dec 17 1990.

Schiller G.E. et al., 1988,
“Thermal environments and comfort in office buildings”,
ASHRAE Transactions, Vol. 94, No.2 pp280-308.

Schiller G.E., 1990,
“A comparison of measured and predicted comfort in office buildings”,
ASHRAE Transactions, Vol. 96, pt.1, pp602-622.

Slater A., 1995,
“Occupant use of lighting controls in offices”,
CIBSE Journal, Vol.17, No.8, pp43.

Smith S,.W. and Rea M.S., 1978,
“Proof-reading under different levels of illumination”,
Journal of the Illuminating Engineering Society, Vol.27, pp47-52.

Stone N.J. and Irvine J.M., 1994,
“Direct or indirect window access, task type, and performance”,
Journal of Environmental Psychology, Vol.14, pp57-63.

Sutherland V. and Cooper C.L., 1997,
“Stress and the changing nature of work”,
Buildings in use ‘97: How buildings really work. Commonwealth Institute, London.

Sutton R., 1990
"Modelling Human Operators in Control System Design"
Research Studies Press Ltd., Taunton

Tennessen C.M. and Cimprich B., 1995,
"Views to nature: Effects on attention",
Journal of Environmental Psychology, Vol.15, pp77-85.

Tregenza P.R., et al., 1974,
"Consistency and variation in preferences for lighting offices",
Lighting Research and Technology, Vol.6, pp205.

Tsoukalas L.H. and Uhrig R.E., 1997
"Fuzzy and Neural Approaches in Engineering"
John Wiley & Sons inc, New York

Van Paassen A.H.C. and Lute P.J., 1992,
"Natural ventilation and automation with manual over-riding are health solution",
Energy and Buildings, Vol.18, No.2, 1992, p. 65-75.

Wang L.X, 1994
"Adaptive Fuzzy Systems and Control – Design and Stability Analysis"
PTR Prentice Hall, New Jersey.

Wells B.W.P., 1965,
"Subjective responses to the lighting installation in a modern office building and their design implications",
Building and Environment, Vol.1, pp57-68.

Weston H.C., 1945,
"The relation between illuminance and visual performance",
Industrial Health Research Board Report No.87, HMSO, London.

Wheeler W.M., 1928,
"Emergent Evolution and the development of societies",
W.W.Norton.

Wilkins A.J., 1993,
"Health and efficiency in lighting practice",
Energy, Vol.18, No.2, pp123-129.

Wilmhurst P. and Hamblin J., 1999,
"Openness – A customer driven evolution" Proceedings of the CIBSE National
Conference 1999, Harrogate, pp128-135.

Wyon D.P., 1974,
"The effects of moderate heat stress on typewriting performance",
Ergonomics, Vol.17, pp309-318.

Appendix I The Intelligent Building and Its Skin

Technology and 'Intelligent Buildings'

An 'intelligent building' can be described as a building that utilises innovative and adaptive technologies to *"provide a responsive, effective and supportive intelligent environment within which an organisation can achieve its business objectives."* [1] Many themes drive innovation; the themes that have guided the concept of 'intelligent buildings' can be listed under the following headings:

(i) *Biology*: a better understanding of biological processes, super-organisms, and the products and processes of genetics;
(ii) *Technology*: the rapid improvement in communications and information transfer and the exponential growth of computing power;
(iii) *Materials*: advances in materials science, engineering and technology; and
(iv) *Ecology*: the need for a more sustainable environment.

Biology

Biological theory has shown us that natural systems are highly complex, forming an intricate web of life that exists all around us. This hypothesis was epitomised by James Lovelock's "Gaia" theory [2], of which he wrote:

> *"The entire range of living matter on Earth, from whales to viruses, from oaks to algae, could be regarded as constituting a single living entity, capable of manipulating the Earth's atmosphere to suit its overall needs and endowed with facilities and powers far beyond those of its constituent part."*

Lovelock's theory of collective responsibility and other similar theories, such as "Hive Mind" [3] [4] and "Neural Network" theory [5], have lead to an initiative, by many, to mimic natural complexity and its inherent robustness within the artificial world. [6]

By applying these ideas to buildings, biomimetics hope to construct a nervous system of sensors that allow a building to be more aware of its surroundings and interact with its users. This goal is epitomised in research by MIT and others, which has explored new forms of interaction between people and computers by creating rooms full of simple automated components that learn to recognising people, their moods, respond to their verbal commands and optimise their energy performance accordingly.[7]

Technology

Before we can apply this biological intricacy to all buildings, we must develop a new rationale for building system inter-connectivity by utilising the very latest communication and computer technologies. A study undertaken by DEGW and Teknibank in 1991-92 [8], explored the utilisation of technology in future buildings. It showed six levels of system integration and proposed a technology evolution model to aid the development of 'Intelligent Buildings'. This model is outlined in Figure I-I.

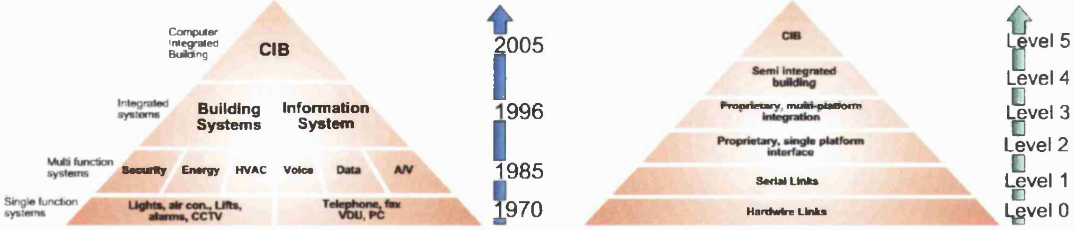


Figure I-I: Models from the "Intelligent Buildings Europe" research project,
Left - technical evolution model; right - Increasing levels of integration

As Figure I-I illustrates, the communications infrastructure of virtually any system within a building can be mapped into two components: Building control systems and information technology systems. The goal of the 'Intelligent Building' is to merge these two components to form a 'Computer Integrated Building'. Although information technology systems are well developed and have a number of standards defining their technological implementation, the evolution of building control networks has fallen about a decade behind. However, with the price of technology rapidly falling and the expectations of users rising, building management and control systems are currently undergoing a sudden evolution (see Appendix V).

In years to come, when the electronic age has become truly established, technology will become completely transparent, thus enabling the designer to view a building as a super-organism. The concept is that each device within the building will behave as an organism that reacts to stimulus and communicates with the others. Alone the electronic components are simple, together they form a community that is attentive, robust and evolving to produce an environment that satisfies both the diversities of individual human needs and the subtleties of modern energy concerns.

Materials

As advances in technology reduce the size of microprocessors year on year, the nervous system of the intelligent building will begin to extend into the microscopic scale through materials, from metals to concrete, that will be designed with built in sensing and controlling logic, including self-diagnostic monitoring systems.[9]

New composite materials will integrate mechanical functions, resulting in fewer parts and removing the need for extra equipment to perform functions. For example, piezoelectric structural materials will dampen vibrations caused by anything from the sway of a tall building to the vibration caused by air conditioning plant, thus reducing the onus on vibration mountings[10] and structures will be developed that use sensors to react to changes in weight loads.[11]

Ecology

Apart from human self-indulgence, one of the main driving forces behind intelligent buildings is the ideal of creating a sustainable environment for the future of the planet by using technology and materials to link the artificial world to the natural world and ensure that they can live in harmony with one another.

For many years, buildings that are deep with a central core, a hermetic skin and a highly engineered environment have been built, which work against nature rather than with it. By automating a number of building processes it is possible to respond to external weather fluctuations to save energy by fully utilising the sun's energy and natural air currents when needed, and optimising the use of the precious resources such as water and heat.

Central to this ecological ideal is the notion that the building skin should act like a filter, designed to give people the enjoyment of nature and moderate the energy exchange between inside and outside. The rest of this appendix concentrates on this aspect of 'intelligent buildings' – 'the intelligent facade'.

The Changing Role of the Facade

The Search for Transparency

A facade can be defined as the element of a building that separates the internal and external environment. Traditionally it has many roles, from protecting the building from intrusion or the diversities of the weather, to decorating the building to celebrate status. But with the development of glass manufacturing over the course of the last century, a new role for the facade has become apparent; that of a tool to exploit a new set of spatial dynamics. The proponents of this position were the pioneers of modern architecture, who considered transparency an aesthetic ideal; a concept epitomised by Mies van der Rohe's futuristic glass skyscraper design in 1921.

In recent years, growing environmental concerns have highlighted the glass facade as an energy problem. In summer, excessive heat gains result in high cooling loads for the exhaustive but necessary mechanical systems. In the winter, excessive heat losses add to the already typically high heating load. In both cases energy is being wasted due to poor facade performance.

A low energy facade must be designed to provide a high performance for a broad range of seasonal changes within a building's natural climatic context. Therefore if the facade was to evolve further, the principles of vernacular bioclimatic architecture had to be transposed onto the glass facade so that the building skin could become adaptable to cope with this change over time.

The 'Intelligent Facade'

'Intelligent facades' place their emphasis on the active control of the functions that a facade performs. This is very different to the conventional passive architectural approach that seems to have prevailed in buildings over the years. Static or fixed systems often provide an average solution to daylight, heat, ventilation and view for average climatological and occupant conditions throughout the year.

The sun offers great potential in terms of passive and active gains for heat, daylight and electricity generation. However, it can also be detrimental to internal comfort conditions and therefore it is often necessary to reject it from a building (see Figure I-II).

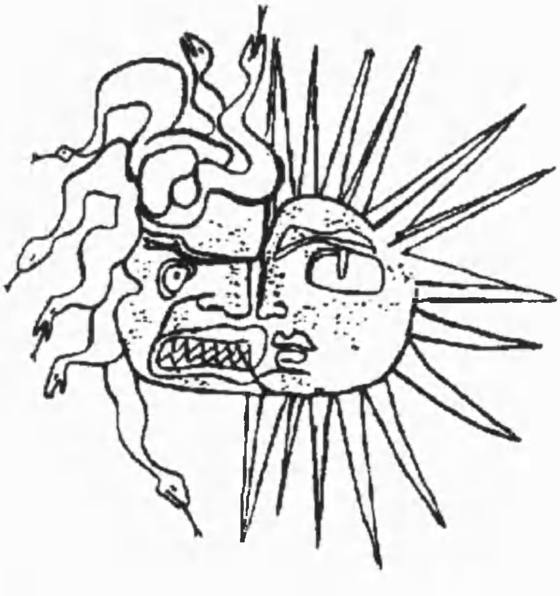


Figure I-II: Sketch by Le Corbusier, with which he wrote, "Part of the year the sun is our friend and part of the year it is our enemy." [12]

Climatic conditions vary between morning and afternoon, between day and night and between seasons. Therefore buildings benefit from an element of variability. Indeed, van der Rohe's glass skyscraper equipped with blinds and shutters would actually perform better than it would as a clear glass tower, but of course that would have defeated the objective of transparency.

The goal of the intelligent facade is to enhance the performance of the building as a whole by combining its functionality with sensors, actuators and building services devices to provide an energy efficient and comfortable environment. They can comprise of a few or many different components, all performing functions that can be individually or cumulatively adjusted to respond to environmental variations. A list of functions the modern day facade is expected to perform can be found in Table I-I.

Be a self supporting and stable structure	Prevent condensation and mould growth
Transfer loads	Control humidity
Keep out wind	Integrate with other building systems
Keep out rain and snow	Keep out intruders and provide security
Provide fire resistance and safety	Keep out insects, vermin, etc.
Be durable	Prevent biological damage from plants, fungi etc.
Control solar radiation	Keep out unwanted noise
Utilisation and control of daylight penetration	Limit weather generated noise
Control glare	Keep out unwanted odours and pollutants
Allow views	Address safety issues
Provide visual privacy	Support maintenance and repair
Provide facility to blackout windows	Maintain a good appearance
Allow controlled natural ventilation	Communicate status
Provide safe operable openings	Address buildability issues, handling, storage, etc
Limit infiltration	Address manufacturing issues
Control winter air temperature	Be at a reasonable cost
Prevent fabric heat loss	

Table I-I: List of modern day expectations from the performance of a facade

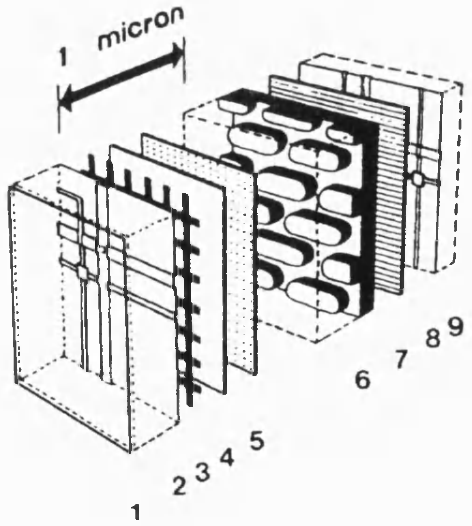
Defining the Intelligent Facade

Modern architects are now faced with the challenge of creating buildings that balance all their customary aesthetic and functional requirements with environmental issues. This goal was first identified by Paul Scheerbart in 1914, when he suggested that one day the 'window' may be abolished altogether and replaced by a variable-transmission, multiple-function wall.[13]

In 1981 Mike Davis offered a theoretical solution to the problem:

"What is needed is an environmental diode, a progressive thermal and spectral switching device, a dynamic interactive multi-capability processor acting as a building skin." [14]

This dynamic skin, which he named the "polyvalent wall" (see Figure I-III), would have the ability to regulate energy flow in either direction depending on conditions, in a similar way to the human skin. The wall would operate on a molecular level rather than a mechanical level giving it the ability to resonate between opaque and transparent states.



1. Silica weather skin and deposition substrate
2. Sensor and control logic layer - external
3. Photoelectric grid
4. Thermal sheet radiator / selective absorber
5. Electro-reflective deposition
6. Micro-pore gas flow layers
7. Electro-reflective deposition
8. Sensor and control logic layer - internal
9. Silica deposition substrate and inner skin

Figure I-III: Mike Davies' Polyvalent Wall

Unfortunately the wide range of performance required from the ideal facade system is not available today and no-one has been able to come up with a wall that meets Davies' objectives. Technologies such as electrochromic and thermochromic glass, photovoltaic cells, Gore-tex materials, and intelligent sensors have been developed but none are at the level required for implementation within an all-purpose polyvalent skin.

Despite this, the perceived role of the facade, among many architects, has starting to change from that of a building envelope, to that of a filter or climate moderator that accepts free energy and rejects unwanted pollutants from the external environment.[15]

While a more elegant realisation of intelligent envelope and lighting control systems may have to wait for the development of thin film variable transmission glazings (e.g. electrochromic glazings), a reasonable level of intelligent control can be achieved with modifications, refinements and ingenious combinations of available hardware components, such as movable internal or external shading devices. The latter is the primary focus of this thesis.

Closure

For many years, technological philosophers and fictional authors have been predicting the coming age of the 'Intelligent Building'.[6] [16] However, today these goals still remain in the realms of science fiction.

The concept of intelligent buildings is new, but the tradition it follows is not. Humans have always maintained a steady evolution of the structures in which society live and work and each generation gradually improves upon its living conditions. Today the rate of change of technology, which has been viewed at various stages of civilisation as an indicator of future progress, is stronger than at any other time. However, by using the term 'intelligent building', proponents are suggesting that mankind is very close to a habitual utopia.

The phrase ‘intelligent buildings’, along with phrases such as ‘intelligent facades’, ‘intelligent architecture’ and ‘intelligent control’ have been used throughout this thesis and originate from a number of building review journals, research papers and architectural books published over the last few years. In all these contexts, the meaning of the term ‘intelligent’ is often vague and generally only refers to a collection of innovative and automated features that jointly make systems easier to use or manage.

The definition of true intelligence, particularly with reference to machines, is in fact a subject of constant debate. When considering the nature of intelligence one may side with the writings of Minsky, the pioneer of artificial intelligence, who refers to the human mind as “a computer made of meat”.[17] Or conversely with those of Penrose, a scientific philosopher, who believes that computers are essentially processing units that have no understanding of their actions and will never be able to appreciate beauty and humour, or demonstrate consciousness and free will. [18] Both philosophies offer reason and possibility, neither are proven nor disproved.

Without being drawn into the debate, it is safe to say that our understanding of the term intelligence originates from our own perception of the human ability to comprehend, reason and learn. Therefore, even with this restrictive definition, we can see that the term ‘intelligent’ does not apply to the majority of the systems described as ‘intelligent’ within the building literature. As a result, a more demanding definition of 'intelligent building control' is needed. The system should: [19]

(i) have the ability to adapt, reason and learn about processes, disturbances and operating conditions;
(ii) acquire knowledge and store it in such a way that it can be used or retrieved;
(iii) autonomously improve upon its performance as experience is gathered.

Over time the intelligent building and its skin should develop some of these characteristics. But can buildings be designed to be responsive to humans in the same way they are responsive to each other and can buildings be designed with sufficient intelligence so that they exist in harmony with nature and are therefore sustainable?

While it could be easy to get carried away with the idea that scientists will solve all our problems for us, it is important as engineers to ensure that technological developments meet current system objectives and do not create revenge effects that require more technological solutions

When an engineer refers to something that works, they refer to something that is reliable and meets its functional specification. Good design produces an object that works for people, in the context of their values and needs, to produce quality results and a satisfying experience. Therefore we can say that a truly intelligent building must use technologies to serve, not dominate.

Most of the technologies and materials described here are still in their theoretical phase and will not be implemented for many years. However, the DEGW study showed that the technology is developing and the networking infrastructure needed for intelligent buildings is very close.

As described in Chapter One, information technology can be part of the problem as well as being the solution. As information technology finds its way into architecture and the building itself becomes a responsive mechanism, anticipating change, predicting patterns of operation, being able to respond to individual demands, it must be developed with care and diligence.

In general, advanced facades have not fulfilled their promise as a key energy efficient strategy that also enhances occupant comfort and performance. Part of the problem can be traced back to a lack of adequate, high performance systems that meet users needs as well as energy targets. Therefore before designers can strive to meet many of the goals set out in this appendix they must first understand how a user interacts with a system and what a user desires. This is the primary focus of this thesis.

---

[1] Grandguillaume J., Read R., Southwood W., 1994, "The future of intelligent buildings", *The Arup Journal*, No.4.

[2] Lovelock J., 1988, "The ages of Gaia: A biography of our living Earth," Oxford Paperbacks, Oxford University Press, Oxford.

[3] Wheeler W.M., 1928, "Emergent Evolution and the development of societies", W.W.Norton.

[4] Minsky M., 1986, "Society of Mind", New York, Simon and Schuster.

[5] Hebb D.E., 1949, "Organisation of behaviour", Wiley.

[6] Kelly K., 1994, "Out of Control: The biology of machines", Fourth Estate Limited, London.

[7] Coen M.H., 1998, "Design principles for intelligent environments", Proceedings of the National Conference on Artificial Intelligence.

[8] Anon, 1992, "The intelligent building in Europe," DEGW (London) and Teknibank (Milan) with the European Intelligent Building Group

[9] Drexler K.E., 1990, "Engines of Creation: The coming era of nanotechnology", Fourth Estate, London

[10] Amato I., 1992, "Animating the material world", *Science*, Vol.255, No. 5042, Jan 17 1992.

[11] Schendler B.R., 1990, "Even your walls will have brains", Fortune, Vol. 122, No. 15, p16, Dec 17 1990.

[12] Le Corbusier, extracted from Boesiger W., 1977, "Oevre Complete: 1938-1944", Vol. 4.

[13] Scheerbart P., 1914, "Glasarchiteckur", translation published in Glass Architecture, Alpine Architecture, November Books Ltd, 1972.

[14] Davis M., 1981, "A wall for all seasons", *RIBA Journal*, Feb 1981.

[15] Construction Industry Research and Information Association, 1992, "Wall technology - Volume A: Performance requirements", CIRIA Special Publication 87, CIRIA, London.

[16] Kerr P., 1995, "Gridiron", Chatto & Windus Ltd, London.

[17] Minsky M.L., 1968, "Matter, mind and models", In "Semantic Information Processing" ed. Minsky M.L., MIT Press, Cambridge, Mass.

[18] Penrose R., 1989, "The Emperors New Mind", Oxford University Press.

[19] Astrom K.J. and McAvoy T.J., 1992, "Intelligent control: An overview and evaluation", in Handbook of Intelligent Control: Neural, Fuzzy and Adaptive Approaches, ed. White D.A. and Softe D.A., Van Nostrand Reinhold, NY.

Appendix II A Study of Thermal and Visual Comfort Theory

Introduction

Primarily a venetian blind is an artificial mechanism for alleviating thermal and visual discomfort. Man's development of artificial techniques for climate control, such as the building envelope, can be attributed to his need for an environment that enables him to thrive and flourish. This is emphasised by research that attempts to define the influence of the indoor climate on productivity, health and well being, in order to quantify the economic benefits of creating the right environment to work. [1] [2] [3] [4] [5] [6] [7] [8] [9] [10] Together such research demonstrates that productivity decreases and accident frequency increases as conditions deviate from a comfort band. However, the definition of this comfort band varies from paper to paper, thus implicating comfort as an ambiguous issue.

When developing a control strategy for an automated venetian blind we must try to create an environment in which our individual is thermally and visually comfortable. Traditionally this assessment would be made at design time using environmental comfort theory, which represents a series of pragmatic exercises aimed at reducing each human perception of comfort to a working engineering model. However by utilising such theories many of the finer nuances of comfort have been lost.

This appendix reviews certain aspects of thermal and visual comfort theory that can be related to the way people control blinds and has been included to supplement the outline information provided in Chapter 2 and Chapter 3.

Thermal Comfort Criteria in Design

Thermal Comfort Indices

Thermal comfort standards are used to define the conditions for thermal comfort and indicate the likely degree of occupant discomfort in a thermal environment. Significant contributions have been made by Fanger [Predicted Mean Vote (PMV), BS EN ISO 7730 [11]] and Gagge [Standard Effective Temperature (SET), ASHRAE Standard 55-92 [12]] toward the development of thermal comfort indices that reflect the combined effect of various environmental variables. Fanger used classical heat transfer theory and observations on a large number of people in laboratory experiments, to relate comfort to a heat balance equation for the human body. His equation captured four environmental variables (air temperature, mean radiant temperature, air velocity, and relative humidity), and two personal variables (activity level and clothing). [13] Gagge used similar methods to define his SET using a two-node (body and core) thermodynamic model of the human body. [14]

By using these two methods of comfort evaluation, a designer can obtain a range of environmental temperatures that are considered comfortable for a particular application. Within an automated blind system, this comfort band could be incorporated within the blind's control logic by developing a numerical model to modulate the slat angle to provide a floating comfort temperature.[15] However, in practice this strategy is never utilised, because the building services are more able to provide such functionality. The blind is instead used to work in harmony with the building services systems, which do utilise comfort bands prescribed by comfort theory, to reduce peak loads.

Although both these indices represent good engineering models for the human physiology, they are somewhat restricted when used to simulate an individual's perception of thermal comfort within a real world situation. Research that has compared field data with predicted values obtained from comfort standards, show that there are discrepancies in results. [16] [17] [18] [19] [20] [21] For example, ISO 7730 states that internal design temperature should be around 20-24°C for winter, 23-26°C for summer, whereas Humphries found that the neutral temperatures preferred by people ranged from 17 to 30°C.[22] An addition, climate chamber studies have suggested that the ISO 7730 PMV (Predicted Mean Vote) equation is internally inconsistent when predicting the discomfort of persons who are not in thermal neutrality.[23]

Psychological and Sociological Factors

One reason for these discrepancies is that any model of such an empirical nature can lead to a crude oversimplification of the interactions between people and their surroundings. [24] [25] The models require estimates of mean clothing levels and metabolic rates and are also based on simplified assumptions about thermal acceptability of the indoor environment.[26] Many researchers now suggest that the perception process is not solely governed by environmental stimuli and primary physiological responses but also by personal and contextual factors not considered by the laboratory studies. [27] [28] [29] [30] These include:

(i) perceived control of the environment; [31] [32]
(ii) stress in the workplace; [32] [33]
(iii) job satisfaction; [32] [33]
(iv) organisational structure, i.e. size of work-groups, layout of offices 32 [34] [35];
(v) expectations [32]
(vi) past experiences [32]

Adaptive Thermal Comfort Theory

In an attempt to quantify some of these psychological factors a few researchers have created, what one set of authors call, adaptive errors, shown within the ellipses of Fig. [28] Adaptive errors are also empirically derived variables that could be applied to correct the environmental comfort equations.

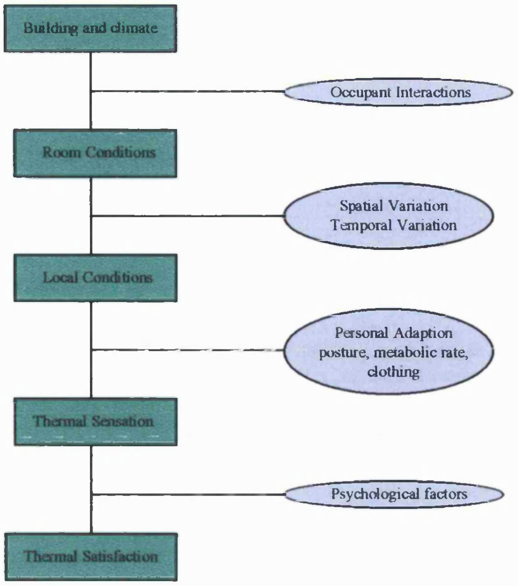


Figure II-I: Adaptive errors for a thermal comfort model.

This work is the next step towards developing a new set of standards that reflect people's adaptive mechanisms. However, its empirical nature means that it cannot account for all the diversities of real world situations and individual preferences; for example, some people prefer to sit in direct sunlight and some people do not. Each person, organisation, task and climate represents a change in psychological variables as much as it does physical variables; this makes an individuals perception of comfort very difficult to predict and thus impossible to guarantee with automatic control alone.

Visual Comfort Criteria in Design

Task Lighting Recommendations

International recommendations exist that specify the level of illuminance required to provide visual comfort for various tasks and locations. [36] [37] As with thermal comfort temperatures, these light levels were derived experimentally and reflect an average person engaged in a specific activity within a defined environment. [38] [39] Specified light levels can be provided by either daylight or artificial light, and venetian blinds can be used to control the amount of natural light used to illuminate a room as well as cut out any glare that may occur.

A misconception that has originated from the international recommendations is that we need to provide tightly controlled uniform lighting levels to achieve occupant visual comfort. In fact it is the contrast in light levels between the front and back of the room that we wish to limit, not the variation in the overall illuminance. Preferences to different combinations of illuminance and colour temperature vary from person to person depending on gender, age and past experience. [40] [41] An individual can have a tolerance to a wide range of light levels, so much so that reading performance does not show any significant variation between 100 and 5000 lux. [42]

Based solely on task performance it would be hard to justify specification of even 150 lux for an office interior. In reality such a level would not be specified, as many would subjectively view it as unpleasant. This shows that what is perceived as comfortable for the viewer is not necessarily related to task performance. Visual preferences are more difficult to predict than visual needs as individual preferences vary widely. Also in terms of human response to lighting, horizontal illuminance is one of a number of factors that contribute to the visual experience, and in terms of effect on preference possibly not particularly important.[43] [44] Thus if designing for comfort adherence to a set illuminance is unlikely to render success and raises the issue of the extent to which standards set for electric lighting quality can be applied to daylight.

The Positive and Negative Aspects of Natural Light

In the same way, different people react to windows, views, natural light and direct sunlight in different ways. Research has shown that such factors can have real psychological benefits to occupants, [45] [46] [47] [48] [49] but whether they have a positive affect on productivity depends on the task being undertaken. [50] [51]

The variable nature of natural light is one of the fundamental parameters of human life; it provides us with a perception of the running time and the conditions for psychological well-being in closed environments. Cool daylight which is filtered through trees or shading devices can provide visual relief to mentally counteract the thermal rigors of hot days. The carefully placed sparkle of sunlight can help relieve the impression of cold in the winter.

Clearly the utilisation of daylight reduces the need for artificial light and thus, along with switching routines, forms a vital strategy for energy conservation. However, the introduction of natural light is not a guarantee of visual comfort. Physiologically, daylight can cause visual discomfort when distributed unevenly in a room, resulting in patterns of high contrast. Outdoor views can make an interior seem dark and gloomy, and direct sunlight can make a room too bright. Both of these examples can cause discomfort glare and in the worst cases disability glare. Such inadequacies lead to occupants closing blinds and switching on lights, resulting in the unnecessary use of electric lighting.

Glare

The Commission Internationale de l'Eclairage (CIE) defines glare as:

"visual conditions in which there is excessive contrast or an inappropriate distribution of light sources that disturbs the observer or limits the ability to distinguish details and objects." [52]

Glare can be quantified by a glare index, depending mainly on window illuminance and reflections within the room. Glare caused by a direct view of the sky is considered to be acceptable if the glare index in a particular point in the room, does not exceed the recommended level for the particular operation. There are various forms of glare indices available for the designer, these include: the British Glare Index, based on research by Hopkinson and Pertherbridge [53]; and the CIE Glare Index proposed by Einhorn. [54] JJ Kim utilised the British Glare Index within his "conceptual framework for the dynamic control of daylighting and electric lighting systems" as a tool to ensure visual comfort. [55] However, in practice obtaining the correct measurement data required for the glare indices equations would prove difficult. In addition, the use of these indices, as with thermal indices, is limited by many assumptions and an analytical method that can not account for the subjective human responses already identified as being associated with visual comfort.

We can conclude from this that glare and view are the most important factors to consider when designing a venetian blind control strategy. However, both these factors are difficult to accommodate within a control strategy, as one affects the other; an occupant's decision on preferred blind angle depends upon a trade off of perceptions. Blinds do not need to modulate to provide a constant level of daylight as variations in natural light are tolerable and desirable.

Vision is the most developed of our senses and it can affect an individual's mood and cognition. [56] [57] It is not adequate to simply provide adequate illumination levels to satisfy the multidimensional nature of visual comfort. Daylight within buildings is provided for people, therefore daylighting design should respond to their visual and perceptual needs. As these needs are so variable and difficult to measure we must allow the occupant the luxury of being able to make adjustments.

The Psychological Benefits of Individual Control

We have shown that design comfort criteria, by definition, are related to an average person engaged in a specific activity within a defined environment. The terms average person or "user" are neutral and completely obscure differences amongst people. Settings and environments in which office workers feel most comfortable are as varied as their individual physiologies, thus making environmental design difficult. However, there is broad agreement among researchers that individual comfort and satisfaction can be attained by providing individual control of the local environment.[3] [58] [59] [60] [61] [62] [63]

Kroner's study of individually controlled environmental systems, which he named Environmentally Responsive Workstations (ERWs), epitomises this belief. [64] [65] By giving each worker their own set of environmental controls, i.e. their own radiant heater, air diffusers and task lighting, the occupants were able to adapt their local environments to individual preferences. The results were a rise in productivity and a higher degree of user satisfaction with the environment. As a check, Kroner then disabled the ERWs, thus taking away individual control, which conversely resulted in a fall in productivity levels.

The perception of being in control appears to be as important as having comfortable conditions. Perceived control depends on the presence, design and placement of control devices, but also on the overall effectiveness of control strategies, the attitudes and actions of the management and the way in which physical and human management systems operate together. A model produced by Bordass and Leaman identifies three primary factors that affect perceived control (see Fig). [66]

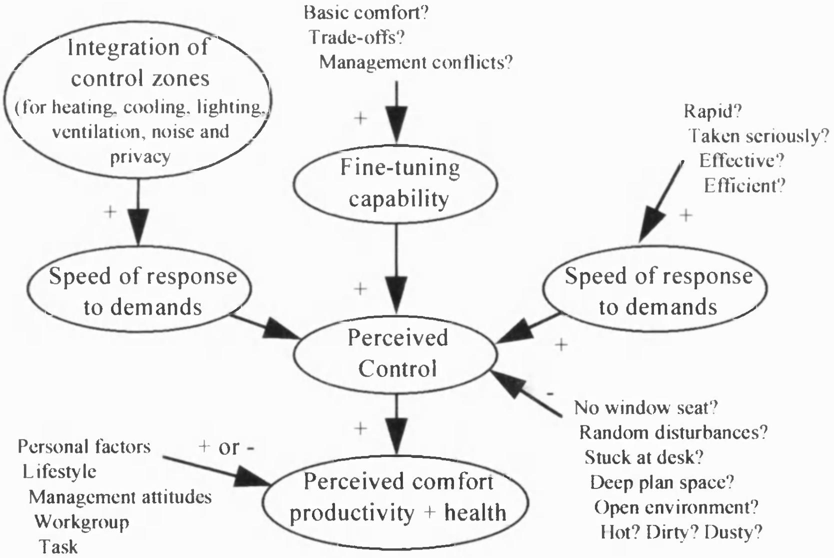


Figure II-II: Factor influencing perceived control.

Despite such advice, designers continue to use the design standards which knowingly are unable to respond to 20-30% of the building's population in terms of environmental comfort and well-being [9] and build essentially static buildings that are centrally controlled and unable to be fine-tuned by individual users. These buildings produce environments of moderate quality that can leave a percentage of the occupants dissatisfied and sometimes ill.

The most common solution to the problem is outlined in BRE IP3/95, which suggests that comfort should be provided by automatic control without energy wastage for a high proportion of the time and a facility should be available to allow the occupants to over-ride a system action so that they can make rapid adjustments to avoid discomfort.[60] [67] Environmental psychology theory tells us that by giving the occupant this form of personal control, we are providing them with three different forms of individual control:[68]

(i) *Decisional control* - this is the opportunity to make various adjustments;
(ii) *Cognitive control* - this is the perception of control, it represents the way in which an event is interpreted, appraised and learnt;
(iii) *Behavioural control* - this is when an occupant makes an adjustment to avoid a threatening event, such as overheating.

According to Veitch and Gifford[69] the availability of decisional control (e.g. adjustment of thermostats, window blinds etc.) and the perception of control, both contributed to user satisfaction. However, they also suggest that the exercise of control (behavioural control) decreased occupant satisfaction.

From this we can conclude that even by providing user over-ride on a control system, the occupant can still get frustrated with the system if they have to over-ride it constantly and this fact could have an affect on occupant comfort as well as productivity. Such frustration would largely stem from the inability of the automated system to predict uncomfortable conditions for the occupant. This would be especially true if those conditions were caused by factors the occupant believed the control system should respond to, such as glare due to the position of the sun.

References

[1] Lorsch H.G. and Abdou O.A., 1994, "The impact of the building indoor environment on occupant productivity: Part I: Recent studies, measures and costs", *ASHRAE Transactions*, Vol.100, Part 2, pp741-749.

[2] Wyon D.P.,1993, "The economic benefits of a healthy indoor environment", Proceedings of Healthy Air '94, Italy, pp.405-416.

[3] Pepler R.D. and Warner C.G., 1968, "Temperature and learning: an experimental study," *ASHRAE Transactions*, Vol.74, Part 2, pp211-219.

[4] Wyon D.P., 1974, "The effects of moderate heat stress on typewriting performance", *Ergonomics*, Vol.17, pp309-318.

[5] Burge S.A. et al., 1987, "Sick Building Syndrome: A study of 4373 office workers", Ann. Occ. Hygiene, Vol. 31, pp493-504.

[6] Wilkins A.J., 1993, "Health and efficiency in lighting practice", *Energy*, Vol.18, No.2, pp123-129.

[7] Dorgan C.E. et al., 1994, "Productivity link to the indoor environment estimated relative to ASHRAE 62-1989", Proceedings of Healthy Buildings '94, Budapest, pp461-472.

[8] Lorsch H.G. and Abdou O.A., 1994, "The impact of the building indoor environment on occupant productivity: Part II: Effects of temperature", *ASHRAE Transactions*, Vol.100, Part 2, pp895-901.

[9] Abdou O.A. and Lorsch H.G., 1994, "The impact of the building indoor environment on occupant productivity: Part III: Effects of Indoor Air Quality", *ASHRAE Transactions*, Vol.100, Part 2, pp902-913.

[10] Dorgan C.E. and Dorgan P.E., 1997, "Productivity link assessment to indoor air quality", Creating the Productive Workplace Conference, Workplace Comfort Forum, London.

[11] International Standards Organisation, 1995, "Moderate Thermal Environments - Determination of PMV and PDD indices and specification of the conditions for thermal comfort", ISO Standard 7730, Geneva, Switzerland.

[12] ASHRAE, 1992, "Thermal environmental conditions for human occupancy", ANSI/ASHRAE Standard 55-1992, ASHRAE.

[13] Fanger P.O., 1970, "Thermal comfort analysis and applications in environmental engineering", McGraw-Hill, New York.

[14] Gagge A.P., Stolwijk J.A.J. and Nishi Y., 1971, "An effective temperature scale based on a simple model of human physiological regulatory response", *ASHRAE Transactions*, Vol. 77, No.1, pp247-262.

[15] Rheault S. and Bilgen E., 1987, "Heat transfer optimisation of an automated venetian blind window system", ASES 12th National Passive Solar Conference, Portland, OR, pp122-128.

[16] Schiller G.E. et al., 1988, "Thermal environments and comfort in office buildings", *ASHRAE Transactions*, Vol. 94, No.2 pp280-308.

[17] Schiller G.E., 1990, "A comparison of measured and predicted comfort in office buildings", *ASHRAE Transactions*, Vol. 96, pt.1, pp602-622.

[18] Busch J., 1992, "A tale of two populations: thermal comfort in air-conditioned and naturally ventilated offices in Thailand", *Energy and Buildings*, Vol.18, pp235-249.

[19] Humphreys M.A., 1993, "Field studies and climate chamber experiments in thermal comfort research", Proc. Conf. Thermal comfort: Past, present and future, Building Research Establishment.

[20] de Dear R.J. and Fountain M.E, 1994, "Field experiments on occupant comfort and office building thermal environments in a hot-humid climate", *ASHRAE Transactions*, Vol. 100, No.2 pp457-475.

[21] Baker N. And Standeven M., 1996, "Thermal comfort in free running buildings", *Energy and Buildings*, Vol. 23, pp175-182.

[22] Humphreys M.A., 1976, "Field studies of thermal comfort compared and applied", *Building Services Engineer*, Vol. 44, pp5-27.

[23] Parsons K.C., Webb L.H., McCartney K.J., Humphreys M.A. and Nicol J.F., 1997, "A climatic chamber study into the validity of Fanger's PMV/PPD thermal comfort index for subjects

wearing different levels of clothing insulation", Proceedings of the CIBSE National Conference 1997, Alexandra Palace, Vol. I, 5-7 Oct 1997.

[24] Proshansky H.M., 1972, "Methodology in environmental psychology; problems and issues", *Human Factors*, 14, pp451-460.

[25] Russel J.A. and Ward L.M., 1982, "Environmental psychology", *Ann. Rev. Psychol.*, 33, pp651-688.

[26] Newsham G.R., 1997, "Clothing as a thermal comfort moderator and the effect on energy consumption", Energy and Buildings, Vol.26, pp283-291.

[27] McIntyre D.A.,1982, "Chamber studies - reductio ad absurdum?", *Energy and Buildings*, Vol. 5, pp89-96.

[28] Baker N. And Standeven M., 1994, "Comfort criteria for passively cooled buildings a PASCOOL task", *Renewable Energy*, Vol.5, Part II, pp977-984.

[29] Davies A.D.M. and Davies M.G., 1995, "The adaptive model of thermal comfort", *Building Services Research and Technology*, Vol.16, No.1, pp51-53.

[30] Cena K.M., 1993, "Thermal and non-thermal aspects of comfort surveys in homes and offices", Thermal Comfort: Past, Present and Future, BRE 264, Edited by N.A. Oseland and M.A. Humphries, BRE, Garston, 9-10 June 1993.

[31] Leaman A. and Bordass W. 1997, "Productivity in buildings: The killer variables", Creating the Productive Workplace Conference, Workplace Comfort Forum, London.

[32] Aronoff S. and Kaplan A., 1995, "Total workplace performance: Rethinking the office environment", WDL Publications, Ottawa.

[33] Sutherland V. and Cooper C.L., 1997, "Stress and the changing nature of work", Buildings in use '97: How buildings really work. Commonwealth Institute, London.

[34] Donald I., 1994, "Management and change in office environments", *The Journal of Environmental Psychology*, Vol.14, pp21-30.

[35] Loftness V. et al, 1994, "Beyond the open plan: New space planning concepts to support organisational, technical, and environmental change", Center for Building Performance and Diagnostics, Carnegie Mellon University.

[36] Chartered Institution of Building Services Engineers, 1984, "Code for interior lighting", CIBSE, London.

[37] Commission Internationale de l'Eclairage, 1975, "Guide on interior lighting", CIE Publication 29, Vienna.

[38] Beutell A.W., 1934, "An analytical basis for a lighting code", *Journal of the Illuminating Engineering Society*, Vol.27, pp5-.

[39] Weston H.C., 1945, "The relation between illuminance and visual performance", Industrial Health Research Board Report No.87, HMSO, London.

[40] Wells B.W.P., 1965, "Subjective responses to the lighting installation in a modern office building and their design implications", *Building and Environment*, Vol.1, pp57-68.

[41] Tregenza P.R., et al., 1974, "Consistency and variation in preferences for lighting offices", *Lighting Research and Technology*, Vol.6, pp205.

[42] Smith S,.W. and Rea M.S., 1978, "Proof-reading under different levels of illumination", *Journal of the Illuminating Engineering Society*, Vol.27, pp47-52.

[43] Loe D.L., Mansfield K.P. and Rowlands E., 1994, "Appearance of lit environment and its relevance in lighting design", Lighting Research and Technology, Vol.26, No.3, pp119-133.

[44] Bernecker C.A. et al, 1993, "Task lighting in the open office: A visual comfort perspective", Journal of the Illuminating Engineering Society, Winter 1993, pp18-25.

[45] Markus T.A., 1967, "The significance of sunshine and view for office workers", in Sunlight in Buildings, ed. Hopkinson R.G., Boeurcentrum International, Rotterdam.

[46] Butler D.L. and Biner P.M., 1989, "Effects of setting on window preferences and factors associated with those preference", *Environment and Behaviour*, 21, pp17-31.

[47] Heerwagen J.H. and Heerwagen D.R., 1986, "Lighting and psychological comfort", *Lighting Design & Application*, Vol.16, pp47-51.

[48] Ne'eman E., 1974, "Visual aspects of sunlighting in buildings", *Lighting Research and Technology*, Vol.6, pp159-164.

---

[49] Ne'eman E. and Hopkinson R.G., 1970, "Critical minimum acceptable window size: A study of window design and provision of a view", *Lighting Research and Technology*, Vol.2, pp17-27.

[50] Tennessen C.M. and Cimprich B., 1995, "Views to nature: Effects on attention", *Journal of Environmental Psychology*, Vol.15, pp77-85.

[51] Stone N.J. and Irvine J.M., 1994, "Direct or indirect window access, task type, and performance", *Journal of Environmental Psychology*, Vol.14, pp57-63.

[52] Commission Internationale de l'Eclairage Comittee E-1.1, 1987, "International lighting vocabulary", CIE Publication 17.4, Central Bureau of the CIE, Vienna.

[53] IES Technical Committee, 1967, "Evaluation of discomfort glare: the IES Glare Index System for artificial lighting installations", Tech. Rep. 10, illuminating Engineering society, London.

[54] Einhorn H.D., 1979, "Discomfort glare: a formula to bridge differences", *Lighting Research and Technology*, Vol. 11, pp90-94.

[55] Kim J.J.and Jones J., 1993, "A conceptual framework for dynamic control of daylighting and electric lighting systems", Conference Record of the 1993 IEEE Industrial Applications Society 28th Annual Meeting, Toronto.

[56] Butler D.L. and Biner P.M., 1987, "Preferred lighting levels: Variability among settings, behaviours and individuals", *Environment and Behaviour*, 6, pp695-721.

[57] Knez I., 1995, "Effects of indoor lighting on mood and cognition", *Journal of Environmental Psychology*, Vol.15, pp39-51.

[58] Haigh D., 1981, "User response in environment control", in The Architecture of Energy (Eds. D Hawkes and J Owers), Construction Press, London.

[59] Preller L. Et al., 1990, Indoor Air '90, Fifth International Conference on Indoor Air Quality and Climate, pp227-230.

[60] Bordass W.T., Bromley A.K.R. and Leaman A.J., 1995, "Comfort, control and energy efficiency in offices", BRE Information Paper, IP3/95.

[61] Slater A., 1995, "Occupant use of lighting controls in offices", *CIBSE Journal*, Vol.17, No.8, pp43.

[62] Bauman F., Arens E., 1996, "Task/ambient conditioning systems: Engineering and application guidelines", Center for Environmental Design Research, University of California, Berkeley, CA.

[63] Bordass W., Leaman A., 1997, "From feedback to strategy", Buildings in use '97: How buildings really work. Commonwealth Institute, London.

[64] Kroner W.M. and Stark-Martin J.A., 1994, "Environmentally responsive workstations and office-worker productivity", *ASHRAE Transactions*, Vol.100, Part 2, pp750-755.

[65] Kroner W.M., 1997, "An intelligent and responsive architecture", Automation in Construction, Vol.200, pp1-13.

[66] Bordass W., Leaman A., 1997, "Future buildings and their services", *Building Research and Information*, Vol.25, No.4, pp190-195.

[67] Van Paassen A.H.C. and Lute P.J., 1992, "Natural ventilation and automation with manual overriding are health solution", Energy and Buildings, Vol.18, No.2, 1992, p. 65-75.

[68] Averill J.R., 1973, "Personal control over aversive stimuli and its relationship to stress", *Psychological Bulletin*, Vol.80, 286-303.

[69] Veitch J.A. and Gifford R., 1996, "Choice, perceived control, and performance decrements in the physical environment", *Journal of Environmental Psychology*, Vol.16, pp268-276.

Appendix III Automated Blind Control: A State of the Art Review

Introduction

This Appendix is included to provide additional information about the variety of blind control strategies installed in buildings today in order to supplement the research and manufacturer information provided in Chapter Three. The study is not intended to be an exhaustive review of installed automated blind control system, as many exist. The study's aim is to give the reader a sample of the variety of different systems that are used and the nature of their operation and design intent.

The information in this appendix has been obtained largely from journals, case studies, and in some case conversations with designers. The majority of the information provided for the first eleven systems was obtained from 'a case study review of the intelligent skin' carried out by Harris and Wigginton.[1] Over a period of two years the authors visited and reviewed, in detail, a number of buildings whose skins exhibited, what the authors termed, 'intelligent features'. The final output of the project was a report that summarised the features of 23 buildings.

Due to the nature of the information sources, namely architectural journals, the systems reviewed here mainly represent the high end of blind control installations, which are often specifically designed for each particular brief rather than being off the shelf systems. Therefore this information should always be viewed as supplementary to Chapter Three.

Examples of louvre-shading systems have also been included, as their control algorithms are very similar to venetian blind control algorithms due to parallels in their geometry and operation.

The Case Study Summary Format

There are two case studies per page. For each case study the name of the building in which the system is installed is given along with a picture of the system or the building, details of the building type, its location, the architect and the date it was completed.

Details of the blind type are also given. Whether it is a venetian or a louvre blind, it's location, external, internal, within a double glazed unit or as part of a twin skin facade, and whether it has both tilt and lift modes.

Details of the control algorithms are then outlined. Which of the three levels of complexity, outlined in Chapter Three, are exhibited, threshold control, sun blocking control, or scene/mode control? Does the system allow an individual over-ride? Does its control strategy consider daylight control, solar heat control and glare control? Does the control system integrate seamlessly with the other control devices within the building?

And finally at the bottom of each case study box, a brief description of the control systems operation is the given along with some references for the information.

---

[1] Harris J. and Wigginton M., 1998, "A case study of the intelligent skin", First Draft, Intelligent Buildings Research Programme, School of Architecture, University of Plymouth, August 1998.

Commerzbank Headquaters

Type: High Rise Office Building
Location: Frankfurt, Germany
Architect: Foster & Partners
Date: 1997

Blind Type	
Venetian Blind	●
Louvre Blind	
External	
Mid-pane	
Mid skin	●
Internal	
Tilt Mode	●
Lift Mode	●

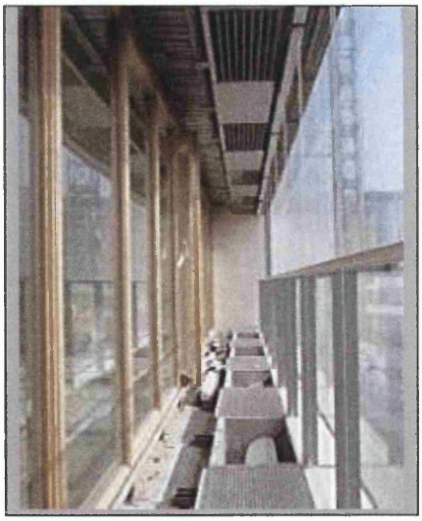
Controller Details	
Threshold	●
Sun Positioning	
Scene/Mode	●
Individual Control	●
Daylight Control	
Heat Control	●
Glare Control	
True Integration	

References

Harris and Wigginton Case Study
Davis C. and Lambot I., 1997, "Commerzbank Frankfurt: Prototype for an ecological high-rise", Birkhauser, Basel.
Connpagno A., 1995, "Intelligent glass facades: Material Practice Design", Birhauser,

Description of System

Blinds located in a 300mm ventilated cavity. Controlled to optimise conditions for natural ventilation or air conditioning modes. Lowered when sun is on facade and full over-ride available.



Stadttor (City Gate)

Type: Specular Office Development
Location: Dusseldorf, Germany
Architect: Petzinka Pink und Partner
Date: 1997

Blind Type	
Venetian Blind	●
Louvre Blind	
External	
Mid-pane	
Mid skin	●
Internal	
Tilt Mode	●
Lift Mode	●

Controller Details	
Threshold	●
Sun Positioning	
Scene/Mode	
Individual Control	●
Daylight Control	
Heat Control	
Glare Control	
True Integration	

References

Harris and Wigginton Case Study

Description of System

The blinds are threshold controlled with sun sensors located on each facade. Once lowered the slat angle is adjusted to 45°. Users have the facility to over-ride the blinds and have a choice of three tilt angles, closed, 45° and horizontal as well as up or down.

BRE Environmental Building

Type: Office Building
Location: Garston, UK
Architect: Feilden Clegg Architects
Date: 1996

Blind Type

Venetian Blind	
Louvre Blind	●
External	●
Mid-pane	
Mid skin	
Internal	
Tilt Mode	●
Lift Mode	

Controller Details

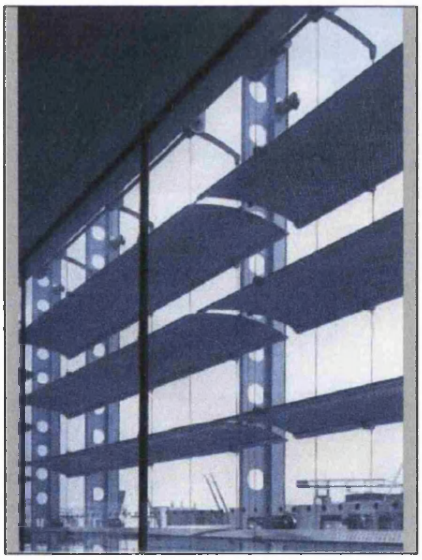
Threshold	
Sun Positioning	●
Scene/Mode	
Individual Control	●
Heat Control	
Daylight Control	●
Glare Control	
True Integration	

References

Harris and Wigginton Case Study 1997, Building Services Journal, Vol.19, No.3, pp18-23.
Anon, 1995, "A performance specification for the energy efficient office of the future", EEBPP, Crown..

Description of System

The translucent glass louvres are adjusted every 15 minutes and are controlled in bays. They are programmed to block solar penetration. When it is dull the louvres adjust to become light shelves. Users have full over-ride using a handheld IR controller.



Helicon Building

Type: Specular Office and Retail Development
Location: London, UK
Architect: Sheppard Robson
Date: 1996

Blind Type

Venetian Blind	
Louvre Blind	●
External	
Mid-pane	
Mid skin	●
Internal	
Tilt Mode	●
Lift Mode	●

Controller Details

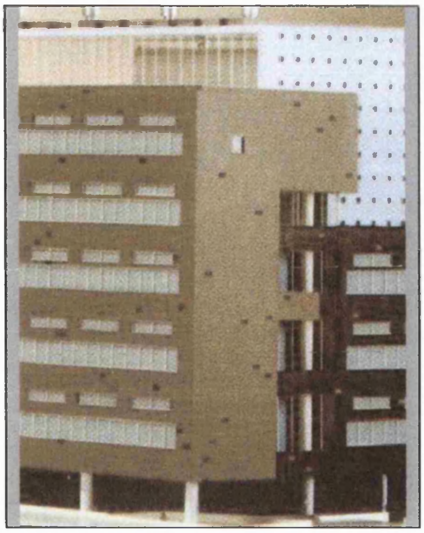
Threshold	●
Sun Positioning	
Scene/Mode	●
Individual Control	
Daylight Control	●
Heat Control	●
Glare Control	
True Integration	

References

Harris and Wigginton Case Study 1996, *Building Services*, Vol.18, No.7, Sept 1996.

Description of System

The perforated 450mm aluminium louvres are controlled in floors. The controller has a 150W/m^2 threshold and a 30-minute time delay to determine when blinds should be lowered. The sun sensors located in skin and on roof. Light and temp optimising.



Tax Office Extension

Type:	Government Office Building
Location:	Enschede, Netherlands
Architect:	Ruurd Roorda (RGA)
Date:	1996

Blind Type

Venetian Blind	●
Louvre Blind	
External	●
Mid-pane	
Mid skin	
Internal	
Tilt Mode	●
Lift Mode	●

Controller Details

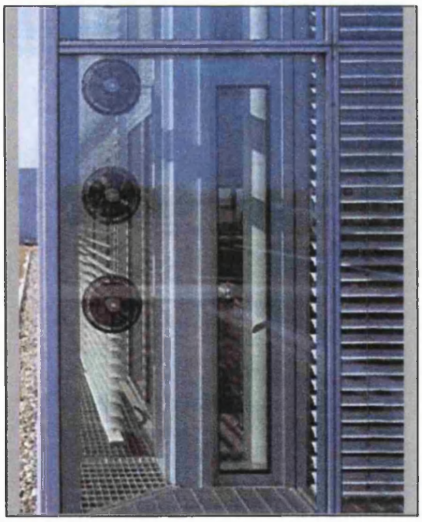
Threshold	●
Sun Positioning	
Scene/Mode	
Individual Control	●
Daylight Control	
Heat Control	
Glare Control	
True Integration	●

References

Harris and Wigginton Case Study
1996, "Future Buildings", *Building Services*, March 1996

Description of System

The central BMS system lowers the blinds in the morning if the solar irradiance rises above a pre-determined level. After this point the blinds are left to the occupants to control. In the evening the blinds are raised again ready for the next day.



Gotz Headquarters

Type:	Owner Occupied Office Building
Location:	Wurzburg, Germany
Architect:	Webler + Geissler
Date:	1995

Blind Type

Venetian Blind	●
Louvre Blind	
External	
Mid-pane	
Mid skin	●
Internal	
Tilt Mode	●
Lift Mode	●

Controller Details

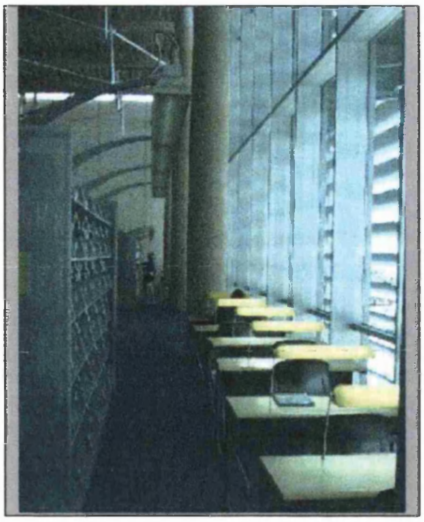
Threshold	●
Sun Positioning	●
Scene/Mode	
Individual Control	●
Daylight Control	●
Heat Control	
Glare Control	
True Integration	●

References

Harris and Wigginton Case Study
Webler M. et al, 1996, "A low energy intelligent building", 4th European Conf. on Arch. 26-29 March 1996

Description of System

Each bay of the building has two independently controlled blinds housed within the 600mm double skin cavity. The upper blinds redirect daylight and sunlight using a solar positioning algorithm. The lower blinds have a dark coating and are manual or automatic.



Phoenix Central Library

Type: Public Library
Location: Phoenix, Arizona, USA
Architect: Bruder/DWL Architects
Date: 1995

Blind Type

Venetian Blind	
Louvre Blind	●
External	●
Mid-pane	
Mid skin	
Internal	
Tilt Mode	●
Lift Mode	

Controller Details

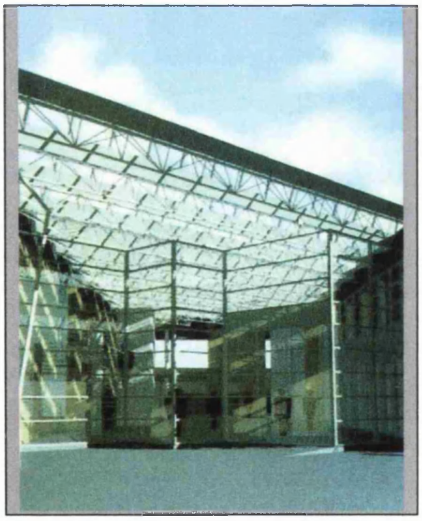
Threshold	
Sun Positioning	●
Scene/Mode	
Individual Control	
Daylight Control	
Heat Control	
Glare Control	
True Integration	

References

Harris and Wigginton Case Study
www.public.asu.edu/~bah24/c-index.htm

Description of System

The external louvers on the south glazed facade alter their tilt depending on a solar positioning algorithm. A row of six sensors mounted on the roof measure sky conditions for brightness.



Brundtland Centre

Type: Exhibition and Conference Centre
Location: Toftlund, Denmark
Architect: KHR Architects
Date: 1994

Blind Type

Venetian Blind	●
Louvre Blind	
External	
Mid-pane	●
Mid skin	
Internal	
Tilt Mode	●
Lift Mode	

Controller Details

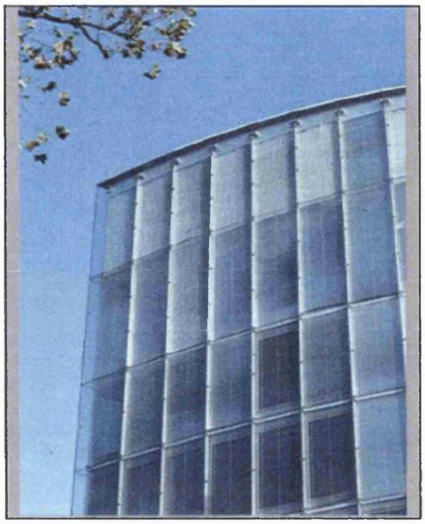
Threshold	
Sun Positioning	●
Scene/Mode	
Individual Control	●
Daylight Control	●
Heat Control	
Glare Control	
True Integration	

References

Harris and Wigginton Case Study
Esbensen T. et al, 1996, "Brundtland Centre Denmark", 4th European Conf. on Arch. 26-29 March 1996
Anon, 1994, "Focus 21 - A CEC Joule II project", *Building Research Information*, Vol.22, No.1.

Description of System

Central vision window incorporated thin inverted venetian blinds integrated within the double glazed units. These are controlled by the BMS to reflect daylight onto the ceiling of the office spaces. User presses "+ light" blinds open then lights on.



Duisburg Business Promotion Centre

Type: Office and Exhibition Space
Location: Duisburg, Germany
Architect: Foster & Partners
Date: 1993

Blind Type

Venetian Blind	●
Louvre Blind	
External	
Mid-pane	
Mid skin	●
Internal	
Tilt Mode	●
Lift Mode	

Controller Details

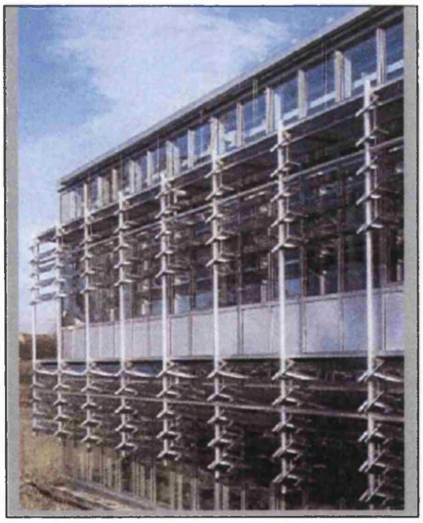
Threshold	
Sun Positioning	
Scene/Mode	●
Individual Control	●
Daylight Control	●
Heat Control	●
Glare Control	
True Integration	

References

Harris and Wigginton Case Study
Wigginton M., 1996, "Glass in Architecture", Phaidon.
Conmpagno A., 1995, "Intelligent glass facades": Material Practice Design", Birhauser, Basel

Description of System

The blinds are incorporated within a cavity and are 7% perforated to allow a view through when closed. The blind controller is linked to light and temperature sensors that determine the blind's tilt position.



Gartner Design Office

Type: Owner Occupied Office Building
Location: Gundelfingen, Germany
Architect: Kurt Ackermann und Partner
Date: 1991

Blind Type

Venetian Blind	
Louvre Blind	●
External	●
Mid-pane	
Mid skin	
Internal	
Tilt Mode	●
Lift Mode	

Controller Details

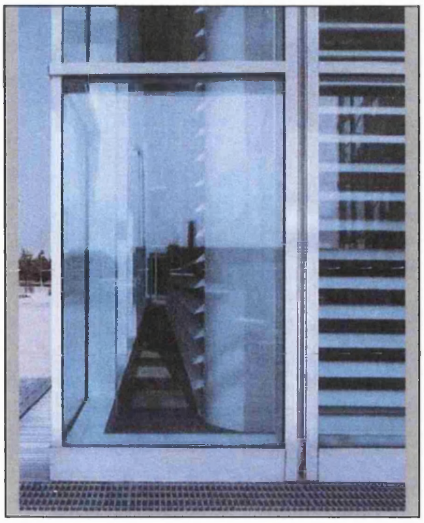
Threshold	●
Sun Positioning	●
Scene/Mode	●
Individual Control	
Daylight Control	●
Heat Control	●
Glare Control	
True Integration	●

References

Harris and Wigginton Case Study
Conmpagno A., 1995, "Intelligent glass facades": Material Practice Design", Birhauser, Basel.

Description of System

The BMS adjusts semi-transparent louvres every 5 minutes. A decision is made from sensor readings whether the shades should be positioned in a light-transmitting mode, a light guiding mode or a shading mode. A solar positioning algorithm is utilised.



Occidental Chemical Center

Type: Corporate Office Building
Location: Niagara Falls, New York, USA
Architect: Cannon Design
Date: 1981

Blind Type

Venetian Blind	
Louvre Blind	●
External	
Mid-pane	
Mid skin	●
Internal	
Tilt Mode	●
Lift Mode	

Controller Details

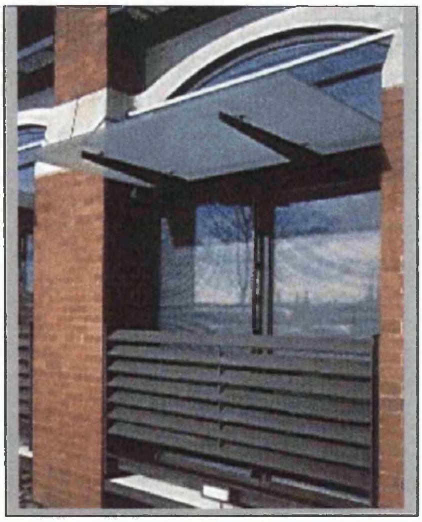
Threshold	
Sun Positioning	●
Scene/Mode	
Individual Control	●
Daylight Control	
Heat Control	
Glare Control	
True Integration	

References

Harris and Wigginton Case Study
Wigginton M., 1996, "Glass in Architecture", Phaidon.

Description of System

The louvres track the sun during the occupied hours. The controller uses feedback control to obtain the correct louvre tilt angle. Each building face is fitted with two photocell sensors which when shaded stop the motion of the sensor.



Inland Revenue Building

Type: Government Office Building
Location: Nottingham, UK
Architect: Michael Hopkins & Partners
Date: 1994

Blind Type

Venetian Blind	●
Louvre Blind	
External	
Mid-pane	●
Mid skin	
Internal	●
Tilt Mode	●
Lift Mode	

Controller Details

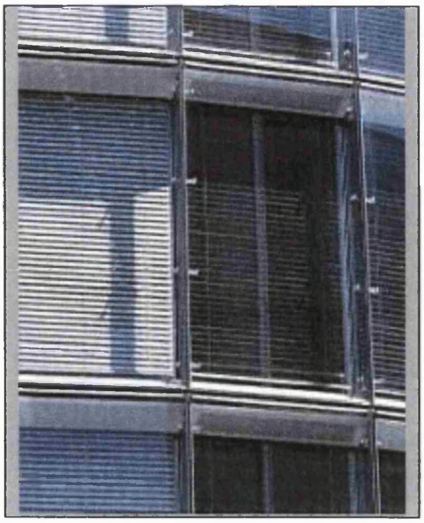
Threshold	●
Sun Positioning	●
Scene/Mode	
Individual Control	●
Daylight Control	●
Heat Control	
Glare Control	
True Integration	

References

Twinn C., 1997, "Specifying environmental conditions for buildings", in Naturally Ventilated Buildings ed. D Clements Croome E & FN Spon.
Anon, "Natural ventilation in non-domestic buildings" CIBSE, AM10, 1997, Balham

Description of System

Blinds in celestory window set at the optimum angle to allow in reflected light from light shelf but preclude direct solar radiation. Blinds in the main viewing window alter depending on solar illuminance and position.



RWE Headquaters

Type: High Rise Office Development
Location: Essen, Germany
Architect: Ingenhoven, Overdiek und Partner
Date: 1996

Blind Type

Venetian Blind	●
Louvre Blind	
External	
Mid-pane	●
Mid skin	
Internal	
Tilt Mode	●
Lift Mode	●

Controller Details

Threshold	●
Sun Positioning	
Scene/Mode	
Individual Control	●
Daylight Control	
Heat Control	
Glare Control	
True Integration	

References

Connpagno A., 1995, "Intelligent glass facades": Material Practice Design", Birhauser, Basel.

Description of System

Blinds can be controlled on a room by room basis.

Refurbished DTI Headquarters

Type: Government Office Building
Location: Westminster, London, UK
Architect: DEGW
Date: 1995

Blind Type

Venetian Blind	●
Louvre Blind	
External	
Mid-pane	●
Mid skin	
Internal	
Tilt Mode	●
Lift Mode	●

Controller Details

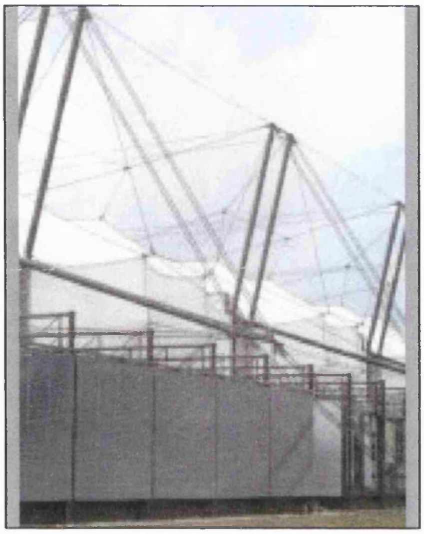
Threshold	●
Sun Positioning	
Scene/Mode	
Individual Control	●
Daylight Control	
Heat Control	
Glare Control	
True Integration	

References

Brister A., 1995, "Industrial Revolution", Building Services, Vol.17, No.9, Sept 1995

Description of System

Ventilated mid-pane venetian blinds come down in response to illuminances > 50,000lux on the South, East and West facades. When the blinds are lowered they are tilted to 40 to the horizontal. Manual control of up/down and tilt. Lighting control by telephone.



Schlumberger Research Centre

Type: Research Facility
Location: Cambridge, UK
Architect: Michael Hopkins Associates
Date: 1981

Blind Type

Venetian Blind	●
Louvre Blind	
External	●
Mid-pane	
Mid skin	
Internal	
Tilt Mode	●
Lift Mode	●

Controller Details

Threshold	●
Sun Positioning	
Scene/Mode	
Individual Control	
Daylight Control	
Heat Control	●
Glare Control	
True Integration	

References

Technical Blinds, 1986, T80 Sun Controller – 80mm Retractable Louvre Blind Brochure.

Description of System

System automatically lowers tilts and raises the blind according to the sun's intensity. Incorporates a digital clock to raise the blinds at the end of a working day and anemometer override.

Rank Xerox Research Facility

Type: Office Building
Location: Welwyn Garden City, UK
Architect: DEGW
Date: 1994

Blind Type

Venetian Blind	
Louvre Blind	●
External	●
Mid-pane	
Mid skin	
Internal	
Tilt Mode	●
Lift Mode	

Controller Details

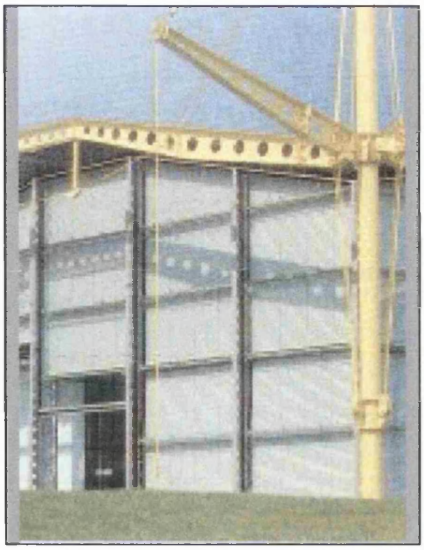
Threshold	
Sun Positioning	
Scene/Mode	●
Individual Control	
Daylight Control	●
Heat Control	
Glare Control	
True Integration	

References

Technical Blinds, 1993, product selection Brochure.

Description of System

Uses automatic constant daylight level control to tilt blinds and over-ride lights.



Renault Product Development Centre

Type: Research and Office Facility
Location: Swindon, UK
Architect: Foster Associates
Date: 1982

Blind Type		**Controller Details**	
Venetian Blind	●	Threshold	●
Louvre Blind		Sun Positioning	
		Scene/Mode	
External			
Mid-pane		Individual Control	
Mid skin		Daylight Control	
Internal	●	Heat Control	●
		Glare Control	
Tilt Mode	●		
Lift Mode	●	True Integration	

References

Technical Blinds, 1986, T80 Sun Controller – 80mm Retractable Louvre Blind Brochure.

Description of System

System automatically lowers tilts and raises the blind according to the sun's intensity. Incorporates a digital clock to raise the blinds at the end of a working day and anemometer override.

SONY Headquarters, Berlin

Type: Office Building
Location: Berlin, Germany
Architect: Murphy/Jahn
Date: 2000

Blind Type		**Controller Details**	
Venetian Blind		Threshold	●
Louvre Blind	●	Sun Positioning	●
		Scene/Mode	
External			
Mid-pane	●	Individual Control	●
Mid skin		Daylight Control	
Internal		Heat Control	
		Glare Control	
Tilt Mode	●		
Lift Mode	●	True Integration	

References

Anon, 2000, "Levolux puts Europe in the shade – with 1200 blinds" Design Lines, Levolux Newsletter No.5, p4.

Description of System

Sun blinds track the sun but bad detailing leaves gaps around the edges for glare to penetrate.

Appendix IV A Comparison of Solar Positioning Algorithms for Use in Blind Control Systems

Introduction

Initial studies show that the relative position of the sun in the sky compared to the orientation of the window is one of the most important factors in determining patterns in blind use (see Chapter Two). Therefore in order to be successful, an automated blind system must try and incorporate the sun's position within its control logic. Chapter Three showed that one way of achieving this is through the use of solar positioning algorithms.

This appendix reviews a series of algorithms that can be used to determine the position of the sun in terms of its altitude and azimuth and illustrates how these variables are often used to control the blind tilt angle to block the sun.

Calculating the Sun's Position

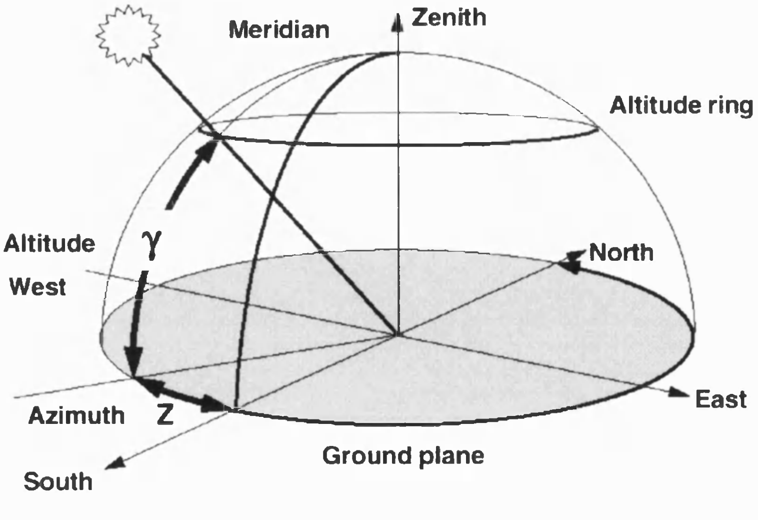


Figure IV-I: Position of the sun in the hemisphere of the sky

The position of the sun in relation to a point on earth can be described using Solar Altitude and Azimuth. The Solar Altitude is the angle of the sun in relation to the ground or horizon plane. The Solar Azimuth is the compass direction of the sun and can be measured in degrees clockwise from North, or degrees East or West of South (see Figure IV-I)

The equation for calculating Solar Altitude [γ] is as follows:

$$\sin\gamma = \sin\delta_s \sin\varphi - \cos\delta_s \cos\varphi \cos\xi \qquad \text{[Degrees]}$$

Where δ_s - is the solar declination, the angle of the sun's rays to the equatorial plane (positive in summer, see Figure IV-II);

φ - is the latitude of the site, the angle from the equator to a position on Earth's surface (see Figure IV-II);

ξ - is the hour angle, the angle the Earth needs to rotate to bring the meridian to noon. Each hour of time is equivalent to 15 degrees.

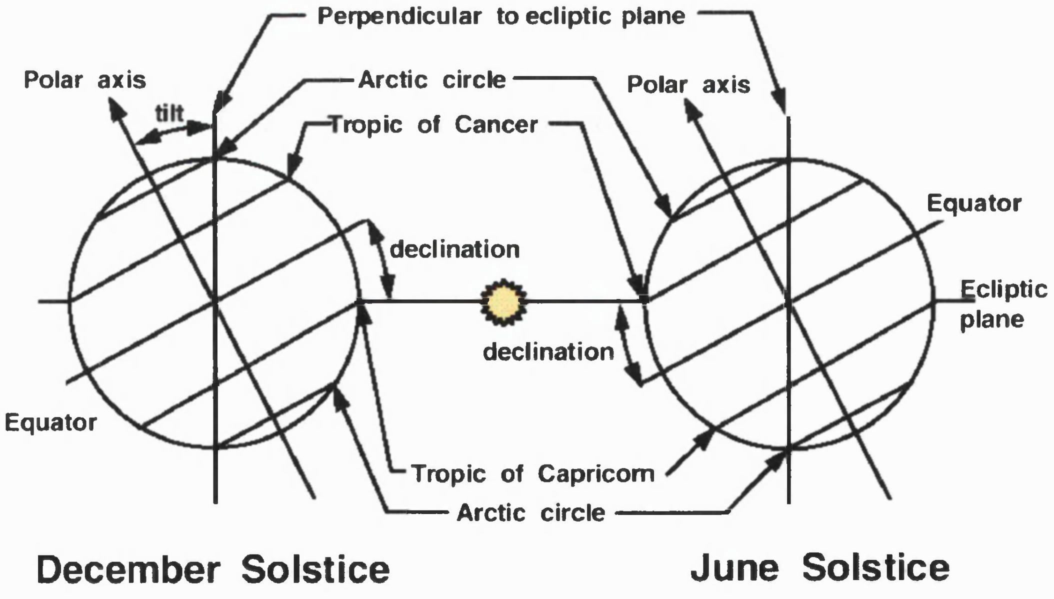


Figure IV-II: Cross section of the ecliptic plane showing the solar declination angle.

The equation for calculating Solar Azimuth [α] is as follows:

$$\cos\alpha = \frac{\cos\delta_s(\cos\varphi\tan\delta_s + \sin\varphi\cos\xi)}{\cos\gamma} \qquad \text{[Degrees]}$$

This equation gives an azimuth angle measured from North, where East is positive and West is negative. In order to improve the usability of any resulting algorithm, this result should then be converted to a geographical orientation expressed as a clockwise measurement from North so that it can be compared mathematically to a façade orientation value entered as part of the commissioning process.

As we can see from their equations, both the Solar Altitude and Azimuth are dependent on the Solar Hour Angle and Solar Declination. Both of these values vary throughout the day and year and a number of different methods of calculation are available, which vary in accuracy and complexity.

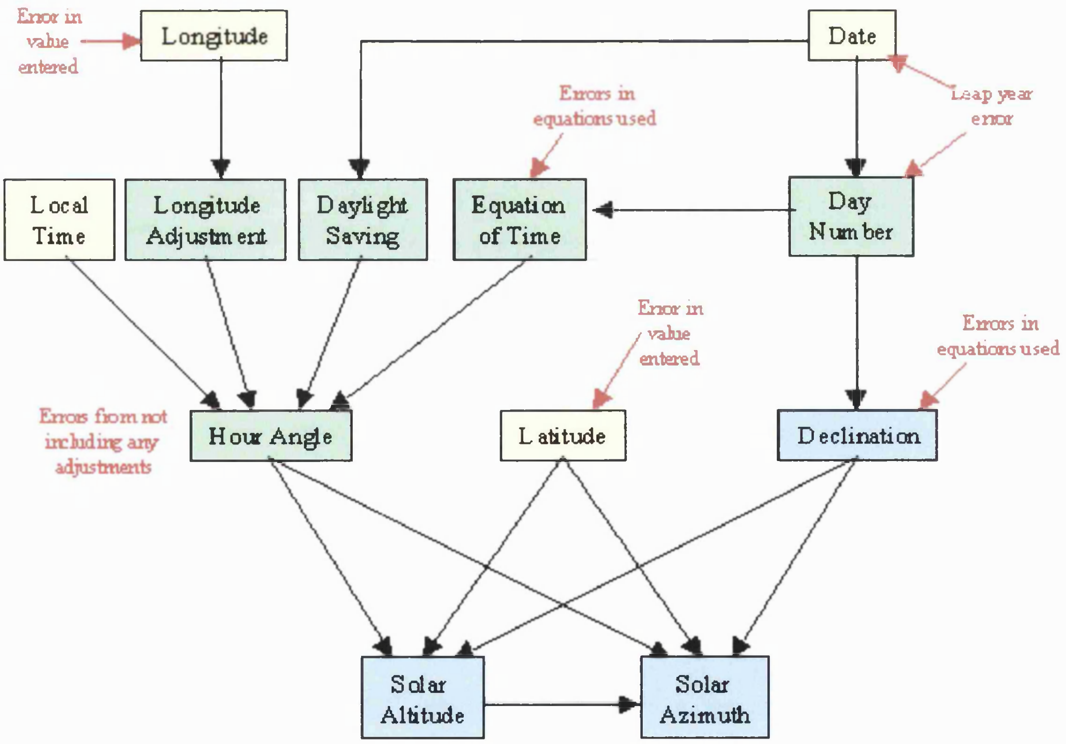


Figure IV–III: A flow diagram showing the stages of the solar altitude and azimuth calculation and the errors associated with each stage.

Solar positioning errors associated with Hour Angle errors

Small inaccuracies in the hour angle can have a large affect on altitude and azimuth calculations. The graph in Figure IV-IV shows the worst case scenario for altitude error resulting from True Solar Time error in the UK. This occurs on the equinoxes when inaccuracies effect the accuracy of around the sunrise and sunset periods. We can see from the graph that a 60 minute error in True Solar Time gives a 10 degree error in solar altitude, this error increases to a 15 degree error as our latitude approaches the equator.

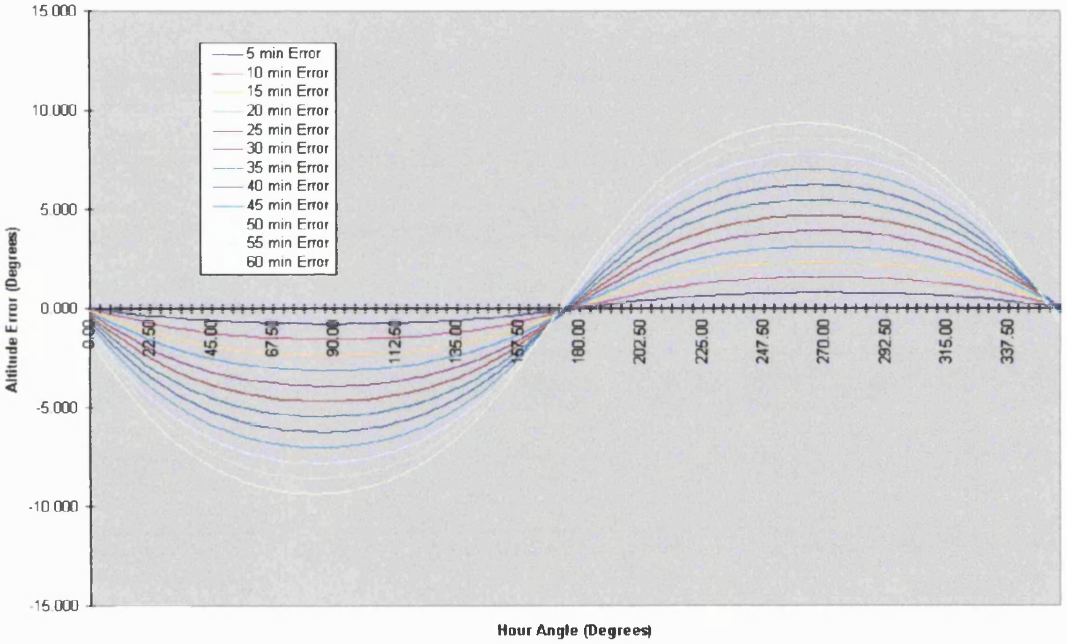


Figure IV-IV: Graph showing the effect of True Solar Time error on Altitude calculations during the Equinox in UK

The graph in Figure IV-V gives similar information for the effect of True Solar Time error on Solar Azimuth determination. For this parameter the worst case occurs during the summer solstice. We can see from the graph that a 60 minute error in Hour Angle gives a 29 degree error in solar altitude, this increases to a 36.6 degree error as our latitude approaches 45 degrees.

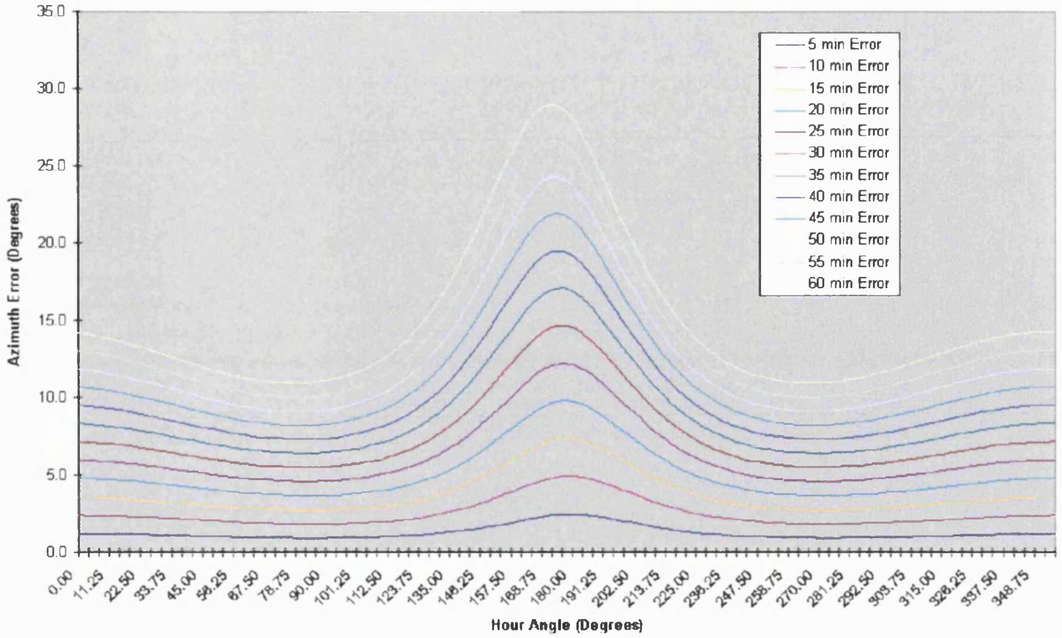


Figure IV-V: Graph showing the effect of True Solar Time error on Azimuth Calculations during the summer solstice in UK

Solar positioning errors associated with errors in the Solar Declination

Errors in solar declination have less of an effect on the solar altitude and azimuth. The graph in Figure IV-VI shows the worst case scenario for altitude error resulting from various declination errors. This occurs on the equinoxes when the rate of change of equinox is highest. We can see from the graph that a one-degree error in declination can result in a one-degree error in solar altitude. The lines on the graph become flatter as you approach the latitude of the North Pole.

The graph in Figure IV-VII gives similar information for the effect of solar declination error on the Solar Azimuth calculation. In the UK, the worst case also occurs during the summer solstice where errors are high and the rate of change of azimuth is also high. We can see from the graph that a one-degree error in declination can cause a 0.9-degree error in azimuth. As the latitude approaches the equator the errors can become a lot larger around noon, but these are often over a short period of time.

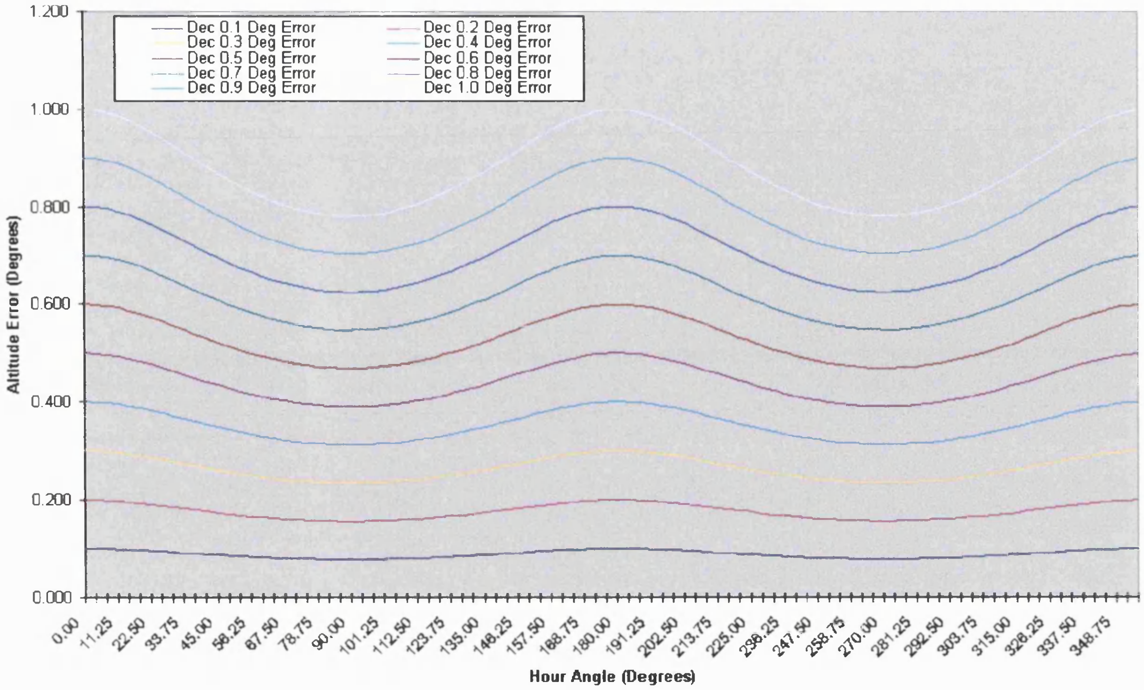


Figure IV-VI: Graph showing the effect of Declination Error on an Altitude calculation during the Equinox

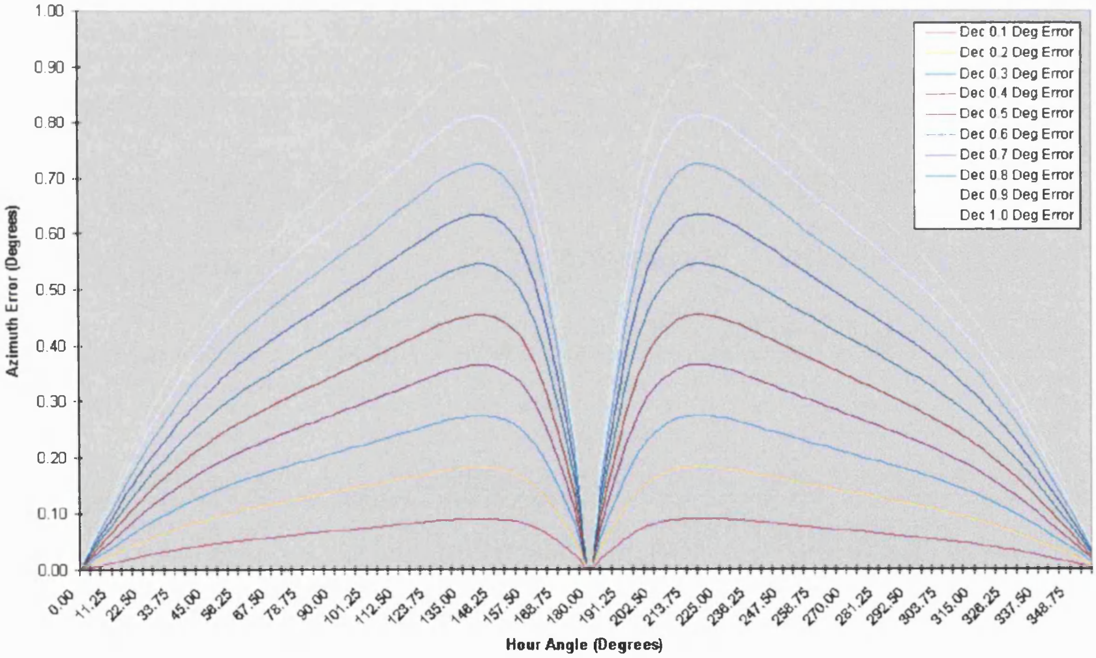


Figure IV-VII: Graph showing the effect of Declination Error on an Azimuth calculation during the Summer Solstice.

The Balance Between Simplicity and Accuracy

When choosing the control algorithm to be implemented within a real life control strategy, one must try to balance program speed, i.e. length and complexity of the algorithm, with program accuracy. Therefore in this appendix, as we examine each method of calculating the Solar Declination and Hour Angle, we shall consider the complexity and accuracy of each algorithm and how any inaccuracy could effect the final altitude and azimuth readings. This information will then be used with information gathered about the margins of error on a actual automated blind system in Appendix VII to determine which algorithm is best suited to the automated venetian blind applications. Both of these appendices are then used to aid the algorithm development for the test cell blind controller described in Chapter Seven and outlined in Appendix IX.

Algorithm Accuracy

The error associated with each method of calculating the solar declination and hour angle is determined by comparing each algorithm's results over the year 2002 to the accurate values given by the Royal Greenwich Observatory for that year (published in Muneer 1997).[1] A comparison of accuracy is then made by, firstly using Simpson's rule to determine the area under the graph of the absolute errors and then considering the maximum error.

Although algorithm accuracy is the most important factor to consider, less complex algorithms will be preferred to more complex algorithms where both demonstrate an equivalent accuracy.

Algorithm Length and Complexity

In this thesis it is shown that a LonWorks distributed control system is required to achieve the functionality and flexibility required for developing blind control algorithms (see Chapter Five and Six, and Appendix V). Distributed control is made possible in a LonWorks system by programming the application code for each device into a local processor chip or Neuron (see AppendixVI).

A variety of chips are available but the two most common types of chips support application code of either 2 Kbytes for small devices or 42Kbytes for larger devices.[2] The application code is written in a variation of ANSI C, called Neuron C and 6 pages of Neuron C equal approximately 300 bytes. With a limited capacity on a chip it is wise to ensure that program codes are as efficient as possible in terms of computational accuracy, speed and complexity.

In mathematics, the complexity of an algorithm is defined in terms of the number of mathematical operations needed to solve it (Coveney 1995).[3] Therefore in this text, in order to compare algorithms in terms of complexity, the number of variables and mathematical functions in an algorithm have been summed to give a value that represents the algorithm length and complexity.

Unfortunately, the Neuron C firmware in the Neuron Chip does not support the ANSI C math.h include file, which is used to define certain mathematical functions.[4] Most simple algebraic functions, such as multiplication and division are included in some form but the code for other mathematical functions must be defined and included within the application code. In the case of trigonometric functions, which rely on a series calculation whose length is dependent on the number of decimal places of accuracy required, this can take up vital space.

An example of a trigonometric series function is the cosine series:

$$\cos x = x - \frac{x^2}{2!} + \frac{x^4}{4!} - \frac{x^6}{6!}$$

Where x is converted from degrees to radians by multiplying by 180/π

This length of series will give a reasonable level of accuracy and its complexity rating is about 57. Therefore each time a trigonometric function is used within an algorithm, fifty will be added to the value to reflect the series iterations needed to perform such a function.

Pi values can be treated as a constant and therefore will incur an addition of four only once in the algorithm, to represent the definition of pi as a number to certain number of decimal places, e.g.

PI = 3.1415927

Calculation Procedures for Hour Angle

The Hour Angle is related to the Time of Day. In most control systems, devices either have access to the current time from within their own control chips or current local civil time via a real time keeper on the network. However, the Solar Hour Angle is dependent on the True Solar Time rather than the local clock time.

We can convert Local Clock Time *(LT)* [in hours] to True Solar Time *(TST)* [in hours] by using the three factors outlined in the following Equation:

$$TST = LT - TD + \frac{\lambda_s - \lambda}{15} + EOT \qquad \text{[Hours]}$$

Where TD - correction for summer time and daylight saving conventions (for example British Summer Time (BST), positive by 1 hour when clock time later than standard time GMT)

λ_s - longitude of the standard meridian (for example 0°, Greenwich meridian);

λ - longitude of the site;

EOT- equation of time, describes the variation between clock time and solar time due to eccentricity in the earth's orbit;

The Hour Angle is zero at noon of the TST, negative before noon and positive after noon. From the change in the True Solar Time we can find the change in the Hour Angle $(\Delta\xi)$ by using:

$$\Delta\xi = 15\ \Delta TST \qquad \text{[Degrees]}$$

The inclusion and exclusion of any of the three factors used to determine the true solar time could greatly effect the speed and accuracy of the overall algorithm.

Daylight Saving Adjustment

The inclusion of this adjustment depends upon the nature of the clock used by the algorithm. A LonWorks device can either use one of the timers included within the chip or use standard network variable type (SNVT) inputs such as SNVT_timestamp from other devices on the control network.[2]

If we use the former method we can get away with not using this adjustment within our algorithms if we ensure that the:

(i) time is set correctly to GMT (not BST) during the commissioning stage;
(ii) time keeping is not power supply dependent during power loss scenarios;
(iii) time is checked as part of the routine monitoring process.

If the latter method is used then it is likely that the daylight adjustment will be needed as the LonMark standard functional profile for a real time keeper takes into account daylight saving. [5] This is because other devices on the network such as plant schedulers and security systems need to know the local time. It is wise to use the real time keepers on the network if potentially you could have the solar positioning algorithm in a large number of devices, as time setting during the commissioning stage would be extremely time consuming.

In the UK, the correction for daylight saving has by far the biggest affect on the accuracy of the TST conversion giving a 60 minute error for a large proportion of the year, which leads to a 15 degree error in the Hour Angle. We can see from Figures IV-IV and IV-V that this can lead to a 9 degree error in solar altitude calculation a 29 degree error in solar azimuth calculation. This scale of inaccuracy is of course unacceptable and therefore it is vital that any solar positioning algorithm takes into account the BST hour change.

The Eighth European Parliament and Council Directive on Summer Time Arrangements states that from 1998, summer (or daylight saving) time will be kept between the last Sunday in March and the last Sunday in October. With this information it is possible to write a simple algorithm that ensures that *TD* is +1 during this period. This algorithm works in a similar way to the one that automatically adjusts the clock on your PC. Examples of how this can be achieved can be found in Muneer (1997)[1] and the controller code found in Appendix IX.

Longitude Adjustment

The longitude adjustment does not add many more lines of code to the algorithm, but it does add two extra input variables to the system at commissioning time (the latitude of site and the latitude of the time meridian). In the UK, errors of up to about 30 minutes or 7.5 degrees of Hour Angle do not cause too much concern. However in other countries, where time zones are spread across many a wide range of longitude, these errors can be far greater. For example in China, errors can be up to 3 hours or 45 degrees of Hour Angle, and this can have a serious impact on the accuracy of the algorithm.

The test room study showed that including a few extra global configuration properties (such as the latitude of site and the latitude of the time meridian) during the installation of a LonWorks device would not decrease the cost effectiveness of that device in terms of network installation and commissioning. Therefore if we wish to provide an algorithm a certain degree of flexibility, including the longitude adjustment parameter will not effect the overall system complexity and usability.

Equation of Time Adjustment

The final adjustment, the equation of time adjustment, has the smallest impact on the accuracy of the algorithm. It has a maximum of 15 minutes error as shown in Figure IV-VIII below:

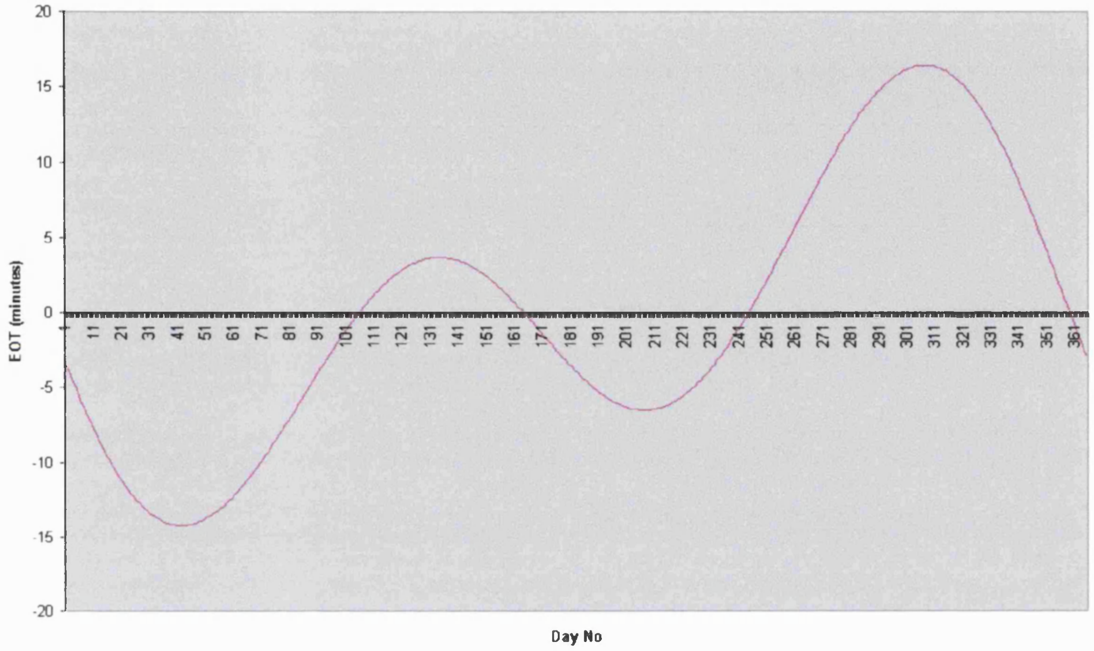


Figure IV-VIII: Graph showing the value of Equation of Time throughout the year

It is often tempting to leave this adjustment out of the solar positioning algorithm, but we have already seen from Figures IV-IV and IV-V that a 15 minute error can cause up to a 2.5 degree error in solar altitude and a 7 degree error in solar azimuth. Although these errors are only experienced for part of the year they are unacceptable when considering the degree of accuracy needed due to limitations in the blind positioning and therefore we concluded that it was important to include some form of EOT adjustment.

There are various algorithms available for calculating the EOT. A few of the most widely used algorithms are reviewed below.

Whillier's EOT equation

A low accuracy algorithm by Whillier (1979) can be found in Duffie and Beckman (1980).[6]

$$EOT = 9.87\sin 2B - 7.53\cos B - 1.5\sin B$$

Where B - the day angle [degrees] is given by

$$B = \frac{360(J = 81)}{364}$$

J - is the day number, where $J = 1$ on 1st Jan and $J = 365$ on 31st Dec

Woolf's EOT equation

A simple and relatively accurate method proposed by Woolf in 1968 (taken from Muneer 1997)[1] uses the following equation, which is accurate to within 50secs: [1]

$$EOT = -01236\sin\tau_d + 0.0043\cos\tau_d - 0.1538\sin 2\tau_d - 0.0608\cos 2\tau_d \qquad \text{[Hours]}$$

Where τ_d - the day angle [degrees] given by $\tau_d = \frac{360(J = 1)}{365}$

The Illuminating Engineering Society of North America EOT equation

An alternative medium accuracy algorithm for calculating the equation of time can be obtained from the Illuminating Engineering Society of North America (IES).[7] This equation uses the day number more directly and is accurate to within 40 seconds.

$$EOT = 0.170\sin\left[\frac{4\pi(J-80)}{373}\right] - 0.129\sin\left[\frac{2\pi(J-8)}{355}\right] \qquad \text{[Hours]}$$

Yallop's EOT equation

A high precision accurate algorithm for equation of time by Yallop (1992) can be found in Muneer 1997.[1] This algorithm is accurate to within 3 seconds and is valid for the period 1980-2050:

$$t = \frac{\left\{\left(\frac{UT}{24}\right) + D + (30.6m + 0.5) + [365.25(y - 1976)] - 8707.5\right\}}{365.25}$$

Where D - day of the month

m - month of the year

y - year

If $m > 2$ then $y = y$ and $m = m - 3$, else $y = y - 1$ and $m = m + 9$.

UT - Universal Time, time with no daylight adjustments, in UK this is GMT [hours].

The following terms are then determined:

$$G = 357.528 + 35999.05t$$
$$C = 1.915 \sin G + 0.020 \sin 2G$$
$$L = 280.460 + 36000.770t + C$$
$$\alpha = L - 2.466 \sin L + 0.053 \sin 4L$$

$$EOT = \frac{(L - C - \alpha)}{15}$$

Therefore:

Greenwich Hour Angle (GHA) = $15UT - 180 - C + L - \alpha$

and

$\xi = 15UT - 180 - C + L - \alpha + (\lambda_s - \lambda)$

A comparison of the EOT equations

The accuracy of these four equation of time formulae throughout the year can be demonstrated by comparing their results with highly accurate predicted values provided by the Royal Greenwich Observatory for the year 2002 and published in Muneer 1997.[1] The graph in Figure IV-IX illustrates the errors associated with each method. The slight jaggedness of the lines is due to the fact that the Royal Greenwich Observatory Data used was only accurate to the nearest minute, but this is not important for the information we wish to ascertain. Table IV-I compares these four algorithms in terms of accuracy and complexity.

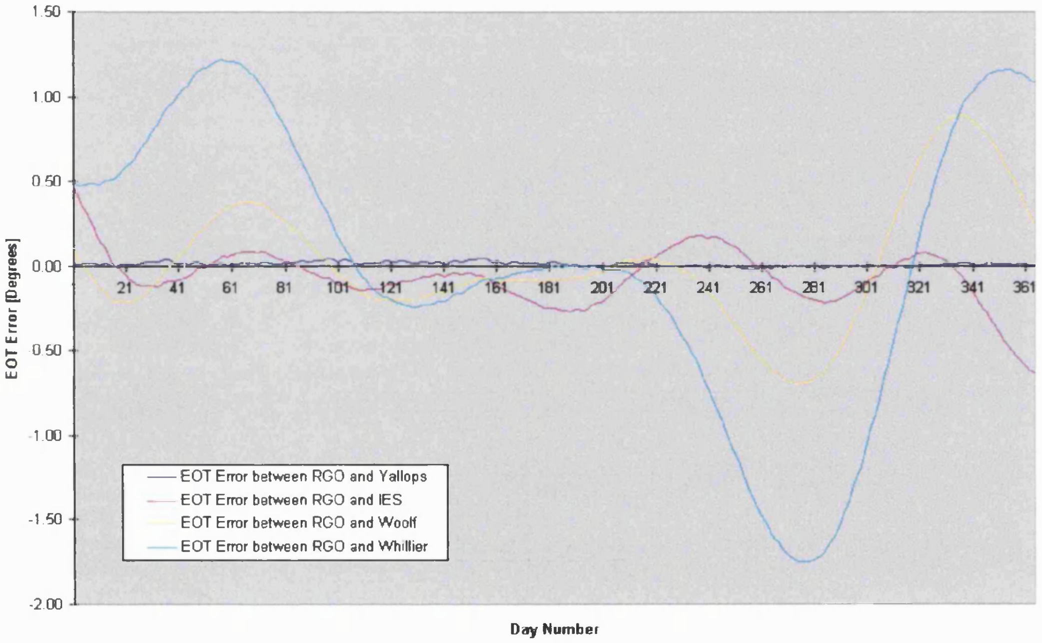


Figure IV-IX: Graph showing errors associated with each of the above method of calculating the EOT

Calculation Method	Complexity	Max Error	Accuracy (area under error curve)
Whillier	160	-1.7	78.54
Woolf	214	0.89	32.44
IES	111	-0.63	16.01
Yallop	257	0.04	1.78

Table IV-I: A comparison between the accuracy and complexity of the four methods of calculating the EOT

From Figure IV-IX and Table IV-I we can see that the IES formula is better than Whillier's and Woolf's formulae in terms of both accuracy and complexity. However, if the space is available Yallop's formula provides high accuracy.

Calculation Procedures for Solar Declination

The solar declination varies throughout the year and is dependent on the time of year (see Figure IV-X). Its value peaks at 23.44 degrees during the summer solstice and dips to –23.44 degrees during the winter solstice. A number of calculation techniques are available and some of the most common forms are reviewed below.

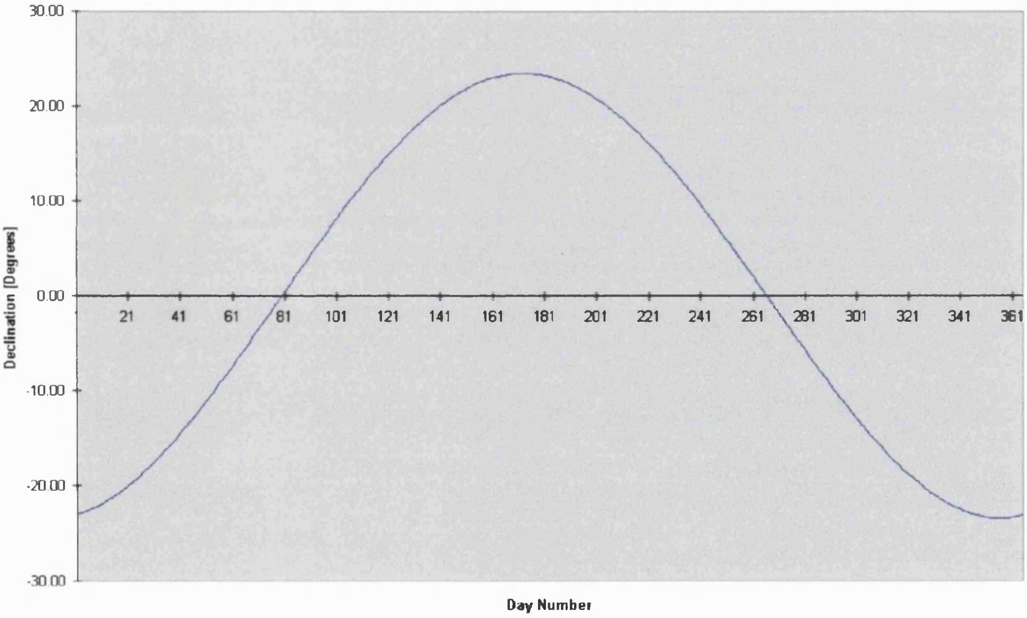


Figure IV-X: The variation in Solar Declination during the year

Boes' solar declination equation

A simple low accuracy value for solar declination can be found by using Boes' equation found in Kreider and Kreith (1981):[8]

$$\delta_s = \sin^{-1}\{0.39795\cos[0.98563(J-1)]\} \qquad \text{[Degrees]}$$

Cooper's solar declination equation

A similar equation by Copper (1969) can be found in Duffie and Beckman (1974)[6]:

$$\delta_s = 23.4\sin\left[\frac{360(284+J)}{365.25}\right] \qquad \text{[Degrees]}$$

Kreider's solar declination equation

A less accurate version of Cooper's equation can be found in Kreider (1987):[9]

$$\delta_s = -23.4\cos\left[\frac{360(J+10.5)}{365.25}\right]$$

Spencer's solar declination equation

A slightly more complex formula by Spencer (1971) that utilises the day angle to form a Fourier series equation for solar declination can be found in Tregenza (1993)[10].

$$\delta_s = 0.006918 - 0.399912\cos\tau_d + 0.070257\sin\tau_d - 0.006758\cos 2\tau_d + 0.000907\sin 2\tau_d \\ - 0.002697\cos 3\tau_d + 0.001480\sin 3\tau_d \qquad \text{[Degrees]}$$

Goulding's solar declination equation

A formula found in Goulding (1986) has a similar accuracy with slightly less complexity: [11]

$$\delta_s = \sin^{-1}\left\{0.39795\sin\left[J^* - 80.2 + 1.92\sin\left(J^* - 2.89\right)\right]\right\}$$

Where $J^* = \dfrac{360J}{365.25}$

Yallop's solar declination equation

An example of a high precision algorithm again originates from Yallop and can be found in Muneer (1997).[1] It is a continuation of the EOT formula and has an accuracy of 1 minute of arc:

Obliquity of ecliptic (ε) = $23.4393 - 0.013t$ [Degrees]

$\delta_s = \tan^{-1}(\tan\varepsilon \sin\alpha)$ [Degrees]

A comparison of the solar declination equations

The errors associated with all six methods of calculating solar declination can be assessed by comparing the results of each method to values given by the Royal Greenwich Observatory for the year 2002 (Muneer 1997).[1] The graph in Figure IV-XI illustrates the errors associated with all six methods. In addition, Table IV-II compares the six algorithms in terms of accuracy and complexity.

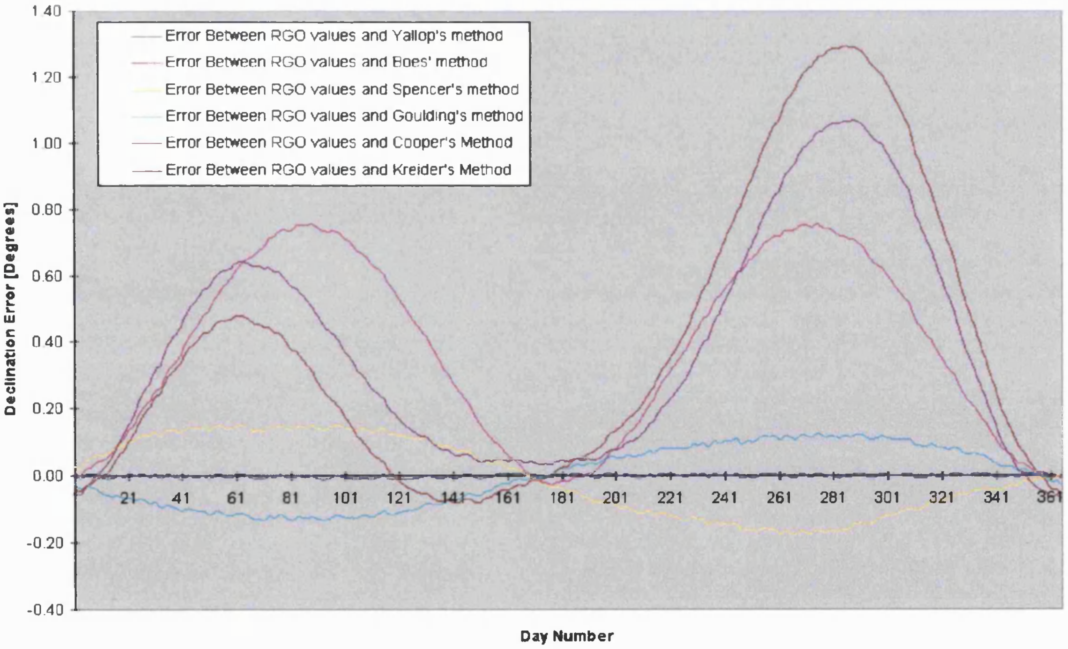


Figure IV-XI: Graph showing errors associated with each of the above methods of calculating solar declination

Calculation Method	Complexity	Max Error	Accuracy (area under error curve)
Boes	103	0.75	46.25
Cooper	54	1.07	46.62
Kreider	54	1.29	49.35
Spencer	362	-0.17	12.46
Goulding	158	-0.13	9.88
Yallop	409 (155 if EOT used)	-0.01	0.56

Table IV-II: A comparison between the accuracy and complexity of the six methods of calculating the solar declination reviewed

From Figure IV-XI and Table IV-II we can see that for the low accuracy algorithms, the Boes' and Cooper's formulae give similar accuracy but Cooper's formula is preferable as it is less complex. For the medium accuracy algorithms Spencer's and Goulding's formulae also provide similar levels of accuracy, but again Goulding's formula is preferred as it is less complex. However, if the space is available Yallop's formula provides a high level of accuracy and if Yallop's EOT formula is adopted the solar declination calculation only adds 155 functions.

Calculating the Blind Angle

The required angle of the blind necessary to block direct sunlight is directly related to the wall solar altitude angle, the angle the sun's rays make perpendicular to the window/blind plane. This incident angle is in turn related to the solar altitude and azimuth and can be calculated using the following equation:

$$\tan\theta_s = \frac{\tan\gamma}{\cos(\Delta\alpha)}$$

Where θ_s - wall solar altitude angle perpendicular to the window/blind plane;

γ - solar altitude angle;

$\Delta\alpha$- solar wall azimuth angle - difference between the solar azimuth angle and the orientation of window/blind azimuth.

Figure IV-XII below illustrates how these angles relate to blind and sun position.

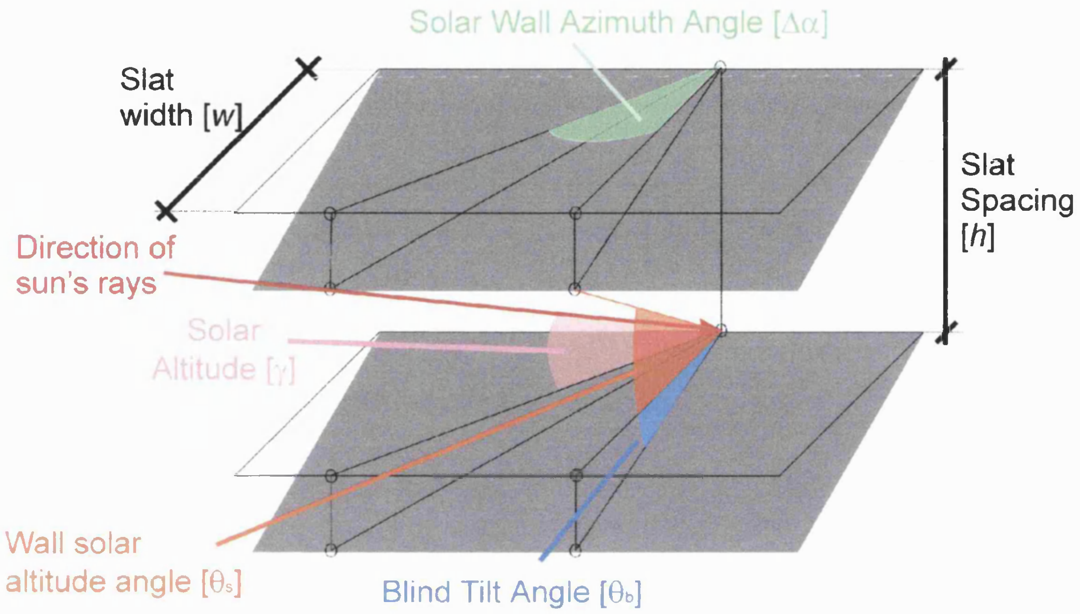


Figure IV-XII: Diagram showing the angles used to calculate the wall solar altitude angle perpendicular to the window/blind plane

In addition to the wall solar altitude angle, the required blind angle is also related to the geometry of the blinds, in particular the blind slat width [*w*] to blind slat spacing [*h*] (see Figure IV-XII). The formula for calculating the blind angle [θ_b], where the vertical blocking angle is 0 degrees, the horizontal slat angle is 90 degrees and the vertical daylight-redirecting angle is 180 degrees, is as follows:

$$\theta_b = \theta_s + 90 - \sin^{-1}\left(\frac{h}{w}\sin(90 - \theta_s)\right) \qquad \text{[Degrees]}$$

The majority of venetian blinds have a slat spacing to slat width ratio of 1:1.25. Figure IV-XIII shows the relationship between the solar incident angle and the required blind angle for a 1:25 ratio blind and a 1:1 ratio blind.

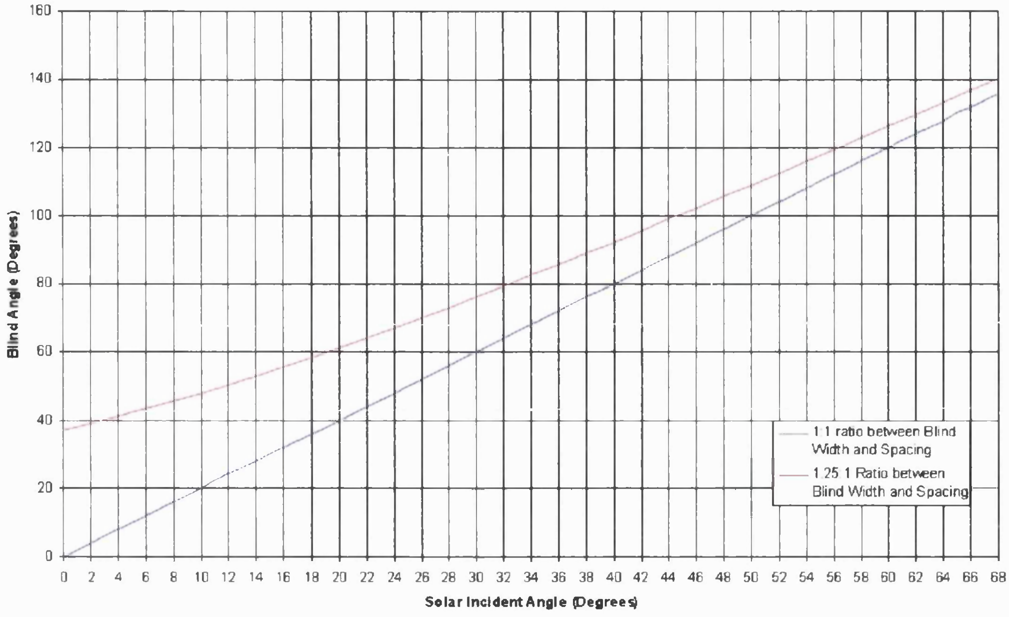


Figure IV-XIII: Graph showing the relationship between solar incident angle and blind angle

The area below a line on this graph represents the blind's ability to shade against solar penetration. Whereas the areas above represent the solar incident angles that would allow sunlight to penetrate through the various blind angles.

Determining the accuracy in solar positioning required for sun blocking

Throughout this appendix, the effects of a range of small errors on a number components of the solar positioning algorithms have been determined and illustrated in a graphical format. However, determining the accuracy required from a solar positioning algorithm, for the purposes of sun blocking, is dependent on the nature of the blind mechanism and the method of control.

Appendix VIII reviews the inaccuracies associated with the blind mechanism and the method of control adopted in the test room study. The information in this appendix is then used to determine the most appropriate solar positioning algorithms for use in the test room application.

---

[1] Muneer T., 1997, "Solar Radiation & Daylight Models for the energy efficient design of buildings", Architectural Press, Oxford UK.

[2] Anon, 1997, "LonWorks: Technology Device Data", Revision 4, Motorola Literature Distribution, Colorado, USA.

[3] Coveney P. and Highfield R., 1995, Frontiers of Complexity: The search for order in the chaotic world", Faber and Faber, London

[4] Neuron C Programmer's Guide, Document No. 29300, Echelon Corporation, 4015 Miranda Avenue, Palo Alto, California.

[5] Anon, 1997, "LonMark Functional Profile: Real Time Keeper", Rev1, LonMark Interoperability Association, www.lonmark.org.

[6] Duffie J.A. and Beckman W.A., 1980, "Solar engineering of thermal processes", John Wiley & Sons, New York.

[7] IES Calculation Procedures Committee, 1984, "Recommended practice for the calculation of daylight availability", Journal of the Illuminating Engineering Society of North America, Vol.13 (4) pp 381-392

[8] Kreider J.F. and Kreith F., 1981, "Solar Energy Handbook", McGraw-Hill, New York.

[9] Kreider J.F. Hoogendoorn C.J. and Kreith F., 1987, "Solar design: Components, systems, economics", Hemisphere Publishing Corporation, New York.

[10] Tregenza P. and Sharples S., 1993, "Daylight algorithms", ETSU S 1350,ETSU, DTI, London.

[11] Goulding J.R., Lewis J.O. and Steemers T.C., 1986, "Energy in architecture: The European passive solar handbook", B.T.Batsford Limited, London, UK.

Appendix V An Overview of Field Bus Systems

Introduction

This appendix reviews the concept of integrated field bus control and accompanies Appendix VI, which provides more detailed information into the workings of LonWorks, the field bus system used in this study. This appendix finishes by providing a state of the art review of integrated control systems and prediction for the future.

The Development of Field Bus Control

Traditional Propriety Control Systems

Until recently, nearly all of a building's automated installations were proprietary in nature, with all of the components within each individual subsystem being supplied by a single manufacturer. For example, a lighting manufacturer would supply all the controls for the lighting system and the HVAC manufacturer would supply all the controls for their heating and ventilation system. This arrangement was largely due to the fact that the development of building control systems had been based around a concept of centralised control, where each individual subsystem had its own substation that utilised self contained control operations to drive dumb field devices (see Figure V-I). Therefore the devices in each subsystem had little or no interaction with the devices in other subsystems.

Appendix I reviewed the integration model produced by DEGW to show the levels of technological and commercial development required before designers are able to attain truly integrated buildings. Centralised propriety control systems represent Level Two of that model.

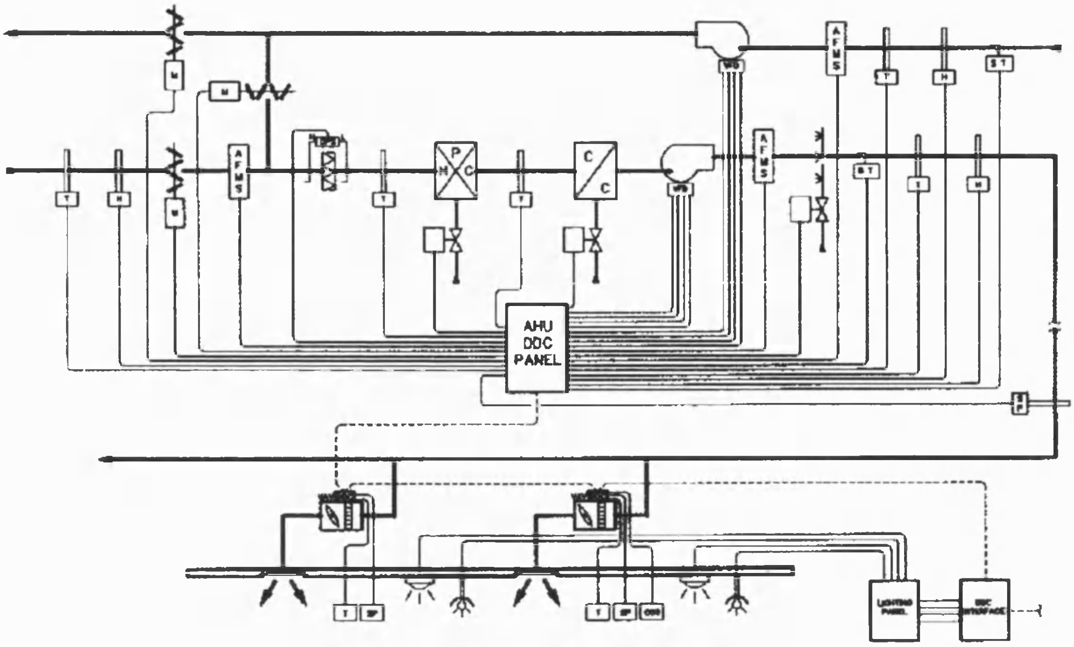


Figure V-I: A centralised control solution for an air conditioning and lighting system, each with separate control substations

The disadvantages of the proprietary centralised control for the building designer and user can be summarised as follows:

- building owner is locked into a particular supplier for the life of their building services. This results in high on-going costs and a limited choice of service and changes;
- systems are expensive to install due to the large amounts of wiring required and the fact that some systems require duplicate parts because there is no exchange of information between subsystems;
- systems are expensive to change as they require lots of custom programming and they must be changed individually.

This form of control was considered inappropriate for the kind of system being developed within this thesis as it was unable to incorporate learning on an individual blind basis simply and effective whilst still offering flexibility.

Open Interoperable Building Control Networks

The advent of cheap network devices has facilitated the development of bus architectures for implementation in control systems at the field device level. In contrast to a centralised control system, a field bus system has local processing at each device, whether it is a simple switch or a complicated operator interface. This methodology enables a device from one manufacturer to communicate in a peer to peer fashion across a bus media with a device from another manufacturer by using a common communications protocol (see Figure V-II). Each device understands its role and takes decisions according to the information it sees across the network.

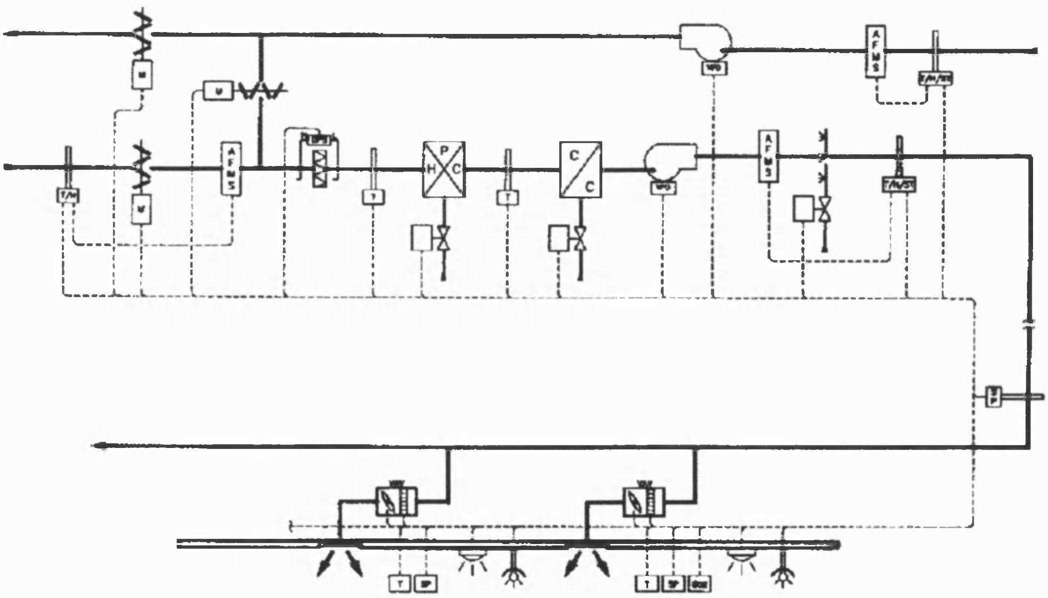


Figure V-II: A digital field bus solution for an air conditioning and lighting system

The advantages of this kind of interoperability over the proprietary nature of traditional systems include:

- *Lower cost* - due to the need for less wiring;
- *Easier to commission, manage and maintain* – since the controls electronics is embedded and housed in the plant, the electrical interface problem are eliminated by the manufacturer and not left to the controls system installer. Also one operator can look after many systems as they look and feel the same;

- *More flexible and adaptable* - no proprietary manufacturers locked into system and the system's logical connections can be changed at any time without physically changing the wiring;
- *Interoperable plug and play functionality* – data sharing simple between HVAC, lighting, fire and security systems, offering possible safety benefits;
- *More information* – the data transparency of the network allows unprecedented levels of information to be obtained about a process in real time;
- *Better fault diagnostics* – a modular approach means that faults can be quickly identified and traced to specific locations;

The ISO/OSI 7-Layer Reference Model

Of course the key to interoperability is to define a standard communications protocol for all of the controls devices to use. To help with such a process, the International Organisation of Standardisation derived a framework which both information technology and control communications protocol development should reference. The framework was called the Open Systems Interconnection (OSI) Basic Reference Model and it enabled the designer to divide protocol requirements into seven hierarchical layers of functionality that could be made available to a networked device. The layers are outlined in the table below.

Layer	Name	Purpose	Operation
USER			
7	Application	Device Application Compatibility	This is the code written by the user that governs the operation of the device.
6	Presentation	Data Interpretation	Deals with the common application orientated functions for instance dealing with known data structures such as ASCII strings or conversion between codes. (SNVTs)
5	Session	Remote Access	Takes the communication from the transport layer below and supports application orientated functions (BINDING)
4	Transport	End to End Reliability	Accepts data from the above and passes it down to the network layer in convenient packets for transmission
3	Network	Destination Addressing	Controls the operation of the subnet to ensure that packets of information are presented to the correct devices
2	Datalink	Media Access and Framing	Takes the raw transmission from below and presents it above as error free data frames. Deals also with acknowledgements and other error checking.
1	Physical	Electrical Inter-connection	Concerned with transmitting over the physical medium of the network, e.g. twisted pair, optical fibre, power line etc.

Table V.I: The ISO/OSI 7-Layer Reference Model[1]

Building Automation Communications Protocols

The requirements of control network systems are slightly different to information technology systems because control loops are not usually restricted to a single small circuit board. Several field bus protocols are currently competing for the building automation market. These include:

- *LONWORKS*
- *European Installation Bus (EIB)*
- *BATIBUS*
- *PROFIBUS*
- *WORLDFIP - The Factory Instrumentation Protocol (FIP)*
- *Firm Neutral Data Transmission (FND)*
- *ASHRAE's BACnet*

The Need for Standardisation

The availability of all these systems and their individual communications protocols has produced a somewhat complex marketplace, with industry unwilling to commit to a specific bus before the emergence of a clear market leader. The situation has compounded by the standardisation process that is been unable to define a clear standard.

CEN (European Committee for Standardisation) Technical Committee 247, Working Group 4 is responsible for arriving at a standard for communications between HVACR products. Their draft agreement states that a building control system should be broken down into three different types of communication:

- *Management* between PCs – using BACnet or the German FND;
- *Automation* between intelligent controllers - using BACnet, Profibus or WorldFIP;
- *Field* between terminal controllers and sensors – using LonWorks, Batibus or EIB.

Protocol Requirements and Selection

In terms of communications protocols, this thesis is primarily concerned with providing decentralised peer to peer control with each peer having the ability to facilitate learning at the field level. In order to make the choice of which protocol to use for this research, a list of protocol requirements were drawn up and each protocol was examined for its ability to meet those requirements. The requirements were as follows:

- *Robustness* – built in redundancy, easy adaptation and addition, error checking and message acknowledgements for reliable message transfer and fault tolerance isolation and recovery, number of layers of OSI model satisfied;
- *Flexibility* – sufficient capacity for a large number of devices, spaced far apart and connected using different topologies and media, as well as a number of application resources to ensure democratic operation;
- *Interconnectivity* – capacity of channels and division of the network and a flexible addressing service;
- *Interoperability* – a need for a consistent protocol stack, consistent application behaviour, tools to do system level verification and tools to migrate transparently from development to field systems;
- *Manageability* – simple but effective installation, commissioning, management and monitoring capabilities;
- *Usability* – plug and play functionality, easy to use human machine interfaces;
- *Suitability* - to field level peer to peer control;
- *Availability* – the number of UK HVAC controls manufacturers that were already using the protocol as well as blind and vent actuator manufacturers;
- *Approachability* – the openness of the protocol in terms of information available on the web and the learning courses available through their respective organisations;

The following forms summarise the decision making process.

---

[1] ISO 7498 Information processing systems: Open systems interconnection basic reference model, International Organisation of Standardisation, Geneva

LONWORKS

Creator:	Echelon
Country of Origin:	United States
CEN Level	Field

ECHELON
THE LONWORKS COMPANY

General Description

Mike Markkula, the co-founder of Apple, created LonWorks and it's supporting US Company Echelon, in 1986. A LonWorks system consists of a network of micro-controllers, which have peer to peer communications over a variety of different media.

Robustness

OSI 7 Layer Model			
7	Application Layer	Application Code	●
6	Presentation Layer	Common language	●
5	Session Layer	Controls Dialogue	●
4	Transport Layer	Optimises Network	●
3	Network Layer	Data transport	●
2	Data Link layer	Message structure	●
1	Physical Layer	Physical media	●

Notes

Robust communications protocol which conforms to all 7 layers of the OSI model

Robustness Rating ●●●●●

Flexibility and Interconnectivity

Media		*Topologies*	
●	Twisted Pair	●	Free Topology
●	Power Line	●	Point to Point
●	Infra Red	●	Bus Topology
●	Radio Frequency	●	Tree Topology
●	Optical Fibre	●	Star Topology
●	Coaxial	●	Ring Topology
●	Telephone Line		

It has the ability to be able to support free topology networks and a variety of media

Flexibility Rating ●●●●●

Interoperability

Field Interoperability	●●●●●
Automation Interoperability	●●●●●
Management Interoperability	●●●●
Interoperability Association	LonMark

LNS Architecture provides facility to provide interoperability at the management level whilst control modules deal with automation level.

Interoperability Rating ●●●●●

Manageability

The LNS Architecture and the associated tools provide an easy to use plug and play interface for all aspects of network management. However, each node installed using LNS requires a $2 license.

Manageability Rating ●●●●

Suitability

LonWorks is not aimed specifically at any market, however the system is flexible and robust enough to adapt to most distributed control applications.

Suitability Rating ●●●●●

Availability

Due to its robustness, LonWorks has been utilised by a large number of HVAC and building automation controls manufactures, therefore a variety of products are available

Availability Rating ●●●●●

Approachability

A wealth of information about the finer details and the workings of LonWorks is available on the web and on CD. A large number of free courses are provided by user groups and by Echelon themselves.

Approachability Rating ●●●●●

Conclusions

LonWorks offered a complete protocol that can handle, field, automation and management levels of integration and is robust, flexible and simple to use. Information was also freely available.

Overall Rating ●●●●●

Additional Information

Additional information can be found at www.echelon.com and www.lonmark.org.

European Installation Bus (EIB)

Creator:	Siemens
Country of Origin:	Germany
CEN Level	Field

General Description

EIB is a decentralised, peer to peer network operating system, which uses a serial transmission protocol. When a decision was being made about the protocol to use for this research, EIB seemed to be lagging behind LonWorks. It has caught up somewhat, but the score given here reflects the past.

Robustness

OSI 7 Layer Model			
7	Application Layer	Application Code	●
6	Presentation Layer	Common language	
5	Session Layer	Controls Dialogue	
4	Transport Layer	Optimises Network	●
3	Network Layer	Data transport	●
2	Data Link layer	Message structure	●
1	Physical Layer	Physical media	●

Notes

The system uses building blocks known as Bus Access Units (BAUs) and these are defined by a standard specification and can be implemented on any chip or processor platform chosen. For this reason the protocol claims to be totally open as it is a specification and not an implementation (like a chip or transceiver).

Robustness Rating ●●●●

Flexibility and Interconnectivity

	Media		*Topologies*
●	Twisted Pair		Free Topology
	Power Line		Point to Point
	Infra Red	●	Bus Topology
	Radio Frequency	●	Tree Topology
	Optical Fibre	●	Star Topology
	Coaxial	●	Ring Topology
	Telephone Line		

Originally EIB offered little flexibility. It had limited network capacity and zoning required higher level buses. However, since then it has acquired the ability to run on the Ethernet it has recently formed an alliance with BATIBUS, a free topology system. Now a variety of transceivers can be used for a variety of media.

Flexibility Rating ●●●

Interoperability

Field Interoperability	●●●●●
Automation Interoperability	●●●●
Management Interoperability	●●●●
Interoperability Association	EIBA

The Bus Access Unit locally implements the operating system and caters for user RAM and EEPROM space.

Interoperability Rating ●●●●

Manageability

Management tools seemed primitive compared to LonWorks. However, since obtaining ethernet compatibility this has changed.

Manageability Rating ●●●

Suitability

It is very much a field level protocol.

Suitability Rating ●●●●

Availability

It was not taken up by the HVAC industry and seemed more suited to Home Automation. But this again is changing slowly.

Availability Rating ●●●

Approachability

There was a limited amount of information on the internet unless you were a member of the EIBA. Most of the structure is based in Europe and not much support in UK.

Approachability Rating ●●●

Conclusions

A good protocol which has improved with time but which did not offer the same flexibility and support as LonWorks at the time the research was commencing

Overall Rating ●●●●

Additional Information

More information can be found at www.eiba.com

BATIBUS

Creator:	Merlin Gerin
Country of Origin:	France
CEN Level	Field

General Description

Batibus is a relatively simple low cost protocol also not relying on dedicated chips. However, its simplicity only allows it to be implemented in small to medium sized buildings

Robustness

OSI 7 Layer Model			
7	Application Layer	Application Code	●
6	Presentation Layer	Common language	
5	Session Layer	Controls Dialogue	
4	Transport Layer	Optimises Network	
3	Network Layer	Data transport	
2	Data Link layer	Message structure	●
1	Physical Layer	Physical media	●

Notes

Simple protocol that only offers levels 1,2 and 7 of ISO Model. Batibus recently formed an alliance with EIB.

Robustness Rating ●●

Flexibility and Interconnectivity

Media		*Topologies*	
●	Twisted Pair	●	Free Topology
	Power Line	●	Point to Point
	Infra Red	●	Bus Topology
	Radio Frequency	●	Tree Topology
	Optical Fibre	●	Star Topology
	Coaxial	●	Ring Topology
	Telephone Line		

Connection structure allows any arrangement of cables making it extremely flexible in matching building requirements. However, the protocol is limited by the number of devices it can accommodate.

Flexibility Rating ●●●

Interoperability

Field Interoperability	●●●●
Automation Interoperability	●●
Management Interoperability	●
Interoperability Association	Batibus Club

The Batibus protocol standard was originally proposed by Merlin Gerin and is now open to 'club' membership.

Interoperability Rating ●●●

Manageability

Management tools are available from a variety of manufacturers, but it is not apparent whether these are interoperable.

Manageability Rating ●●●

Suitability

The protocol seemed to cater well for small field networks, but automated blinds seemed to predominately used in large high profile offices.

Suitability Rating ●●

Availability

It is predominately used in France and not many compatible products were available in the UK.

Availability Rating ●

Approachability

Not a great deal of information on the internet unless you are a member of the Batibus Club. Most of the structure is based in Europe and not much support in UK.

Approachability Rating ●

Conclusions

The protocol was lacked the flexibility required for this research

Overall Rating ●●

Additional Information

More information can be found at www.batibus.com and www.invirtou.com/batibus.

PROFIBUS

Creator:	Siemens
Country of Origin:	Germany
CEN Level	Automation

General Description

PROcess FIeld BUS (PROFIBUS) resulted from collaboration between German industry and technical institutions also spearheaded by Siemens. It defines the technical characteristics of a serial bus system with which distributed digital programming controllers can be networked.

Robustness

OSI 7 Layer Model			
7	Application Layer	Application Code	•
6	Presentation Layer	Common language	
5	Session Layer	Controls Dialogue	
4	Transport Layer	Optimises Network	
3	Network Layer	Data transport	
2	Data Link layer	Message structure	•
1	Physical Layer	Physical media	•

Notes

It is a multiple master, multiple slave distributed network system which operates on a cyclic token system.

Robustness Rating	•••

Flexibility and Interconnectivity

Media		*Topologies*	
•	Twisted Pair		Free Topology
	Power Line	•	Point to Point
	Infra Red	•	Bus Topology
	Radio Frequency	•	Tree Topology
	Optical Fibre		Star Topology
	Coaxial	•	Ring Topology
	Telephone Line		

A master can send messages without an external request, when it holds the bus access rights (token). A slave does not have bus access rights and can only acknowledge received signals or send messages to the master when requested to do so. However, systems are limited to 32 active devices.

Flexibility Rating	••

Interoperability

Field Interoperability	•
Automation Interoperability	••••
Management Interoperability	••
Interoperability Association	PI

The user organisation has over 100 members across Europe.

Interoperability Rating	•••

Manageability

Management tools are available from a variety of manufacturers, but it is not apparent whether these are interoperable.

Manageability Rating	•••

Suitability

It is not particularly well suited to flat field level control, but provides a good means of communicating between IO modules and programmable controllers

Suitability Rating	••

Availability

It was primarily developed for manufacturing and process automation and was established as the DIN standard in 1991 (DIN 19 245). Not expected to be implemented a great deal outside Germany.

Availability Rating	•

Approachability

Not a great deal of information on the internet unless you are a member of the Profibus Club. Most of the structure is based in Europe and not much support in UK.

Approachability Rating	•

Conclusions

Better suited to automation level management across Ethernet than field level management.

Overall Rating	••

Additional Information

More information can be found at www.profibus.com.

WORLDFIP

Creator:	Honeywell
Country of Origin:	France
CEN Level	Field

General Description

The Factory Instrumentation Protocol (FIP) is the French national standard and has been expanded to WORLDFIP with support from Honeywell.

Robustness

OSI 7 Layer Model			
7	Application Layer	Application Code	•
6	Presentation Layer	Common language	
5	Session Layer	Controls Dialogue	
4	Transport Layer	Optimises Network	
3	Network Layer	Data transport	
2	Data Link layer	Message structure	•
1	Physical Layer	Physical media	•

Notes

The field network is a low-level network which employs a centralised token management structure and provides layers 1,2 and 7 of the OSI Model.

Robustness Rating ••

Flexibility and Interconnectivity

Media		*Topologies*	
•	Twisted Pair		Free Topology
	Power Line	•	Point to Point
	Infra Red	•	Bus Topology
	Radio Frequency		Tree Topology
	Optical Fibre		Star Topology
	Coaxial		Ring Topology
	Telephone Line		

WorldFIP has a rich family of supporting chips which uses twisted pair wiring at between 31kBits/s and 2.5Mbits/s.

Flexibility Rating ••

Interoperability

Field Interoperability	•
Automation Interoperability	•••
Management Interoperability	•
Interoperability Association	WorldFIP org

Interoperability Rating ••

Manageability

Low-level management tools are available from the WorldFIP Organisation.

Manageability Rating ••

Suitability

This French standard was mainly designed for applications in the production and process industries and does not support any HVAC or building specific functions.

Suitability Rating •

Availability

No building products available.

Availability Rating •

Approachability

Very little information available on the internet.

Approachability Rating •

Conclusions

Not suitable for building applications at this stage and unlikely to catch the others in the future.

Overall Rating •

Additional Information

More information can be found at www.worldfip.org.

FND	
Creator:	German Government
Country of Origin:	Germany
CEN Level	Management

FND

General Description

Firm Neutral Data Transmission (FND) is a German DIN standard intended for linking BMS central stations.

Robustness

OSI 7 Layer Model			
7	Application Layer	Application Code	•
6	Presentation Layer	Common language	
5	Session Layer	Controls Dialogue	
4	Transport Layer	Optimises Network	
3	Network Layer	Data transport	
2	Data Link layer	Message structure	•
1	Physical Layer	Physical media	•

Notes

The FND protocol specifies in its application layer general data items and functional properties of these data items. General data items include: point objects, AI, AO, DI, DO, set points, multi-state inputs and multi-state outputs.

Robustness Rating ••

Flexibility and Interconnectivity

Media		*Topologies*	
•	Twisted Pair	•	Free Topology
•	Power Line	•	Point to Point
•	Infra Red	•	Bus Topology
•	Radio Frequency	•	Tree Topology
•	Optical Fibre	•	Star Topology
•	Coaxial	•	Ring Topology
•	Telephone Line		

FND only specifies an common interface to the data network which itself may be implemented using any relatively high speed technology.

Flexibility Rating •••

Interoperability

Field Interoperability	•
Automation Interoperability	•
Management Interoperability	•••
Interoperability Association	None

Represents a only management structure

Interoperability Rating ••

Manageability

A good level of network management for proprietary systems.

Manageability Rating •••

Suitability

FND is designed to connect islands of automation (i.e. complete proprietary systems) with supervisory computers.

Suitability Rating •

Availability

The protocol is used primarily in German and Switzerland. Very few manufacturers supply compatible products in this country.

Availability Rating •

Approachability

Very little information on the internet.

Approachability Rating •

Conclusions

Since there is now very little market demand CEN/TC247 has recommended the withdrawal of FND. It is thought there will not be many future applications for this protocol.

Overall Rating •

Additional Information

BACNET	
Creator:	ASHRAE
Country of Origin:	United States
CEN Level	Management and Auto

General Description

ASHRAE's Building Automation and Control Networks (BACnet) standard is essentially a complete BMS protocol defined solely for use within building services.

Robustness

OSI 7 Layer Model			
7	Application Layer	Application Code	●
6	Presentation Layer	Common language	
5	Session Layer	Controls Dialogue	
4	Transport Layer	Optimises Network	
3	Network Layer	Data transport	●
2	Data Link layer	Message structure	●
1	Physical Layer	Physical media	●

Notes

It is now formally defined in the ANSI/ASHRAE standard 135-1995[1]. BACnet is essentially a complete BMS protocol, but it is not just limited to HVAC Applications. Because it a standard way of representing the functions of any device the protocol can also be applied to other building services.

Robustness Rating ●●●●

Flexibility and Interconnectivity

	Media		*Topologies*
●	Twisted Pair	●	Free Topology
●	Power Line	●	Point to Point
●	Infra Red	●	Bus Topology
●	Radio Frequency	●	Tree Topology
●	Optical Fibre	●	Star Topology
●	Coaxial	●	Ring Topology
●	Telephone Line		

A high level protocol that can use a number of different transmission protocols at the field bus level, such as LonWorks, Ethernet, Arcnet, RS 485 and RS 232.

Flexibility Rating ●●●●●

Interoperability

Field Interoperability	●●●●●
Automation Interoperability	●●●●●
Management Interoperability	●●●●●
Interoperability Association	

Its object-orientated structure makes it particularly suitable for management level communications (i.e. transmitting data to and from operator interfaces).

Interoperability Rating ●●●●●

Manageability

A very versatile management structure (ie transmitting data toand from operator interfaces) since its object-oriented structure means there is far less configuration involved when a system is set up.

Manageability Rating ●●●●●

Suitability

Designed specifically for Building Management Systems this protocol has all the features necessary for successful building integration and automation.

Suitability Rating ●●●●●

Availability

At the beginning of this work, BACnet was still reasonably new and had not really established itself in the market place.

Availability Rating ●●

Approachability

Not a great deal of information was available when this project began. But this has got better as BACnet has turned its attention to the European market.

Approachability Rating ●●●●●

Conclusions

This protocol takes many of the advantages of LonWorks and adds to them on the Management level to the extent that it allows users to demand openness from their existing proprietary legacy systems.

Overall Rating ●●●●

Additional Information

More information can be found at www.bacnet.org

The Slow Emergence of Market Leaders

We can see from the review outline in the tables above that when considering building automation some of the protocols are stronger than other protocols. The simple sensor actuator bus systems, such as Batibus and Profibus were deemed to be unable to provide the functionality required for this study as the emphasis was placed on the protocol that offered the right communications objects to exchange messages, and connect nodes.

World FIP and Profibus are more suited to the process and manufacturing industries for which they were created and therefore their take up is slow. FND was purely developed to link between proprietary systems and therefore is not really in the same market. In fact CEN is about to recommend the withdrawal of FND from its standard.

In terms of standardisation these facts seem to leave BACnet as the clear market leader for the high and medium level protocol. However, some believe the story will not end here. With the advent of the Internet age, Ethernet is figuring more and more in systems integration. For example, building management systems increasingly make use of a building's Ethernet IT network. The latter provides the BMS with a ready-made communications "backbone" which is both very flexible and offers high data transmission speeds.

Links are now being developed between building control systems and internet protocol based IT networks. New products that take data off the controls network and place it on the IT network are now becoming common and this may in fact mean that protocols such as TCP/IP, whose networks currently cover the planet, will form the management level protocol for use in buildings.

At the low level field control market, the news that Batibus, EIB and another smaller plug and play protocol EHSA have merged to form Konnex in an attempt to fight off the challenge of LonWorks means that the fight is now between only two contenders, the Konnex and LonWorks.

LonWorks – The De Facto Standard

At the time this study began, EIB seemed to be lagging behind LonWorks in terms of both technology and marketing. This issue has since been redressed and there now is little between the two systems but at the time it was decided that as LonWorks offered more levels of functionality than the other systems and demonstrated all of the required attributes needed for use in this study.

A LonWorks network is capable of control on a number of levels. At a low level there is enough control capabilities available on a Neuron to deal with low level IO, dealing with sensors, actuators and running control loops. At a middle level the network can be made into an integrated system that co-ordinates low level nodes to give good system performance. At a high level, network management and front-end co-ordination can be achieved through comprehensive Application Programming Interface (API) facilities. Appendix VI provides a more detailed description of LonWorks.

Despite the lack of international agreement on a standard communications protocol and the number of the protocols on the market, most observers now regard LonWorks as the *de facto* standard for the UK and Scandinavia.[2] [3] [4] [5] .

Closure

One of the more popular arguments advanced against the flat control system architecture by those holders of the proprietary banner is that a higher-speed backbone is needed to transfer data. Most of this thought process comes from trying to design control systems using the old paradigm: gather all the information in the big black box and transfer it en masse upon request. Properly designed, few control systems require throughput greater than 1 megabit per second, which field bus technology readily accommodates. A good network control protocol sends short concise messages and it only sends them when they are needed. The messages are only seen within the control device community in which they are required.

If all makes and types of building services system used the same communications protocol there should of course be no need for gateways. Yet, despite years of trying, progress towards the deceptively simple goal of imposing a common standard has been painfully slow, and it now seems extremely unlikely that a single open standard will prevail. A reasonable consensus has been achieved that to satisfy technical and commercial needs at all levels it will be necessary to provide a high level bus (backbone) and a low-level bus (field bus). However, when faced with the prospect of writing a specification and implementing a solution, reality sets in - which standard do we support: BACnet or LONMARK?

Pushing the technology down to the lowest level, where the activity takes place is the best way to make information available. Such a unified architecture can significantly reduce the life-cycle cost of the system, and can enable new functionality by taking advantage of IT technologies such as the Web and Internet. Such functionality is a step towards the goal set out by DEGW in Appendix I of the seamless integration computer and controls technologies to form the computer-integrated building.

---

[1] BACnet – A data communication protocol for building automation and control networks, ANSI/ASHRAE 135-1995, American Society of Heating, Refrigeration and Air Conditioning Engineers, Atlanta, GA.

[2] Wilmhurst P. and Hamblin J., 1999, "Openness – A customer driven evolution" Proceedings of the CIBSE National Conference 1999, Harrogate, pp128-135.

[3] Aldridge A., 1993, "Trends in control system development", Proceedings of the Conference of 'Intelligent buildings today and in the future', UCE, 7th October 1993, pp21-29.

[4] Faithfull M., 1997, "Can we talk?", *CIBSE Journal Building Services Supplement on BMS controls*, November 1997, pp10-11.

[5] Hobson S., 1998, "Who needs integrated building controls?", *M&E Design*, April 1998, pp30-32.

Appendix VI An Introduction To LonWorks

The LonTalk Protocol

LonTalk

The core of the LonWorks system is its communications protocol LonTalk, which is much like Ethernet's TCP/IP protocol but optimised for small data packets (messages). The LonTalk protocol implements all seven layers of the OSI model outlined in the previous appendix, and does so by using a mixture of hardware and firmware on a silicon chip, thus eliminating any possibility of modification. It is a 1-Persistant Carrier Sense Multiple Access (1-Persistent CSMA) protocol modified for peer to peer networks.

The LonTalk protocol is designed as a collection of services that can be optionally invoked. They may be chosen by the programmer and fixed at compile time or changed by an installer during commissioning and maintenance. The services are often broken down into the following:

- *Physical channel management* (layers 1 and 2)
- *Naming, addressing and routing* (layers 3 and 6)
- *Communication services* (layers 2, 4 and 5)
- *Prioritising messages* (layer 2)
- *Network management* (layer 5)
- *Network interface* (layer 5)
- *Data interpretation* (layer 6)
- *Application compatibility* (layer 7)

Table VI - I outlines how these services and their individual components relate to the 7 layers of the ISO Model.

	OSI Layer	Purpose	Services Provided
7	Application	Application Compatibility	• Standard network variables
6	Presentation	Data Interpretation	• Network variable • Foreign frame transmissions
5	Session	Remote Actions	• Request-Response • Sender authentication • Network management • Network interface
4	Transport	End to End Reliability	• Acknowledged & Unacknowledged • Unicast & Multicast authentication • Common ordering • Duplicate message detection • Automatic retries
3	Network	Destination Addressing	• Addressing routers • Unicast/multicast/broadcast addressing
2	Link	Media Access and Framing	• Framing • Data encoding • CRC error checking • Predictive CSMA • Priority transmission • Collision avoidance • Optimal priority and collision detection through mixed data rates
1	Physical	Electrical Interconnections	• Media-specific interfaces and modulation schemes (twisted pair, power line, radio frequency etc.)

Table VI - I: The LonTalk Services provided at each layer of the OSI/ISO Model

The Physical Channel

A channel is a physical transport medium for packets and can contain up to 32,385 nodes maximum. The LonTalk protocol is able to make use of a variety of transceiver types thus allowing LonWorks devices to support a wide range of media, including twisted pair, power line, radio frequency, infrared, coaxial cable and fibre optics.

A network may comprise of many channels and routers can be used to connect multiple channels of multiple media. Channels may be configured for different bit rates to allow trade-offs of distance, throughput and power consumption. Specifications quoting maximum communications distances, bit rates and topologies supported are available for each LonWorks transceiver type (see Table VI-III). [1]

Naming Addressing and Routing

A name, in a LonWorks network, is a unique 48-bit Neuron ID burnt into each individual Neuron at manufacture, and which can not be changed at any time. LonTalk addresses uniquely identify the source node and destination nodes of a LonTalk message (or packet) and may be changed at any time. Routers manage network traffic by learning and using addressing tables.

To simplify routing, the LonTalk protocol defines a hierarchical form of addressing using:

- *Domain* – identifies a subsystem on an open media or in a large installation. There can be 2^{48} domains in a network. A node may simultaneously belong to 2 domains;
- *Subnet* – is a subset of a domain. There can be 255 subnets to a domain. Routing is based on subnets;
- *Node* – uniquely identifies a node within its subset. There can be 127 nodes in a subnet and thus 32,385 nodes in a domain.

This form of addressing allows the protocol to adopt a number of addressing modes:

- *Unicast addressing* (subnet/node) – sends messages to a single node, for example a wall switch sending a message to an individual blind controller;
- *Multicast addressing* (group) – sends a message to a group of nodes, for examplem an external sun sensor send illuminance readings to a series of blind controllers;
- *Broadcast addressing* (subnet/domain) – sends a message to a subnet, or to an entire domain.

Routers can be installed using one of four routing algorithms:

- *Repeater* – is the simplest form of router, simply forwarding all packets between the two channels. Using the repeater, a subnet can exist across multiple channels;
- *Bridge* – simply forwards all the packets that match its domain between the two channels. Using a bridge a subnet can exist across multiple channels;
- *Learning router* – monitors the network traffic and learns the network topology at the domain/subnet level. The learning router then uses its knowledge to selectively route packets between channels. Learning routers cannot learn group topology, so all packets using group addressing are forwarded;
- *Configured router* – selectively routes packets between channels by consulting internal routing tables. Unlike a learning router, a network management tool defines the contents of the internal routing tables.

Configured or learning routers can be used to isolate traffic within a segment to increase total system capacity and improve reliability. For example, an occupancy sensor reading from one zone in an office may not need to be communicated to any devices other than those in that zone. Therefore a router installed at the edge of that zone would ensure that the data packet sent by that sensor did not pass onto the rest of the network, thus freeing the network from excessive amounts of traffic and increasing the systems overall bandwidth.

It is not only the router that has advanced network optimisation functions. The LonTalk protocol is an effective control protocol due to its utilisation of small packets and a random back-off algorithm that monitors channel backlog information. In checking the network for the relevant destination address, a particular node receives and extracts information about the channel backlog information. It uses this information to build a model of the likelihood of being able to reply to its own messages and expands the range over which it randomises a time slot selection in case of collisions. As all the nodes on the network are doing this the likelihood of collision is reduced, enabling more linear network usage at a higher bandwidth. The mechanism for reducing this range in periods of light traffic is based on packet elapse time. Indicating the average time a packet occupies the network. The default time is based on the average LonTalk packet of 10-14 bytes. For each elapsed packet cycle the backlog number is reduced by one, reducing the backlog range. This scheme allows the network to adapt to changing network usage preventing multiple collisions when network traffic is heavy and allowing faster responses when the network traffic is light. All of these features help make LonWorks a robust and near-future proof technology.

Communication Services

The communication services are utilised to provide a trade off between reliable communications, better response times and efficient use of channel bandwidth. The LonTalk protocol offers four types of message service options:

- *Unacknowledged* – fastest possible service and the most commonly used on large networks;
- *Repeated* – also an unacknowledged service but reduces the probability of a collision without sacrificing bandwidth;
- *Acknowledged* – end to end messages that re-try after timeout period if messages are lost due to noise or congestion. Message can be sent to individual nodes, or groups of nodes;
- *Request/Response* – application to application message where the acknowledgement also carries data. Message can be sent to individual nodes, or groups of nodes. Duplicate message detection and reminders are fully implemented.
- *Authenticated* - allows the receiver of a message to determine whether the sender is authorised to send that message. When an authenticated message is sent, the receiver challenges the sender to provide authentication, using a different random challenge (8bytes) every time. The sender then responds with a transformation performed on the challenge, using an authentication key. The receiver compares the reply to the challenge with its own transformation on the challenge. If the transformations match, the transaction goes forward. The transformation used is designed so that it is extremely difficult to deduce what the key is, even if the challenge and response are both known. This service has many application in security systems.

In order, from most reliable to least reliable, the request/response and acknowledged are equally reliable followed by repeated and then unacknowledged. However the most reliable message options utilise more of a channel's capacity and on large networks can cause problems.

The LonTalk protocol also provides duplicate message detection, and normally only delivers a message to the destination once. Duplicate packets can occur when using unacknowledged repeated service or if an acknowledgement or response message has been lost. The duplicate detection capability is provided by a received transaction database in each node.

Priority Messages

The performance of the control networks are measured in response time not data throughput, thus a linear time response at higher bandwidth usage might not be good enough. In critical applications when important messages need to get through no matter what the network traffic conditions then priority message slots are specified. Again these have important uses in fire alarm and security systems, as well as occupant over-rides and such features add to the versitility of the protocol for a number of applications.

Network Management

The LonTalk protocol provides network management services for installing and configuring nodes, downloading software and diagnosing the network. There are many functions provided by the network management services. Two of them, the Modify Address Tables and Modify Net Variables messages may be used to dynamically connect network variables and message tags. This process is termed binding and is used during installation and reconfiguration to establish the addressing information needed to route messages and network variable updates between nodes.

Network Interface

The LonTalk Protocol includes an optional network interface protocol that can be used to support LonWorks applications running on any host processor. A host processor may be any micro-controller, microprocessor or computer. The host processor manages layers 6 and 7 of the LonTalk protocol and uses the LonWorks network interface to manage layers 1 to 5. The LonTalk network interface protocol defines the format of packets exchanged between the network interface and the host.

A host application running on the host processor communicates with the network interface through a network driver. The network driver manages the buffer allocation, buffer transfers to and from the network interface and isolate the host application from any differences in the network interface link layer protocol.

Data Interpretation

The LonTalk protocol employs a data orientated application approach. In this approach, application data items such as temperatures, pressures, states, text strings, etc are exchanged between nodes in standard engineering and other predefined units. Commands are encapsulated within the application programs of the receiver nodes rather than being sent over the network. In this way, the same engineering value can be sent to multiple nodes, which have a different application program for that data item.

The data items in the LonTalk protocol are called network variables. Network variables can be any single data item or a data structure. Each network variable has a data type declared by the application program, in a manner much like local C variables except that the 'network' keyword is used to make the variable available to any other node on the network. When output network variables change via assignment operations within the application program, the Neuron Chip firmware automatically propagates the new value over the network as LonTalk messages while the other LonTalk protocol services within the Neuron chip firmware automatically handle any buffer management, message initialisation and error handling.

Applications that require a different data interpretation model than network variables can send and receive explicit messages. Explicit messages use the messaging services of the LonTalk protocol with the minimum data interpretation. Each explicit message contains a message code that the application can use to determine the type of interpretation to be used on the contents of the message.

LonTalk Packet

All of the LonTalk services discussed are embedded in firmware in each individual node or Neuron on the network (to be discussed in the next section). Each Neuron communicates with other devices on the network through network variables implemented within the small LonTalk packet (message). Also implemented within the packet are other LonTalk services, such as Communications services, priority references etc.

Figure VI - I illustrates the make up of a LonTalk Packet.

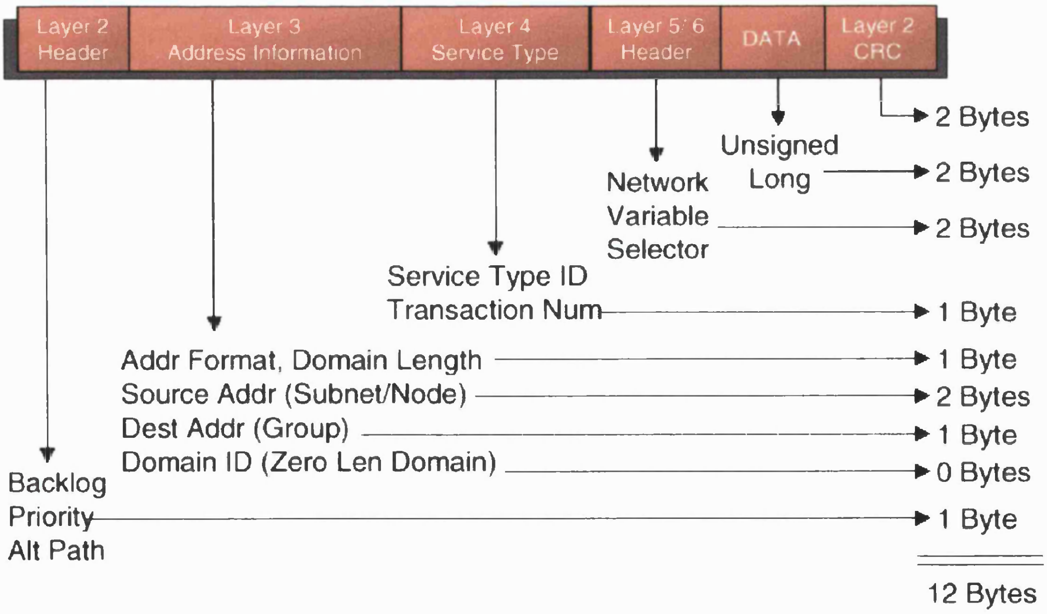


Figure VI - I: Typical LonTalk packet size[2]

The Neuron Hardware and Software

The Anatomy of A LonWorks Neuron

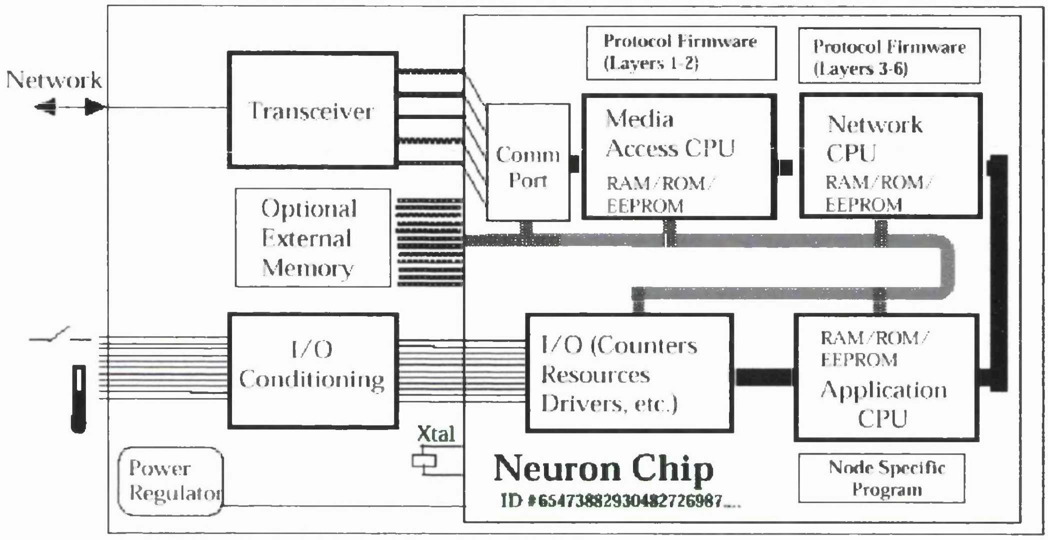


Figure VI - II: A Schematic of a Neuron based node[3]

A basic anatomy of the LonWorks Neuron consists of a Neuron chip, a network transceiver, I/O conditioning, memory, a power supply and some support circuitry such as memory protection devices and a clock source for 16-bit timers/counters

The Neuron Chip

The heart of the Neuron, and thus the LonWorks system, is the Neuron chip. The Neuron chip has been specifically designed by Echelon to run the LonTalk protocol and support LonWorks services with embedded firmware. The Chip is manufactured by, Cypress Semi-conductors, Toshiba, and until recently Motorola, in two basic forms: the 3120 (32 pins) and the 3150 (64 pins). A unique non-changeable 48-bit Neuron ID string is assigned at manufacture to ensure that every Neuron has an individual identity.

The Neuron chip incorporates three pipelined CPUs. Each CPU is an 8-bit processor responsible for different layers of the OSI model:

- *Media Access CPU* – executes LonTalk Protocol Layer 2 (sends and receives messages to and from the network);
- *Network CPU* – executes LonTalk Protocol Layers 3-6 (addressing, reliable message delivery, duplication detection and security);
- *Application CPU* – executes LonTalk Protocol Layer 7 user application written in Neuron C).

Having two processors dedicated to network tasks and one dedicated to application tasks ensures that the complexity of the application does not reduce the network responsiveness and vice versa. This has a number of advantages when considering the implementation of complex adaptive algorithms on a blind by blind basis. The CPUs are connected with memory via an internal 16-bit address bus and an internal 8-bit data bus. Memory options depend upon device.

The 3120 supports applications of up to 2 Kbytes, that is equivalent to about 40 pages of Neuron C code. These chips tend to be used for simple devices such as sensors and un-intelligent actuators. All of the required Read Only Memory (ROM), Random Access Memory (RAM) and Electrically Erasable Programmable Read Only Memory (EEPROM) is included within the chip.

The ROM is the Neuron Chip Firmware, or operating system that runs on the Media Access Control and Network processors. The firmware is written, tested and debugged prior to being burnt onto the chip at manufacture and can not be changed or written to. A number of runtime libraries for the application program running on the application processor can also be included here. The RAM is used for stack space for the three processors, for application variables and communications buffers and is volatile memory. The EEPROM is the memory space for the application code and non-volatile network variables. It can be erased and used over by exposing it to an electric charge.

The 3150 tends to be used for larger applications. It has an external memory interface that allows the flexible configuration of applications with up to 42 Kbytes, equivalent to about 840 pages of Neuron C code. The chip's internal memory only consists of 0.5 Kbytes of EEPROM and 2.0 Kbytes of RAM. But its external memory can be any combination, up to 58 Kbytes of ROM, EEPROM, RAM, Memory-Mapped I/O and Flash Memory (a more flexible erasable memory space than EEPROM). 16 Kbytes of this external memory is reserved for the operating system, this is either stored in ROM or Flash, and the rest is available for the application code. This means that the device control algorithms can be updated as and when improved algorithms are available and adaptive algorithms are able to be accommodated.

The memory allocations for the 3120 family of chips are outlined in Table VI - II:

Device	EEPROM	RAM	ROM
3120	0.5 Kbytes	1.0 Kbytes	10.0 Kbytes
3120E1	1.0 Kbytes	1.0 Kbytes	10.0 Kbytes
3120E2	2.0 Kbytes	2.0 Kbytes	10.0 Kbytes

Table VI - II: Memory allocation for 3120x family [4]

Transceivers

Each Neuron chip has a five-pin communications port that interfaces with a range transceivers for different network media. The transceiver implements layer one of the OSI Model and the LonTalk protocol. A major function of the transceiver is to isolate the device's electronics from harmful energy on the network channel. A list of some of these transceivers and their media is given in Table VI - III below.

Transceiver	Medium	Data Range	Notes
TP/XF-1250	Twisted Pair	1.25 Mbps	Bus Topology
FTT-10A	Twisted Pair	78 kbps	Free Topology
LPT-10	Twisted Pair	78 kbps	Free Topology
PLT-21	Power-Line	5 kbps	World-wide use
PLT-30	Power-Line	2 kbps	European city utility networks
PLT-10A	Power-Line	10 kbps	USA and Japan
RF-100	Radio Frequency	4.883 kbps	

Table VI - III: Some examples of transceivers and their characteristics

IO Circuitry

The IO circuitry is the hardware component that connects the Neuron chip in the device to the element that is sensing or actuating within the device. The Neuron chip has eleven intelligent IO pins rather than a lot of dumb parallel I/O lines. To support these I/O lines, on-chip firmware I/O drivers allow a quick and efficient interface between the Neuron and the application I/O. Specific objects are generated for I/O applications and are supported by two 16-bit timer counters on the Neuron.

There are twenty-five types of I/O firmware available, including bit, nibble, byte, level detect, timer/counter frequency, serial and parallel. An exhaustive list of firmware functions is given in the Neuron C programmers Guide.[5] The use of the low-level embedded firmware I/O features within the LonWorks Neuron allows a certain amount of parallel processing, without which its functionality as a network processor would be very limited.

Neuron C

Neuron C is a programming language designed specifically for the Neuron chip. It is an event driven version of ANSI C utilising certain C libraries, but also including a few extensions introduced to make it more suited to control applications:

- A new statement, the ‘when’ statement, to introduce events and define task execution order;
- 37 additional data types, 35 I/O objects and 2 timer objects, to simplify and standardise device controller usage; [6]
- Integral message-passing mechanisms for both explicit (physical, logical and destination-name addressing) and implicit (network variable) message formats to simplify data sharing across the network.

Device based Neuron C development was not attempted in this study as it required the use of expensive equipment that was not available to the author.

Scheduler

Each Neuron chip within a network has its own scheduler, timers and I/O objects. This allows the application programmer to define tasks that execute as a result of certain events. These events can be pre-defines within the firmware such as reset event or can originate from internal timers or external I/O objects. All these events are defined through ‘when’ clauses within the program body. The program designer can also specify certain priority tasks to be preferentially executed, such as a user override in an automated blind control system. Figure VI - III illustrates this concept.

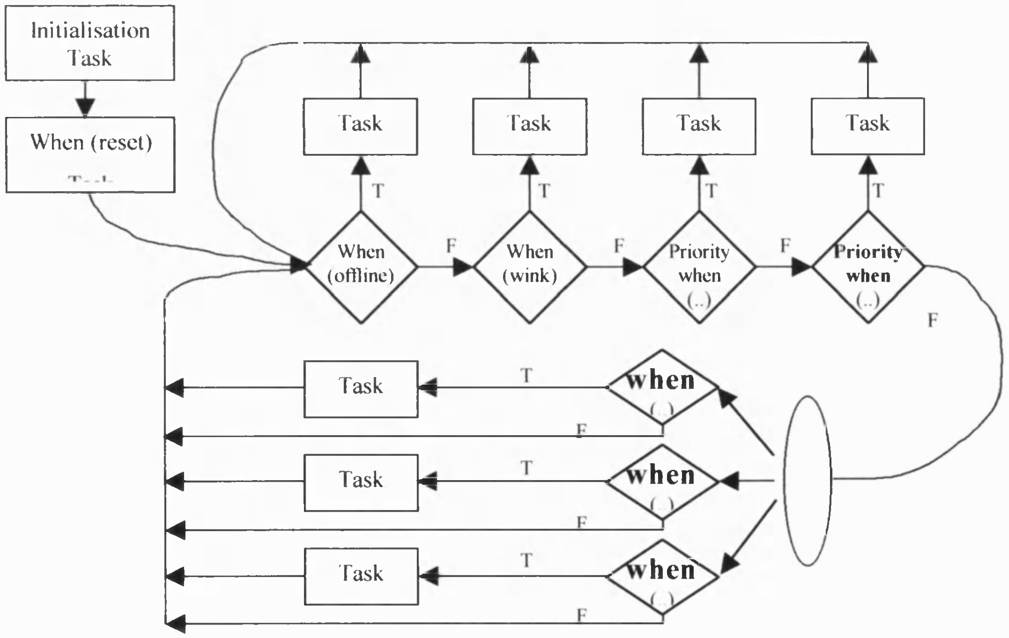


Figure VI – III: Scheduler operation[2]

Interoperability with LonWorks

Application Compatibility

Application compatibility is facilitated through the use of Standard Network Variable Types (SNVTs). A comprehensive list of nearly 100 SNVTs, that covers a wide range of applications, can be found in the SNVT Master List and Programmer's Guide.[7] The definition of each SNVT includes its units, its range and its resolution; some examples are given in the table below. Using the appropriate network management tool can allow a LonWorks node to extract SNVT information (ID# and optional text string) from any other node. Custom structures specific for individual manufacturers' product features can also be defined.

SNVT	Measurement	Units	Range	Resolution
SNVT_temp	Temperature	Degrees Celsius	-274 – 6271	0.1 degree
SNVT_angle	Phase/Rotation	Radians	0 - 65	0.001 rads
SNVT_speed	Speed	Meter/Sec	0 - 6553	0.1 m/s
SNVT_elapsed_tm	Elapsed Time	HH:MM:SS	0 – 65535 days	1 msec
SNVT_lev_count	Continuous Level	Percent	0 – 100%	0.5%
SNVT_ascii	ASCII String	Characters	30 Chars	N/A
SNVT_count	Events	Count	0 – 65535	1 Count

Table VI - IV: Examples of some Standard Network Variable Types

In addition to SNVTs, which can be exchanged from device to device, a Lonworks device can have Standard Configuration Properties Types (SCPTs). SCPTs are used to customise and optimise the performance of a particular node's application program, by utilising parameters such as range, resolution, maximum send time and operating mode.

The SNVTs and SCPTs are often grouped into a series of functional objects, with the nature of those objects depending on the application. See Chapter Six and Appendix VII for a description of the functional blocks of the blind controller used in the test room.

LonMark Interoperability Association

The LonMark Interoperability Association is an open global organisation dedicated to promoting and facilitating interoperability between LonWorks-based products. It has over 200 members who together develop LonMark standards and guidelines for producing interoperable products.

The distinction between LonTalk and LonMark is very subtle. LonMark subsumes the LonTalk protocol and as a whole provides a set of standards that allow independently designed systems to be integrated without the need to develop custom application code. The LonMark approach creates an application layer interoperability interface by the generic description of a series of objects:

- *Node Object* – provides the mechanism for requesting object modes and for reporting status of objects within the node. In addition, the node object includes network variables and configuration properties related to the node as a whole, such as for network management support;
- *Open Loop Sensor Object* – suitable for use with sensing devices that report absolute rather than relative values and for use with devices that do not require feedback information for correct operation, one example of this type of object is the sun sensor used in the test room;
- *Closed Loop Sensor Object* – contains a feedback feature that enables multiple sensors to control a common actuator and a single sensor to control multiple actuators, while retaining synchronisation between the actual and desired states of objects in both sensors and actuators. An example would be when multiple remote dimmer light controls are located round a space to control the same lighting load;
- *Open Loop Actuator Object* – suitable for use in applications where the actuator provides no feedback information;
- *Closed Loop Actuator Object* – contains a feedback feature that enables multiple actuators to be arbitrarily combined with multiple sensors. The feedback feature allows synchronisation between the actual and desired states of the objects in multiple sensors and actuators;

- *Controller Object* – allows control algorithms to be introduced between data producing objects such as sensor objects and data consuming objects such as actuator objects.

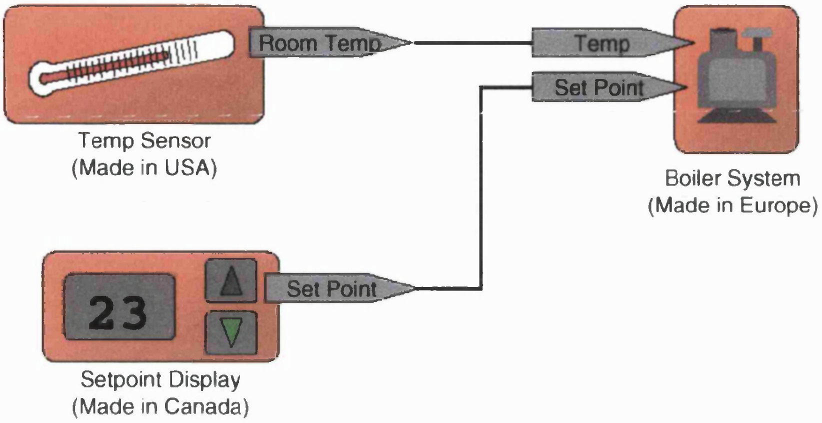


Figure VI - IV: An example of two sensor objects and an actuator object

Sets of interoperability guidelines and standard functional profiles are available from the LonMark Interoperability Association. [8] Devices that conform to the standards set out by the organisation get their products certified with the LonMark Logo.[9]

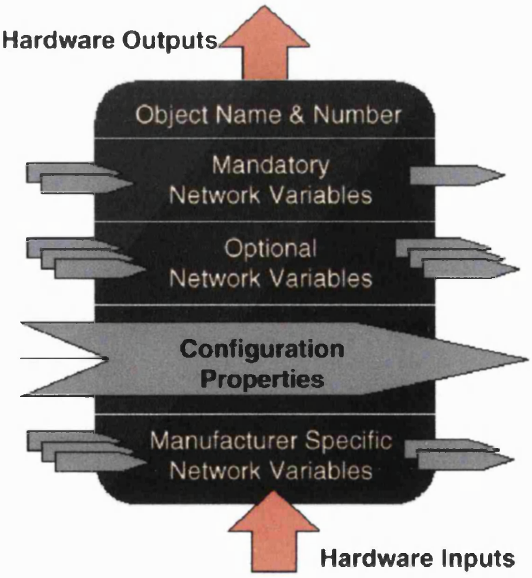


Figure VI - V: A LonMark Functional Profile and The LonMark Logo

Developing LonWorks Devices

This Appendix has demonstrated that all communications services and protocols required to develop LonWorks devices are available on the Neuron chip as a mixture of hardware and firmware. The chip also incorporates features such as watchdog timers, on-board diagnostics, 35 device controller types a distributed real time operating system, run-time libraries, three types of memory and the unique ID number, that make the protocol more robust. The transceivers are also readily available. Therefore all the developer needs to provide to create a LonWorks compatible product is the application code and the I/O devices which are already within their field of expertise. The only new skills that need to be mastered are programming in Neuron C using the development tools available and ensuring devices conform to the interoperability standards.

Development tools are available from a number of companies. Examples include NodeBuilder, which incorporates a compiler, linker, loader and debugger, and LonBuilder, which incorporates a protocol analyser on top of the other devices.

The process of LonWorks product development cab be summarised into the following steps [10]:

- Assess the application, the overall functionality required, the installation method and network resources available;
- Identify devices , sensors, actuators and controllers, and assign functions;
- Define device external interface in terms of the information shared with other devices, network variables, LonMark objects and configuration properties;
- Write device application programs in Neuron C using the LonBuilder or NodeBuilder developers kits;
- Build, debug and test devices using the LonBuilder or NodeBuilder developers kits;
- Integrate devices into a network and test using a network management tool such as LonMaker or a development tool such as LonBuilder.

LONWORKS NETWORK SERVICES (LNS)

The LonWorks Network Services Architecture

The LonWorks Network Services (LNS) architecture provides the foundation for interoperable LonWorks installation, maintenance, monitoring and control tools. Just as the LonTalk protocol provides a comprehensive set of communication services that is the foundation for interoperability between application nodes, the LNS architecture provides a comprehensive set of network services that is the foundation for interoperability between installation and maintenance software tools. Each tool draws upon the services of the LNS architecture to complete its tasks.

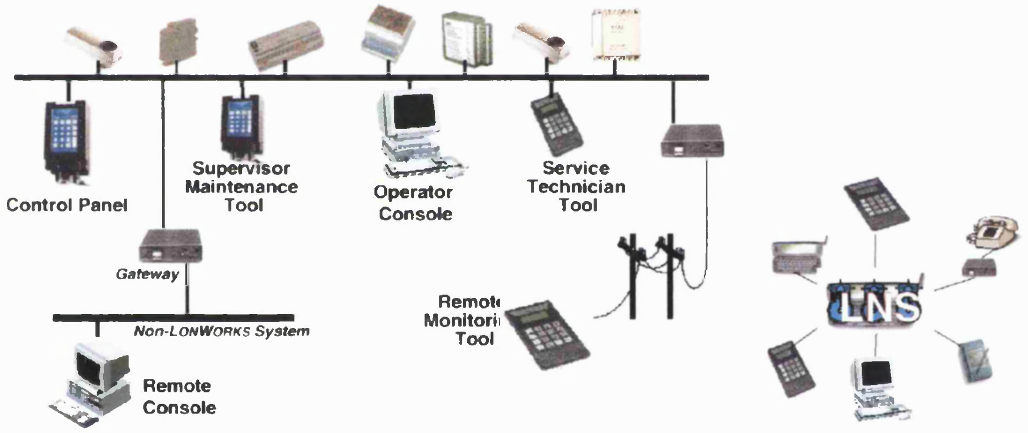


Figure VI - VI: Diagram showing the relationship between monitoring and control devices on the network and the network services server

The LNS architecture allows users to reconfigure the network from any user interface and ensures that all user interfaces used for monitoring and controlling nodes are always up to date with respect to the network's configuration. This is similar to the way a multi-tasking operating system provides a set of services that allow multiple programs shared access to the resources of a processor, the LNS architecture provides tools with shared access to the resources of a LonWorks network.

LNS objects are accessed using the client-server architecture. A client sens a network service request to the server, which performs the service and notifies the client when it is complete. This makes the system extremely flexible because it allows a number of different sets of user profiles to be set up to provide different levels of access to the network, enabling different contractors to be working on different elements of the system at the same time.

A single network services server provides standard network services, but individual application servers can be provided for application specific services. For example, a building management system may have separate application servers for the HVAC system, lighting system and security system.

LNS Components

Physically, LNS network services are provided using two major components which work together to make remote access to the server and its data, transparent to the client:

- *Network Services Server (NSS)* – processes standard network services, maintains the network database, and enables and co-ordinates multiple points of access to its services and data. The host of the NSS can be any microcontroller, microprocessor or PC running any operating system.
- *Network Services Interface (NSI)* – provides the physical connection to the network for each client, manages transactions with the NSS and application servers to give transparent remote access. They can be added and removed from the network as needed. The client can be any microcontroller, microprocessor or PC running any operating system.

The LNS Server supports both LonTalk and TCP/IP protocols at the transport layer meaning that the control system can be seamlessly integrated to the Ethernet or Internet, which often used as a management level transportation layer.

LNS Network Services

The LNS architecture provides network services as collections of objects, each with methods, properties and events:

- *Methods* – used to invoke services on objects. A service is an operation that can be invoked by a client and executed in the server. Methods available include:
    - Installing, commissioning, removing and replacing nodes
    - Connecting and disconnecting network variables and message tags
    - Loading new application code into nodes
    - Access to the LonMark objects and configuration properties in a node
    - Importing node self documentation and self-identification information
    - Copying configuration parameter values from one node to another
    - Querying and setting node properties, such as locations, priority slots, self documentation and network variable attributes;
    - Resetting, winking and testing nodes
- *Properties* – the NSS makes all information related to the network's contents and configuration available to clients through properties. A property is an attribute of a managed object, such as the location of a node. For example, a property can be used to get or set the state of a device;
- *Events* – used to inform applications of significant occurrences such as the receipt of a service pin message. The NSS provides the following events:
    - Service pin message arrived
    - Node network image update complete
    - New, unconfigured node attached to the network
    - Previously installed node removed from the network
    - Node or network variable configuration change

Clients can use events to automate installation or maintenance. For example in a self-contained network, discovery events eliminate the need for the user to press service pins to identify nodes. Events also play a central role in synchronising monitoring and controlling applications during network configuration changes.

The LNS Host API

The LNS architecture is host-independent, supporting clients on any platform, including embedded microcontrollers, Windows PCs and UNIX workstations. LNS servers are based on Windows 95, Windows NT or the Neuron Chip firmware. LNS applications interact with the LNS infrastructure through the LNS host API. The same host API is used by all clients, no matter what the host processor or operating system is. The API also presents a common interface for local and remote access to LNS services.

LonWorks Component Architecture

End users want tools that are tailored to their application. Tools must provide an application view of the system, hiding the network when the user is uninterested in the network details. They must also be tailored for specific devices in the system and the terminology must match end user terminology.

The LonWorks Component Architecture (LCA) is a software architecture that provides co-operation between LNS software components and the host Windows components. LCA represents LNS objects as Microsoft OLE and ActiveX objects, decreasing developing time by allowing the use of the support for these controls built into Windows development tools.

OLE controls are high performance, 32-bit, language independent programmable objects that can be used with a variety of development tools, including rapid application development tools such as Visual Basic, as well as full featured programming environments such as Visual C++. This functionality allowed the author to use Visual Basic as a programming tool in the Test Room study and thus eliminating the need to invest in expensive Neuron C development tools for these initial investigations.

LNS Network Management Tools

A network management tool is any tool that installs, diagnoses or maintains devices on a LonWorks network, or that monitors and controls a LonWorks system. The interconnection of network variables is done in a binding process by a network management tool at installation. This information is then stored in EEPROM in each Neuron chip to allow complete recovery in case of a power failure.

Examples of LNS Network management tools include:

- *LonMaker for Windows* – by Echelon, used in this research and described in more detail in Chapter Five.
- *Unilon* – by Phillips use to configure Phillips Helio lighting nodes and those from other manufacturers.
- *Alex* - by Mentzel & Krutmann Engineering.[11]
- *NL220* – by Newron System. This tool is divided into 2 versions: An installation version and a maintenance version. You only need to purchase the installation tool once and but you must purchase the maintenance tool for your sites.[12]

These LNS tools provide easy to use, plug and play functionality for LonWorks networks. The installer can simply plug in the devices and have then configure themselves with minimal or no user interaction. Failed devices can easily be identified and replaced by a maintenance technician, with replacement devices automatically taking on the application and network configuration of the failed device.

Using the LNS Tool 'LonMaker for Windows' in the Test Room Study

The installation of the control components in the test room study was undertaken using the 'LonMaker for Windows' LNS tool. The study showed that the installation, commissioning and network management procedures, which utilised simple drag and drop functionality and a series of installation wizards, were simple, flexible, fast and effective.

Installing a device simply involved dragging a shape from a stencil (containing shapes representing all the basic components of a LonWorks network) onto a drawing. The LonMaker tool then automatically created a network component represented by that shape and prompts the user for information about that component. For large repetitive installations, creating custom stencils from any number of devices and functional blocks can reduce the time taken on this process.

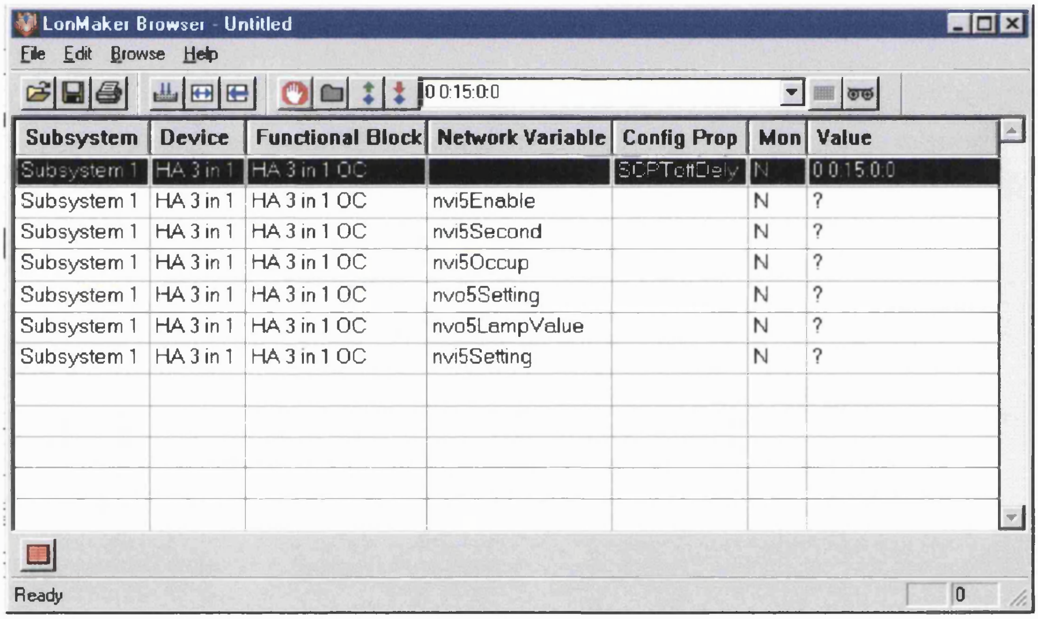
Once the device had been installed in software, the final part of the installation required the installer to identify the physical location of the device on the network. This could be done in one of two ways. The service pin button on the device could be pressed or the Neuron ID of the device could be entered manually (either by typing it in or by bar coding it in). Either method provided a quick and effective way of identifying the device.

Every LonWorks device has an application residing within it that determine how a device functions and contains all of its network variables and configuration properties. During the installation process the option is available to install either the application that is already in the device or to load a new application into the device from some application files (APB and NXE extensions). This functionality gives the installer the option of being able to simply update any application code over the network at any time if the manufacturer issues revisions to the code to deal with known bugs.

Following the installation of the devices, the control system functionality needed to be set up by connecting, or binding, the network variables of these functional blocks together to create one control task. This process was also fairly straightforward and simply involved dragging a connection shape from the stencil until one end sat over the first network variable requiring connection, the unconnected end was then dragged to a second network variable of the same type and the connection was made. Once connected in software the LonMaker tool then instructed the Neuron chips in both devices to talk to one another and the devices became connected in hardware.

In the LonMaker Tool, commissioning and configuration of LonWorks devices can be carried out by using the LonMaker Browser or a device specific LNS ActiveX plug-in. Both methods were used within the test room study.

The LonMaker Browser, shown in Figure VI-IX, had a spreadsheet style interface that listed all the configuration properties and network variables available on each network device. Changes could be made to the variables by simply highlighting a cell and typing in a new value. Unfortunately, in some cases these variables were in Hexadecimal form and unrecognisable to the untrained eye, therefore an experienced commissioning engineer or a detailed device commissioning manual would often be needed.



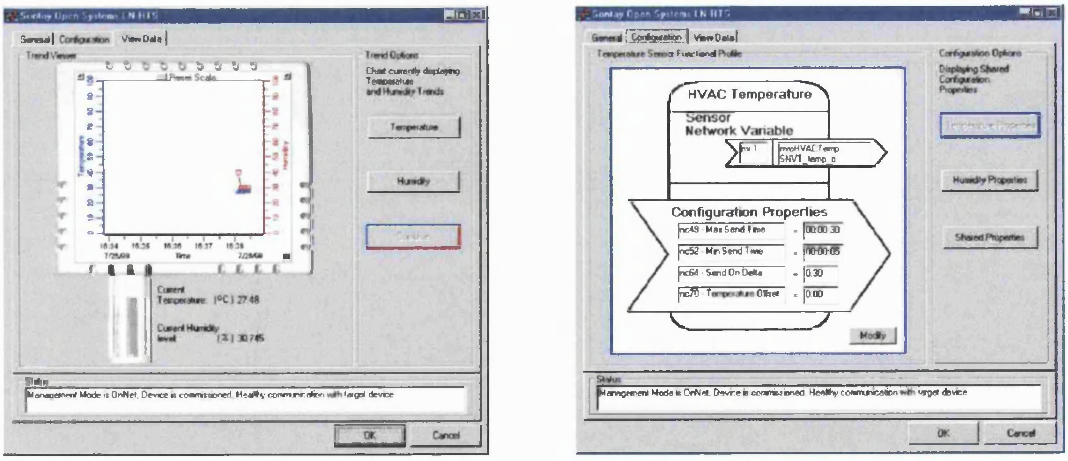
Subsystem	Device	Functional Block	Network Variable	Config Prop	Mon	Value
Subsystem 1	HA 3 in 1	HA 3 in 1 OC		SCPToffDely	N	0 0:15:0:0
Subsystem 1	HA 3 in 1	HA 3 in 1 OC	nvi5Enable		N	?
Subsystem 1	HA 3 in 1	HA 3 in 1 OC	nvi5Second		N	?
Subsystem 1	HA 3 in 1	HA 3 in 1 OC	nvi5Occup		N	?
Subsystem 1	HA 3 in 1	HA 3 in 1 OC	nvo5Setting		N	?
Subsystem 1	HA 3 in 1	HA 3 in 1 OC	nvo5LampValue		N	?
Subsystem 1	HA 3 in 1	HA 3 in 1 OC	nvi5Setting		N	?

Figure VI-IX: The LonMaker Browser

LNS ActiveX Plug-ins on the other hand are created by manufacturers to provide a user-friendly Visual Basic interface for their particular device or product. Examples of plug-ins used in this research included the Sontay Datalogger plug-in, described in the last chapter, and the Sontay LN-HTS Internal Temperature and Humidity Sensor plug-in, shown in Figure 6.2. Both were 'easy to use' configuration interfaces with simple 'click and go' Visual Basic controls that were also supplied with the ability to provide real time trending of device data that could be used to ensure the devices were functioning correctly.

Plug-ins

Plug-ins are a special kind of LNS application, implemented as ActiveX automation servers, that implement the LNS plug-in API to provide add-ons to LNS network management tools. For example, plug-ins allow device manufacturers to provide custom add-ins that simplify configuration, monitoring and control of their devices.[13] Plug-ins can also add new functionality to tools, such as alarming logging and trending.



Current Temperature: (°C) 27.48

Current Humidity level (%) 30.745

Management Mode is OnNet, Device is commissioned, Healthy communication with target device

nv 1 nvoHVACTemp SNVT_temp_p

nc49 - Max Send Time = 00:00:30

nc52 - Min Send Time = 00:00:05

nc64 - Send On Delta = 0.30

nc70 - Temperature Offset = 0.00

Management Mode is OnNet, Device is commissioned. Healthy communication with target device

Figure VI – VIII: The Sontay Open Systems Internal Temperature and Humidity sensor plug-in. Left: Window showing real-time logging; Right: Device configuration properties adjustment window.

If designed correctly plug-ins can make tools easier to use and thus are beneficial to the end user, because they can reduce the cost of training users and reduce the time and cost of installing, configuring and maintaining systems.

LNS and Communication Protocol Standards

The LNS Architecture complements LonWorks, which is primarily a field, level system, by providing simple predefined management and automation layer functionality. However, the LNS architecture has not been recognised by the draft CEN standard, which recommends the use of BACnet, FND and Profibus at the high functionality end. This is because the LNS architecture is not fully open and free.

Each LNS tool has a licensing arrangement built into them that only allows you to install a node once you have paid a $2 license fee for that node. In the case of LonMaker this fee is $5 per node. Many regard this licensing process to be outside the spirit of open systems and as a result comparisons are now being drawn between Echelon, the creator of LonWorks and the receiver of all LNS license fees, and Microsoft.

Impact of LNS on System and Network Tool Design

The advantages of the LNS client-server architecture for both end users and tool developers are as follows:

- Cheaper portable installation tools since the tool can use the NSI the access the services of the NSS.
- Multiple installers can work simultaneously on a single system without database synchronisation problems.
- Different views and user interfaces can exist on the system at the same time. One installer may prefer to look at a problem through their own installation tool, which they are used to.

Non LNS Network Tools

If you do not wish to pay a license to use LNS software tools but wish to use software that enables LonWorks devices to be installed commissioned and managed, there are a number of non-LNS network management tools available. These tools do not carry any license fee per node and once bought can be used freely. Instead of using the LNS Architecture the tools simply use API to communicate with the windows environment.

Examples of API Network Management Tools include:

- *Icelan G* - IEC Intelligent Technologies' Icelan G provides full network binding and management functionality to bind any nodes that use SNVTs.[14]
- *Metravision* - The Metra Corporation's MetraVision provides full network binding and management functionality to bind nodes that use SNVTs. There are different versions of the software with the limits to the maximum no. of nodes to be bound. Metravision also provides graphics capabilities so it can be used as a front-end user interface.[15]

These systems are not as user friendly as LNS systems in terms of multi-user system integration and supporting plug-ins. However, in practice many system integrators and open systems manufacturers choose to use the API systems because LonMaker and LNS have a few well-documented limitations. They fail to allow parts of the building or certain types of equipment e.g. HVAC, fire, access lifts, lighting, etc. to be independently installed, commissioned and proven before integration into one overall system. In so doing, they deny the fundamental contractual splits that exist within the building industry to ensure that clients get proven individual sub systems first.

LonWorks and Field Bus Systems – A State of the Art Review

Current Obstacles to Open System Implementation

Despite the success of setting up the test room, it should be noted that on the whole, system integration is still not easy and that a large number of commercial barriers still inhibit the desired evolution of interoperable control technology.

In reality there is always a certain gap between what is technically possible and what is practically feasible. At present there are a number of obstacles to the wide spread implementation of open field bus systems such as LonWorks. These are:

- *Manufacturers reluctance* - to make their systems totally open and as a result the plug and play functionality is often lost and system integrators are usually needed. Freely configurable outstations are not available while current interoperable products are restricted to pre-configured controllers;
- *Contractual arrangements* – tender documents should define the responsibilities of one contractor with respect to another;[16]
- *Skills shortages* – a lack of skilled installers and specifiers result in high installation costs thus reducing the capital cost benefits.

Manufacturer's Reluctance

At present, commercial pressures are slowing the widespread emergence of fully configurable, interoperable systems. Despite a growing demand, most suppliers are side stepping the issue by putting in gateways to mimic the Lon system within their old proprietary technology. Also a good number of manufacturers have chosen to use LonWorks, not through a commitment to open protocols or interoperability, but because of a desire to find a network technology for their own systems that is fast, flexible, robust and reliable. Indeed they regard LonWorks as a cost-effective way to allow their proprietary devices to share information within their own closed system and wish to leave it at that.

The reason for this lack of commitment can be attributed to the fact that the large control manufacturers are striving to maintain their market share, and seem to prefer to maintain the status quo by keeping their customers boxed in. They fear their market share would be lost if an open protocol allows other suppliers to interoperate directly with their existing or future control equipment and feel their products will become commodities, all offering similar functionality and sold only on price.

This focus is understandable but wrong, as marketing controls equipment shouldn't be about how they communicate, it should be about what they do. However, many manufacturers are not willing to accept the fact, proven over and over in other industries such as information systems, that open systems greatly expand markets, providing plenty of opportunity for many competitors to prosper by delivering new functions and added value to both old and new customers.

Contractual Arrangements

The control package has always been well down in the contractual chain, often leading to a number of on-site problems and the client not getting the system and capability they thought they had bought. Most control systems are rarely commissioned and checked against a performance standard when installed.

Open interoperable systems present a wide new range of contractual problems to both main contractor and the client. For example, the most common type of integration is that between the lighting and the HVAC controls. The lighting tends to be in the electrical contractor's jurisdiction, while HVAC controls are the responsibility of the mechanical contractor. The building system, as a whole, may have been designed with components provided by 50 suppliers, installed by 10 contractors. Because everything is interconnected and there was no individual system proving prior to full binding and commissioning, who is contractually responsible for putting faults right?

On way to tackle this problem is to allow the different building control systems to retain a high degree of independence. By getting the manufacturers or contractors to install their part of the system on their own network and prove its functionality in its own right, it much easier to confirm a system is functioning correctly and diagnose faults. A system integrator or controls contractor then needs to be appointed to take the engineering and contractual responsibility to make the system work whilst creating, managing and organising the system binding database to avoid the sort of contractual problems that would otherwise exist.

This process requires the pre-selection of the controls communications protocol and controls manufacturers at the design stage, and letting the controls contract through competitive tendering, as a works contract alongside the M+E services.

Skills Shortages

The contractual issues related to installing control systems within building have provided the opportunity for a new breed of systems integrator (SI) to evolve. These integrators can either be experts at managing these complex projects or implementing them directly. However, to date there are still very few systems integrators around and because the technology is changing so rapidly, they are often learning on each job. This makes them expensive.

Also if you are not careful about how you word the system specification you can often be as reliant on them as you are a traditional systems house. Each specification should insist that the system data-base used for the installation of the system should be stored on site and not kept by the installer. This way the client is not tied to a particular system integrator and the system remains truly open.

Manufacturer reluctance to make their products fully interoperable does mean that SIs face a big challenge in making open systems work in the UK. One key minefield concerns the definition of supposedly "open" products. The LonMark standard is designed to take care of this problem but in reality does not. Various device applications are defined by the LonMark Interoperability Association and are given a LonMark standard profile.

Problems arise because agreed profiles exist to define only a very small number of the huge variety of plant or items in a given building. Terminal units, such as, fan coils, and VAV units have LonMark profiles defined, while main plant, such as, air handlers, boilers, and chillers do not. At present, this makes it impossible to build a complete system from LonMarked devices.

Even when the LonMark profile does exist, it does not necessarily guarantee interoperability. Some variables, such as setpoint, can be defined by one of a number of variables exclusive to a particular manufacturer's brand of equipment and so inhibiting connection between theoretically compatible open devices from different manufacturers.

Future technologies and products need to be comprehensively tested to determine if they are truly viable and acceptable. Unfortunate evaluation methods are not yet established.

LonWorks and Echelon

Such problems with the LonMark standard and the fact that Echelon, the creator of LonWorks, charge license fees on installed LNS nodes, have led to unfavourable comparison being drawn between Echelon and the computer software giant Microsoft and accusations of LonWorks not being totally open.

This comparison seems slightly unfair as Echelon derives its main revenue from the sale of tools, software, components and services in support of companies wishing to create LonWorks applications. It does not make any money from the sale of chips or protocols and it is possible to install a fully interoperable LonWorks system without purchasing any support or products from Echelon.

Echelon does carry out a fantastic marketing job on LonWorks simple plug and play functionality of LonWorks even though on the practical level there is often no such Utopia. However, if you look behind the Echelon hype and look at the product itself you can see the benefits and the support the company provides to users is substantial. What is more Echelon have achieved what no other company could have achieved and that is to bring manufacturers, specifiers and users together.

Closure

At present, designers are unable to devote substantial resources to determine compatibility of various components, while component-orientated manufacturers lack the market motivation to make the system design transparent to designers and installers because of the lack of volume. As a result, the market is small, products are limited and the risk and costs are currently high.

These non-technological issues are probably largest barriers to the integration of building functions and the development of intelligent buildings. However, they can create business opportunities for people and organisations prepared to work positively with the new technology. As open systems integrators win more projects, the demand for flat architectures will rise, as the demand will be client led because it is to them that the main benefits apply. Thus the real creators of interoperable systems will be the customer, not the controls companies or the committees.

Only a few examples exist of fully integrated buildings in UK, for example, Grampian House and the Bluewater retail development, but the elements are all in place for open systems to develop further. Open control products are available, satisfactory binding tools exist and suitable open system supervisors are also on the market. Equally important, a growing number of SIs exist who are attuned to the finer points of LonWorks control technology.

Although the LonWorks technology may not be all that it was expected to be, a high level of interoperability, that would not be possible with proprietary control systems, is certainly achievable providing the project team can navigate diligently around the various issues. Within the scope of this work, LonWorks and LNS offered the best tool for implementing open field bus systems within the test cell set up. The results of the research itself are not affected by the nature of the management structure, whether it be LNS, BACnet or TCP/IP, and the algorithms developed can be implemented on most major field level protocols.

---

[1] LonMark Layers 1-6 Interoperability Guidelines, Version 3.0, LonMark Interoperability Association, Palo Alto, CA, 1996.

[2] Introduction to LonWorks Device and Network Design, 4-day training Course, Version 4.1, Echelon Corporation, 4015 Miranda Avenue, Palo Alto, California.

[3] Introduction to the LonWorks Platform – An Overview of Principles and Practices, Version 2, Echelon Corporation, 4015 Miranda Avenue, Palo Alto, California.

[4] LonWorks Technology Device Data, Motorola, 1997

[5] Neuron C Programmer's Guide, Document No. 29300, Echelon Corporation, 4015 Miranda Avenue, Palo Alto, California.

[6] Neuron C Reference Guide, Document No. 29350, Echelon Corporation, 4015 Miranda Avenue, Palo Alto, California.

[7] The SNVT Master List and Programmer's Guide, Document No. 005-0027-01, Echelon Corporation, 4015 Miranda Avenue, Palo Alto, California, May 1997.

[8] LonMark Application Layer Interoperability Guidelines, Version 3.0, LonMark Interoperability Association, Palo Alto, CA, 1996. 078-0120-01C.

[9] www.lonmark.org

[10] LonWorks Custom Node Development, LonWorks Engineering Bulletin, January 1995, Echelon Corporation, 4015 Miranda Avenue, Palo Alto, California.

[11] www.mentzel-krutmann.de

[12] www.newron-system.com

[13] Developing LNS Plug-ins using Visual Basic 5, Document No. 39820, , Echelon Corporation, 4015 Miranda Avenue, Palo Alto, California.

[14] www.ieclon.com/ice2info

[15] www.metra.com

[16] Boone W., 1999, "Implementing the latest technology in building automation: A project orientated methodology", Proceedings of "Intelligent and Responsive Buildings" Brugge, 29-30 March 1999, CIB Working Commission WO98, pp223-231.

Appendix VII Details of the Test Room Study

Introduction

This appendix supports Chapters Five, Six and Seven by providing additional information on the devices installed within the test room study. It is divided into two parts. The first part can be found on the CD-ROM and contains a report submitted to the Department of Environment Transport and Regions, by the author, on completion of the CWCT part of the test room study. The report reviews details of the hardware used in the test room and provides general information on how to integrate façade and building services control devices.

The rest of this appendix is devoted to providing the reader with relevant additional information about the devices installed in the test room. It begins by reviewing the functional profile of the blind controller, supplied by the manufacturer, and considers the detail of its operation (as briefly described in Chapter Seven). It goes on to summarise the network variable and configuration property settings for many of the other key devices in order to provide some background on the variables quoted in the Visual Basic code in Appendix IX, and provides a drawing of the functional links.

The CWCT Study Report

This report can be found on the CD-ROM located on the inside of the back cover of this thesis. The report is in an html format and can be open by using a web browser such as Internet Explorer, or Netscape. Simply insert the CD into the CD drive and look at the contents of the CD in the Windows Explorer. Click on 'run.html' file and the CD will then open on the title page of the report. The reader can browse through the report in a linear or non-linear fashion, by using the subject index on the left-hand side of the page.

The Functional Profile of the Blind Controller

A Functional Profile

The functionality of the application code within the Neuron chip within a LonWorks device is defined by a functional profile. A functional profile is made up of a series of network variables and configuration properties.

Network variables allow a device to send and receive data over the network to and from other devices. Configuration properties are properties within a device that are set during installation to determine how the data is manipulated within the device. The application reads the values from the network variables and configuration properties and performs functions upon them. For example, an application may allow an arithmetic function (add, subtract, multiply. or divide) to be performed on two values received from two network variables. The function to be performed could be determined by a configuration property.

This section will complement Chapters Six and Seven by reviewing the functional profile of the blind control application code supplied by the blind manufacturer.[1] By doing this it is hoped that the reader will gain a better understanding of LonWorks, its manageability and the inner workings of current blind control algorithms.

The Blind Controller's Functional Profile

The functional profile for the blind controller is divided into four functional blocks: a node object and four open loop actuator objects (see Figure VII–I).[2] These blocks implement a motor interface for two motors, therefore many of the functional blocks described are duplicated. For example, the input nviSceneCfgMotor [0] configures motor 1 and nviSceneCfgMotor [1] configures motor 2.

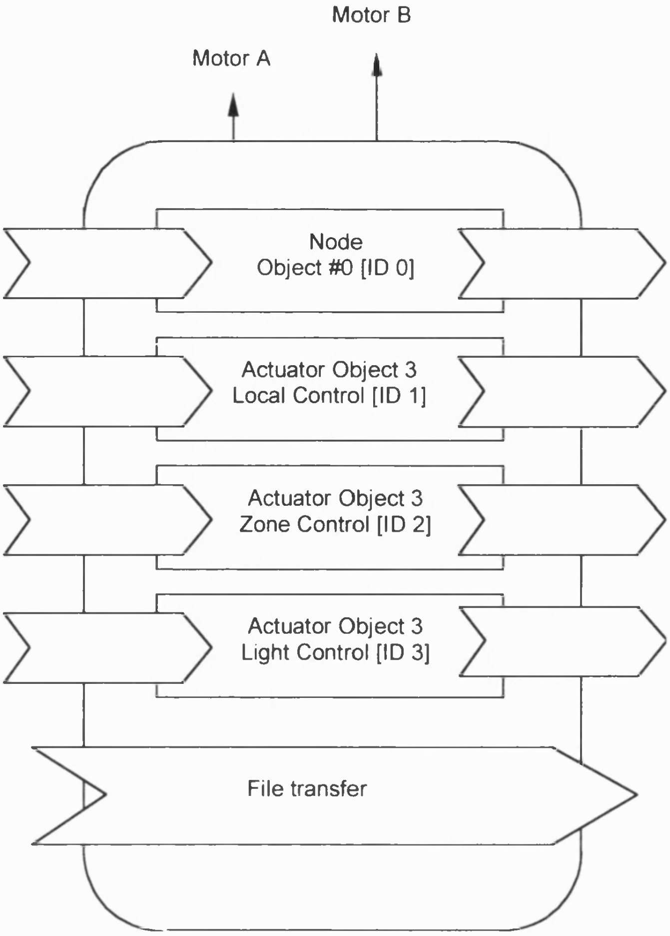


Figure VII-I: The Blind Controller's Functional Profile

The Node Object

Every LonWorks device has a node object. The node object provides the mechanism for requesting and reporting the status of any of the objects within a node. In addition, the node object includes network variables and configuration properties related to the node as a whole, such as network management support. Figure VII-I shows the node object graphical representation for the blind controller and Tables VII-I and VIII-II outline details of the node object's network variables and configuration properties respectively.

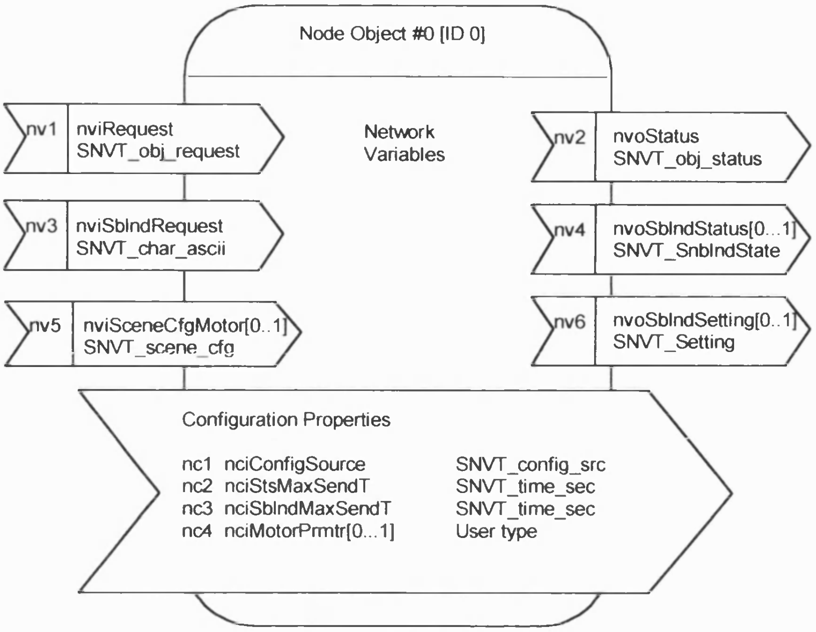


Figure VII-II: A graphical representation of the node object

NV #	Name	In/Out	SNVT Type	Class	Description
1	nviRequest	In	SNVT_obj_request	nv	Object request
2	nvoStatus	Out	SNVT_obj_status	nv	Object status
3	nviSblndRequest	In	SNVT_char_acii	nv	Motor status request
4	nvoSblndStatus[0...1]	Out	SNVT_SblndState	nv	Motor [0...1] status report
5	nviSceneCfgMotor[0..1]	In	SNVT_scene_cfg	nv	Scene configuration
6	nvoSblndSetting[0..1]	Out	SNVT_setting	nv	Setting in progress on motor [A...B]

Table VII-I: SNVT I/O Details for the blind controller's node object

NC #	Name	SNVT type	Description
1	nciConfigSource	SNVT_config_src	Network configuration source
2	nciStsMaxSendT	SNVT_time_sec	Send time, Maximum for the node status
3	nciSblndMaxSendT	SNVT_time_sec	Send time, Maximum for the sunblind status
4	nciMotorPrmtr[0...1]	User type	Motor [0...1] configuration parameters

Table VII-II: Configuration property details for the blind controller's node object

The Node Object Network Variables Inputs

Object Request (nviRequest)

The 'object request' network variable provided the mechanism to request a particular node, or a particular object within a node. The meaning of the function codes for SNVT_obj_request are described in the SNVT Master List and Programmer's Guide.[3]

Motor status request (nviSnbldRequest)

The 'motor status request' network variable provided the mechanism to request either motors' status depending on an index value. The index values are shown in Table VII–III.

SNVT_char_ascii Hex Value	**Description**
0	Request of all motors status
1	Request status motor [0] (motor A)
2	Request status motor [1] (motor B)
FF	Null

Table VII-III: Index values for nviSnbldRequest

Scene configuration (nviSceneCfgMotor[0...1])

This input variable was used to tell an actuator object to save a specified setting as a scene, report the scene data for a specified scene, and manage scene storage space. The SNVT_scene_cfg was defined by the following structure:

```
type enum  function
0       SCF_SAVE,
1       SCF_CLEAR,
2       SCF_REPORT,
3       SCF_SIZE,
4       SCF_FREE,
255 SCF_NUL,

type unsigned short      scene_number ;
type unsigned short      setting ;
type signed long         rotation ;
type unsigned long       fade_time ;
type unsigned long       delay_time ;( unused )
type unsigned short      scene_priority ;( unused )
```

A description of each of these nviSceneCfgMotor components and their valid ranges is shown in Table VII-IV.

function	**scene number**	**setting**	**rotation**	**fade_time**	**Description**
SAVE	1 à 255	0 à 100%	-360° à 360°	1 à 6553 s	to memorize a new scene
CLEAR	1 à 255	0 à 100%	-360° à 360°	1 à 6553 s	to remove a scene
REPORT	1 à 255	0 à 100%	-360° à 360°	1 à 6553 s	to report about the scene
SIZE	1 à 255	0 à 100%	-360° à 360°	1 à 6553 s	to report the number of programmed scenes
FREE	1 à 255	0 à 100%	-360° à 360°	1 à 6553 s	to report the number of free scene storage space
NUL	1 à 255	0 à 100%	-360° à 360°	1 à 6553 s	to release

Table VII-IV: Values for nviSceneCfgMotor

The Node Object Network Variables Outputs

Object status (nvoStatus)

The 'object status' network variable reported the status for any object of the node. The meaning of the SNVT_obj_status field are described in the SNVT Master List and Programmer's Guide.[3]

Motor Status Report (nvoSblndStatus[0...1])

This output network variable reported the status of the motor [x]. The SNVT_SnblndState was defined by the following structure:

```
typedef struct
{
        SNVT_setting    CurrentPosition
        SnblndSource    Source
        SnblndCause     Cause

} SNVT_SnblndState
```

The current position gave the information of the current position (see Table VII-V). The source field indicated the source of the current control of the motor (see Table VII-VI). The cause field indicated what was the cause of the difference between the current position and the position asked (see Table VII-VII).

Command	**Position***	**Angle****	**Interpretation**
SET_OFF (= 0)	XX	XX	unused
SET_ON (= 1)	XX	XX	unused
SET_DOWN (= 2)	invalid	invalid	going down
SET_UP (= 3)	invalid	invalid	going up
SET_STOP (= 4)	invalid	invalid	stopped to unknown position
SET_STATE (= 5)	100% 0% 0%	invalid invalid 0° to 180°	up position down position down position and specified orientation
SET_NUL (= 255)	XX	XX	unused

* invalid data FF

** invalid data 7FFF

Table VII-V: Current Position

Enum	**Name**	**Description**	
0	Sce_Local_control	a software local control input is the source	Used
1	Sce_Group_control	a network control is the source	Used
2	Sce_Wind	a wind sensor is the source	Unused
3	Sce_Sun	a sun sensor is the source	Used
4	Sce_Rain	a rain sensor is the source	Unused
5	Sce_Frost	a frost sensor is the source	Unused
6	Sce_Scheduler_0	a scheduler is the source (timer)	Unused
7	Sce_Scheduler_1	a scheduler is the source (timer)	Unused
8	Sce_Temp_Outdoor	an exterior temperature sensor is the source	Unused
9	Sce_Temp_Indoor	an interior temperature sensor is the source	Unused
10	Sce_Humid_Outdoor	an exterior relative humidity sensor is the source	Unused
11	Sce_ Humid _Indoor	an interior relative humidity sensor is the source	Unused
12	Sce _Light	an interior light sensor is the source	Unused
13	Sce_Scene_Panel	a scene control is the source	Unused
255	Sce_Nul	unknown	Used

Table VII-VI: Source

Enum	**Name**	**Description**	
0	Cse_Well_Executed	Well Executed	Used
1	Cse_In_Progress	In progress	Used
2	Cse_No_Limit_Switch	The limit switches are not set	Unused
3	Cse_Obstacle_Downward	Obstacle occurring during downward movement	Unused
4	Cse_Obstacle_Upward	Obstacle occurring during upward movement	Unused
5	Cse_Thermal_Detection	Thermal detection	Unused
6	Cse_Power_Failure	Mains failure	Unused
255	Cse_Nul		Used

Table VII-VII: Cause

In Table VII-VI, we can see that the majority of the sources quoted are not used, such as an internal light sensor, an external wind speed sensor, an internal temperature sensor and a scheduler. This is also true for some of the values quoted in Table VII-V and Table VII-VII. All of these unused values were included within the logic to enable the control algorithm to remain flexible enough to be adapted for other uses in the foreseeable future.

Setting in progress (nvoSblndSetting[0...1])

This output network variable was the image of the setting in progress on the corresponding output.

The Node Object Configuration Properties

Network config (nciConfigSource)

The 'network config' network variable could be altered so that the node could support either a self-installation mode or an external installation mode. Self-installation meant that the device assigned itself a network address, external installation meant that the network manager assigned the address.

Status max send time (nciStsMaxSendT)

This configuration network variable was used to set the maximum period of time that expired before the object automatically transmitted the current value of the nvoStatus output variable. The minimum and maximum valid values were 1 second and 6553.4 seconds respectively, with a minimum step resolution of 1s. The default value was 0, which meant the property was disabled, because the use of the maximum send time increases the amount of unnecessary traffic on the control network. In fact the parameter is only needed in certain rare circumstances, where integration with other devices is critical. Even when disabled, failures in the device can be detected by the network management tool which polls each device at regular intervals using the 'object request' network input.

Sunblind status max send time (nciSblndMaxSendT)

This configuration network variable was used to set the maximum period of time that expired before the object automatically transmited the current value of the nvoSblndStatus output variable. Again the minimum and maximum valid values were 1 second and 6553.4 seconds respectively, with a minimum step resolution of 1s. The default value was 0, which meant the property was disabled.

Motor configuration parameters (nciMotorPrmtr[0...1])

The motor configuration parameters were used to set the motor functioning parameters. The variable was defined by the following structure.

```
typedef struct
{
        unsigned            RelayOffDelay
        unsigned long       RotationTime
        Unsigned long       RotationDelay
} snvt_somfy_motor_prmtr;
```

The motor *RelayOffDelay* configuration parameter was used to set the lead-time during which the relays powering the motor were closed (time needed to go from down to up positions). In this case value of 21 seconds was measured and used. The minimum and maximum valid values were 2 seconds and 508 seconds respectively, with a minimum step resolution of 2 seconds.

The motor *RotationTime* configuration parameter was used to set the time needed for a 180° orientation of the slats of the blind on the motor. The minimum and maximum valid values were 0 milliseconds and 65535 milliseconds respectively, with a minimum step resolution of 1 millisecond. A value of 5500 milliseconds was measured and used.

The motor RotationDelay configuration parameter was used to set the delay between the start of movement of the motor and the start of orientation of the slats of the blind on the motor. The minimum and maximum valid values were 0 millisecond and 65535 milliseconds respectively, with a minimum step resolution of 1 millisecond. The default value for this parameter was 250 ms and this seemed to perform adequately.

Local Control Object

The local control object provided network variable inputs that could be liked to switching signals from manual switches or IR controller devices, or scene signals from scene controllers or scene switches. The object also allowed the user to customise the system's response to such signals. A graphical representation of the object can be found in Figure VII-II, and Tables VII-VIII and VII-IX outline details of the object's network variables and configuration properties respectively.

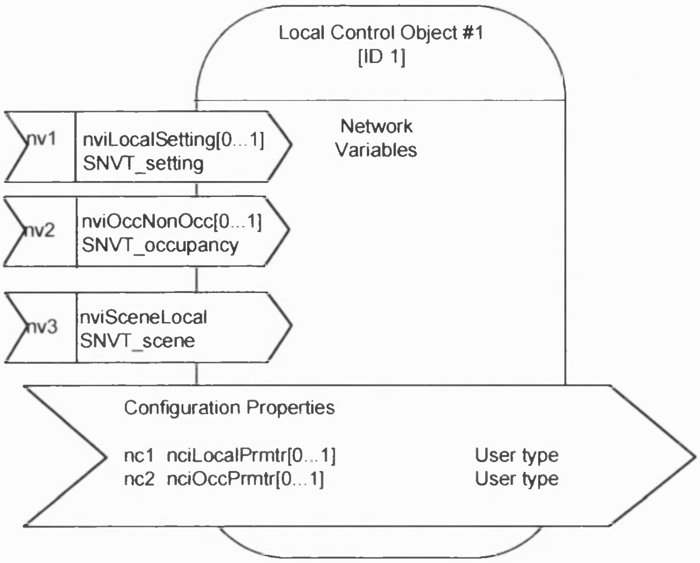


Figure VII-III: A graphical representation of the local control object

NV #	Name	In/Out	SNVT Type	Class	Description
1	nviLocalSetting[0...1]	In	SNVT_setting	nv	Local setting input
2	nviOccNonOcc[0...1]	In	SNVT_occupancy	nv	Automatic or non-automatic functioning mode
3	nviSceneLocal	In	SNVT_scene	nv	Local scene input

Table VII-VIII: SNVT I/O Details for the local control object

NC #	Name	SNVT type	Description
1	nciLocalPrmtr[0...1]	User type	Local control parameters
2	nciOccPrmtr[0...1]	User type	Auto / non-auto parameters

Table VII-IX: Configuration details for the local control object

The Local Control Object Network Variables Inputs

Local setting (nviLocalSetting[0...1])

The 'local setting' network variable contained the desired action to be carried out and/or position to be reached. Although the setting comes from the network, it is considered as a local control, like controls coming from the hardwired inputs. The SNVT_setting was defined by the following structure:

```
type enum  function
0       SET_OFF,
1       SET_ON,
2       SET_DOWN,
3       SET_UP,
4       SET_STOP,
5       SET_STATE,
255     SET_NUL ,

type unsigned short  setting   (position)
type unsigned long  rotation   (angle)
```

A description of each of these nviLocalSetting components and their valid ranges are similar to those shown in Table VII-V.

Automatic / non-automatic functioning mode (nviOccNonOcc[0...1])

This input network variable was used to set the operating mode depending on occupancy sensed by an occupancy sensor. Each time a non-occupancy signal was received, the corresponding outputs were set to the automatic mode. The corresponding outputs were set to the non-automatic mode when an occupancy signal was received from the occupancy sensor or a signal was received from a local control device.

In the non-automatic mode, all objects with lower priority than the local control priority were disabled. If this input variable was not bound, the corresponding outputs were set into the automatic mode. Therefore it was decided not to bind this variable to the occupancy sensor.

Local Scene input (nviSceneLocal)

This input variable was used to tell an actuator object to save its current setting as a scene, or recall a previously saved scene. The SNVT_scene was defined by the following structure:

```
type enum  function
0       SC_RECALL,
1       SC_LEARN,
255     SC_NULL,
```

function	scene number	Description
RECALL	1 à 255	to recall a memorized scene
LEARN	1 à 255	to learn a new scene
NUL	1 à 255	to release

Table VII-X: Values for nviSceneLocal

The Local Control Object Configuration Properties

Local control parameters (nciLocalPrmtr[0...1])

This configuration network variable was used to set the local controls configuration parameters.

```
typedef struct
{
                unsigned        PrioLevel
                unsigned        PrioDuration
                unsigned        TiltingTime
                unsigned        MotorControl
                SNVT_setting    State
} snvt_somfy_local_prmtr;
```

The hardwired Push Buttons and the network local setting (nviLocalSetting) didn't have the same configuration parameters.

	Hardware BP	network local
PrioLevel	X	X
PrioDuration	X	X
TiltingTime	X	
MotorControl	X	X
State	X	

Table VII-XI: Configuration parameters used for hardwired and network controls

The hard-wired inputs had the following characteristics:

- a pulse less than 0.5seconds actuated a full up or down operation (down to the position defined by configuration);
- an action between 0.5 seconds and the configured tilting time was a momentary action (the motor stopped turning when the corresponding button was released);
- an action longer than the configured tilting time commanded a full up or down operation (down to the actual down limit position).

The local control *PrioLevel* configuration network variable was used to set a Priority level for the local controls (see Table VII-XII). The default was a low priority.

Value	**Priority level**
255	No Priority
200	Low Priority
150	Medium Priority
100	High Priority

Table VII-XII: Values for the priority level configuration property

The local control *PrioDuration* configuration parameter was used to enable or disable local controls and it defined whether the local controls would have permanent (last throughout the occupied period) or temporary one-shot priority. This parameter could also provide an opt out clause for the device to bypass the automatic operation if it began to annoy the occupants.

The local control *TiltingTime* configuration parameter was used to define the hardware local tilting time for the slow speed of the dual speed controller. This slow speed setting allowed the user to have greater control of the blind tilt angle by using the manual switches. The minimum and maximum valid values were 0 seconds and 12 seconds respectively, with a minimum step resolution of 1 second. The value used was 5 seconds, similar to the value used for the nciMotorPrmtr[0...1].

The local control *MotorControl* configuration parameter defined which motor had to take into account which nviLocalSetting.

The local control *State* configuration parameter defined the action to be carried out when a pulse action on the down hardwired button is pressed. Its values were similar to those shown in Table VII-V.

Occupancy parameters (nciOccPrmtr[0...1])

These configuration network variables were used to set different operational parameters for the automatic / non-automatic function.

```
typedef struct
{
        unsigned                NonOccDisAppTime
        unsigned                MotorControl

} snvt_somfy_occupancy_prmtr;
```

The occupancy *NonOccDisAppTime* configuration parameter was used to define the non-occupancy time. This was the delay imposed on the change between the occupied state and the non-occupied state. The minimum and maximum valid values were 0 minutes and 255 minutes respectively. The default value was 15 minutes but because an occupancy delay time had been included within the occupancy sensor set-up this was changed to 0 minutes.

The occupancy *MotorControl* configuration parameter defined which motor has to take into account which nviOccNonOcc value.

Zone control object

The zone control object provided the facility for the blinds to be controlled in a large control zones or scene zones, as well as in the small local zones discussed in the last section. A graphical representation of the object can be found in Figure VII-IV, and Tables VII-XIII and VII-XIV outline details of the object's network variables and configuration properties respectively.

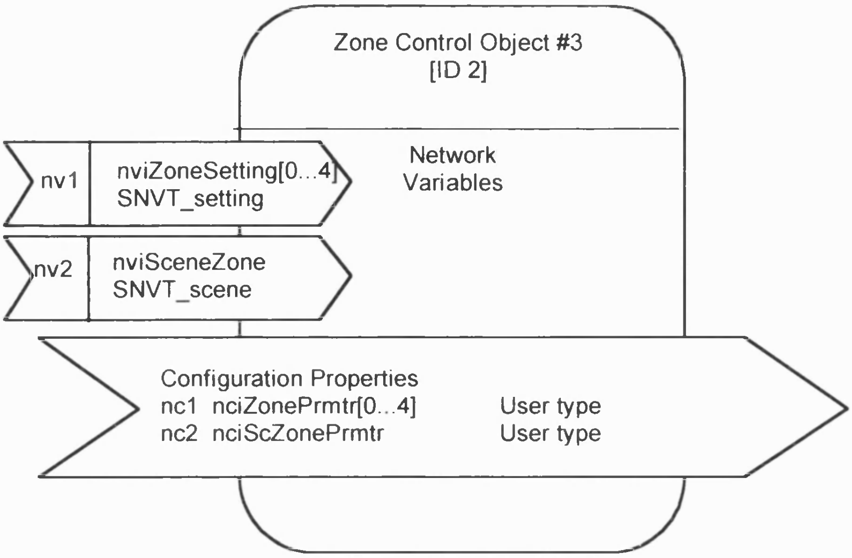


Figure VII-IV: A graphical representation of the zone control object

NV #	Name	In/Out	SNVT Type	Class	Description
1	nviZoneSetting[0...4]	In	SNVT_setting	nv	Zone setting input
2	nviSceneZone	In	SNVT_scene	nv	Scene zone input

Table VII-XIII: SNVT I/O Details for the zone control object

NC #	Name	SNVT type	Description
1	nciZonePrmtr[0...4]	User type	Zone parameters
2	nciScZonePrmtr	User type	Scene zone parameters

Table VII-XIV: Configuration details for the zone control object

The Zone Control Object Network Variables

Zone setting input (nviZoneSetting[0...4])

This input network variable contained the desired action to be carried out and/or position to be reached. The SNVT_setting was defined using the same structure as the current position of the nvoSblndStatus shown in Table VII-V.

Zone Scene input (nviSceneZone)

This input network variable was essentially the same as the Zone setting input variable but allowed the zone controller to also control the device in terms of pre-set scenes.

The Zone Control Object Configuration Properties

Network controls configuration parameters (nciZonePrmtr[0...4])

This configuration network variable was used to set the network controls configuration parameters.

```
typedef struct
{
        unsigned          PrioLevel
        unsigned long     MaxRcvTime
        unsigned          AppTimeDelay
        unsigned          NoPrioRedo
        Unsigned          MotorControl

} snvt_somfy_zone_prmtr ;
```

The zone control *PrioLevel* configuration parameter made it possible to set a Priority level for the network control. The values for this configuration parameter were the same as those shown for nciLocalPrmtr *PrioLevel* in Table VII-XII.

The zone control *MaxRcvTime* configuration parameter was used to control the maximum period of time that expired before the object automatically didn't take into account the value of the control setting if a value was not received. The minimum and maximum valid values were 0 seconds and 65535 seconds respectively, with a minimum step resolution of 1 second. The default value was 0 seconds.

The zone control *AppTimeDelay* configuration parameter set an appearing time delay that the node had to take into account before the motor moved. The minimum and maximum valid values were 0 seconds and 1275 seconds respectively, with a minimum step resolution of 5 second. The default value was 0 seconds.

The zone control *NoPrioRedo* configuration parameter was a true or false parameter that defined whether the node had to take into account sensor actions with no Priority when all other settings had disappeared.

The zone control *MotorControl* configuration parameter made it possible to define which motor was linked to which nviZoneSetting.

Scene zone configuration parameters (nciScZonePrmtr)

This configuration network variable was used to set the scene network controls configuration parameters.

```
typedef struct
{
        unsigned          NoPrioRedo
} snvt_somfy_scene_zone_prmtr ;
```

The true or false *NoPrioRedo* scene zone control parameter defined if the node had to take into account sensor actions with no Priority when all other scene settings had disappeared.

Light Control Object

The light control object provided the facility for the blind to respond to variations in light level sensed by an external sensor. A graphical representation of the object can be found in Figure VII-V, and Tables VII-XV and VII-XVI outline details of the object's network variables and configuration properties respectively.

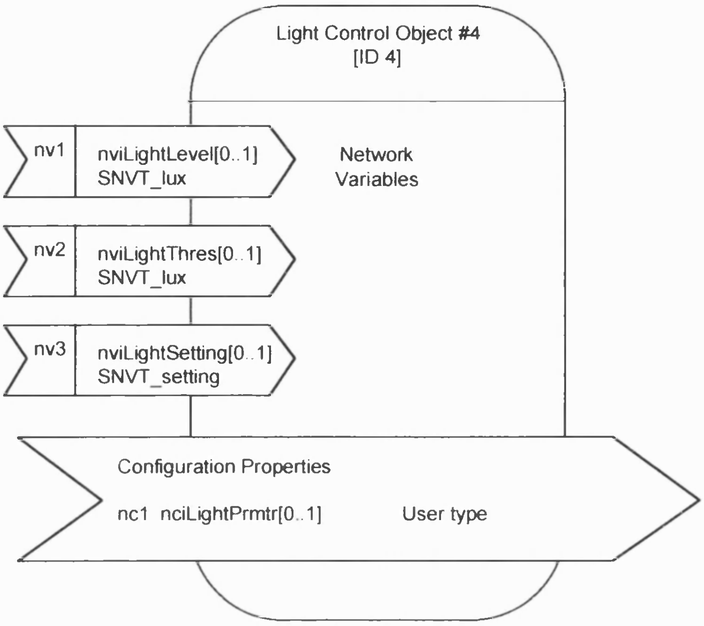


Figure VII-V: A graphical representation of the light control object

NV #	**Name**	**In/Out**	**SNVT Type**	**Class**	**Description**
1	nviLightLevel[0..1]	In	SNVT_lux	nv	Light level input
2	nviLightThres[0..1]	In	SNVT_lux	nv	Light level threshold
3	nviLightSetting[0..1]	In	SNVT_setting	nv	Light control setting

Table VII-XV: SNVT I/O Details for the light control object

NC #	**Name**	**SNVT type**	**Description**
1	nciLightPrmtr[0..1]	User type	Light control parameters

Table VII-XVI: Configuration details for the light control object

The Light Control Object Network Variables Inputs

Light level input (nviLightLevel[0..1])

This input network variable indicated the light level measured by a light sensor (in this case a sun sensor). The reaction of the blind to this reading was defined by nviLightThres, nviLightSetting and the operational parameters set in nciLightPrmtr.

Light level threshold (nviLightThres[0..1])

This input network variable set the light level threshold that determined whether the sun was considered to be out and the blind should be lowered.

Light level setting (nviLightSetting[0..1])

This input network variable set the position to be reached or the action to be carried out when the actual light level became higher than the light threshold. The SNVT_setting was defined using the same structure as the current position of the nvoSblndStatus shown in Table VII-V.

The Light Control Object Configuration Properties

Light control parameters (nciLightPrmtr[0..1])

This configuration network variable was used to set different operational parameters of the light control object, such as its priority level and time delays.

```
typedef struct
{
        unsigned            PrioLevel
        unsigned long       MaxRcvTime
        unsigned long       AppThres
        unsigned long       AppTimeDelay
        SNVT_setting        AppSetting
        unsigned long       DisThres
        unsigned long       DisTimeDelay
        SNVT_setting        DisSetting
        unsigned            AppDisPrioDurations
        unsigned            MotorControl
} snvt_somfy_light_prmtr ;
```

The light *PrioLevel* configuration parameter made it possible to set a Priority level for the corresponding control setting values (AppSetting and DisSetting). The values for this configuration parameter were the same as those shown for nciLocalPrmtr *PrioLevel* in Table VII-XII. The default for this parameter was no priority, but this was changed to low priority.

The light *MaxRcvTime* configuration parameter was used to control the maximum period of time that expired before the object automatically did not take into account the present sun value if a new value had not been received. The minimum and maximum valid values were 0 seconds and 65535 seconds respectively, with a minimum step resolution of 1 second. The default value for this parameter was 0 seconds or disabled.

The light *AppThres* configuration parameter made it possible to set a brightness threshold that triggered the appearance timers. When the nviLightlevel value became higher than this threshold an action defined by AppSetting was carried out. The default was 24,000 lux and this seemed to function well in the months when the blind was installed and configured.

The light *AppTimeDelay* configuration parameter set a sun appearing time delay that the sunblind interface had to take into account when the value of SNVT_lux became higher than the appearing threshold. The minimum and maximum valid values were 0 seconds and 65535 seconds respectively, with a minimum step resolution of 1 second. The default value was 180 seconds, but 30 seconds seemed to perform just as well.

The light *AppSetting* configuration parameter defined the action to be carried out when the value of SNVT_lux became higher than the appearing threshold and when the sun appearing time delay had ended. The SNVT_setting was defined using the same structure as the current position of the nvoSblndStatus shown in Table VII-V. The default was 'STATE, 0%, 45' meaning lower the blinds and tilt them to 45 degrees.

The light *DisThres* configuration parameter made it possible to set a brightness threshold that triggered the disappearance timers. When the nviLightLevel value became lower than this threshold an action defined by DisSetting was carried out. The default for this parameter was the same as for the default value of *AppThres* configuration parameter, 24,000 lux, therefore the same value was used.

The light *DisTimeDelay* configuration parameter set a sun disappearing time delay that the sunblind interface had to take into account when the value of SNVT_lux became lower than the disappearing threshold. The minimum and maximum valid values were 0 seconds and 65534 seconds respectively, with a minimum step resolution of 1 second. The default value was 900 seconds but 600 seconds seemed to perform adequately.

The light *DisSetting* configuration parameter defined the action to be carried out when the value of SNVT_lux became lower than the disappearing threshold and when the sun disappearing time delay had ended. The valid range was any value defined in the SNVT_setting (see Sun configuration parameter: AppSetting). The default was 'UP, inv, inv' meaning raise the blinds.

The light *AppDisPrioDurations* configuration parameter set the control linked to be maintained or released priority duration, when the nviLightLevel Value became higher than the AppThres value or lower than the DisThres. The default values were that the appearing priority was maintained and the disappearing priority was released.

The light *MotorControl* configuration parameter made it possible to set which motor has to take into account the which AppSetting and DisSetting.

Network Variable and Configuration Properties used in Test Room

This section list the network variables and configuration properties used within some of the key control devices installed within the test room. Both are shown to give some idea of the variable forms used by the devices and the DDE server. The configuration properties shown are those used within the test cell set up.

The Blind Controller

OBJ	Name	SNVTs / *SCPTs*	Description of Functionality	Value Format
INPUTS				
Ob	nviRequest	SNVT_obj_request	Object Request - allows an object to be placed in one of several functional modes (normal, update-status, disable, report_mask).	0,RQ_NORMAL
Ob	nviSblndRequest	SNVT_char_ascii	Motor status request - provides the mechanism to request the motor status depending on the index value (HEX value 1)	0 = request all motor status, 1 = request motor 0, 2 = request motor 1
Ob	nviSceneCfgMotor[0..1]	SNVT_scene_cfg	Input variable use to tell an actuator object to save a specified setting as a scene, report the scene data for a specified scene and manage the scene storage space	SCF_SAVE 0 0.0 0.00 0.0 0.0 0
Ob	nciConfigSource	SNVT_config_src / *SCPT_nwrk_cnfg*	Network configuration source - allows nodes to either support self-installation or to be installed externally	CFG_EXTERNAL
Ob	nciStsMaxSendT	SNVT_time_sec / *SCPT_max_snd_t*	Maximum send time for the node status (seconds)	0.0
Ob	nciSblindMaxSendT	SNVT_time_sec / *SCPT_max_snd_t*	Maximum send time for the sunblind status (seconds)	0.0
Ob	nciMotorPrmtr[0..1]	User Type	Sets the motor function parameters, RelayOffDelay (seconds), RotationTime (milliseconds) and RotationDelay (milliseconds)	20|21|124|0|250
Lc	nviLocalSetting[0..1]	SNVT_setting	Contains the desired action to be carried out and/or position to reach. Although the setting comes from the network, it is considered as a local control, like controls coming from the hardwired inputs.[function, position, angle] only valid when in operation otherwise null.	SET_OFF 0.0 0.00
Lc	nviOccNonOcc[0..1]	SNVT_occupancy	This network variable is used to set the operating mode. If non-occupancy data received the corresponding outputs are set to automatic mode. The outputs go to non-automatic mode when a local control (from network or hardwired inputs) is received. In the non-automatic mode all objects with lower priority than the local control priority are disabled. If this input variable is not bound, the outputs go to automatic mode	OC_OCCUPIED
Lc	nviSceneLocal	SNVT_scene	Tells the actuator to save it's current setting as a scene or recall a previously saved scene	SC_RECALL 0

<table>
<tr><td>Lc</td><td>nciLocalPrmtr[0]</td><td>User Type</td><td>This is used to set the local controls configuration parameters, PrioLevel, PrioDurartion, TiltingTime, MotorControl, State.</td><td>200|0|5|2|5|0|0|0</td></tr>
<tr><td>Lc</td><td>nciLocalPrmtr[1]</td><td>User Type</td><td>This is used to set the local controls configuration parameters, PrioLevel, PrioDurartion, TiltingTime, MotorControl, State. The hardwired PushButtons and the network local setting (nviLocalSetting) don't have the same configuration parameters</td><td>200|0|3|2|5|0|0|0</td></tr>
<tr><td>Lc</td><td>nciOccPrmtr</td><td>User Type</td><td>Used to set different operational parameters for the automatic / non-automatic function, NonOccDisAppTime and MotorControl</td><td>15|2</td></tr>
<tr><td>Zc</td><td>nviZoneSetting[0..4]</td><td>SNVT_setting</td><td>Contains the desired action to be carried out and/or position to reach.</td><td>SET_OFF 0.0 0.00</td></tr>
<tr><td>Zc</td><td>nviSceneZone</td><td>SNVT_scene</td><td>Scene Zone Input</td><td>SC_RECALL 0</td></tr>
<tr><td>Zc</td><td>nciZonePrmtr[0]</td><td>User Type</td><td>Used to set the network controls configuration parameters, PrioLevel, MaxRcvTime, AppTimeDelay, NoPrioRedo, MotorControl</td><td>200|0|0|0|0|3</td></tr>
<tr><td>Zc</td><td>nciZonePrmtr[1]</td><td>User Type</td><td>Used to set the network controls configuration parameters, same as nciZonePrmtr[0]</td><td>255|0|0|0|0|3</td></tr>
<tr><td>Zc</td><td>nciZonePrmtr[2]</td><td>User Type</td><td>Used to set the network controls configuration parameters, same as nciZonePrmtr[0]</td><td>255|0|0|0|0|3</td></tr>
<tr><td>Zc</td><td>nciZonePrmtr[3]</td><td>User Type</td><td>Used to set the network controls configuration parameters, same as nciZonePrmtr[0]</td><td>255|0|0|0|0|3</td></tr>
<tr><td>Zc</td><td>nciZonePrmtr[4]</td><td>User Type</td><td>Used to set the network controls configuration parameters, same as nciZonePrmtr[0]</td><td>100|0|0|0|0|3</td></tr>
<tr><td>Zc</td><td>nciScZonePrmtr</td><td>User Type</td><td>Used to set the scene network controls configuration parameters, NoPrioRedo - which defines if a node has to take into account sensors actions with no Priority when all other scene settings have disappeared</td><td>18</td></tr>
<tr><td>Ltc</td><td>nviLightLevel[0..1]</td><td>SNVT_lux</td><td>Light level input - measured by light sensor</td><td></td></tr>
<tr><td>Ltc</td><td>nviLightThres[0..1]</td><td>SNVT_lux</td><td>Light level threshold impacted on the light control object output</td><td>24000</td></tr>
<tr><td>Ltc</td><td>nviLightSetting[0..1]</td><td>SNVT_setting</td><td>Light control setting that blind must reach when threshold passed</td><td>SET_STATE 0.0 45.00</td></tr>
<tr><td>Ltc</td><td>nciLightPrmtr[0]</td><td>User Type</td><td>Light control parameters - used to set the different operational parameters of the light control object, PrioLevel, MaxRcvTime, AppThres, AppTimeDelay, AppSetting, DisThres, DisTimeDelay, DisSetting, AppDisPrioDurations, MotorControl</td><td>200|0|0|93|0|30|5|0|8|203|93|192|3|132|5|200|127|255|1|3</td></tr>
<tr><td>Ltc</td><td>nciLightPrmtr[1]</td><td>User Type</td><td>Light control parameters - used to set the different operational parameters of the light control object, same as nciLightPrmtr[0]</td><td>255|0|0|93|0|180|5|0|8|203|93|192|3|132|5|200|127|255|1|0</td></tr>
</table>

OUTPUTS

<table>
<tr><td>Ob</td><td>nvoStatus</td><td>SNVT_obj_status</td><td>Object status - reports the status for any object on a node (15 fields, 6 used, 9 unused)</td><td>0 0,0,0,0,0,0,0,0, 0,0,0,0,0, 0,0,0,0,0,0,0</td></tr>
<tr><td>Ob</td><td>nvoSblindStatus[0..1]</td><td>SNVT_SnBlndState</td><td>Motor [1..2] status report - used to communicate current position, source and cause (see literature)</td><td>0|0|0|0|0|0</td></tr>
<tr><td>Ob</td><td>nvoSBlndSetting[0]</td><td>SNVT_Setting</td><td>This is the image of the setting in progress on the corresponding output</td><td>SET_STATE 0.0 0.00</td></tr>
<tr><td>Ob</td><td>nvoSBlndSetting[1]</td><td>SNVT_Setting</td><td>This is the image of the setting in progress on the corresponding output</td><td>SET_OFF 0.0 0.00</td></tr>
</table>

The Sun Sensors

OBJ	Name	SNVTs / SCPTs	Description of Functionality	Value Format
INPUTS				
	nviInstallDel	SNVT_lev_disc	LED Installation Input - sets the state of the LED of the sun sensor interface to be on or off during the process of intuitive installation	ST_OFF
	nciSnMinRange	SNVT_lux	High Limit of Transmitted Value - specifies the low limit of the transmitted values (linked to the type of solar cell)	**3000**
	nciSnSendOnDelta	SNVT_lev_count	Sun brightness send on delta - specifies the relative variation of lux to take into account before updating nvoSun	10 0
	nciSnMaxSendTime	SNVT_elapsed_tm	Sun brightness max send time - specifies the maximum period of time that expires before the object automatically transmits the current value of nvoSun	0 0:1:0:0
	nciSnMinSendTime	SNVT_elapsed_tm	Sun brightness min send time - specifies the minimum period of time between nvoSun transmissions	0 0:0:1:0
	nciSnAverage	SNVT_count	Filtering on Measurements - specifies the filtering ratio of measurements versus transmission (linked to the type of solar cell)	**1**
	nciSnThresTable	SNVT_lux	Threshold Table Value - value specified for test mode and generates the transmission of nviTestMode set to test	8000
	nciSnThresTable	SNVT_lux	Threshold Table Value - set of 16 threshold values that can be manually set on the device	3000
	nciSnThresTable	SNVT_lux	Threshold Table Value - set of 16 threshold values that can be manually set on the device	3700
	nciSnThresTable	SNVT_lux	Threshold Table Value - set of 16 threshold values that can be manually set on the device	4500
	nciSnThresTable	SNVT_lux	Threshold Table Value - set of 16 threshold values that can be manually set on the device	5600
	nciSnThresTable	SNVT_lux	Threshold Table Value - set of 16 threshold values that can be manually set on the device	6900
	nciSnThresTable	SNVT_lux	Threshold Table Value - set of 16 threshold values that can be manually set on the device	8500
	nciSnThresTable	SNVT_lux	Threshold Table Value - set of 16 threshold values that can be manually set on the device	10000
	nciSnThresTable	SNVT_lux	Threshold Table Value - set of 16 threshold values that can be manually set on the device	13000
	nciSnThresTable	SNVT_lux	Threshold Table Value - set of 16 threshold values that can be manually set on the device	16000
	nciSnThresTable	SNVT_lux	Threshold Table Value - set of 16 threshold values that can be manually set on the device	20000
	nciSnThresTable	SNVT_lux	Threshold Table Value - set of 16 threshold values that can be manually set on the device	24000
	nciSnThresTable	SNVT_lux	Threshold Table Value - set of 16 threshold values that can be manually set on the device	30000
	nciSnThresTable	SNVT_lux	Threshold Table Value - set of 16 threshold values that can be manually set on the device	36000
	nciSnThresTable	SNVT_lux	Threshold Table Value - set of 16 threshold values that can be manually set on the device	45000
	nciSnThresTable	SNVT_lux	Threshold Table Value - set of 16 threshold values that can be manually set on the device	55000
	nciConfigSource	SNVT_config_src	Network configuration source - allows nodes to either supprot self-installation or to be installed externally	CFG_EXTERNAL
	nviRequest	SNVT_obj_request	Object Request - allows an object to be be placed in one of several functional modes (normal, update-status, disable, report_mask).	0,RQ_NORMAL

	nciThresMaxSdT	SNVT_elapsed_tm	Threshold Max Send Time - used to specify the period of time that expires before the object automatically transmits the current setting of nvoSnThreshold	0 1:0:0:0
	nciTestMdeMaxSdT	SNVT_elapsed_tm	Test Mode Max Send Time - used to specify the period of time that expires before the object automatically transmits the value of nvoTestMode when it is set to test mode	0 0:0:10:0
	nciStsMaxSendT	SNVT_elapsed_tm	Status Max Send Time - used to set the maximum period of time that expires before the object automatically transmits the current value of the nvoStatus output variable	0 0:0:0:0
	nciSnGain	SNVT_muldiv	Gain of the Sensor - sets the gain of the sensor (linked to the type of solar cell) DON'T CHANGE	**56923/256**
	nciSnOffset	SNVT_lux	Offset of the sensor - sets the offset of the sensor (linked to the type of solar cell) DON'T CHANGE	**0**
	nciSnMaxRange	SNVT_lux	High Limit of Transmitted Value - specifies the high limit of the transmitted values (linked to the type of solar cell) DON'T CHANGE	**55000**

OUTPUTS

	nvoInstallDel	SNVT_lev_disc	LED Installation Input - sets the state of the LED of the devices the sun sensor interface is bound to during the process of intuitive installation	ST_OFF
	nvoStatus	SNVT_obj_status	Object status - reports the status for any object on a node (15 fields, 6 used, 9 unused)	0 0,0,0,0,0,0,0,0,0, 0,0,0,0, 0,0,0,0,0,0,0
	nvoSun	SNVT_lux	Sun Brightness - the brightness measured by the solar cells connected to the device	13786
	nvoSnThreshold	SNVT_lux	Sun Brightness Threshold - the brightness threshold set on the node by the hardware selector	20000
	nvoTestMode	SNVT_lev_disc + SNVT_elapsed_tm	Functioning Mode Setting - sets the functioning mode of nodes it is bound to, to be a test mode or a normal one	0|0|0|0|0|0|0|0

The 8-Way Lighting Controller

OBJ	Name	SNVTs / *SCPTs*	Description of Functionality	Value

INPUTS

Ob	nviRequest	SNVT_obj_request	Object Request - allows an object to be be placed in one of several functional modes (normal, update-status, disable, report_mask).	0,RQ_NORMAL
Ob	nviholiday	????		0|0|0|0|0|0|0|0|0|0
Ob	nvigenerate	SNVT_switch		0.0 0
Ob	nviemergcy	SNVT_switch		0.0 0
Ob	nvievent	????		0|0|0|0|0|0|0|0|0
Ob	nviTimeSet	SNVT_time_stamp		0/0/0 0:0:0
Ob	nvisecurity	SNVT_switch		0.0 0
Ob	nviDay	SNVT_date_day		DAY_SUN
Ob	nvifunc	????		0|0|0|0|0
C1	nvi01pri	SNVT_switch		0.0 0
C1	nvi01sec	SNVT_switch		0.0 0
C1	nvi01tri	SNVT_switch		0.0 0
C2	nvi02pri	SNVT_switch		0.0 0
C2	nvi02sec	SNVT_switch		0.0 0
C2	nvi02tri	SNVT_switch		0.0 0

OUTPUTS

Ob	nvoStatus	SNVT_obj_status	Object status - reports the status for any object on a node (15 fields, 6 used, 9 unused)	0 0,0,0,0,0,0,0,0,0, 0,0,0,0, 0,0,0,0,0,0,0
Ob	nvoholiday	????		0|0|0|0|0|0|0|0|0|0
Ob	nvofunc	????		0|0|0|0|0
Ob	nvorunhours	????		0|0|0|0|0|0|0|0|0|0|0|0|0|0|0|0
Ob	nvoevent	????		0|0|0|0|0|0|0|0|0
C1	nvotime1	SNVT_switch		0.0 0
C1	nvo01ValueFb	SNVT_switch		0.0 0
C2	nvotime2	SNVT_switch		0.0 0
C2	nvo02ValueFb	SNVT_switch		0.0 0
S1	nvo090Value	SNVT_switch		0.0 0

The 3 in 1 Sensor

INPUTS

OBJ	Name	SNVTs / *SCPTs*	Description of Functionality	Value
Ob	nciConfigure	SNVT_config_src / *SCP_Type1*	Network configuration source - allows nodes to be configured at install time to implement global functions or object functions	0000000000000
Ob	nviRequest	SNVT_obj_request	Object Request - allows an object to be placed in one of several functional modes (normal, update-status, disable, report_mask).	0,RQ_NORMAL
Ls	nci1MaxSendT	SNVT_time_sec	Internal Illuminance max send time - specifies the maximum period of time that expires before the object automatically transmits the current value of nvoLuxLevel	2.0 [0 0:1:0:0]
Ls	nciLocation	SNVT_str_asc	Location - 32 character string is used to denote the physical location of the device	
Ls	nci1Reflect	SNVT_lev_percent / *SCP_Type89*	Reflections factor - percentage reflections factor for the environment the unit is installed (used when the desk lux output function is selected)	800
Ls	nci1FieldCal	SNVT_lux / *SCP_Type90*	Field Calibration - used to calibrate the light object for a particular piece of hardware (set at manufacture)	4074
Cl	nci2MaxSendTime	SNVT_time_sec / *SCPTmaxSendTime*	Constant Light Controller max send time - specifies the max period of time between nvoXlampValue transmissions	0 0:0:1:0
Cl	nvi2Setting	SNVT_setting	This input controls the modulation of the object. The object can regulate to the light level specified in the configuration property nciXLuxSetpt, or it can temporarily over-ride with this value	SET_ON 0.0 0.00
Cl	nvi2LuxLevel	SNVT_lux	The light level control input to the node (fed from the light sensor or similar object)	300
Cl	nci2LuxSetpt	SNVT_lux / *SCPTluxSetpoint*	Sets the light level deman for the object in it's normal operating state	300
Cl	nci2Step	SNVT_lev_count / *SCPTstep*	Controls the percentage change from the constant light controller when it is regulating light	5
Cl	nci2PowerUSt	SNVT_setting / *SCP_Type87*	Controls the default power up value loaded to the nviSetting. It is used to control the initial state of the Constant Light Controller	1000000
Os	nci4Location	SNVT_str_asc / *SCPTlocation*	Location - 32 character string is used to denote the physical location of the device	
Os	nci4Heartbt	SNVT_time_sec / *SCPTmaxRcvTime*	Defines the update rate for the object assuming there is no change of state in the mean time	50
Oc	nvi5Occup	SNVT_occupancy	Primary input to the Occupancy Controller Object for the Occupancy sensor	OC_UNOCCUPIED
Oc	nvi5Second	SNVT_occupancy	Secondary input to the Occupancy Controller Object for the Occupancy sensor	OC_UNOCCUPIED
Oc	nvi5Setting	SNVT_setting	Used to control the operation of the device	SET_ON 0.0 0.00
Oc	nvi5Enable	SNVT_switch	Has a number of different functions depending on the value selected for this object in nciConfigure	100.0 1
Oc	nci5TimeOut	SNVT_elapsed_tm / *SCPToffDelay*	This configuration property sets the time delay needed before the occupancy controller can be turned off	0 0:5:0:0
Ir	nvi7SwitchFb	SNVT_switch	Used to modulate the output from the infra-red switch object	100.0 1
Ir	nci7irCode	UNVT_7631IRCode / *SCP_Type2*	Channel that this object will respond to	1
Ir	nci7time	SNVT_elapsed_tm / *SCPToffDelay*	Optional configuration property. If not set to zero then the switch object will work in pulse mode with the time pulse setting specified	0 0:0:1:0

Ir	nci7preset	UNVT_7631Preset / *SCP_Type3???*	Allows the selection of the preset lamp values when the IR switch object is configured to Toggle/Level Preset. Each level corresponds to the 8 buttons on the handset	????
Ir	nvi8SwitchFb	SNVT_switch	Used to modulate the output from the infra-red switch object	100.0 1
Ir	nci8irCode	UNVT_7631IRCode / *SCP_Type2*	Channel that this object will respond to	5
Ir	nci8time	SNVT_elapsed_tm / *SCPToffDelay*	Optional configuration property. If not set to zero then the switch object will work in pulse mode with the time pulse setting specified	0 0:0:1:0
Ir	nci8preset	UNVT_7631Preset	Allows the selection of the preset lamp values when the IR switch object is configured to Toggle/Level Preset. Each level corresponds to the 8 buttons on the handset	????
Lx	nvi9SwitchFb	SNVT_switch	Used to modulate the output from the infra-red switch object. In passthrough function it is used to control and enable the object	100.0 1
Lx	nvi9LuxLevel	SNVT_lux	Input from the light sensor object	872
Lx	nci9LuxSetP	SNVT_lux / *SCPTluxSetpoint*	Sets the switching threshold for the Lux Switch Object	300
Lx	nci9Hyst	SNVT_lev_cont / *SCPTonoffHysteresis*	Sets the hysteresis for the switching threshold (in %)	5
Ts	nci10MaxST	SNVT_time_sec		
Ts	nci10MinST	SNVT_time_sec		
Ts	nci10MinDlt	SNVT_temp_p		

OUTPUTS

Ob	nvoStatus	SNVT_obj_status	Object status - reports the status for any object on a node (15 fields, 6 used, 9 unused)	0 0,0,0,0,0,0,0,0,0, 0,0,0,0, 0,0,0,0,0,0,0
Ls	nvo1LuxLevel	SNVT_lux	Internal Illuminance - the brightness measured by the sensor connected to the device (can be in two modes, sensor lux output, desk lux output [taking into account reflections])	1160
Cl	nvoXLampValue	SNVT_lux	This is the control output from the constant light controller, used to control lamp loads	0.0 0
Os	nvo4Occup	SNVT_occupancy	Two occupancy sensor values which are the same but allow the occupancy sensor device to be attached to an object that does not allow aliasing	OC_UNOCCUPIED
Os	nvo4OccupSec	SNVT_occupancy	Two occupancy sensor values which are the same but allow the occupancy sensor device to be attached to an object that does not allow aliasing	OC_UNOCCUPIED
Oc	nvoXLampValue	SNVT_switch	This is then control output of the Occupancy Controller object for controlling lamp loads	100.0 1
Oc	nvoXSetting	SNVT_setting	Used to control the setting inputs on other objects. Occupied = SET_ON.	SET_ON 0.0 0.00
Ir	nvo7Switch	SNVT_switch	Control output for the IR switch object	100.0 0
Ir	nvo7Setting	SNVT_setting	used to control the operation of other LonMark objects, supports the SET_ON, SET_OFF, SET_UP and SET_DOWN functions	SET_ON 0.0 0.00
Ir	nvo8Switch	SNVT_switch	Control output for the IR switch object	100.0 0
Ir	nvo8Setting	SNVT_setting	used to control the operation of other LonMark objects, supports the SET_ON, SET_OFF, SET_UP and SET_DOWN functions	SET_ON 0.0 0.00
Lx	nvo9Switch	SNVT_switch	Control output from the Lux Sensor object used to control lamp loads	0.0 0

The Lumen Maintenance Sensor

OBJ	Name	SNVTs	Description of Functionality	Value

INPUTS

Ob	nviRequest	SNVT_obj_request	Object Request - allows an object to be be placed in one of several functional modes (normal, update-status, disable, report_mask).	0, RQ_NORMAL
	nviright_adjust1	SNVT_lev_cont	Alters the preset lighting level at nvoright1	50.0
	nviright_adjust2	SNVT_lev_cont	Alters the preset lighting level at nvoright2	33
	nviright_adjust3	SNVT_lev_cont	Alters the preset lighting level at nvoright3	16.5
	nviright_adjust4	SNVT_lev_cont	Alters the preset lighting level at nvoright4	0.0
	nvileft_adjust1	SNVT_lev_cont	Alters the preset lighting level at nvoleft1	50.0
	nvileft_adjust2	SNVT_lev_cont	Alters the preset lighting level at nvoleft2	33
	nvileft_adjust3	SNVT_lev_cont	Alters the preset lighting level at nvoleft3	16.5
	nvileft_adjust4	SNVT_lev_cont	Alters the preset lighting level at nvoleft4	0.0
	nviright_reqd1	SNVT_lux	Required Lux level for nviright1	300
	nviright_reqd2	SNVT_lux	Required Lux level for nviright2	300
	nviright_reqd3	SNVT_lux	Required Lux level for nviright3	300
	nviright_reqd4	SNVT_lux	Required Lux level for nviright4	300
	nvileft_reqd1	SNVT_lux	Required Lux level for nvileft1	400
	nvileft_reqd2	SNVT_lux	Required Lux level for nvileft2	400
	nvileft_reqd3	SNVT_lux	Required Lux level for nvileft3	300
	nvileft_reqd4	SNVT_lux	Required Lux level for nvileft4	300
	nviswitch1	SNVT_switch		100.0 1
	nviswitch2	SNVT_switch		100.0 1
	nviswitch3	SNVT_switch		100.0 1
	nviswitch4	SNVT_switch		100.0 1
	nviswitch_core	SNVT_switch		100.0 1
	nvitime1	SNVT_switch		100.0 1
	nvitime2	SNVT_switch		100.0 1
	nvitime3	SNVT_switch		100.0 1
	nvitime4	SNVT_switch		100.0 1
	nvitime_core	SNVT_switch		100.0 1
	nvichain_core	SNVT_switch		100.0 1
	run_lumen	SNVT_switch		100.0 0
	nvifunc	????		0|0|0|0|0
	maintime	????		0|30
	dimtime	????	Time all lights will stay dimmed at 0% before turning off	0|60
	exttime	????	Time the lights will stay on for after a light switch has been switched on	60|0
	path_loss	????	Variable that can be adjusted to calibrate the value of photo_new to a reading taken with a lux meter on the working plane directly below the sensor	1|0
	lux_to_level	????		1|0
	sample_time	????		7|208
	calibrate	????		1|44
	photo_pulse	????		39|140
	photo_allow	SNVT_lux		1000
	nviphoto_windowR	SNVT_lux	Input from angle sensor	0
	nviphoto_windowL	SNVT_lux	Input from angle sensor	0

OUTPUTS

	nvoright1	SNVT_switch		0.0 0
	nvoright2	SNVT_switch		0.0 0
	nvoright3	SNVT_switch		0.0 0
	nvoright4	SNVT_switch		0.0 0
	nvoleft1	SNVT_switch		0.0 0
	nvoleft2	SNVT_switch		0.0 0
	nvoleft3	SNVT_switch		0.0 0

	nvoleft4	SNVT_switch		0.0 0
	nvofunc	????		0|0|0|0|0
	nvochain_core	SNVT_switch		100.0 1
	run_complete	SNVT_switch		100.0 1
	nvoStatus	SNVT_obj_status	Object status - reports the status for any object on a node (15 fields, 6 used, 9 unused)	0 0,0,0,0,0,0,0,0,0, 0,0,0,0, 0,0,0,0,0,0,0,
	photo_new	SNVT_lux	Light level directly below the sensor on the working plane (used for adjustment)	149
	photo_level	SNVT_lux		149

The Angle Sensor

OBJ	Name	SNVTs	Description of Functionality	Value

INPUTS

	nviRequest	SNVT_obj_request	Object Request - allows an object to be be placed in one of several functional modes (normal, update-status, disable, report_mask).	0,RQ_NORMAL
	nvi020ValueFb	SNVT_switch		0.0 0
	nvi021ValueFb	SNVT_switch		0.0 0
	nvi022ValueFb	SNVT_switch		0.0 0
	nvi023ValueFb	SNVT_switch		0.0 0
	nvi024ValueFb	SNVT_switch		0.0 0
	nvi025ValueFb	SNVT_switch		0.0 0
	nvi026ValueFb	SNVT_switch		0.0 0
	nvi027ValueFb	SNVT_switch		0.0 0
	nvi028ValueFb	SNVT_switch		0.0 0
	nvi029ValueFb	SNVT_switch		0.0 0
	nvi02AValueFb	SNVT_switch		0.0 0
	nvi02BValueFb	SNVT_switch		0.0 0
	nvi02CValueFb	SNVT_switch		0.0 0
	nvi02DValueFb	SNVT_switch		0.0 0
	nvi02EValueFb	SNVT_switch		0.0 0
	nvi02FValueFb	SNVT_switch		0.0 0
	filter_count	????		0
	calibrate	????		1|44
	photo_pulse	????		144|136
	path_lossA	????	Variable that can be adjusted to calibrate the value of photo_windowA to a reading taken with a lux meter on the working plane near window B the sensor	1|0
	sample_time	????		39|116
	path_lossB	????	Variable that can be adjusted to calibrate the value of photo_windowB to a reading taken with a lux meter on the working plane near window B	1|0
	nvifunc	????		0|0|0|0|0

OUTPUTS

	nvo020Value	SNVT_switch		100.0 0
	photo_windowA	SNVT_lux	Light level from window A	926
	photo_windowB	SNVT_lux	Light level from window A	926
	photo_level	SNVT_lux		926
	photo_new	SNVT_lux		681
	nvoStatus	SNVT_obj_status	Object status - reports the status for any object on a node (15 fields, 6 used, 9 unused)	0 0,0,0,0,0,0,0,0,0, 0,0,0,0, 0,0,0,0,0,0,0

The IR Receiver

OBJ	Name	SNVTs / *SCPTs*	Description of Functionality	Value
INPUTS				
	nviRequest	SNVT_obj_request	Object Request - allows an object to be be placed in one of several functional modes (normal, update-status, disable, report_mask).	0,RQ_NORMAL
	nvifunc			0|0|0|0|0
	channel_code			0
	nvi010chan	UNVT_7631IRCode / *SCP_Type2*	Channel that this object will respond to	0
	nvi011chan	UNVT_7631IRCode / *SCP_Type2*	Channel that this object will respond to	0
	nvi012chan	UNVT_7631IRCode / *SCP_Type2*	Channel that this object will respond to	0
	nvi013chan	UNVT_7631IRCode / *SCP_Type2*	Channel that this object will respond to	0
	nvi014chan	UNVT_7631IRCode / *SCP_Type2*	Channel that this object will respond to	0
	nvi015chan	UNVT_7631IRCode / *SCP_Type2*	Channel that this object will respond to	0
	nvi016chan	UUNVT_7631IRCode / *SCP_Type2*	Channel that this object will respond to	0
	nvi017chan	UNVT_7631IRCode / *SCP_Type2*	Channel that this object will respond to	0
	nvi020chan	UNVT_7631IRCode / *SCP_Type2*	Channel that this object will respond to	0
	nvi030chan	UNVT_7631IRCode / *SCP_Type2*	Channel that this object will respond to	0
	nvi02Preset0	UNVT_7631Preset / *SCP_Type3???*	Allows the selection of the preset lamp values when the IR switch object is configured to Toggle/Level Preset. Each level corresponds to the 8 buttons on the handset	0.0 1
	nvi02Preset1	UNVT_7631Preset / *SCP_Type3???*	Allows the selection of the preset lamp values when the IR switch object is configured to Toggle/Level Preset. Each level corresponds to the 8 buttons on the handset	12.5 1
	nvi02Preset2	UNVT_7631Preset / *SCP_Type3???*	Allows the selection of the preset lamp values when the IR switch object is configured to Toggle/Level Preset. Each level corresponds to the 8 buttons on the handset	25.0 1
	nvi02Preset3	UNVT_7631Preset / *SCP_Type3???*	Allows the selection of the preset lamp values when the IR switch object is configured to Toggle/Level Preset. Each level corresponds to the 8 buttons on the handset	37.5 1
	nvi02Preset4	UNVT_7631Preset / *SCP_Type3???*	Allows the selection of the preset lamp values when the IR switch object is configured to Toggle/Level Preset. Each level corresponds to the 8 buttons on the handset	50.0 1
	nvi02Preset5	UNVT_7631Preset / *SCP_Type3???*	Allows the selection of the preset lamp values when the IR switch object is configured to Toggle/Level Preset. Each level corresponds to the 8 buttons on the handset	62.5 1
	nvi02Preset6	UNVT_7631Preset / *SCP_Type3???*	Allows the selection of the preset lamp values when the IR switch object is configured to Toggle/Level Preset. Each level corresponds to the 8 buttons on the handset	75.0 1
	nvi02Preset7	UNVT_7631Preset / *SCP_Type3???*	Allows the selection of the preset lamp values when the IR switch object is configured to Toggle/Level Preset. Each level corresponds to the 8 buttons on the handset	100.0 1
	nvi03Preset0	UNVT_7631Preset / *SCP_Type3???*	Allows the selection of the preset lamp values when the IR switch object is configured to Toggle/Level Preset. Each level corresponds to the 8 buttons on the handset	0.0 1
	nvi03Preset1	UNVT_7631Preset / *SCP_Type3???*	Allows the selection of the preset lamp values when the IR switch object is configured to Toggle/Level Preset. Each level corresponds to the 8 buttons on the handset	12.5 1

	nvi03Preset2	UNVT_7631Preset / *SCP_Type3???*	Allows the selection of the preset lamp values when the IR switch object is configured to Toggle/Level Preset. Each level corresponds to the 8 buttons on the handset	25.0 1
	nvi03Preset3	UNVT_7631Preset / *SCP_Type3???*	Allows the selection of the preset lamp values when the IR switch object is configured to Toggle/Level Preset. Each level corresponds to the 8 buttons on the handset	37.5 1
	nvi03Preset4	UNVT_7631Preset / *SCP_Type3???*	Allows the selection of the preset lamp values when the IR switch object is configured to Toggle/Level Preset. Each level corresponds to the 8 buttons on the handset	50.0 1
	nvi03Preset5	UNVT_7631Preset / *SCP_Type3???*	Allows the selection of the preset lamp values when the IR switch object is configured to Toggle/Level Preset. Each level corresponds to the 8 buttons on the handset	62.5 1
	nvi03Preset6	UNVT_7631Preset / *SCP_Type3???*	Allows the selection of the preset lamp values when the IR switch object is configured to Toggle/Level Preset. Each level corresponds to the 8 buttons on the handset	75.0 1
	nvi03Preset7	UNVT_7631Preset / *SCP_Type3???*	Allows the selection of the preset lamp values when the IR switch object is configured to Toggle/Level Preset. Each level corresponds to the 8 buttons on the handset	100.0 1
	nvi010ValueFb	SNVT_switch	Used to modulate the output from the infra-red switch object	100.0 1
	nvi011ValueFb	SNVT_switch	Used to modulate the output from the infra-red switch object	100.0 1
	nvi012ValueFb	SNVT_switch	Used to modulate the output from the infra-red switch object	100.0 1
	nvi013ValueFb	SNVT_switch	Used to modulate the output from the infra-red switch object	100.0 1
	nvi014ValueFb	SNVT_switch	Used to modulate the output from the infra-red switch object	100.0 1
	nvi015ValueFb	SNVT_switch	Used to modulate the output from the infra-red switch object	100.0 1
	nvi016ValueFb	SNVT_switch	Used to modulate the output from the infra-red switch object	100.0 1
	nvi017ValueFb	SNVT_switch	Used to modulate the output from the infra-red switch object	100.0 1
	nvi020ValueFb	SNVT_switch	Used to modulate the output from the infra-red switch object	100.0 1
	nvi030ValueFb	SNVT_switch	Used to modulate the output from the infra-red switch object	100.0 1

OUTPUTS

	nvoStatus	SNVT_obj_status	Object status - reports the status for any object on a node (15 fields, 6 used, 9 unused)	0 0,0,0,0,0,0,0,0,0, 0,0,0,0, 0,0,0,0,0,0,0
	nvofunc			0|0|0|0|0
	nvo010Value	SNVT_switch	Control output for the IR switch object	100.0 0
	nvo011Value	SNVT_switch	Control output for the IR switch object	100.0 0
	nvo012Value	SNVT_switch	Control output for the IR switch object	100.0 0
	nvo013Value	SNVT_switch	Control output for the IR switch object	100.0 0
	nvo014Value	SNVT_switch	Control output for the IR switch object	100.0 0
	nvo015Value	SNVT_switch	Control output for the IR switch object	100.0 0
	nvo016Value	SNVT_switch	Control output for the IR switch object	100.0 0
	nvo017Value	SNVT_switch	Control output for the IR switch object	100.0 0
	nvo020Value	SNVT_switch	Control output for the IR switch object	100.0 0
	nvo030Value	SNVT_switch	Control output for the IR switch object	100.0 0

The Simple Wind Speed Sensor

OBJ.	Name	SNVTs / *SCPTs*	Description of Functionality	Value
INPUTS				
	nviInstallDel	SNVT_lev_disc	LED Installation Input - sets the state of the LED of the wind sensor interface to be on or off during the process of intuitive installation	ST_OFF
	nciWdMinRange	SNVT_speed	High Limit of Transmitted Value - specifies the low limit of the transmitted values (linked to the type of anemometer) DON'T CHANGE	2.8
	nciWdSendOnDelta	SNVT_speed	Wind Speed send on delta - specifies the relative variation of m/s to take into account before updating nvoWind	1.4
	nciWdMaxSendTime	SNVT_elapsed_tm	Wind Speed max send time - specifies the maximum period of time that expires before the object automatically transmits the current value of nvoWind	0 0:1:0:0
	nciWdMinSendTime	SNVT_elapsed_tm	Wind Speed min send time - specifies the minimum period of time between nvoWind transmissions	0 0:0:1:0
	nciWdAverage	SNVT_count	Filtering on Measurements - specifies the filtering ratio of measurements versus transmission (linked to the type of anemometer) DON'T CHANGE	1
	nciWdThresTable	SNVT_speed	Threshold Table Value - value specified for test mode and generates the transmission of nviTestMode set to test	2.8
	nciWdThresTable	SNVT_speed	Threshold Table Value - set of 16 threshold values that can be manually set on the device	2.8
	nciWdThresTable	SNVT_speed	Threshold Table Value - set of 16 threshold values that can be manually set on the device	4.1
	nciWdThresTable	SNVT_speed	Threshold Table Value - set of 16 threshold values that can be manually set on the device	5.6
	nciWdThresTable	SNVT_speed	Threshold Table Value - set of 16 threshold values that can be manually set on the device	6.9
	nciWdThresTable	SNVT_speed	Threshold Table Value - set of 16 threshold values that can be manually set on the device	8.3
	nciWdThresTable	SNVT_speed	Threshold Table Value - set of 16 threshold values that can be manually set on the device	9.7
	nciWdThresTable	SNVT_speed	Threshold Table Value - set of 16 threshold values that can be manually set on the device	11.1
	nciWdThresTable	SNVT_speed	Threshold Table Value - set of 16 threshold values that can be manually set on the device	12.5
	nciWdThresTable	SNVT_speed	Threshold Table Value - set of 16 threshold values that can be manually set on the device	13.9
	nciWdThresTable	SNVT_speed	Threshold Table Value - set of 16 threshold values that can be manually set on the device	15.3
	nciWdThresTable	SNVT_speed	Threshold Table Value - set of 16 threshold values that can be manually set on the device	16.7
	nciWdThresTable	SNVT_speed	Threshold Table Value - set of 16 threshold values that can be manually set on the device	18.1
	nciWdThresTable	SNVT_speed	Threshold Table Value - set of 16 threshold values that can be manually set on the device	19.4
	nciWdThresTable	SNVT_speed	Threshold Table Value - set of 16 threshold values that can be manually set on the device	20.8
	nciWdThresTable	SNVT_speed	Threshold Table Value - set of 16 threshold values that can be manually set on the device	22.2
	nciConfigSource	SNVT_config_src / *SCPT_nwrk_cnfg*	Network configuration source - allows nodes to either supprot self-installation or to be installed externally	CFG_EXTERNAL
	nviRequest	SNVT_obj_request	Object Request - allows an object to be be placed in one of several functional modes (normal, update-status, disable, report_mask).	0,RQ_NORMAL

	nciThresMaxSdT	SNVT_elapsed_tm	Threshold Max Send Time - used to specify the period of time that expires before the object automatically transmits the current setting of nvoWdThreshold	0 1:0:0:0
	nciTestMdeMaxSdT	SNVT_elapsed_tm	Test Mode Max Send Time - used to specify the period of time that expires before the object automatically transmits the value of nvoTestMode when it is set to test mode	0 0:0:10:0
	nciStsMaxSendT	SNVT_elapsed_tm / *SCPT_max_snd_t*	Status Max Send Time - used to set the maximum period of time that expires before the object automatically transmits the current value of the nvoStatus output variable	0 0:0:0:0
	nciWdGain	SNVT_muldiv	Gain of the Sensor - sets the gain of the sensor (linked to the type of anemometer)	16/9
	nciWdOffset	SNVT_speed	Offset of the sensor - sets the offset of the sensor (linked to the type of anemometer)	0.7
	nciWdMaxRange	SNVT_speed	High Limit of Transmitted Value - specifies the high limit of the transmitted values (linked to the type of anemometer) DON'T CHANGE	22.2

OUTPUTS

	nvoInstallDel	SNVT_lev_disc	LED Installation Input - sets the state of the LED of the device the wind sensor interface is bound to during the installation process	ST_OFF
	nvoStatus	SNVT_obj_status	Object status - reports the status for any object on a node (15 fields, 4 used, 11 unused)	0 0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0
	nvoWind	SNVT_speed	Wind Speed - the wind speed level measured by the anemometer connected to the device	0
	nvoWdThreshold	SNVT_speed	Wind Speed Threshold - the wind speed threshold set on the node by the hardware selector	15.3
	nvoTestMode	????????	Functioning Mode Setting - sets the functioning mode of nodes it is bound to, to be a test mode or a normal one	0|0|0|0|0|0|0|0

Outdoor Temperature and Humidity Sensor

OBJ	Name	SNVTs / *SCPTs*	Description of Functionality	Value

INPUTS

	nciOffsetT	SNVT_temp_p	Temp Offset - sets the offset of the sensor (for calibration purposes) DON'T CHANGE	0
	nciOffsetRH	SNVT_lev_percent	RH Offset - sets the offset of the sensor (for calibration purposes) DON'T CHANGE	0.000
	nciMinSendT	SNVT_time_sec3	Temp min send time - specifies the minimum period of time (secs) between nvoHVACTemp transmissions	10.0
	nciMinDeltaRH	SNVT_lev_percent	RH send on delta - specifies the relative variation of RH to take into account before updating nvoHVACRelHum	0.000
	nciMinDeltaT	SNVT_temp_p	Temp send on delta - specifies the relative variation of Temp to take into account before updating nvoHVACTemp	0.00
	nciMaxSendT	SNVT_time_sec3	Temp max send time - specifies the maximum period of time (secs) that expires before the object automatically transmits the current value of nvoHVACTemp	30

OUTPUTS

	nvoHVACRelHum	SNVT_lev_percent	Relative Humidity Value - the external relative humidity measured by the sensor	50.150 (variable)
	nvoHVACTemp	SNVT_temp_p	Temperature Value - the external temperature measured by the sensor	10.19 (variable)

Indoor Temperature and Humidity Sensor

OBJ	Name	SNVTs / *SCPTs*	Description of Functionality	Value
INPUTS				
Te	nciMaxSendT	SNVT_temp_p / *SCPTmaxSendTime*	Temp max send time - specifies the maximum period of time that expires before the object automatically transmits the current value of nvoHVACTemp	30
Te	nciMinDeltaT	SNVT_lev_percent / *SCPTsndDelta*	Temp send on delta - specifies the relative variation of Temp to take into account before updating nvoHVACTemp	0.00
Te	nciMinSendT	SNVT_time_sec3 / *SCPTminSendTime*	Temp min send time - specifies the minimum period of time between nvoHVACTemp transmissions	10.0
Rh	nciMaxSendRH	SNVT_lev_percent / *SCPTmaxSendTime*	RH max send time - specifies the maximum period of time that expires before the object automatically transmits the current value of nvoHVACRelHum	30
Rh	nciMinDeltaRH	SNVT_temp_p / *SCPTsndDelta*	RH send on delta - specifies the relative variation of RH to take into account before updating nvoHVACRelHum	10.0
Rh	nciOffsetRH	SNVT_time_sec3 / *SCPTminSendTime*	RH Offset - sets the offset of the sensor (for calibration purposes) DON'T CHANGE	0.000
OUTPUTS				
Rh	nvoHVACRelHum	SNVT_lev_percent	Relative Humidity Value - the internal relative humidity measured by the sensor	52
Te	nvoHVACTemp	SNVT_temp_p	Temperature Value - the internal temperature measured by the sensor	22.3

Wind Speed and Direction Sensor

OBJ	Name	SNVTs / *SCPTs*	Description of Functionality	Value
INPUTS				
Ws		SNVT_time_sec / *SCPTmaxSendTime*	Wind Speed max send time - specifies the maximum period of time that expires before the object automatically transmits the current value of nvoWindSpeed	30.0
Ws		SNVT_speed / *SCPTsndDelta*	Wind Speed send on delta - specifies the relative variation of wind speed to take into account before updating nvoWindSpeed	0.100
Ws		SNVT_time_sec / *SCPTminSendTime*	Wind Speed min send time - specifies the minimum period of time between nvoWindSpeed transmissions	1.0
Wd		SNVT_time_sec / *SCPTmaxSendTime*	Wind Direction max send time - specifies the maximum period of time that expires before the object automatically transmits the current value of nvoWindDirection	30.0
Wd		SNVT_angle_deg / *SCPTsndDelta*	Wind Direction send on delta - specifies the relative variation of wind direction to take into account before updating nvoWindDirection	45.00
Wd		SNVT_time_sec / *SCPTminSendTime*	Wind Direction min send time - specifies the minimum period of time between nvoWindDirection transmissions	1.0
Fb		*UCP_Type_0*		00
OUTPUTS				
Ws	nvoWindSpeed	SNVT_speed	Wind Speed Value - the wind speed measured by the sensor	5.4
Ws	nvoWindSpeedmil	SNVT_speed_mil	Wind Speed Value - the wind speed measured by the sensor (to 3 decimal places)	4.479
Wd	nvoWindDirection	SNVT_angle_deg	Wind Direction Value - the wind direction measured by the sensor	0.00
Fb	nvoVersionNo	SNVT_count		257

The LonMaker Drawing

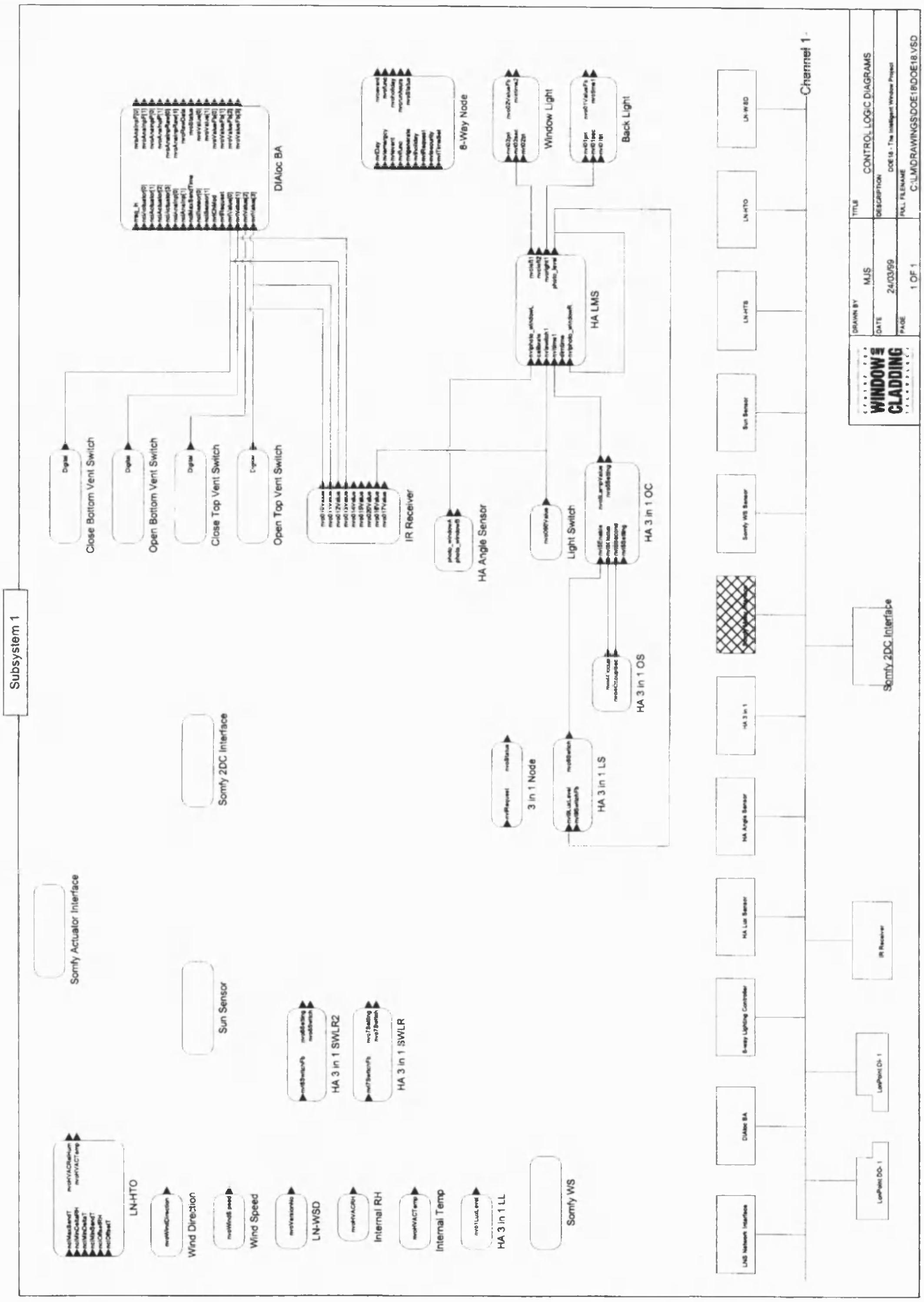


Figure VII-VI: The LonMaker drawing showing the functional links that were maintained at the end of the test room study. Any unlinked items were linked using the Visual Basic code.

---

[1] Paulme C., 1999, Functional Profile BIU-2DC Lon 2, Version 2, Ref 157809, Somfy, Fr.

[2] LonMark Application Layer Interoperability Guidelines, Version 3.0, LonMark Interoperability Association, Palo Alto, CA, 1996. 078-0120-01C.

[3] The SNVT Master List and Programmer's Guide, Document No. 005-0027-01, Echelon Corporation, 4015 Miranda Avenue, Palo Alto, California, May 1997.

Appendix VIII Errors Associated with Automated Blind Control within the Test Room Study

Introduction

Chapter Six reviewed the findings of a study into the practical constraints associated with the automated blind control systems. This appendix supplements the findings outlined in Chapter Six by reviewing the process undertaken to quantify the acceptable ranges of error for different parts of the system. It finishes by using its findings to select a solar positioning algorithm from Appendix IV for use in the visual basic algorithm developed in Chapter Seven.

A Review of the Sun Blocking Blind Control Algorithm Errors

In order for the control system to perform adequately, the algorithms used to calculate the position of the sun, and the resulting position of the blind, must be fit for the purpose at hand. In this study, the purpose of the control system was to alter the blind slat angle in order to prevent direct sunlight from entering the space, whilst also providing maximum views out (see Chapter Seven).

Figure VIII - I illustrates a break down of all the processes involved in controlling blind angle to block the sun and the possible errors associated with each stage. This appendix investigates each of these processes and the errors associated with them and starts at the bottom of Figure VIII – I with the blind positioning errors so as to work it's way up to tie in with the solar positioning algorithms reviewed in Appendix IV.

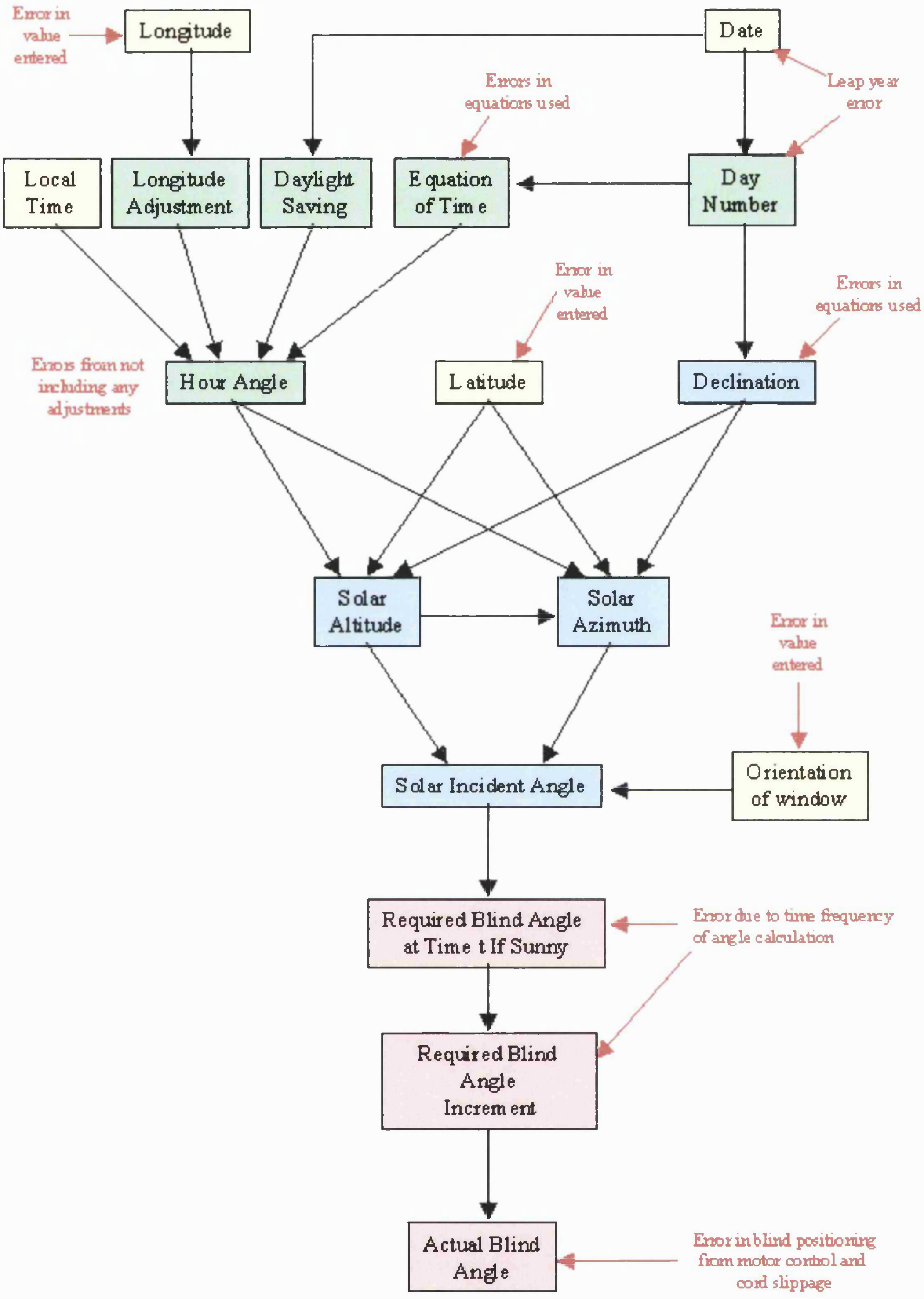


Figure VIII - 1: Flow diagram illustrating the processes required to control blind tilt angle so as to block direct sunlight and the errors associated with each stage.

Blind Positioning Errors

The Blind Motor and Pick Up Mechanism

The blind motor system used in this study incorporated a tubular motor that motorised both cord lift and tilt. Both of these functions were considered important when designing an energy efficient and comfortable venetian blind system and this system was the only system available that allowed the automation of both functions for the mini blind chosen. A description of the blind control mechanism and some photographs can be found in Chapter Five.

The end of the pick up system that collected the lift cord had a rotating grove that created enough friction with the tilt cord to alter blind tilt angle when the blind was between the end limits of its tilting movement. When the had blind reached the end limits of tilting motion the blind simply raised or lowered, depending on the motor direction.

By incorporating a dual speed motor, it was possible to obtain greater control over the dual functions of the pick up system. The motor rotated at a slow speed for a certain period of time at the beginning of it's movement, when the tilting action was likely, and then at a much higher speed after a certain period, when the motor was raising or lowering the blind. The total tilting time for the mechanism when running at the lower speed was around four seconds. This made it very difficult to control the blind position in any less that 22.5degree increments (half second increments).

Errors in Blind Motion and Positioning

The nature of the tilting mechanism meant that there was irregular slippage and cord movements at the start of the friction mechanism motion that made positioning the blind from angle to angle very difficult and accumulative inaccuracies often unacceptable. This, and the fact that there was no control feedback, meant that these inherent errors and control issues required consideration before algorithms could be created to relate motor run time to blind position.

The errors in blind motion were measured by making continuous 22.5-degree incremental adjustments to the blind throughout its movement using the network management tool. For each movement the resultant angle of the blind was measured using the technique illustrated in Figure VIII – II. The method was repeated until 1000 measurements had been taken.

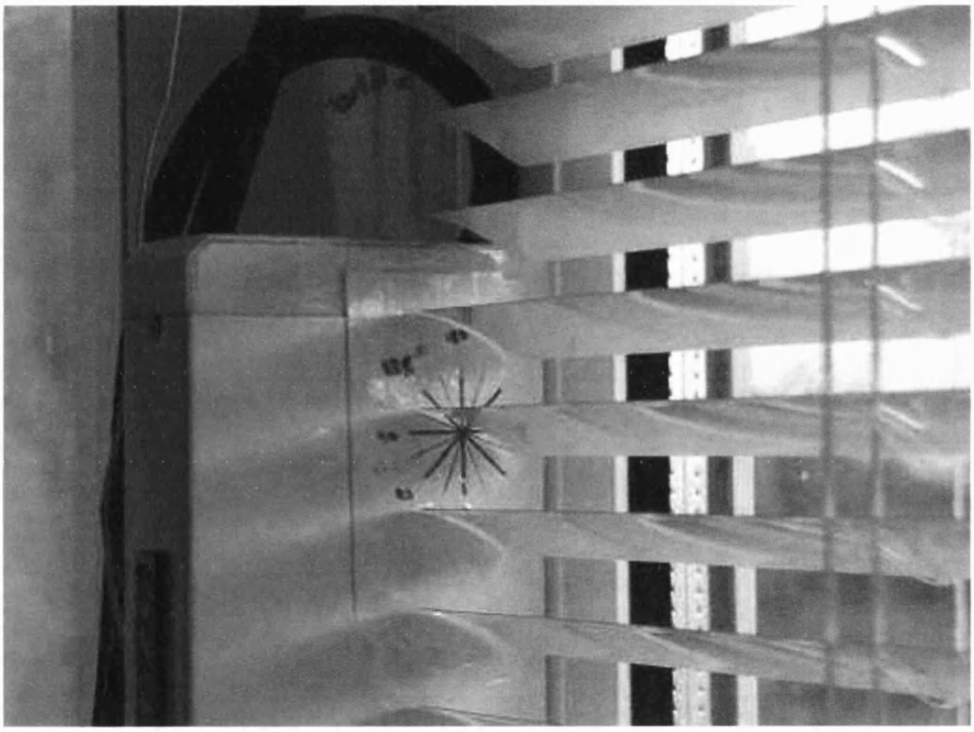


Figure VIII - II: Method of measuring blind position.

Once the blind configuration properties had been commissioned satisfactorily, the results became fairly consistent, due to the fact that once the blind had reached the end of its travel in one direction it effectively reset itself. The graph in Figure VIII - III shows the average errors associated with the manufacturer's blind positioning algorithm.

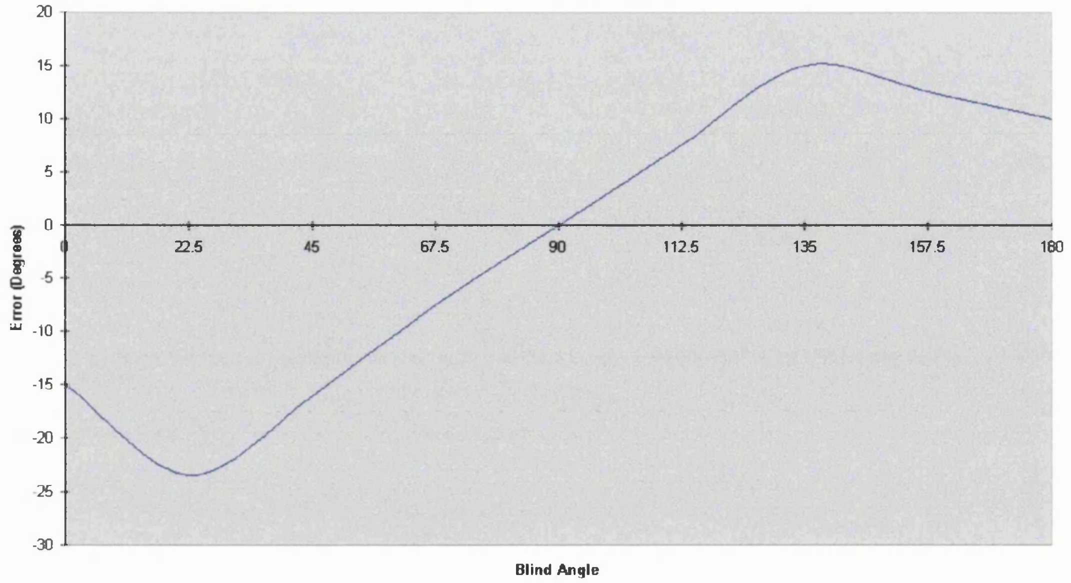


Figure VIII - III: Graph showing the average errors associated with positioning blind in 22.5 Degree increments using the manufacturer's positioning control.

From the graph, we can see that the blind could be positioned fairly accurately at the horizontal angle of 90 Degree, but errors were associated with the positioning the blind either side of horizontal. At the end of the blind tilt range (0 and 180 degrees), the errors were due to the fact that the blind was curved and the closed position was about 15 degrees off vertical. Therefore these errors can be effectively discounted.

Overcoming the blind positioning errors

The blind motor manufacturer's solution to the problem of accumulative blind positioning error was to close the blind, in order to reset the system, every time an adjustment was needed. However, this operation was quite distracting, especially when considering the fact that the lighting system reacted to light levels from the window. Therefore it was decided to by-pass this in built functionality by incorporating a blind positioning algorithm within the visual basic program used to control the test room blind (see Chapter Seven). This algorithm also used a timer to position the blind but it reset itself only twice a day, during periods when the occupant was out of the room.

The line between 22.5 degrees and 135 degrees in Figure VIII – III shows that there could be possible improvements to the manufacturer's blind positioning algorithm to take into account the slight non-linearity of the blind tilting action resulting from the tilt cords being at the edge of the slats. However, rather than make these adjustments to the algorithm, which may not be applicable to other blind mechanisms, it was decided to incorporate these errors into the overall blind positioning algorithm using a methodology that was easily adaptable for systems of all levels of accuracy.

The methodology used was to alter the slat ratio within the blind tilting algorithm outlined in Appendix IV to ensure that any errors in blind positioning did not result in the sun penetrating the blind. To illustrate this, Figure VIII – IV plots the expected and actual blind positions for a blind positioning controller using a 1:1 ratio of slat width to slat spacing onto a graph similar to that shown in Figure IV – XIII in Appendix IV.

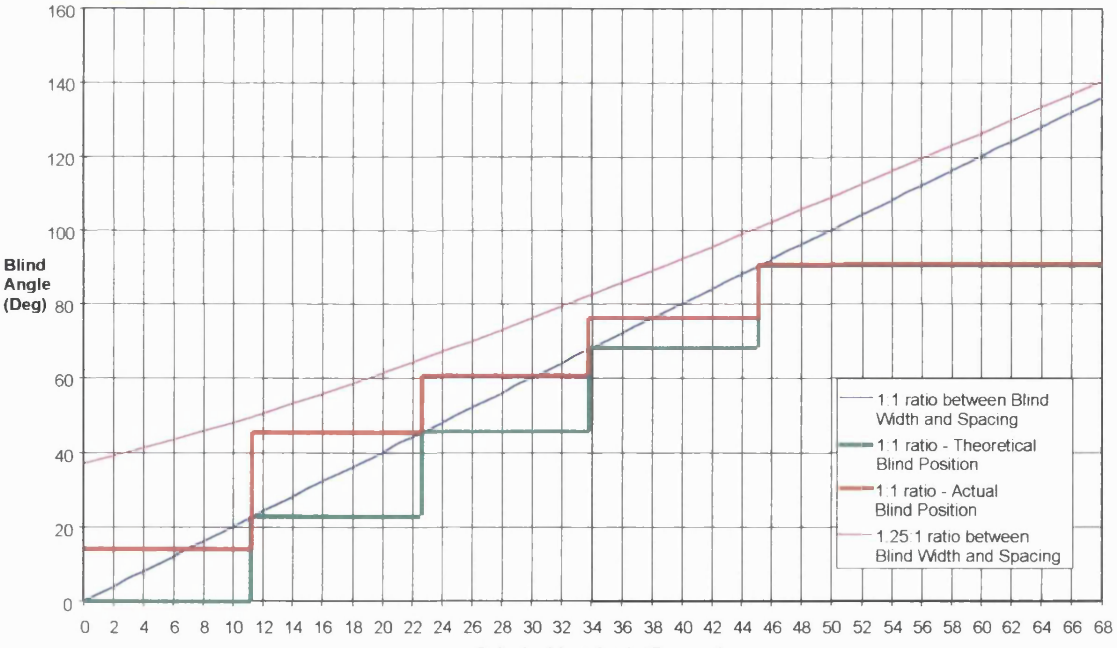


Figure VIII - IV: Graph showing the theoretical and actual blind positions when controlling the blind in 22.5 degree increments for a 1:1 ratio blind controller

The blind moves in 22.5 Degree increments and the inaccuracies in the blind positioning cause the actual blind position line to be above the 1:1 ratio line, thus resulting in unacceptable periods of solar penetration through a true 1:1 ratio blind. However, if we use the 1:1 ratio algorithm to control the 1:1.25 ratio blinds we can see that the line falls below the line 1:1.25 line and thus we can build the blind positioning error into the algorithm and still block the sun.

The additional advantage of using the 1:1 ratio for this particular system was that we were able to simplify the blind angle [θ_b] formula from:

$$\theta_b = \theta_s + 90 - \sin^{-1}\left(\frac{h}{w}\sin(90 - \theta_s)\right) \qquad \text{[Degrees]}$$

to:

$$\theta_b = 2\theta_s \qquad \text{[Degrees]}$$

By simplifying the formula in this way we would be able to make dramatic savings on computational resources required from a Neuron chip compared to the full formula (see Appendix IV).

To ensure that views out were maximised, a rule had to be added to both these algorithms to ensure that they were only executed when the solar wall incident angle was below 45 degrees. This was because at angles above that threshold the slats could stay in the horizontal position.

Allowable error in the solar incident angle calculation

From Figure VIII - IV we can determine the allowable error in solar incident angle calculation associated with each blind increment change. These errors are outlined in Table VIII - I.

Blind Transition	Allowable Error in Solar Incident Angle [Degrees]
90.0 - 67.5	6
67.5 – 45.5	4
45.0 – 22.5	4
22.5 – 0.0	3

Table VIII - I: Allowable errors in solar incident angle calculation related to the transition between slat angles.

From this table we can see that, to obtain an adequate performance from our control strategy in the worst case condition, we can only allow an error in solar incident angle of 3 degrees.

Open Loop Control Errors

The Effect of Calculation Timers on the Allowable Solar Incident Angle Error

An algorithm that angles the slats of a blind depending upon the position of the sun is inherently an open loop control algorithm, where the major inputs controlling the system are related to time. Such an algorithm could be designed to run in a continuous loop calculating the altitude and azimuth for each fraction of a second that passes. However, this would be computationally exhaustive, because it is unnecessary for the degree of control required within the system and because the Neuron chip, on which the application runs, may have other tasks to perform. Therefore it is necessary to define the time interval between each solar positioning assessment.

To understand how we might select such a time interval we must first understand the likely rate change in solar incident angle to ensure that solar penetration does not occur. This is especially important when the solar incident angle has passed its peak and is in its descent. If the time interval is too long and the rate of solar incident angle decrease is high, then sun may be allowed to penetrate through the blind before the algorithm can re-adjusts itself.

Quantifying the Open Loop Error

The worst case scenario for the UK is the west-facing window during the winter solstice, where the solar incident angle falls quickly in relation to the window. Figure VIII - V is a stereographic projection for this orientation illustrating the relationship between solar incident angle, altitude and relative azimuth for the window facing hemisphere and the sun path expected in the UK.

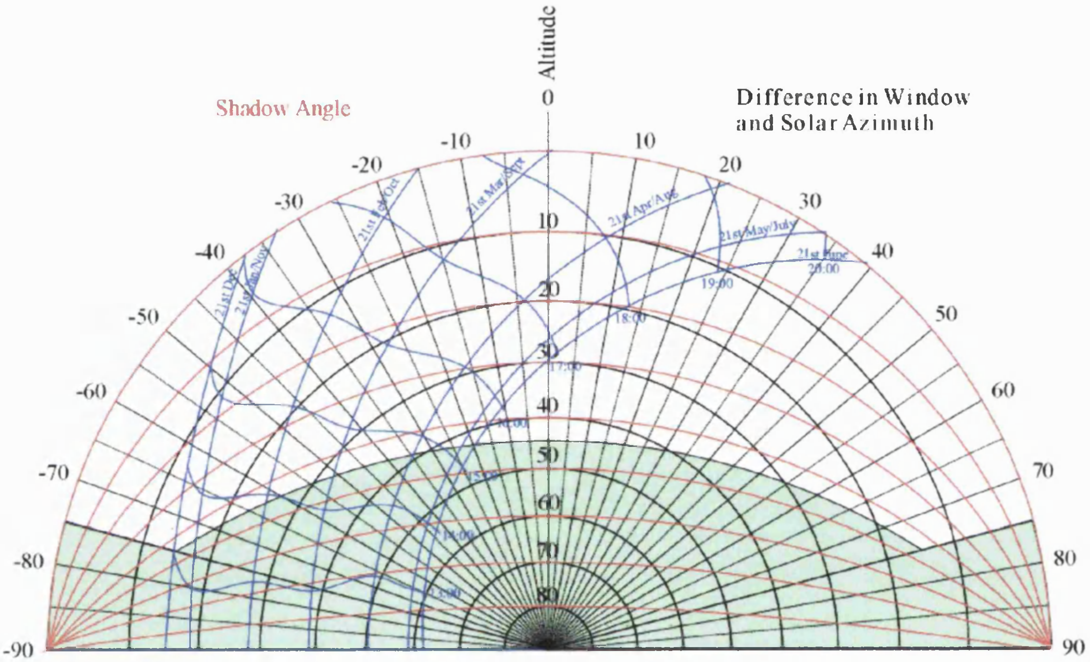


Figure VIII - V: Diagram showing the relationship between solar incident angle, solar altitude and the relative azimuth of the sun, with a sunpath diagram from a west facing window in Bath UK.

If we imagine moving the sunpath around on this projection we can see that the rate of change of solar incident angle would be greatest as the relative azimuth approaches 90 or –90 degrees. However as the solar position approaches these angles, we tend to find solar penetration becomes less of a problem as many buildings often have some form of reveal or mullion to obscure this part of the sky, and if not, occupants would have to be very close to the facade to be affected by direct sunlight. In addition, the solar movement would become so fast that any discomfort caused by solar penetration would be very short lived. As a result, the rest of this study concentrated on the design of algorithms to cater to the relative azimuth angles up to ±75 degrees.

Also if the solar incident angles is greater than 45 degrees then the blind stays at horizontal, therefore this study also did not considered the region above a 45 degrees solar incident angle. Both these exclusion zones are shown in green in Figure VIII - V.

If we plot the solar incident angles shown in Figure VIII – V for the winter solstice and on a two dimensional graph (shown in Figure VIII – VI), we can show that the largest gradient, in the pre-specified region, is at the point where blind needs to alter from 90 degrees (horizontal) to 67.5 degrees (where the line passes through the 45 degree mark). The rate of change of the solar incident angle at this point is 0.544 degrees per minute. By plotting on the solar incident angles experienced at other latitudes we can show that this value is a reasonably good representation of maximum rate of change expected.

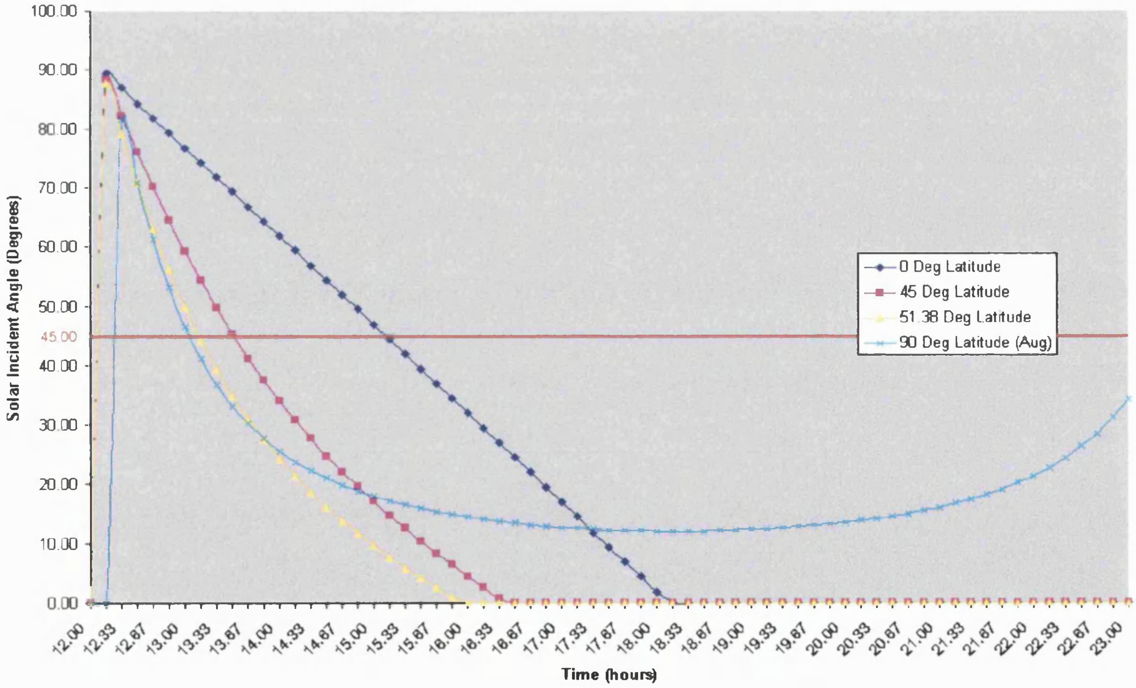


Figure VIII - VII: Graph showing the solar incident angle against time for various latitudes in December

At a peak rate of 0.544 degrees per minute, we can say that if the time interval between solar positioning calculations were 5 minutes, then the blind positioning would perform adequately assuming that the solar positioning algorithms were accurate to within 3.28 degrees solar incident angle (see Table VIII – II). This is because, in five minutes the solar incident angle could have changed by 2.72 degrees and the allowable error defined by the blind positioning algorithm studies for this blind angle was 6 degrees.

Adjusting the open loop interval time

The requirements set out by a five minute time interval were considered slightly too onerous for the solar positioning algorithm selection, as most Neuron chip applications were easily able to provide network variable updates on a minute by minute basis. Therefore it was decided to reduce the time interval to one minute. Table VIII – II compares the allowable incident angle error resulting from this change for each incremental change of blind angle. The Table shows that the minimum allowable wall solar incident angle error for a one minute interval is approximately 2.8 degrees.

Blind transition [Degree increments]	Allowable incident angle error before time effect taken into account [Degrees]	Rate of change of solar incident angle [Degrees /min]	Allowable incident error for 1 min time interval [Degrees]	Error for 5 minute time interval [Degrees]	Allowable incident error for 5 min time interval [Degrees]
90.0 – 67.5	6	0.544	5.456	2.72	3.28
67.5 – 45.0	4	0.394	3.606	1.97	2.03
45.0 – 22.5	4	0.291	3.709	1.45	2.55
22.5 – 0.0	3	0.202	2.798	1.01	1.99

Table VIII - II: The errors associated with the calculation of solar incident angle depending on various time interval for Bath UK (Latitude 51.36N) and the affect of the allowable solar incident angle error

It should be noted that by setting the time interval to 1 minute, the blind will not adjust every minute causing annoyance to occupants, because the 22.5 degree increment in the stepped blind angle response requires certain thresholds to be passed before the slat angle is adjusted. This meant that the blind adjustment was fairly infrequent, except on short occasions when the sun was falling perpendicular to the facade.

This point demonstrates how it is important to get the balance right between having small blind increments, with short adjustment time intervals, and large blind increments, with longer adjustment time intervals. Chapter Seven showed that there was little environmental advantage in reducing the blind increments to increase the control over views and daylight penetration. Therefore the right balance could be achieved at the largest blind increment movement considered unnoticeable by the occupant. In this case, this angle was 22.5 degrees.

The Acceptable Solar Positioning Algorithms to Use

The Effect of Solar Altitude and Azimuth Errors on the Solar Incident Angle

By looking at the stereographic projection in Figure VIII – V, we can see that there is a non-linear relationship between solar altitude and azimuth errors and solar incident angle errors. If we look at the worst case scenario for the zone we are interested in, i.e. on the 75 degree relative azimuth line, we can get some idea of the degree of accuracy required from our solar positioning algorithms (See Table VIII - III). This accuracy largely depends on the procedures used for determining the Hour Angle and the Solar Declination outlined in Appendix IV.

Allowable Solar Incident Angle Error [Degrees]	Allowable Solar Altitude Error [Degrees]	Allowable Solar Azimuth Error [Degrees]
2.5	0.75	1.0

Table VIII - III: Allowable Errors in Solar Altitude and Azimuth derived from the stereographic projection

Determining the Correct Solar Positioning Algorithms to Use

We saw in Appendix IV that the errors in the Hour angle had the greatest effect on the accuracy of the solar positioning algorithms. Using the values in Table VIII – III, we can see from Figures IV – IV and IV – V in Appendix IV that the maximum allowable error in True Solar Time calculation was approximately 5 minutes, an Hour Angle of 1.25 Degrees. This means that it was very important that the algorithm used included the corrections for daylight saving, longitude and Equation of Time within the true solar time calculation (daylight saving only required if the timer used uses a local clock, such as a scheduler).

We can also see from Table IV-I in Appendix IV that it is possible to use any of the EOT algorithms other than the low accuracy Whillier equation. It was therefore decided that the IES formula for the equation of time offered the most benefits in terms of acceptable accuracy at a low computational complexity. This provided a potential hour angle error of 0.63 degrees that occurs at the beginning and end of the year.

Figures IV – VI and IV – VII in Appendix IV show that the maximum allowable error in solar declination calculation for a 0.75 degree error in solar altitude and a 1 degree error in solar azimuth is about 0.6 degrees. Therefore we can see from Table IV-II in Appendix IV that Goulding's equation for solar declination offered the most benefits in terms of acceptable accuracy for a low computational complexity. This gives a potential error in solar declination of 0.13 degrees, which would not occur at the same time of year as the EOT error.

Both these formulae were used when developing the Visual Basic control code for controlling the blind, outlined in Appendix IX.

It should be noted that if both the highly accurate Yallop formulae had been used for calculating the equation of time and the solar declination the total units of complexity (412 units) would only have been 143 units above the sum of chosen algorithms. Although it is difficult to predict whether that such a accurate formula would be able to be added to the limited space available on a automated blind controller's Neuron chip, such an algorithms could probably be added to devices with less functionality, such as the sun sensors. The values of altitude and azimuth could then be broadcast over the LonWorks network as network variables and the blind controller could process them as it sees fit.

Appendix IX The Visual Basic Control Code

Introduction

The appendix contains the Visual Basic code used in the test cell control system. The code provided visual feedback of the control actions as well as control decision making. To simplify the reading of the code it has been divided into the following sections that symbolise its functional modules:

- *Form 1 Loading Code* – defines and loads global variables in user interface form
- *Form 2 Loading Code* – defines and loads global variables in monitoring from
- *Reset System and Second Timer* – a second timer that times many operations
- *Automatic Control Timer* – a minute timer that controls many operations
- *Blind Sun Threshold Control* – determines whether pre-set threshold is passed
- *Solar Positioning* – determines the position of the sun
- *Automatic Blind Control Code* – positions blind to block sun
- *Blind State and Stop* – determines the blind position
- *Blind User Over-ride Control* – code used to translate user over-rides
- *Automatic Vent Control Code* – code used to control the vents
- *Wind Assessment* – assesses the wind conditions for the vent controller
- *Vent User Over-ride Control* – code used to translate user over-rides
- *The User Interface Graphics* – updates the graphical interface with changes

The code was created using Microsoft's Visual Basic Professional Version 5 and was used to control both the blinds reviewed in this thesis and the vents used in the CWCT study. All of the code has been included in this appendix and has been arranged so that the modules that largely refer to the vents are at the back. Any code not used for blind control is shown in grey. Notes have been added in green to make the code easier to follow.

Form 1 Loading Code

```
'This module is the one of the sets of code that is run when the program first starts.  It is based within the user interface form and declares and sets all of the global variables located within that form.

Public PositionTV As Integer 'Position of Top Vent in Percentage of Total Opening
Public PositionBV As Integer 'Position of Bottom Vent in Percentage of Total Opening

Public ButtonOpenTV As Boolean ' State of Relay For Opening Top Vent
Public ButtonCloseTV As Boolean ' State of Relay For Closing Top Vent
Public ButtonOpenBV As Boolean ' State of Relay For Opening Bottom Vent
Public ButtonCloseBV As Boolean ' State of Relay For Closing Bottom Vent

Dim RaiseBlind As Boolean ' State of the Relay for raising the blind
Dim LowerBlind As Boolean ' State of the Relay for raising the blind

Public LockingTV As Boolean 'States whether Top Vent is Locking or Unlocking
Public LockingBV As Boolean 'States whether Bottom Vent is Locking or Unlocking

Public Occupancy As Boolean 'States whether room occupied or not
Public IntTemp As Single 'Variable for internal temperature
Public ExtTemp As Single 'Variable for external temperature
Public SunThresholdState As Boolean

Option Explicit

Private Sub Form_Load()

  ' Ensure all the buttons start in the Off state
    LockingTV = False
    LockingBV = False
    ButtonOpenTV = False
    ButtonCloseTV = False
    ButtonOpenBV = False
    ButtonCloseBV = False
    RaiseBlind = False
    LowerBlind = False
    Form1.Occupancy = True 'set occupancy state as true to begin
    Form2.txtStatus_Occupancy = Form1.Occupancy 'display occupancy state in Form2 display box
    Form1.ExtTemp = Val(txtTemp_Ext.Text) 'get external temperature value form the DDE linked text box
```

```
    Form1.IntTemp = Val(txtTemp_Int.Text) 'get internal temperature value form the
DDE linked text box
    Load Form2     ' Loads form 2 into memory

End Sub
```

```
Private Sub Form_QueryUnload(Cancel As Integer, UnloadMode As Integer)

'checking code that is executed when the user tries to leave the program
  Dim X%
  X% = MsgBox("Are you sure you want to end the program?", vbYesNo)
  If X% = vbNo Then
    Cancel = True
  Else
    Unload Form2
    Unload Me
    End
  End If
End Sub
```

Form 2 Loading Code

```
'This module is the second set of code that is run when the program first starts.  It is
based within the monitoring and commissioning form and declares and sets all of the
global variables located within that form.

Public StatusLockTV As Boolean ' State of Locking Mechanism for Top Vent
Public StatusLockBV As Boolean ' State of Locking Mechanism for Bottom Vent
Public ResetState As Boolean 'Allows the program to assess whether it needs to go
into reset mode

Public Altitude As Single 'The altitude of the sun
Public Azimuth As Single 'The azimuth of the sun
Public Pi As Double 'The mathematical constant Pi

Public VentUserOveride As Boolean 'States whether an vent occupant overide mode
on
Public VentAdjustModeTV As Boolean 'States whether the top vent is adjusting or
not
Public VentAdjustModeBV As Boolean 'States whether the bottom vent is adjusting
or not
Public DesiredTV As Integer 'The desired top vent position
Public DesiredBV As Integer 'The desired bottom vent position
```

```
Dim OccupancyPeriod As Boolean 'States whether time within a predefined occupancy period
Dim WeekEndDay As Boolean 'States whether it is the weekend

Dim FirstSet As Boolean
Dim PreHeatPeriod As Boolean 'States whether night ventilation is in the preheat period
Dim AfternoonTemp As Single 'States the mean afternoon temperature between predefined hours
Dim TotalTemp As Single 'States the accumulative afternoon temperature for the mean calculation
Dim WindyConditions As Boolean 'States whether external conditions are windy or not

Public MoveUp As Boolean 'States whether blind is moving up
Public MoveDown As Boolean 'States whether blind is moving down
Dim RotationBlind As Single 'States the tilt angle of the blind
Dim PositionBlind As Single 'States the vertical position of the blind
Dim DesiredRotation As Single 'States the desired tilt angle of the blind
Dim BlindUserOveride As Boolean 'States whether the user override has got priority over automatic control
Dim BlindAdjustMode As Boolean 'States whether blind adjusting
Public SunDelay As Boolean 'States the controller is within the sun in time delay
Public SunOutDelay As Boolean 'States the controller is within the sun out time delay

Option Explicit

Private Sub Form_Load()

  Pi = 4 * Atn(1#) 'Defines Pi
'Set variables and display in the Form 2 windows
  Form2.VentUserOveride = False
  Form2.txtStatus_VentOR = Form2.VentUserOveride
  Form2.VentAdjustModeTV = False
  Form2.VentAdjustModeBV = False
  Form2.txtStatus_VentAdjustTV = Form2.VentAdjustModeTV
  Form2.txtStatus_VentAdjustBV = Form2.VentAdjustModeBV
  Form2.ResetState = True
  'ResetState = False
  txtTmr_ResetState = Form2.ResetState
  BlindUserOveride = True
  Form2.txtStatus_BlindOR = BlindUserOveride
  Call Form2.BlindControl("SET_UP", "0.0", "0.00")
  FirstSet = True
  AfternoonTemp = 19
  MoveDown = False
  txtState_MoveDown = MoveDown
  MoveUp = False
```

```
  txtState_MoveUp = MoveUp
  BlindAdjustMode = False
  PositionBlind = 0
  RotationBlind = 0

End Sub
```

Reset System and Second Timer

Private Sub cmdReset_Click()

```
'This code resets the system when a button is pressed on the visual basic interface
  Form2.ResetState = True
  txtTmr_ResetState = Form2.ResetState

End Sub
```

Private Sub tmrSecond_Timer()

```
'This is timer code that set to repeat every second
  Static Counter As Integer

  ' Procedure to calculate the position of the Bottom Vent
  Dim TimerOpenBV As Integer 'Duration (secs) of Open Bottom Vent Signal to nviValue[3]
  Dim TimerCloseBV As Integer 'Duration (secs) of Close Bottom Vent Signal to nviValue[2]
  Static TimerPositionBV As Integer ' Position of Bottom Vent along TimeLine Characteristic

  ' Procedure to calculate the position of the Top Vent
  Dim TimerOpenTV As Integer 'Duration (secs) of Open Top Vent Signal to nviValue[3]
  Dim TimerCloseTV As Integer 'Duration (secs) of Close Top Vent Signal to nviValue[2]
  Static TimerPositionTV As Integer 'Position of Top Vent along TimeLine Characteristic

'This code is the reset code that operates when the reset state becomes true
  If Form2.ResetState = True Then
    Form3.Show 'Form 3 was a visual counter that appeared to inform the user the system was resetting itself
    Form2.StatusLockTV = False 'ensures vents are in unlocked state
    Form2.StatusLockBV = False
    Counter = Counter + 1 'counts the seconds of reset
```

```
    Call Form1.CloseTVRelaySwitch("100.0", "1") 'calls the vent positioning function
    Call Form1.CloseBVRelaySwitch("100.0", "1")
    Call Form2.BlindControl("SET_UP", "0.0", "0.00") 'calls function to raise the blind
    If Counter = 27 Then 'counts to 27 secs and then stops reset
      Form2.StatusLockTV = True
      Form2.StatusLockBV = True
      txtStatus_LockTV = Form2.StatusLockTV
      txtStatus_LockBV = Form2.StatusLockBV
      Call Form1.CloseTVRelaySwitch("0.0", "0")
      Call Form1.CloseBVRelaySwitch("0.0", "0")
      Counter = 0 'resets counter
      Form2.ResetState = False
      txtTmr_ResetState = Form2.ResetState
      Call Form2.SolarPosition 'starts to re-initiate automatic control
      Call Form2.WorkHours
      If OccupancyPeriod = True Then Call Form2.OccVentControl
      BlindUserOveride = False
      Form2.txtStatus_BlindOR = BlindUserOveride
      Call Form1.SunThreshold
      Unload Form3
    End If
    Form3.ProgressBar1.Value = Counter Mod 27
    txtTmr_ResetTime.Text = Counter
    txtStatus_LockTV = Form2.StatusLockTV
    txtStatus_LockBV = Form2.StatusLockBV
  End If

'Routine that uses the second timer to only open and position the bottom vent when
not in the process of unlocking
    If Form2.txtState_OpenBtmVent = 1 And Form1.LockingBV = False Then
      Form2.StatusLockBV = False ' for switches and IR
      Form2.txtStatus_LockBV = Form2.StatusLockBV
      TimerOpenBV = TimerPositionBV
      TimerPositionBV = TimerOpenBV + 1
      Form2.txtTmr_PosBtmVent.Text = TimerPositionBV
   '  TimerOpenBV = 0
        If TimerPositionBV >= 25 Then
          TimerPositionBV = 25    ' sets maximum TimerPositionBV as 25
          Form2.txtTmr_PosBtmVent.Text = TimerPositionBV
          Form1.ButtonOpenBV = False    ' routine that turns of the relay
          Form2.VentAdjustModeBV = False
          Form2.txtStatus_VentAdjustBV = Form2.VentAdjustModeBV
          Call Form1.OpenBVRelaySwitch("0.0", "0")
        End If
    End If
```

```
' Routine that Unlocks the bottom vent
    If Form2.txtState_OpenBtmVent = 1 And Form1.LockingBV = True Then
      TimerOpenBV = TimerPositionBV
      TimerPositionBV = TimerOpenBV + 1
      Form2.txtTmr_PosBtmVent.Text = TimerPositionBV
   '  TimerOpenBV = 0
        If TimerPositionBV = 8 Then    ' sets maximum TimerPositionTV as 8
          Form1.LockingBV = False
          Call Form1.OpenBVRelaySwitch("0.0", "0") ' routine that turns off the relay
        End If
   End If

'Routine that only closes and positions the bottom vent when not locking
   If Form2.txtState_CloseBtmVent = 1 And Form1.LockingBV = False Then
      TimerCloseBV = TimerPositionBV
      TimerPositionBV = TimerCloseBV - 1
      Form2.txtTmr_PosBtmVent.Text = TimerPositionBV
    ' TimerCloseBV = 0
        If TimerPositionBV <= 8 And Form2.ResetState = False _
            And Form2.StatusLockBV = False Then 'needed to preserve zero after reset
'          TimerPositionBV = 8     ' sets minimum TimerPositionBV as 8
          Form1.ButtonCloseBV = False   ' routine that turns of the relay
          Form2.VentAdjustModeBV = False
          Form2.txtStatus_VentAdjustBV = Form2.VentAdjustModeBV
          Call Form1.CloseBVRelaySwitch("0.0", "0")
        Else
          If TimerPositionBV <= 0 Then
            TimerPositionBV = 0
            Form2.txtTmr_PosBtmVent.Text = TimerPositionBV
            If Form2.ResetState = False Then Call Form1.CloseBVRelaySwitch("0.0",
"0")
            'maintains TimerPositionBV minimum in reset mode
          End If
        End If
    End If

' Routine that locks the bottom vent
    If Form2.txtState_CloseBtmVent = 1 And Form1.LockingBV = True Then
      TimerCloseBV = TimerPositionBV
      TimerPositionBV = TimerCloseBV - 1
  '   TimerCloseBV = 0
      Form2.txtTmr_PosBtmVent.Text = TimerPositionBV
        If TimerPositionBV <= 0 Then
          TimerPositionBV = 0 ' sets minimum TimerPositionBV as 0
          Form2.txtTmr_PosBtmVent.Text = TimerPositionBV
          Form1.LockingBV = False
          Call Form1.CloseBVRelaySwitch("0.0", "0") ' routine that turns of the relay
        End If
```

```
    End If

'Routine that calculated and displays the position of the bottom vent
    Form1.PositionBV = (6.66666666 * TimerPositionBV) - 54
      If Form1.PositionBV < 0 Then Form1.PositionBV = 0
      If Form1.PositionBV > 100 Then Form1.PositionBV = 100
    Form1.txtStatus_BtmVent = Form1.PositionBV

    If Form2.VentAdjustModeBV = True And Form2.DesiredBV - Form1.PositionBV
= 0 Then
      Form2.VentAdjustModeBV = False
      Form2.txtStatus_VentAdjustBV = Form2.VentAdjustModeBV
      Call Form1.CloseBVRelaySwitch("0.0", "0")
      Call Form1.OpenBVRelaySwitch("0.0", "0")
    End If

'Routine that only works when not in the process of locking/unlocking
    If Form2.txtState_OpenTopVent = 1 And Form1.LockingTV = False Then
      Form2.StatusLockTV = False ' for switches and IR
      Form2.txtStatus_LockTV = Form2.StatusLockTV
      TimerOpenTV = TimerPositionTV
      TimerPositionTV = TimerOpenTV + 1
      Form2.txtTmr_PosTopVent.Text = TimerPositionTV
      'TimerOpenTV = 0
        If TimerPositionTV >= 25 Then
          TimerPositionTV = 25    ' sets maximum TimerPositionTV as 25
          Form2.txtTmr_PosTopVent.Text = TimerPositionTV
          Form1.ButtonOpenTV = False    ' routine that turns of the relay
          Form2.VentAdjustModeTV = False
          Form2.txtStatus_VentAdjustTV = Form2.VentAdjustModeTV
          Call Form1.OpenTVRelaySwitch("0.0", "0")
            'don't add anything after this as disable used
        End If
    End If

' Routine that Unlocks the top vent
    If Form2.txtState_OpenTopVent = 1 And Form1.LockingTV = True Then
      TimerOpenTV = TimerPositionTV
      TimerPositionTV = TimerOpenTV + 1
      Form2.txtTmr_PosTopVent.Text = TimerPositionTV
      'TimerOpenTV = 0
        If TimerPositionTV = 8 Then ' sets maximum TimerPositionTV as 8
          Form1.LockingTV = False
          Call Form1.OpenTVRelaySwitch("0.0", "0") ' routine that turns of the relay
        End If
    End If
```

```
'Routine that only closes and positions the top vent when not in the process of locking
   If Form2.txtState_CloseTopVent = 1 And Form1.LockingTV = False Then
     TimerCloseTV = TimerPositionTV
     TimerPositionTV = TimerCloseTV - 1
     Form2.txtTmr_PosTopVent.Text = TimerPositionTV
     'TimerCloseTV = 0
       If TimerPositionTV <= 8 And Form2.ResetState = False _
          And Form2.StatusLockTV = False Then 'needed to preserve zero after reset
         TimerPositionTV = 8      ' sets minimum TimerPositionBV as 8
         Form2.txtTmr_PosTopVent.Text = TimerPositionTV
         Form1.ButtonCloseTV = False     ' routine that turns of the relay
         Form2.VentAdjustModeTV = False
         Form2.txtStatus_VentAdjustTV = Form2.VentAdjustModeTV
         Call Form1.CloseTVRelaySwitch("0.0", "0")
       Else
         If TimerPositionTV < 0 Then TimerPositionTV = 0
         Form2.txtTmr_PosTopVent.Text = TimerPositionTV
       End If
   End If

' Routine that locks the top vent
   If Form2.txtState_CloseTopVent = 1 And Form1.LockingTV = True Then
     TimerCloseTV = TimerPositionTV
     TimerPositionTV = TimerCloseTV - 1
     Form2.txtTmr_PosTopVent.Text = TimerPositionTV
     'TimerCloseTV = 0
       If TimerPositionTV <= 0 Then
         TimerPositionTV = 0      ' sets minimum TimerPositionTV as 0
         Form2.txtTmr_PosTopVent.Text = TimerPositionTV
         Form1.LockingTV = False
         Call Form1.CloseTVRelaySwitch("0.0", "0") ' routine that turns of the relay
       End If
   End If

'Routine that calculated and displays the position of the bottom vent
   Form1.PositionTV = (6.66666666 * TimerPositionTV) - 54
     If Form1.PositionTV < 0 Then Form1.PositionTV = 0
     If Form1.PositionTV > 100 Then Form1.PositionTV = 100
   Form1.TxtStatus_TopVent = Form1.PositionTV

   If Form2.VentAdjustModeTV = True And Form2.DesiredTV - Form1.PositionTV
= 0 Then 'stops vent at desired position for automatic control
     Form2.VentAdjustModeTV = False
     Form2.txtStatus_VentAdjustTV = Form2.VentAdjustModeTV
     Call Form1.CloseTVRelaySwitch("0.0", "0")
     Call Form1.OpenTVRelaySwitch("0.0", "0")
   End If
End Sub
```

Automatic Control Timer

Private Sub tmrMin_Counter_Timer()

```
'This module is a minute counter and the operation of many of the open loop control
functions and checks are determined by this counter

  Static SolarCounter As Integer 'counter for calculation of the solar position
  Dim SunReadingsTime As Integer 'interval between solar positioning calculations

  Static ResetCounter As Integer 'counter for the reset of the system

  Dim VentAutoDelay As Integer 'duration of occupant vent override priority
  Static VentORMinCounter As Integer 'counter for occupant vent override priority

  Static VentAutoCounter As Integer 'counter for the vent control interval
  Dim VentIntervalTime As Integer 'interval between vent control actions

  Static BlindAutoCounter As Integer 'counter for the blind control interval
  Dim BlindIntervalTime As Integer 'interval between blind control actions

  Dim BlindAutoDelay As Integer 'duration of occupant blind override priority
  Static BlindORMinCounter As Integer 'counter for occupant blind override priority

  Dim BlindSunOutDelay As Integer 'duration of the sun out delay time
  Static SunOutThresMinCounter As Integer 'counter for the sun out delay time

  Dim BlindSunDelay As Integer 'duration of the sun in delay time
  Static SunThresMinCounter As Integer 'counter for the sun in delay time

'The solar positioning counter
    SolarCounter = SolarCounter + 1
    SunReadingsTime = Val(txtTime_SunReadings)
    If SolarCounter = SunReadingsTime Then
      Call Form2.SolarPosition 'calls the solar positioning algorithm routine
      SolarCounter = 0
    End If

'The reset counter
    ResetCounter = ResetCounter + 1
      If ResetCounter = 1440 And Form1.Occupancy = False Then
        Form2.ResetState = True
        txtTmr_ResetState = Form2.ResetState
        ResetCounter = 0
      End If
```

```
'The vent override priority counter
    If Form2.VentUserOveride = True Then 'starts counter when user adjusts vent
      VentORMinCounter = VentORMinCounter + 1
      VentAutoDelay = Val(txtTime_VentDelay.Text)
      If VentORMinCounter = VentAutoDelay Then
        Form2.VentUserOveride = False
        txtStatus_VentOR = Form2.VentUserOveride
        VentORMinCounter = 0
      End If
    Else
      VentORMinCounter = 0
    End If

'the vent control interval counter
    VentIntervalTime = Val(txtVent_Interval.Text) 'gets user defined vent interval time
    VentAutoCounter = VentAutoCounter + 1
      If VentAutoCounter >= VentIntervalTime Then
'when it expires it starts control sequence
        VentAutoCounter = 0
        Call Form2.WorkHours
        Call WindAssessment
        If WindyConditions = True Then 'set back positioning for windy conditions
          If Form1.PositionTV > 53 Then DesiredTV = 53
          If Form1.PositionBV > 0 Then DesiredBV = 0
          Call OpenOrCloseVents 'calls vent positioning routine
        Else
          If OccupancyPeriod = True Then
'If in occupied period modulate vents to control temperature, else close vents during
the weekend day and night cool at night
            If Form2.VentUserOveride = False Then Call Form2.OccVentControl
          Else
            If WeekEndDay = True Then
              If Form1.Occupancy = True Then
                Call Form2.OccVentControl
              Else
                Call Form1.CloseTVRelaySwitch("100.0", "1")
                Call Form1.CloseBVRelaySwitch("100.0", "1")
              End If
            Else
              If Form1.Occupancy = True Then
                Call Form2.OccVentControl
              Else
                Call Form2.NightCooling
              End If
            End If
          End If
        End If
      End If
```

```
'The blind override priority counter
   If BlindUserOveride = True Then
     BlindORMinCounter = BlindORMinCounter + 1
     BlindAutoDelay = Val(Form2.txtTime_BlindDelay.Text)
     If BlindORMinCounter >= BlindAutoDelay Then
       BlindUserOveride = False
       Form2.txtStatus_BlindOR = BlindUserOveride
       BlindORMinCounter = 0
     End If
   Else
     BlindORMinCounter = 0
   End If

'The sun out time delay counter
   If SunOutDelay = True Then
     SunOutThresMinCounter = SunOutThresMinCounter + 1
     BlindSunOutDelay = Val(Form2.txtTime_SunOutThreshold.Text)
'when the counter expires and the blind isn't moving or the over-ride has priority then
the threshold is passed
     If SunOutThresMinCounter >= BlindSunOutDelay And BlindUserOveride = False
       _ And Form2.MoveUp = False And Form2.MoveDown = False Then
       Form1.SunThresholdState = True
       SunOutThresMinCounter = 0
       SunOutDelay = False
       Form2.txtState_SunThreshold = Form1.SunThresholdState
       txtStatus_SunOutDelay = SunOutDelay
     End If
   Else
     SunOutThresMinCounter = 0
   End If

'The sun in time delay counter
   If SunDelay = True Then
     SunThresMinCounter = SunThresMinCounter + 1
     BlindSunDelay = Val(Form2.txtTime_SunThreshold.Text)
'same of the sun out delay but with a different user defined time delay
     If SunThresMinCounter >= BlindSunDelay And BlindUserOveride = False _
         And Form2.MoveUp = False And Form2.MoveDown = False Then
       Form1.SunThresholdState = False
       SunThresMinCounter = 0
       SunDelay = False
       txtStatus_SunDelay = SunDelay
       Form2.txtState_SunThreshold = Form1.SunThresholdState
     End If
   Else
     SunThresMinCounter = 0
   End If
```

```
'the blind control interval counter
  BlindIntervalTime = Val(txtBlind_Interval.Text)
  BlindAutoCounter = BlindAutoCounter + 1

    If BlindAutoCounter >= BlindIntervalTime And BlindUserOveride = False Then
     BlindAutoCounter = 0
     If Form1.Occupancy = False Then 'during unoccupied hours
      If Form2.AfternoonTemp > 18 Then 'in summer the blind comes down in day
       If Form2.Altitude > 0 Then
        Call Form2.BlindControl("SET_STATE", "0.0", "0.00")
       Else                             'and goes up at night
        Call Form2.BlindControl("SET_UP", "0.0", "0.00")
       End If
      Else
       If Form2.Altitude > 0 Then 'in the winter the blind goes up in the day
        Call Form2.BlindControl("SET_UP", "0.0", "0.00")
       Else                          'and comes down at night
        Call Form2.BlindControl("SET_STATE", "0.0", "0.00")
       End If
      End If
     Else
      If Form1.SunThresholdState = True Then Call Form2.AlterSlatAngle
      Else
       If SunDelay = False Then Call Form2.BlindControl("SET_UP", "0.0", "0.00")
      End If
     End If
    End If

End Sub
```

Private Sub txtOcc_Sensor_Change()

```
' Routine to translate occupancy status from the text box linked by DDE to the
occupancy sensor
 If txtOcc_Sensor.Text = "SET_OFF" Then
  Form1.Occupancy = False
 Else
  Form1.Occupancy = True
 End If
 Form2.txtStatus_Occupancy = Form1.Occupancy

End Sub
```

```
Public Sub WorkHours()

Dim CurrentHour As Integer, CurrentDay As Integer, CurrentTime As Date
Dim Start As Integer, Finish As Integer, PreHeatStart As Integer
Dim OldTotalTemp As Single, NewTemp As Single
Static TempReadingsCounter As Integer

  CurrentTime = Now()
  CurrentHour = Hour(CurrentTime)
  CurrentDay = WeekDay(CurrentTime, vbMonday)
  Start = Val(txtOccStartHour) 'start of occupied period, set by the users in Form 2
  Finish = Val(txtOccFinishHour)'end of occupied period, set by the users in Form 2
  PreHeatStart = Val(txtPreHeatHour) 'start of preheat period, set by the users in
Form 2
  txtCurrentHour = Val(CurrentHour)
  txtCurrentDay = Val(CurrentDay)

'routine that acts as a scheduler using the computers internal clock to determine
whether the time is within the predefined occupancy hours
  If CurrentHour > Start And CurrentHour < Finish Then
    If CurrentDay > 5 Then
      OccupancyPeriod = False
      WeekEndDay = True
    Else
      OccupancyPeriod = True
      WeekEndDay = False
    End If
  Else
    OccupancyPeriod = False
    WeekEndDay = False
  End If

  txtStatus_OccPeriod = OccupancyPeriod

'routine to assess whether time is within the preheat period defined by the user after a
night cooling strategy
  If CurrentHour >= PreHeatStart And CurrentHour < Start Then
    PreHeatPeriod = True
  Else
    PreHeatPeriod = False
  End If

  txtStatus_Preheat = PreHeatPeriod

  If CurrentHour = 11 And FirstSet = True Then
    FirstSet = False
    TempReadingsCounter = 0
  End If
```

```
'routine to calculate average afternoon temperature for the night cooling strategy
  If CurrentHour >= 12 And CurrentHour < 17 Then
    If FirstSet = False Then
      TotalTemp = Form1.ExtTemp
      FirstSet = True
    Else
      OldTotalTemp = TotalTemp
      NewTemp = Form1.ExtTemp
      TotalTemp = NewTemp + OldTotalTemp
    End If
    TempReadingsCounter = TempReadingsCounter + 1
    AfternoonTemp = TotalTemp / TempReadingsCounter
    txtAfternoonTemp = AfternoonTemp
  End If

  txtTotalTemp = TotalTemp
  txtTempReadingsCounter = TempReadingsCounter

End Sub
```

Blind Sun Threshold Control

Private Sub txtLux_Ext_Change()

```
'if the external illuminance changes the Sun Threshold routine is started
Call Form1.SunThreshold

End Sub
```

Public Sub SunThreshold()

```
'routine that compares the measure external illuminance to the user defined external
illuminance threshold and activates the timers if the sun is on the façade.
  Dim SunThreshold As Integer, ExtIllum As Single, Orientation As Integer
  Dim DeltaAzimuth As Integer

  SunThreshold = Val(Form2.txtSunThreshold.Text) 'set in Form 2
  ExtIllum = Val(Form1.txtLux_Ext.Text) 'from DDE link with sun sensor
  Orientation = Val(Form2.txtFacadeOrientation.Text) ' of the blind set in Form 2
  DeltaAzimuth = Abs(Form2.Azimuth - Orientation)

  If (ExtIllum >= SunThreshold Or ExtIllum >= 54990) And DeltaAzimuth < 90 Then
    If SunThresholdState = False Then
      Form2.SunOutDelay = True
```

```
    Else
      Form2.SunDelay = False
    End If
  Else
    If SunThresholdState = False Then Form2.SunOutDelay = False
    If SunThresholdState = True Then Form2.SunDelay = True
  End If
  Form2.txtStatus_SunOutDelay = Form2.SunOutDelay
  Form2.txtStatus_SunDelay = Form2.SunDelay

End Sub
```

Private Sub txtState_SunThreshold_Change()

```
If Form1.SunThresholdState = True Then
    If PositionBlind < 100 Then Call Form2.BlindControl("SET_STATE", "0.0",
"0.00") 'lower blind if threshold state = true and raise if false
  Else
    If PositionBlind > 0 Then Call Form2.BlindControl("SET_UP", "0.0", "0.00")
  End If

End Sub
```

Solar Position

Public Static Function ArcSin(X) As Double

```
'defines the inverse of the sine
  ArcSin = Atn(X / Sqr(-X * X + 1))

End Function
```

Public Function ArcCos(X) As Double

```
'defines the inverse of the cosine
If (X > 0.999 And X < 1.001) Or (X < -0.999 And X > -1.001) Then
    ArcCos = -Atn(0) + Pi
  Else
    ArcCos = Atn(X / Sqr(-X * X + 1)) + (Pi / 2)
  End If

End Function
```

```
Public Sub SolarPosition() 'IES's and Goulding's formulae

'routine that calculated the solar position in terms of altitude and azimuth using the
current time and date from the computers time clock and IES's equation of time
formula an Goulding's Solar Declination Formula as defined in Appendix IV

Dim CurrentDay As Integer, CurrentMonth As Integer, CurrentYear As Integer
Dim CurrentHour As Integer, CurrentMinute As Integer

Dim INTYR As Integer, IFAC As Integer, M As Integer, IMNEW As Integer,
DayNo As Integer, Dim EOT As Double

Dim SolarDec As Single, TimeMeridian As Single, TST As Single, HourAngle As
Single

Dim Latitude As Single, Dim Longitude As Single, SouthAzimuth As Double
Dim Summertime As Boolean, Dim TD As Integer

   CurrentDay = Day(Now)
   CurrentMonth = Month(Now)
   CurrentYear = Year(Now)
   CurrentHour = Hour(Now)
   CurrentMinute = Minute(Now)
   DayOfWeek = WeekDay(CurrentTime, vbMonday)

   Latitude = txtLatitude
   Longitude = txtLongitude
   TimeMeridian = txtTimeMerid

'code to work out day number taking into account leap year
   INTYR = 4 * Int(0.25 * CurrentYear)
   If CurrentYear > INTYR Then
     IFAC = 63
   Else
     IFAC = 62
   End If
   If CurrentMonth > 2 Then
     M = CurrentMonth + 1
     IMNEW = (Int(30.6 * M)) - IFAC
   Else
     M = CurrentMonth - 1
     IMNEW = Int(0.5 * M * IFAC)
   End If
   DayNo = CurrentDay + IMNEW

'Goulding's Solar declination formula
   SolarDec = (ArcSin(0.39795 * Sin(DayNo – 80.2 + (1.92 * Sin(DayNo – 2.89)))))
```

```
'code that works out whether it is British Summer Time.  This is needed because the
algorithm is running off the computers clock which reads local time.  It would be
needed in a neuron chip if you to use internal timers
  If CurrentMonth <= 3 Then
   If CurrentMonth = 3 And DayOfWeek > 6 And CurrentDay >24 Then
    SummerTime = True
   Else
    SummerTime = False
   End If
  Else
   SummerTime = True
  End If
  If CurrentMonth = 10 And DayOfWeek > 6 And CurrentDay >24 Then
    SummerTime = False
  End If
  If CurrentMonth > 11 Then
   SummerTime = False
  End If
  If SummerTime = True Then
   TD = 1
  Else
   TD = 0
  End If

'The IES EOT Formula
EOT = (0.170 * Sin((4*Pi*(DayNo - 80)) / 373)) – _
               (0.129 * Sin((2*Pi*(DayNo – 8)) / 355))

'routine to calculate the hour angle
  TST = (CurrentHour + (CurrentMinute / 60)) – TD + ((TimeMeridian – _
          Longitude) / 15) + EOT
  If TST = 12 Then
   HourAngle = 0
  Else
   If TST < 12 Then
    HourAngle = -(12 - UT) * 15
   Else
    HourAngle = (UT - 12) * 15
   End If
  End If

  Form2.Altitude = (ArcSin((Sin(Latitude * (Pi / 180)) * Sin(SolarDec * (Pi / 180)))
- (Cos(Latitude * (Pi / 180)) * Cos(SolarDec * (Pi / 180)) * Cos((HourAngle + 180)
* (Pi / 180))))) * (180 / Pi) 'calculates the altitude

  If Form2.Altitude <= 0 Then Form2.Altitude = 0
```

```
    SouthAzimuth = (ArcCos(((Cos(SolarDec * (Pi / 180)) * Cos(HourAngle * (Pi /
180)) * Sin(Latitude * (Pi / 180))) - (Sin(SolarDec * (Pi / 180)) * Cos(Latitude * (Pi /
180)))) / Cos(Altitude * (Pi / 180)))) * (180 / Pi) 'calculates the azimuth from south

    If HourAngle > 0 Then 'converts the azimuth to an the azimuth from north
      Form2.Azimuth = 360 - SouthAzimuth
    Else
      Form2.Azimuth = SouthAzimuth
    End If

    Form1.txtAlt_Sun = Form2.Altitude
    Form2.txtAlt_Sun = Form2.Altitude
    Form1.txtAzmth_Sun = Form2.Azimuth
    Form2.txtAzmth_Sun = Form2.Azimuth

End Sub
```

Automatic Blind Control Code

Public Sub AlterSlatAngle()

```
'routine that determines the desired slat angle from the sun's position
Dim Orientation As Integer, FacadeAltitude As Single, SlatAngle As Single
  Dim DeltaAzimuth As Integer, BlindAngle As Integer

  Orientation = Val(txtFacadeOrientation) 'of the blind set in Form 2
  DeltaAzimuth = Abs(Form2.Azimuth - Orientation)
  FacadeAltitude = Atn((Tan(Form2.Altitude * (Pi / 180))) / (Cos(DeltaAzimuth * (Pi
/ 180)))) * (180 / Pi) 'calculates the solar wall altitude angle
  txtFacadeAltitude = FacadeAltitude

'selects blind angle to block sun
  If FacadeAltitude >= 45 Then
    BlindAngle = 90
  Else
    SlatAngle = 90 - (2 * FacadeAltitude)
    BlindAngle = 90 - SlatAngle
    If BlindAngle >= 67.5 Then
      BlindAngle = 67.5
    Else
      If BlindAngle >= 45 Then
        BlindAngle = 45
      Else
        If BlindAngle >= 22.5 Then
          BlindAngle = 22.5
```

```
        Else
          BlindAngle = 0
        End If
      End If
    End If
  End If

  DesiredRotation = 180 - BlindAngle
  txtDesiredRotation = DesiredRotation

'once angle decided –calls routine to open or close blinds
  Call RaiseOrLowerBlinds

End Sub
```

Public Sub RaiseOrLowerBlinds()

```
'routine that decides which way to move the blinds to obtain the desired tilt angle
  If DesiredRotation - RotationBlind <> 0 Then
    BlindAdjustMode = True
    If DesiredRotation - RotationBlind < 0 Then
      If Form2.MoveUp = False Then Call BlindControl("SET_STATE", "0.0", "0.00")
    Else
      If Form2.MoveDown = False Then Call BlindControl("SET_UP", "0.0", "0.00")
    End If
  Else
    Exit Sub
  End If
End Sub
```

Blind State and Stop

Private Sub txtBlindFunc_Change()

```
'routine that translates the lonworks setting network variables to the visual basic
variables so that the blind positioning algorithm can keep track of user overrides

  If Form2.txtBlindFunc.Text = "SET_STATE" Then
    MoveDown = True
    txtState_MoveDown = MoveDown
  Else
    If Form2.txtBlindFunc.Text = "SET_UP" And Form2.ResetState = False Then
      MoveUp = True
      txtState_MoveUp = MoveUp
    Else
```

```
   If Form2.txtBlindFunc.Text = "SET_STOP" And PositionBlind = 100 _
     And (RotationBlind = 0 Or RotationBlind = 180) Then Call
Form2.AlterSlatAngle
  End If
 End If

End Sub

Private Sub tmrBlindPos_Timer()

'timer routine, counting in half seconds, used to determine the blind position
 Dim Slow As Boolean
 Static TimerCounter As Integer
 Dim Tilting As Boolean

'routine to work out whether the blind is tilting or not
 If (MoveUp = True Or MoveDown = True) And ResetState = False Then
  Slow = True
  If RotationBlind > 0 And RotationBlind < 180 And PositionBlind > 0 Then
   Tilting = True
  Else
   If RotationBlind = 0 And MoveUp = True And PositionBlind > 0 Then
    Tilting = True
   Else
    If RotationBlind = 180 And MoveDown = True And PositionBlind > 0 Then
     Tilting = True
    Else
     Tilting = False
    End If
   End If
  End If
  txtState_Tilting = Tilting
  TimerCounter = TimerCounter + 1 'timer counting in half seconds
  If TimerCounter >= 8 Then Slow = False 'slow speed stops after 4 secs
  txtState_Slow = Slow
  If MoveUp = True Then
   If Tilting = True And Slow = True Then 'when speed is slow
    RotationBlind = RotationBlind + 22.5 'blind tilts 22.5 degrees in 0.5 secs
   Else
    If Tilting = False And Slow = True Then 'when speed is slow
     PositionBlind = PositionBlind - 1.25 'blind raises 1.25% in 0.5 secs
     If PositionBlind <= 0 Then 'stops blind and rests counter at end of travel
      PositionBlind = 0
      RotationBlind = 0
      TimerCounter = 0
      Call Form2.StopBlinds
     End If
    Else
```

```
     If Tilting = False And Slow = False Then 'when speed is fast
      PositionBlind = PositionBlind - 2.4 'blind raises 2.4% in 0.5 secs
      If PositionBlind <= 0 Then 'stops blind and rests counter at end of travel
       PositionBlind = 0
       RotationBlind = 0
       TimerCounter = 0
       Call Form2.StopBlinds
      End If
     End If
    End If
   End If
  Else
   If Tilting = True And Slow = True Then 'when speed is slow
    RotationBlind = RotationBlind - 22.5 'blind tilts 22.5 degrees in 0.5 secs
   Else
    If Tilting = False And Slow = True Then 'when speed is slow
     PositionBlind = PositionBlind + 1.25 'blind lowers 1.25% in 0.5 secs
     If PositionBlind >= 100 Then 'stops blind and rests counter at end of travel
      PositionBlind = 100
      TimerCounter = 0
      Call Form2.StopBlinds
     End If
    Else
     If Tilting = False And Slow = False Then 'when speed is fast
      PositionBlind = PositionBlind + 2.6 'blind lowers 2.6% in 0.5 secs
      If PositionBlind >= 100 Then 'stops blind and rests counter at end of travel
       PositionBlind = 100
       TimerCounter = 0
       Call Form2.StopBlinds
      End If
     End If
    End If
   End If
  End If

'routine to stop blinds when they are within 5 degrees of their desired angle
  If BlindAdjustMode = True Then
   If Abs(DesiredRotation - RotationBlind) < 5 Then
    TimerCounter = 0
    BlindAdjustMode = False
    Call Form2.StopBlinds
   End If
  End If

  Form1.txtPositionBlind = PositionBlind
  Form2.txtPositionBlind = PositionBlind
  Form1.txtRotationBlind = RotationBlind
  Form2.txtRotationBlind = RotationBlind
```

```
  Else
    TimerCounter = 0
  End If

  Text1.Text = TimerCounter

End Sub
```

Public Sub StopBlinds()

```
'a subroutine that stops the blinds
  MoveDown = False
  Form2.txtState_MoveDown = MoveDown
  MoveUp = False
  Form2.txtState_MoveUp = MoveUp
  Call Form2.BlindControl("SET_STOP", "0.0", "0.00")
End Sub
```

Blind User Override Control

Private Sub cmdLowerBlind_Click()

```
'routine for when the user clicks on lower blind button on the VB user interface
..LowerBlind = Not (LowerBlind)
  If LowerBlind = True And Form2.MoveUp = False Then
    Call Form2.BlindUserOR
    Call Form2.BlindControl("SET_STATE", "0.0", "0.00")
  Else
    Call Form2.StopBlinds
  End If
End Sub
```

Private Sub cmdRaiseBlind_Click()

```
'routine for when the user clicks on raise blind button on the VB user interface
  RaiseBlind = Not (RaiseBlind)
  If RaiseBlind = True And Form2.MoveDown = False Then
    Call Form2.BlindUserOR
    Call Form2.BlindControl("SET_UP", "0.0", "0.00")
  Else
    Call Form2.StopBlinds
  End If
End Sub
```

Private Sub txtIRBlindDown_Change()

```
'routine to control the blind down motion with the IR controller
 If txtIRBlindDown = "1" And Form2.MoveUp = False Then
   Call BlindUserOR
   Call Form2.BlindControl("SET_STATE", "0.0", "0.00")
 Else
   Call Form2.StopBlinds
 End If
End Sub
```

Private Sub txtIRBlindUp_Change()

```
'routine to control the blind up motion with the IR controller
 If txtIRBlindUp = "1" And Form2.MoveDown = False Then
   Call BlindUserOR
   Call Form2.BlindControl("SET_UP", "0.0", "0.00")
 Else
   Call Form2.StopBlinds
 End If
End Sub
```

Public Sub BlindUserOR()

```
'routine that ensures that the blind user over-ride timer always starts at the last user
override
 BlindUserOveride = False ' in case user makes adjustment before UserOR timer
expires
 Form2.txtStatus_BlindOR = BlindUserOveride
 BlindUserOveride = True
 Form2.txtStatus_BlindOR = BlindUserOveride
End Sub
```

Private Sub txtStatus_BlindOR_Change()

```
'Routine to start the over-ride timer
 If txtStatus_BlindOR.Text = "True" Then
   Form2.tmrBlindUserOR.Enabled = True
 Else
   Form2.tmrBlindUserOR.Enabled = False
 End If
 End Sub
```

```
Public Sub BlindControl(ValueA As String, ValueB As String, ValueC As
String)

'DDE code sent to the blind actuator to move the blind
  Form2.txtBlindFunc = ValueA
  Form2.txtBlindSetting = ValueB
  Form2.txtBlindRotation = ValueC
  Form2.txtBlindFunc.LinkPoke
  Form2.txtBlindSetting.LinkPoke
  Form2.txtBlindRotation.LinkPoke

End Sub
```

Automatic Control of Vents

```
Private Sub txtTemp_Ext_Change()

'routine that recognises any change in temperature from the DDE linked text box
  Form1.ExtTemp = Val(txtTemp_Ext.Text)

End Sub

Private Sub txtTemp_Int_Change()

'routine that recognises any change in temperature from the DDE linked text box
  Form1.IntTemp = Val(txtTemp_Int.Text)

End Sub

Public Sub OccVentControl()

'main occupied vent control code
  If Form1.IntTemp > 20 Then
    Call Form2.ModulateVents
  Else
    Call Form1.CloseTVRelaySwitch("100.0", "1")
    Call Form1.CloseBVRelaySwitch("100.0", "1")
  End If

End Sub
```

Public Sub ModulateVents()

```
'occupied vent control subroutine that controls the vent opening depending on a linear
relationship with the internal temperature
 Dim MinuteFraction As Single
 Dim X As Single, Y As Single
 Dim CurrentHour As Integer, CurrentMinute As Integer

 If Form1.ExtTemp < 10 And Form1.IntTemp < 22 Then
  If Form1.PositionTV > 0 Then Call Form1.CloseTVRelaySwitch("100.0", "1")
  If Form1.PositionBV > 0 Then Call Form1.CloseBVRelaySwitch("100.0", "1")
 Else
  CurrentHour = Hour(Now)
  CurrentMinute = Minute(Now)
  MinuteFraction = CurrentMinute / 60
  X = CurrentHour + MinuteFraction
  Y = (0.25 * X) + 18.5
  If Y > Form1.IntTemp Then
   Form2.DesiredTV = 0
   Form2.DesiredBV = 0
  Else
   If Form1.IntTemp - Y <= 0.5 Then
    Form2.DesiredTV = 26
    Form2.DesiredBV = 0
   Else
    If Form1.IntTemp - Y <= 1.5 Then
     Form2.DesiredTV = 53
     Form2.DesiredBV = 0
    Else
     If Form1.IntTemp - Y <= 2.5 Then
      Form2.DesiredTV = 100
      Form2.DesiredBV = 0
     Else
      If Form1.IntTemp - Y <= 3 Then
       Form2.DesiredTV = 100
       Form2.DesiredBV = 26
      Else
       If Form1.IntTemp - Y <= 4 Then
        Form2.DesiredTV = 100
        Form2.DesiredBV = 53
       Else
        If Form1.IntTemp - Y > 4 Then
         Form2.DesiredTV = 100
         Form2.DesiredBV = 100
        End If
       End If
      End If
     End If
```

```
        End If
      End If
    End If
    Form2.txtDesiredTV = Form2.DesiredTV
    Form2.txtDesiredBV = Form2.DesiredBV
    Call Form2.OpenOrCloseVents
  End If

End Sub
```

Public Sub NightCooling()

```
'night cooling routine that opens vents depending on internal temperature
If AfternoonTemp > 18 And Form1.IntTemp > Form1.ExtTemp And
Form1.ExtTemp > 12 _
      And PreHeatPeriod = False Then
    If Form1.IntTemp > 26 And Form1.PositionBV < 100 Then Call
Form1.OpenBVRelaySwitch("100.0", "1")
    If Form1.IntTemp > 17 And Form1.PositionTV < 100 Then Call
Form1.OpenTVRelaySwitch("100.0", "1")
    If Form1.IntTemp < 26 And Form1.PositionBV > 0 Then Call
Form1.CloseBVRelaySwitch("100.0", "1")
    If Form1.IntTemp < 14 And Form1.PositionTV > 0 Then Call
Form1.CloseTVRelaySwitch("100.0", "1")
  Else
    If Form1.PositionTV > 0 Then Call Form1.CloseTVRelaySwitch("100.0", "1")
    If Form1.PositionBV > 0 Then Call Form1.CloseBVRelaySwitch("100.0", "1")
  End If

End Sub
```

Public Sub OpenOrCloseVents()

```
'routine that works out whether the vents should be opened or closed to get to their
desired position
If Abs(Form2.DesiredTV - Form1.PositionTV) > 5 Then
    Form2.VentAdjustModeTV = True
    Form2.txtStatus_VentAdjustTV = Form2.VentAdjustModeTV
    If Form2.DesiredTV - Form1.PositionTV < 0 Then
      Call Form1.CloseTVRelaySwitch("100.0", "1")
    Else
      Call Form1.OpenTVRelaySwitch("100.0", "1")
    End If
  End If

  If Abs(Form2.DesiredBV - Form1.PositionBV) > 5 Then
    Form2.VentAdjustModeBV = True
    Form2.txtStatus_VentAdjustBV = Form2.VentAdjustModeBV
```

```
    If Form2.DesiredBV - Form1.PositionBV < 0 Then
      Call Form1.CloseBVRelaySwitch("100.0", "1")
    Else
      Call Form1.OpenBVRelaySwitch("100.0", "1")
    End If
  End If
End Sub
```

Wind Assessment

Public Sub WindAssessment()

```
'routine that defines when the wind conditions are too strong for normal vent control.
Dim WindSpeed As Single, WindDirection As Integer, Orientation As Integer
Dim WindAngle As Integer

  WindSpeed = Val(Form1.txtWind_SpeedExt)
  WindDirection = Val(Form1.txtWind_Direction)
  Orientation = Val(txtFacadeOrientation)
  WindAngle = Abs(WindDirection - Orientation)

  If WindSpeed > 11 And WindAngle < 90 Then
    WindyConditions = True
    txtWindyConditions = WindyConditions
  Else
    WindyConditions = False
    txtWindyConditions = WindyConditions
  End If
End Sub
```

Vent User Override Control

Private Sub cmdClose_TopVent_Click()

```
'routine for when the user clicks on close top vent button in the VB user interface
  If Form1.PositionTV = 0 And ButtonCloseTV = False Then
    Beep
    MsgBox "The vent is fully closed. It can not be closed anymore", vbOKOnly + _
        vbExclamation, "Bottom Vent"
  Else
    ButtonCloseTV = Not (ButtonCloseTV)
    If ButtonCloseTV = True And ButtonOpenTV = False Then 'stops both relays
going on
```

```
      Call Form1.CloseTVRelaySwitch("100.0", "1")
    Else
      ButtonCloseTV = False 'needed for when ButtonOpenTV = True
      Call Form1.CloseTVRelaySwitch("0.0", "0")
      Call Form2.VentUserOR ' switches user override delay on
    End If
  End If
End Sub

Private Sub txtState_CloseTVIR_Change()

'routine to ensure the vent override timer is activated when the IR controller is used
  If txtState_CloseTVIR.Text = "0" Then Call Form2.VentUserOR
End Sub

Private Sub txtState_CloseTVSth_Change()

'routine to ensure the vent override timer is activated when the switch is used
  If txtState_CloseTVSth.Text = "0" Then Call Form2.VentUserOR
End Sub

Private Sub cmdOpen_TopVent_Click()

'routine for when the user clicks on open top vent button in the VB user interface
  If Form2.StatusLockTV = True Then
    Beep
    MsgBox "This vent is locked. You must unlock the vent before opening", _
            vbOKOnly + vbExclamation, "Top Vent"
  Else
    If Form1.PositionTV = 100 And ButtonOpenTV = False Then
      Beep
      MsgBox "The vent is fully open. It can not be opened anymore", _
          vbOKOnly + vbExclamation, "Top Vent"
    Else
      ButtonOpenTV = Not (ButtonOpenTV)
      If ButtonOpenTV = True And ButtonCloseTV = False Then 'stops both relays
going on
        Call Form1.OpenTVRelaySwitch("100.0", "1")
      Else
        ButtonOpenTV = False ' needed for when ButtonOpenTV = True
        Call Form1.OpenTVRelaySwitch("0.0", "0")
        Call Form2.VentUserOR ' switches user override delay on
      End If
    End If
  End If
End Sub
```

```
Private Sub txtState_OpenTVIR_Change()

'routine to ensure the vent override timer is activated when the IR controller is used
  If txtState_OpenTVIR.Text = "0" Then Call Form2.VentUserOR
End Sub
```

```
Private Sub txtStateOpenTVSth_Change()

'routine to ensure the vent override timer is activated when the switch is used
  If txtState_OpenTVSth.Text = "0" Then Call Form2.VentUserOR
End Sub
```

```
Private Sub cmdLock_TopVent_Click()

'Routine to toggle the top vent lock button on and off and lock and unlock vent
  If Form1.PositionTV <> 0 Then
    Beep
    MsgBox "The vent must be fully closed before it can be locked", _
          vbOKOnly + vbExclamation, "Top Vent"
  Else
    ' Routine that stops the vent from being locked and unlocked at the same time
    If Form1.LockingTV = False Then
      Form2.StatusLockTV = Not (Form2.StatusLockTV)
      Form2.txtStatus_LockTV.Text = Form2.StatusLockTV
      Form1.LockingTV = Not (Form1.LockingTV)
      If Form2.StatusLockTV = False Then
        Call Form1.OpenTVRelaySwitch("100.0", "1")
      Else
        Call Form1.CloseTVRelaySwitch("100.0", "1")
      ' Reset to False at the end of the timer routine
      End If
    End If
  End If
End Sub
```

```
Private Sub cmdClose_BtmVent_Click()

'routine for when the user clicks on close bottom vent button in the VB user interface
If Form1.PositionBV = 0 And ButtonCloseBV = False Then
    Beep
    MsgBox "The vent is fully closed. It can not be closed anymore", vbOKOnly + _
        vbExclamation, "Bottom Vent"
  Else
    ButtonCloseBV = Not (ButtonCloseBV)
    If ButtonCloseBV = True And ButtonOpenBV = False Then 'stops both relays
going on
      Call Form1.CloseBVRelaySwitch("100.0", "1")
    Else
```

```
      ButtonCloseBV = False 'needed for when ButtonOpenBV = True
      Call Form1.CloseBVRelaySwitch("0.0", "0")
      Call Form2.VentUserOR ' switches user override delay on
    End If
  End If
End Sub
```

Private Sub txtState_CloseBVIR_Change()

```
'routine to ensure the vent override timer is activated when the IR controller is used
  If txtState_CloseBVIR.Text = "0" Then Call Form2.VentUserOR
End Sub
```

Private Sub txtState_CloseBVSth_Change()

```
'routine to ensure the vent override timer is activated when the switch is used
  If txtState_CloseBVSth.Text = "0" Then Call Form2.VentUserOR
End Sub
```

Private Sub cmdOpen_BtmVent_Click()

```
'routine for when the user clicks on open bottom vent button in the VB user interface
If Form2.StatusLockBV = True Then
    Beep
    MsgBox "This vent is locked. You must unlock the vent before opening", _
        vbOKOnly + vbExclamation, "Bottom Vent"
  Else
    If Form1.PositionBV = 100 And ButtonOpenBV = False Then
      Beep
      MsgBox "The vent is fully open. It can not be opened anymore", _
          vbOKOnly + vbExclamation, "Bottom Vent"
    Else
      ButtonOpenBV = Not (ButtonOpenBV)
      If ButtonOpenBV = True And ButtonCloseBV = False Then 'stops both relays
going on
        Call Form1.OpenBVRelaySwitch("100.0", "1")
      Else
        ButtonOpenBV = False ' needed for when ButtonOpenTV = True
        Call Form1.OpenBVRelaySwitch("0.0", "0")
        Call Form2.VentUserOR ' switches user override delay on
      End If
    End If
  End If
End Sub
```

Private Sub txtState_OpenBVIR_Change()

```
'routine to ensure the vent override timer is activated when the IR controller is used
  If txtState_OpenBVIR.Text = "0" Then Call Form2.VentUserOR
End Sub
```

Private Sub txtState_OpenBVSth_Change()

```
'routine to ensure the vent override timer is activated when the switch is used
  If txtState_OpenBVSth.Text = "0" Then Call Form2.VentUserOR
End Sub
```

Private Sub cmdLock_BtmVent_Click()

```
'Routine to toggle the bottom vent lock button on and off and lock and unlock vent
  If Form1.PositionBV <> 0 Then
    Beep
    MsgBox "The vent must be fully closed before it can be locked", _
          vbOKOnly + vbExclamation, "Bottom Vent"
  Else
    ' Routine that stops the vent from being locked and unlocked at the same time
    If Form1.LockingBV = False Then
      Form2.StatusLockBV = Not (Form2.StatusLockBV)
      Form2.txtStatus_LockBV.Text = Form2.StatusLockBV
      Form1.LockingBV = True
      If Form2.StatusLockBV = False Then
        Call Form1.OpenBVRelaySwitch("100.0", "1")
      Else
        Call Form1.CloseBVRelaySwitch("100.0", "1")
    ' Reset to False at the end of the timer routine
      End If
    End If
  End If
End Sub
```

Public Sub VentUserOR()

```
'routine that ensures that the vent user over-ride timer always starts at the last user
override
  Form2.VentUserOveride = False
  Form2.txtStatus_VentOR = Form2.VentUserOveride
  Form2.VentUserOveride = True
  Form2.txtStatus_VentOR = Form2.VentUserOveride
End Sub
```

```
Public Sub CloseTVRelaySwitch(ValueA As String, ValueB As String)

'DDE link between VB program and the Close Top Vent Relay switch
  Form2.txtSth_CloseTVV = ValueA
  Form2.txtSth_CloseTVS = ValueB
  Form2.txtSth_CloseTVV.LinkPoke
  Form2.txtSth_CloseTVS.LinkPoke

End Sub

Public Sub OpenTVRelaySwitch(ValueA As String, ValueB As String)

'DDE link between VB program and the Open Top Vent Relay switch
  Form2.txtSth_OpenTVV = ValueA
  Form2.txtSth_OpenTVS = ValueB
  Form2.txtSth_OpenTVV.LinkPoke
  Form2.txtSth_OpenTVS.LinkPoke

End Sub

Public Sub CloseBVRelaySwitch(ValueA As String, ValueB As String)

'DDE link between VB program and the Close Bottom Vent Relay switch
  Form2.txtSth_CloseBVV = ValueA
  Form2.txtSth_CloseBVS = ValueB
  Form2.txtSth_CloseBVV.LinkPoke
  Form2.txtSth_CloseBVS.LinkPoke

End Sub

Public Sub OpenBVRelaySwitch(ValueA As String, ValueB As String)

'DDE link between VB program and the Open Bottom Vent Relay switch
  Form2.txtSth_OpenBVV = ValueA
  Form2.txtSth_OpenBVS = ValueB
  Form2.txtSth_OpenBVV.LinkPoke
  Form2.txtSth_OpenBVS.LinkPoke

End Sub
```

The User Interface Graphics

Private Sub txtStatus_BtmVent_Change()

```
'this routine changes the image of on the user interface to reflect the actual changes in
the bottom vent
 If Form1.PositionBV < 5 Then _
   imgBtmVent.Picture = ImageList2.ListImages(1).Picture
 If Form1.PositionBV >= 5 And Form1.PositionBV < 15 Then _
   imgBtmVent.Picture = ImageList2.ListImages(11).Picture
 If Form1.PositionBV >= 15 And Form1.PositionBV < 25 Then _
   imgBtmVent.Picture = ImageList2.ListImages(10).Picture
 If Form1.PositionBV >= 25 And Form1.PositionBV < 35 Then _
   imgBtmVent.Picture = ImageList2.ListImages(9).Picture
 If Form1.PositionBV >= 35 And Form1.PositionBV < 45 Then _
   imgBtmVent.Picture = ImageList2.ListImages(8).Picture
 If Form1.PositionBV >= 45 And Form1.PositionBV < 55 Then _
   imgBtmVent.Picture = ImageList2.ListImages(7).Picture
 If Form1.PositionBV >= 55 And Form1.PositionBV < 65 Then _
   imgBtmVent.Picture = ImageList2.ListImages(6).Picture
 If Form1.PositionBV >= 65 And Form1.PositionBV < 75 Then _
   imgBtmVent.Picture = ImageList2.ListImages(5).Picture
 If Form1.PositionBV >= 75 And Form1.PositionBV < 85 Then _
   imgBtmVent.Picture = ImageList2.ListImages(4).Picture
 If Form1.PositionBV >= 85 And Form1.PositionBV < 95 Then _
   imgBtmVent.Picture = ImageList2.ListImages(3).Picture
 If Form1.PositionBV >= 95 Then _
   imgBtmVent.Picture = ImageList2.ListImages(2).Picture
 imgBtmVent2.Picture = imgBtmVent.Picture

End Sub
```

Private Sub TxtStatus_TopVent_Change()

```
'this routine changes the image of on the user interface to reflect the actual changes in
the top vent
 If PositionTV < 5 Then _
   imgTopVent.Picture = ImageList1.ListImages(1).Picture
 If PositionTV >= 5 And PositionTV < 15 Then _
   imgTopVent.Picture = ImageList1.ListImages(11).Picture
 If PositionTV >= 15 And PositionTV < 25 Then _
   imgTopVent.Picture = ImageList1.ListImages(10).Picture
 If PositionTV >= 25 And PositionTV < 35 Then _
   imgTopVent.Picture = ImageList1.ListImages(9).Picture
 If PositionTV >= 35 And PositionTV < 45 Then _
```

```
    imgTopVent.Picture = ImageList1.ListImages(8).Picture
  If PositionTV >= 45 And PositionTV < 55 Then _
    imgTopVent.Picture = ImageList1.ListImages(7).Picture
  If PositionTV >= 55 And PositionTV < 65 Then _
    imgTopVent.Picture = ImageList1.ListImages(6).Picture
  If PositionTV >= 65 And PositionTV < 75 Then _
    imgTopVent.Picture = ImageList1.ListImages(5).Picture
  If PositionTV >= 75 And PositionTV < 85 Then _
    imgTopVent.Picture = ImageList1.ListImages(4).Picture
  If PositionTV >= 85 And PositionTV < 95 Then _
    imgTopVent.Picture = ImageList1.ListImages(3).Picture
  If PositionTV >= 95 Then _
    imgTopVent.Picture = ImageList1.ListImages(2).Picture
  imgTopVent2.Picture = imgTopVent.Picture

End Sub
```

```
Private Sub txtStatus_LockTV_Change()

'routine that changes the symbol on the top vent lock button
  If Form2.txtStatus_LockTV.Text = "True" Then
    Form1.cmdLock_TopVent.Picture =
LoadPicture("D:\mjs\DOE18\Programs\Secur02b.gif")
  Else
    Form1.cmdLock_TopVent.Picture =
LoadPicture("D:\mjs\DOE18\Programs\Secur02a.gif")
  End If

End Sub
```

```
Private Sub txtStatus_LockBV_Change()

'routine that changes the symbol on the bottom vent lock button
  If Form2.txtStatus_LockBV.Text = "True" Then
    Form1.cmdLock_BtmVent.Picture = _
LoadPicture("D:\mjs\DOE18\Programs\Secur02b.gif")
  Else
    Form1.cmdLock_BtmVent.Picture = _
LoadPicture("D:\mjs\DOE18\Programs\Secur02a.gif")
  End If

End Sub
```

```
Private Sub txtRotationBlind_Change()

'this routine changes the image of on the user interface to reflect the actual changes in
the blind's tilt angle
  If PositionBlind < 5 Then
    Form1.imgBlind2.Picture = ImageList3.ListImages(1).Picture
  Else
'this continues in the same vain as the vents but for many more pages because of the
many different blind position combinations
Form1.imgBlind1.Picture = Form1.imgBlind2.Picture

End Sub
```

```
Private Sub txtPositionBlind_Change()

'this routine changes the image of on the user interface to reflect the actual changes in
the blind's vertical position
  If MoveDown = True Then
    If PositionBlind < 5 Then
      Form1.imgBlind2.Picture = ImageList3.ListImages(1).Picture
    Else
'this continues in the same vain as the vents but for many more pages because of the
many different blind position combinations
Form1.imgBlind1.Picture = Form1.imgBlind2.Picture

End Sub
```

Appendix X Details of Learning System Exploration

Introduction

This appendix reviews work that was undertaken, using the Matlab Fuzzy Logic Toolbox, to determine the best components of a Fuzzy system to include within the Associative Memory Network control architecture proposed in Chapter 10. This part of the work forms a starting point for future development of a learning architecture for automated blinds.

The appendix starts by describing the various methods of mathematically representing fuzzy sets in a control system and goes on to experiment with the effect of various mathematical options that can be used to form the control system's relational surface.

Membership Functions and Fuzzy Sets

Discrete or Continuous Fuzzy Sets?

A fuzzy set can be represented by using either discrete points arranged into a matrix or a continuous set formed from a mathematical formula. Discrete matrices or look-up tables require large amounts of memory storage within the control micro-processor to provide acceptable membership curve representation, whereas continous sets require storage for just a few points per set and use mathematical formula to interpolate between these points.

The cost and performance issues related to additional storage requirement for discrete fuzzy sets, described in Appendix IV, and the added flexibility and managebility of continuous fuzzy sets led to a decision to use only mathematically derived continous fuzzy sets for any further investigations.

Triangular membership functions (2^{nd} Order –B Splines)

Triangular membership functions are the most popular form of fuzzy set membership functions. When using the mathematical formula for a continuous set, each set is represented by a three term Knot vector [*a,p,c*] (where *p* is usually the peak of the membership function, μ) that can be processed by a second order basis function. The basis function is as follows:[1]

$$\mu = N_h^2(x) = \frac{(x-a)}{(p-a)} N_{h-1}^1(x) + \frac{(c-x)}{(c-p)} N_h^1(x)$$

where $N_{h-1}^1(x) = \begin{cases} 1 & a \le x < p \\ 0 & \text{otherwise} \end{cases}$

and $N_h^1(x) = \begin{cases} 1 & p \le x < c \\ 0 & \text{otherwise} \end{cases}$

See Figure X.I for an example of the Relative Azimuth input space represented as triangular membership functions. The first set has a knot vector of (-101.25,-78.75,-56.25). The input space is divided to ensure the zones correspond to blind tilt angles of 22.5 Degrees, as used in the test room.

Generalised Bell Curve Membership functions

The second type of membership function reviewed in this study was the generalised Bell Curve function. This function also uses a three term Knot vector (*a,b,p*), which is mirrored on the peak line. The mathematical formula is as follows:[2]

$$\mu = \frac{1}{\left((1 + ABS((x-p)/a))^{2b}\right)}$$

See Figure X.I for an example of the Relative Azimuth input space represented as generalised bell membership functions. The first set has a knot vector of (-112.5,22.5,-78.75).

Gaussian Curve Membership Functions

The third type of membership function is the Gaussian Curve function which uses a two term Knot vector (σ,p), also mirrored on the peak line. The mathematical formula is as follows:[2]

$$\mu = e^{\left(\frac{-(x-p)^2}{2\sigma^2}\right)}$$

See Figure X.I for an example of the Relative Azimuth input space represented as Gaussian membership functions. The first set has a knot vector of (-89.5,-78.75).

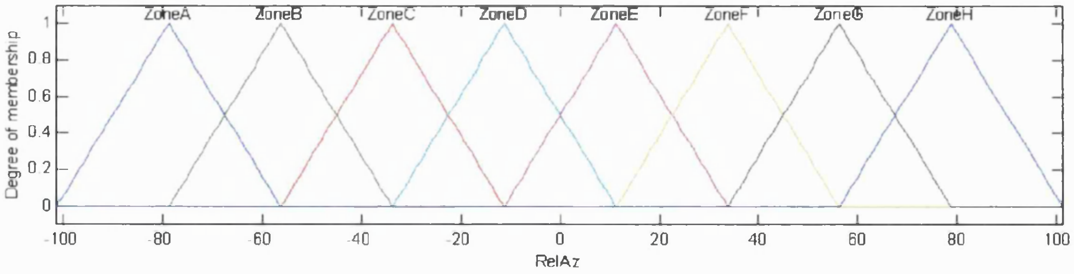
Sigmoid Curve Membership Function

The third type of membership function is the Gaussian Curve function which uses a four term Knot vector (a_1,p_1,a_2, p_2). The mathematical formula is as follows:[2]

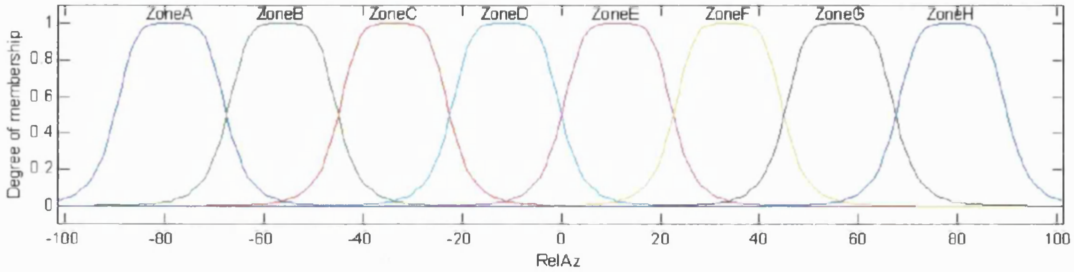
$$\mu = abs\left(\frac{1}{\left(1+e^{-a_1(x-p_1)}\right)}\right)-\left(\frac{1}{\left(1+e^{-a_2(x-p_2)}\right)}\right)$$

See Figure X.I for an example of the Relative Azimuth input space represented as sigmoid membership functions. The first set has a knot vector of (10,-90.0,10,-67.5).

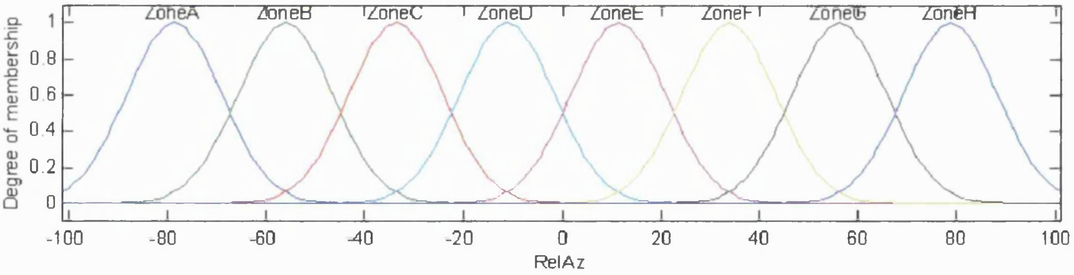
Triangular Fuzzy Membership Functions



Generalised Bell Fuzzy Membership Functions



Gaussian Fuzzy Membership Functions



Sigmoid Fuzzy Membership Functions

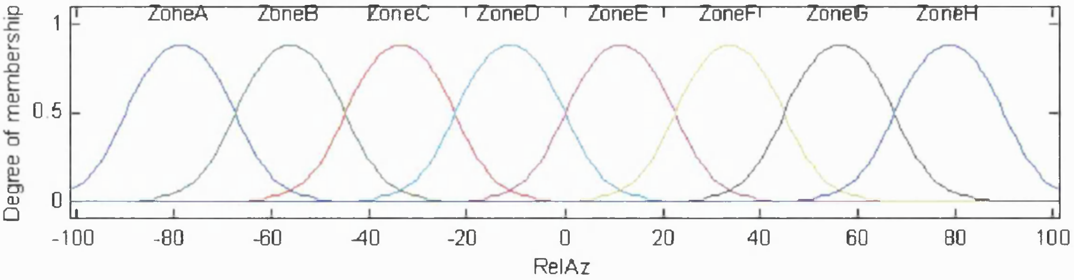


Figure X.1: The four different membership functions investigated representing the Relative Azimuth Input Space

Fuzzy Intersection (The 'AND' Statement)

The process of building knowledged-based systems from fuzzy sets requires the formation of linguistic statements that reflect an expert's knowledge. The first part of this process is to use the 'AND' statement, or fuzzy intersection, that links various fuzzy sets together in rules. For example in the framework proposed a zone of the sky is described as being a member of a relative azimuth set AND a solar wall altitude set.

Two methods of fuzzy intersection are commonly used: The 'Minimum Operator' and the 'Product Operator'.

Minimum Operator

The minimum operator is simply the minimum of the two fuzzy sets over a two dimensional universe of discourse.

$$\mu_{A \cap B} = \min(\mu_A, \mu_B)$$

Product Operator

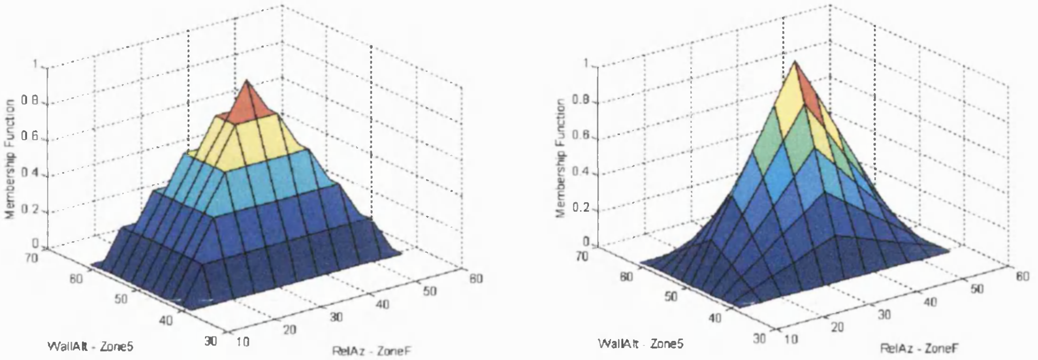
The product operator is simply the product of the two fuzzy sets over a two dimensional universe of discourse.

$$\mu_{A \cap B} = \mu_A \bullet \mu_B$$

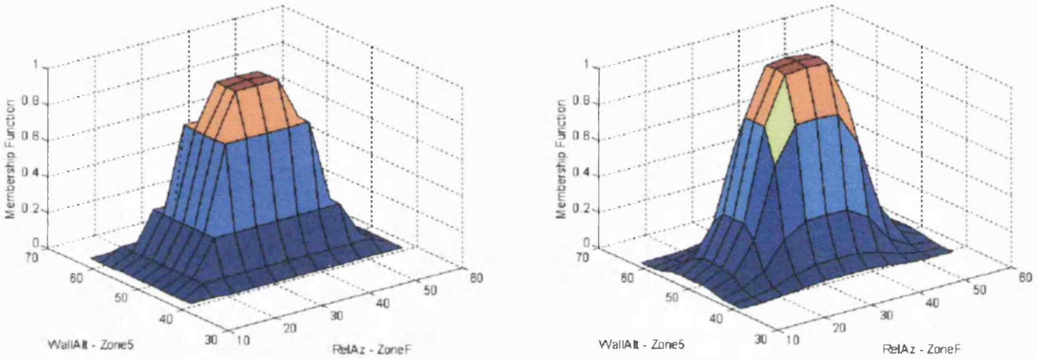
Each method has a slightly different effect on the multidimensional fuzzy set result. An illustation of this can be seen in Figure X-II, which shows the different shapes resulting from different fuzzy intersection operations on different membership function types.

Figure X-II: The different shapes formed by using the minimum and product methods on different membership function methods (next page).

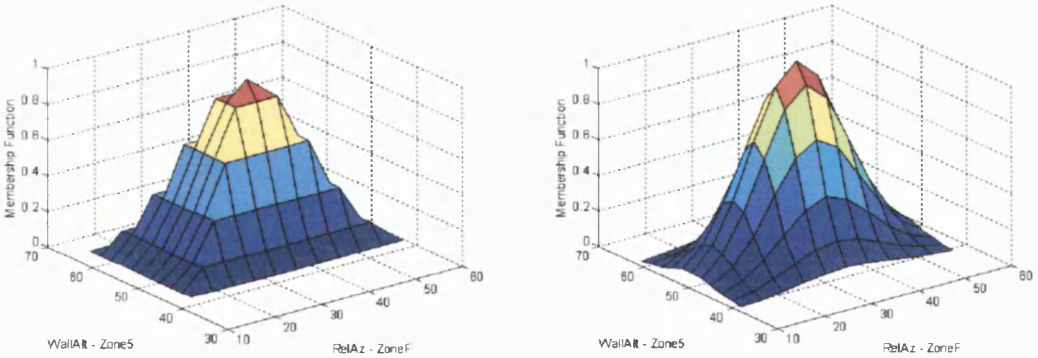
MINIMUM PRODUCT



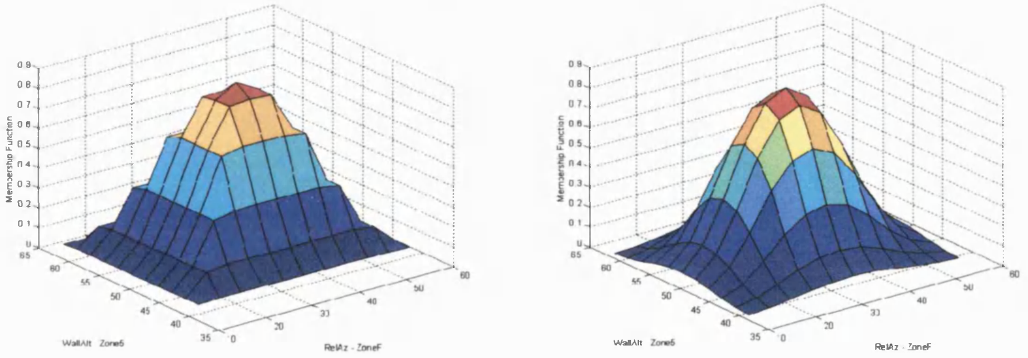
Second Order B-Spline Membership Functions



Generalised Bell-Shaped Membership Functions



Gaussian Curve Membership Functions



Sigmoidally Shaped Membership Functions

Fuzzy Union (The 'OR' Statement)

The second part of the rule making process is to use the 'OR' statement, or fuzzy union, to link single or multidimensional fuzzy intersections together. For example in the framework proposed, the blind could be lowered as a result of the sun being in Zone A OR Zone B OR Zone C.

There are two types of fuzzy union operators commonly used, the 'maximum' and the 'probabilistic OR'.

Maximum Operator

The most common form of fuzzy union is the 'Maximum Operator'. The maximum operator is simply the maximum of the two fuzzy sets over a two dimensional universe of discourse.

$$\mu_{A \cup B} = \max(\mu_A, \mu_B)$$

Probabilistic OR Operator

The Probabilistic OR is the algebraic sum of the membership functions of the sets being united minus the product of those sets.

$$\mu_{A \cup B} = \mu_A + \mu_B - (\mu_A \bullet \mu_B)$$

The effect of these two operators on the output surface will be shown in the next section

Fuzzy Implication (IF antecent THEN Consequent)

No fuzzy logic system would be complete without the fuzzy implication or rule that make up the rules of expert system itself, the IF …THEN statements.

These statements can be built by either using the 'Minimum Operator' or the 'Product Operator', both of which were used for the fuzzy intersection described earlier. The one difference that results from using these operators for fuzzy implication is that the output from their functions produces a relational surface rather than a membership function surface. Figures X-III to X-VI show the relational surface for Solar Wall Altitude Angle and Blind Angle implemented within the test room energy efficient control strategy, for a number of different combinations of implication and union operators. The figures show how the blind alters its angle as the sun rises and falls and hold its angle at horizontal when the sun rises above 45°.

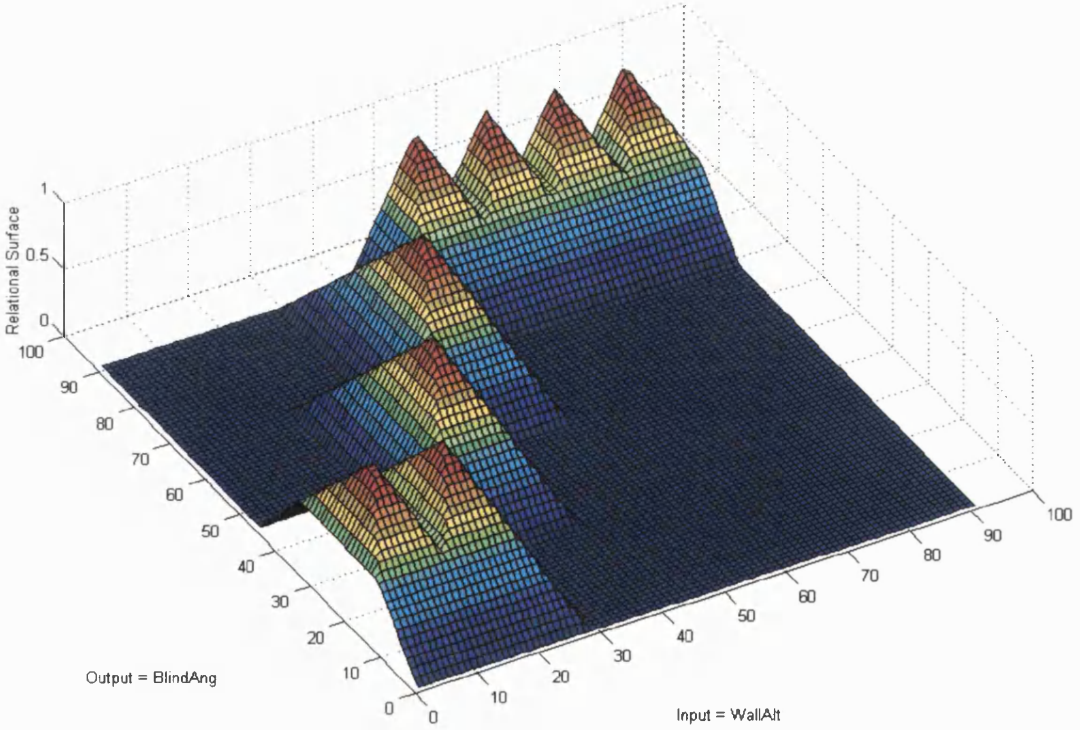


Figure X-III: Relational Surface formed using the Minimum Implication Method and the Maximum Union Method

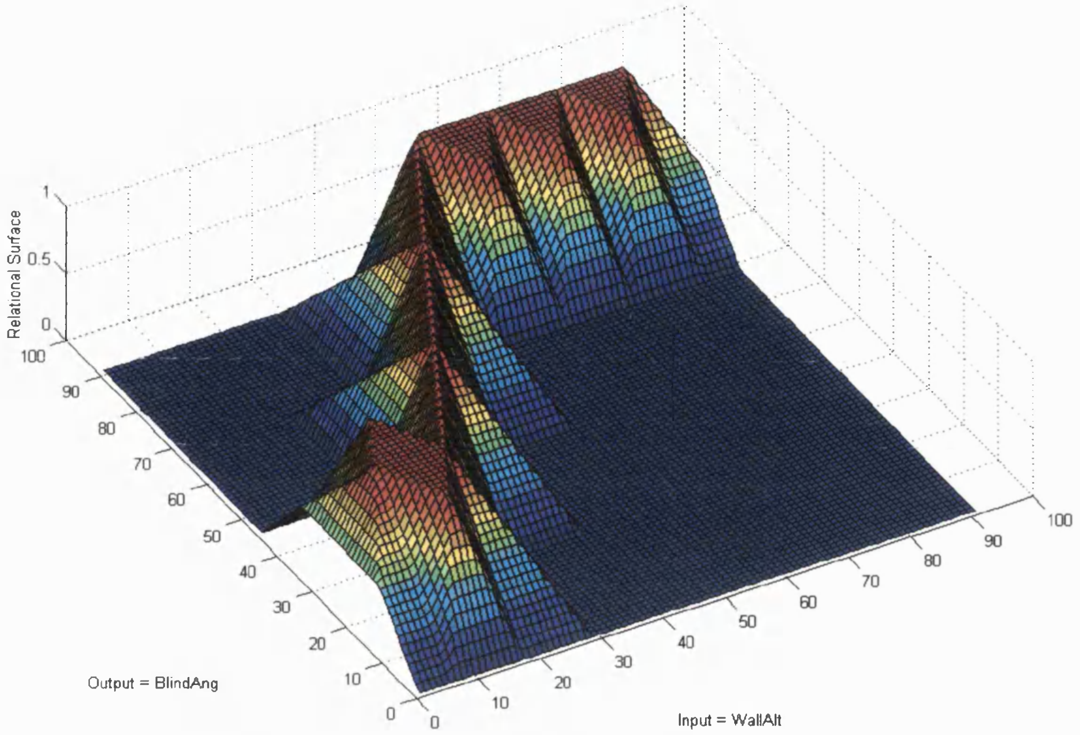


Figure X-IV: Relational Surface formed using the Minimum Implication Method and the ProbOr Union Method

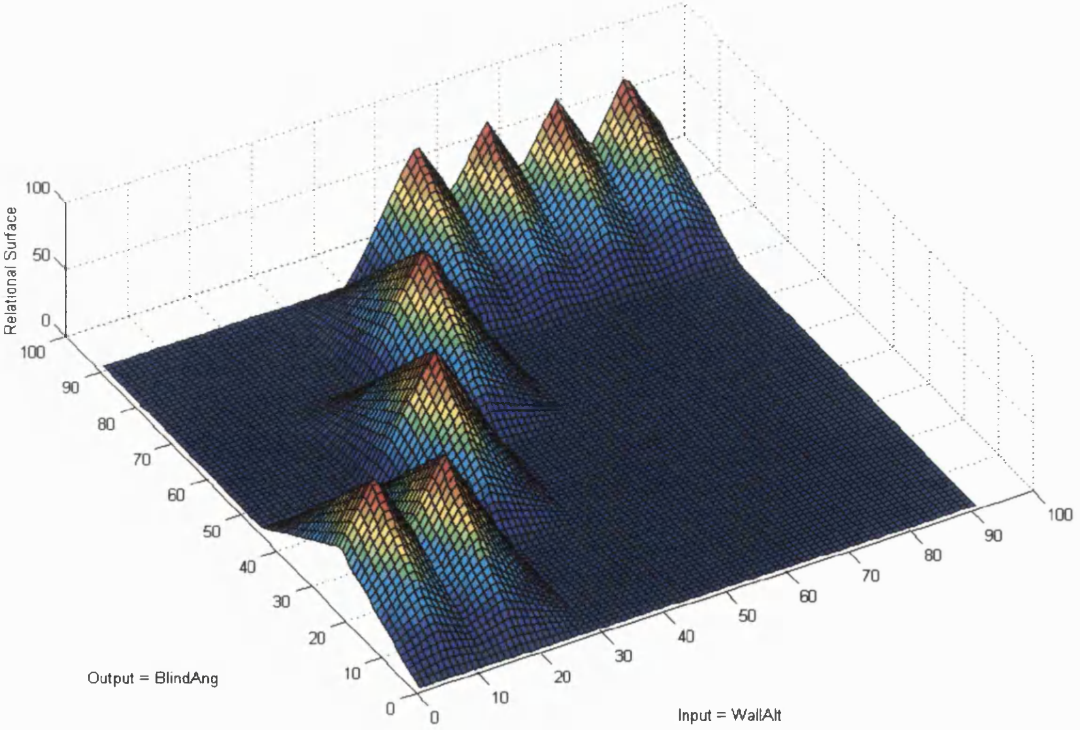


Figure X-V: Relational Surface formed using the Product Implication Method and the Max Union Method

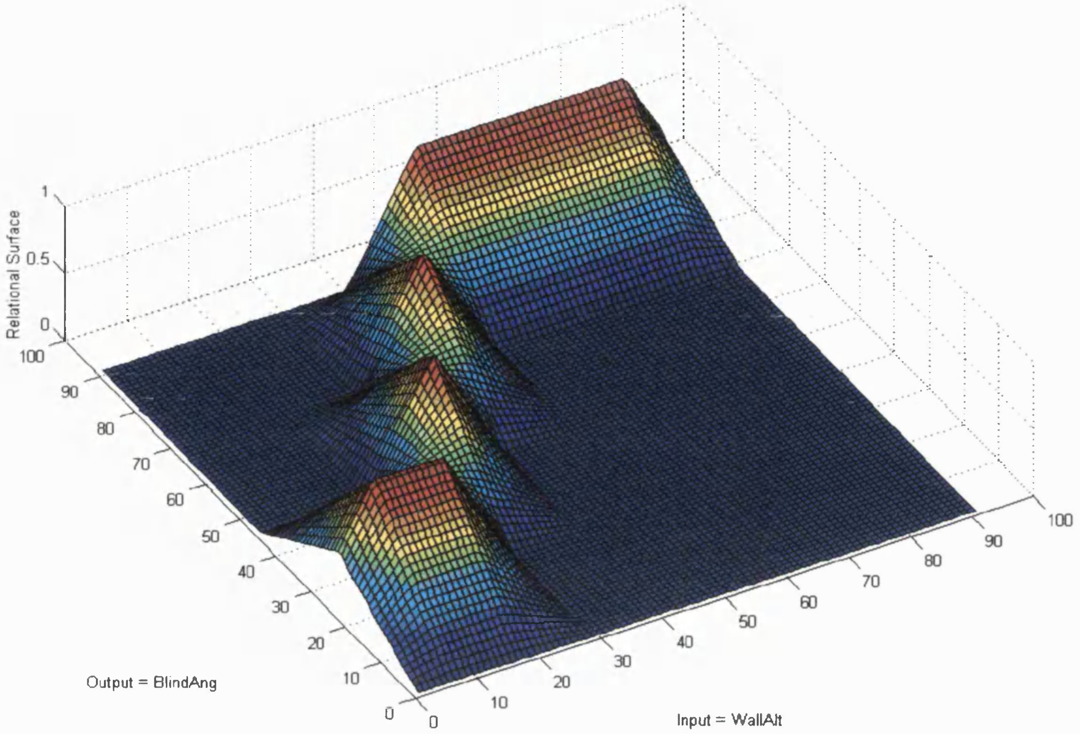


Figure X-VI: Relational Surface formed using the Product Implication Method and the ProbOr Union Method

Confidences

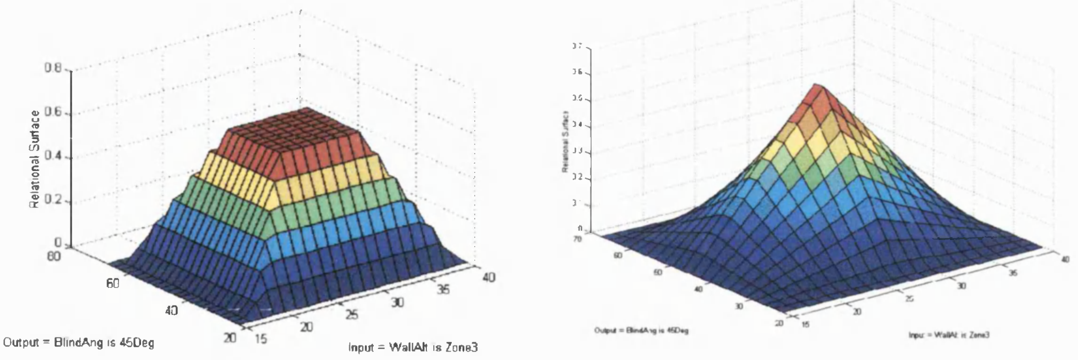
As discussed in Chapter Ten, the confidence of a rule being correct can be used to determine the nature of the final relational surface and adapt rules in real time control situations.

Figure X-IV shows how multiplying the rule by a confidence can effect the relational surface output for each of the types of membership functions considered in this Appendix so far.

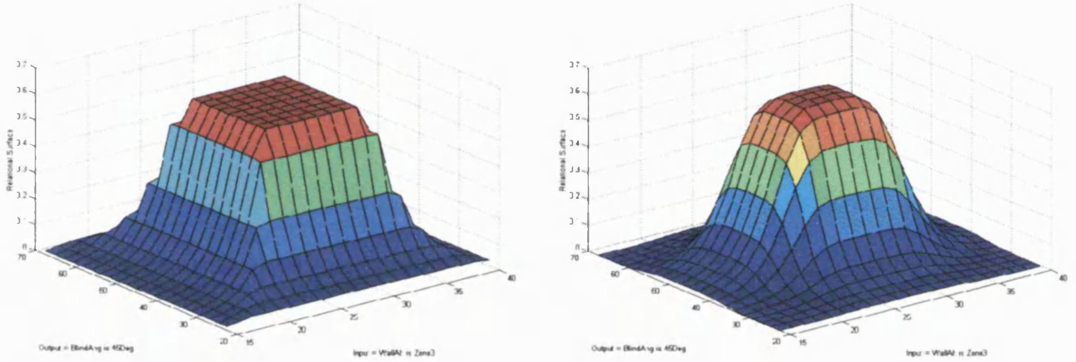
Figure X-VII: The effect of a confidence value of 0.6 on fuzzy inference for each type of membership function (see next page)

This is followed by Figures X-VIII to X-XI, which show the effect of altering the confidence of a rule on the relational surfaces illustrated in the last section.

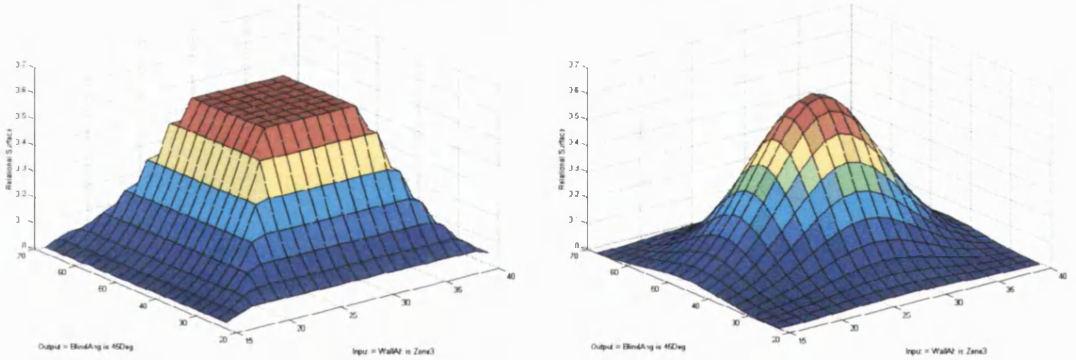
MINIMUM PRODUCT



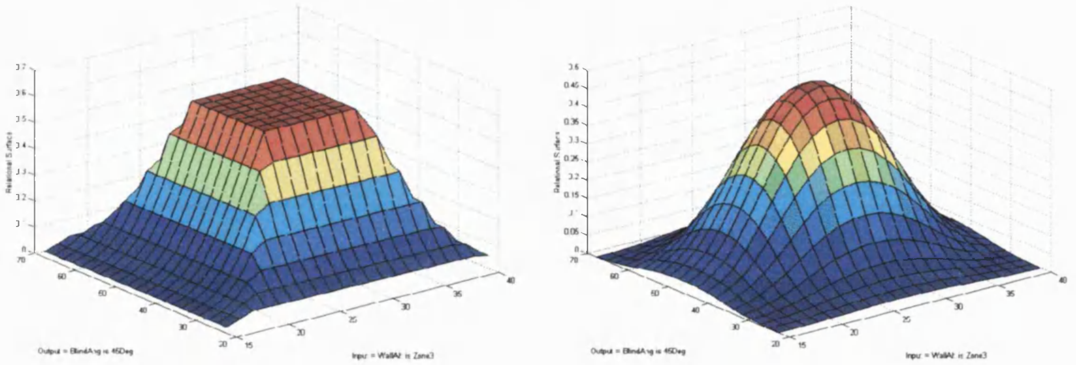
Second Order B-Spline Membership Functions



Generalised Bell-Shaped Membership Functions



Gaussian Curve Membership Functions



Sigmoidally Shaped Membership Functions

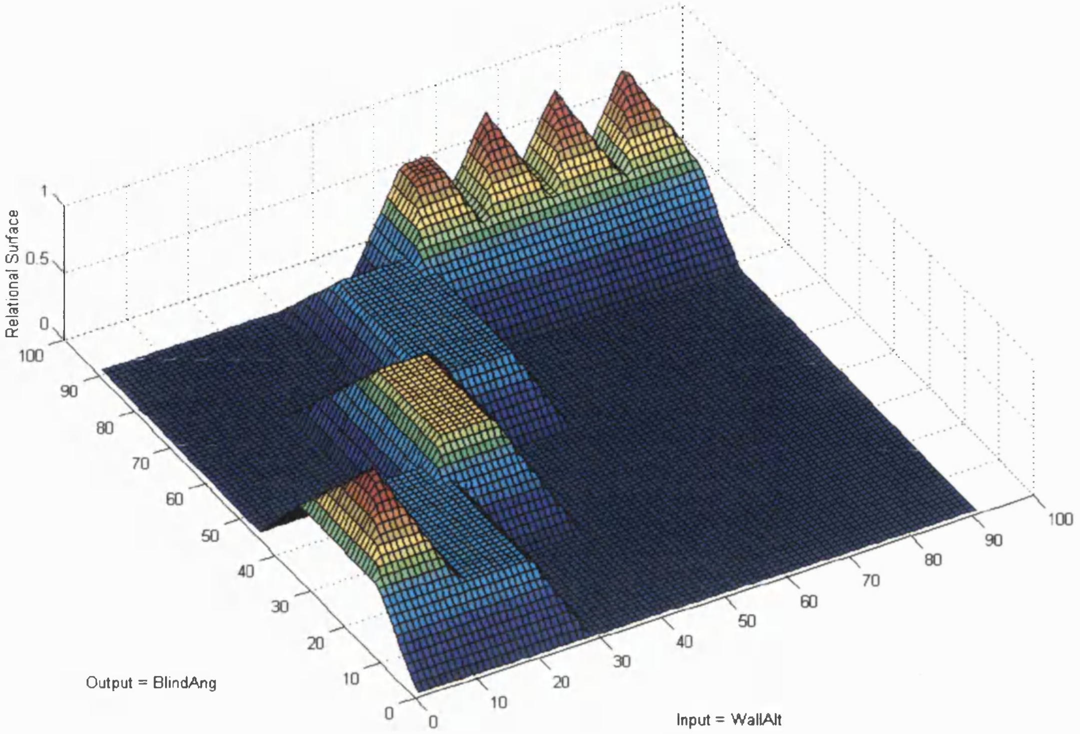


Figure X-VIII: Relational Surface formed using the Minimum Implication Method and the Maximum Union Method with variable rule confidences

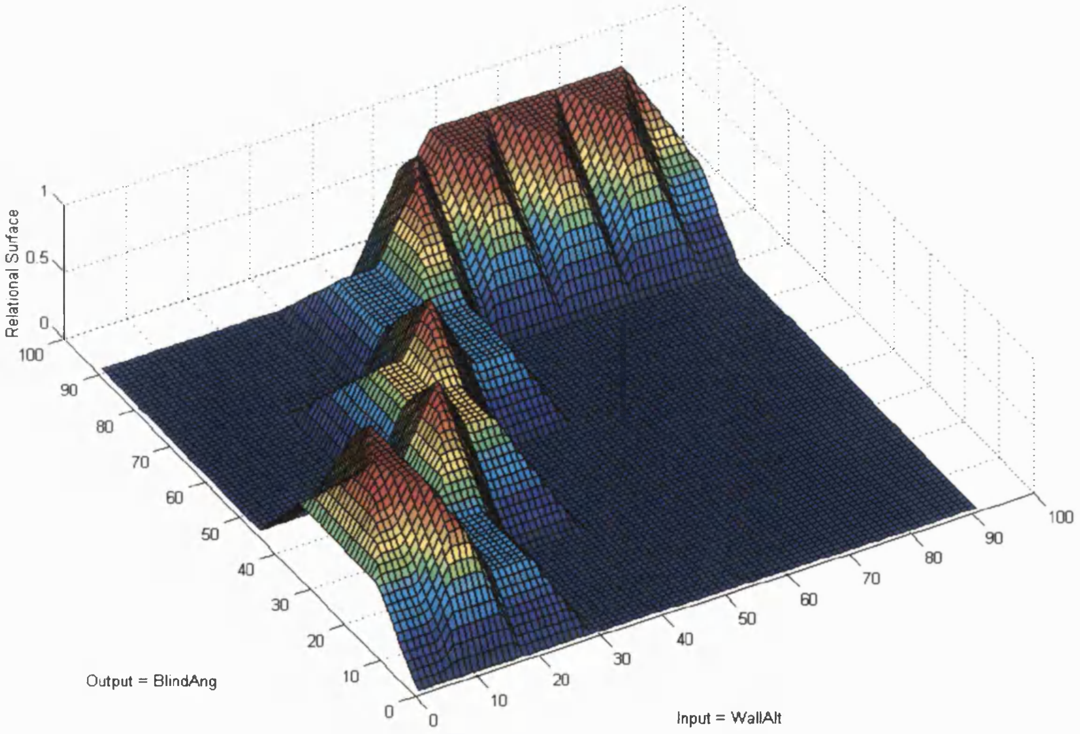


Figure X-IX: Relational Surface formed using the Minimum Implication Method and the ProbOr Union Method with variable rule confidences

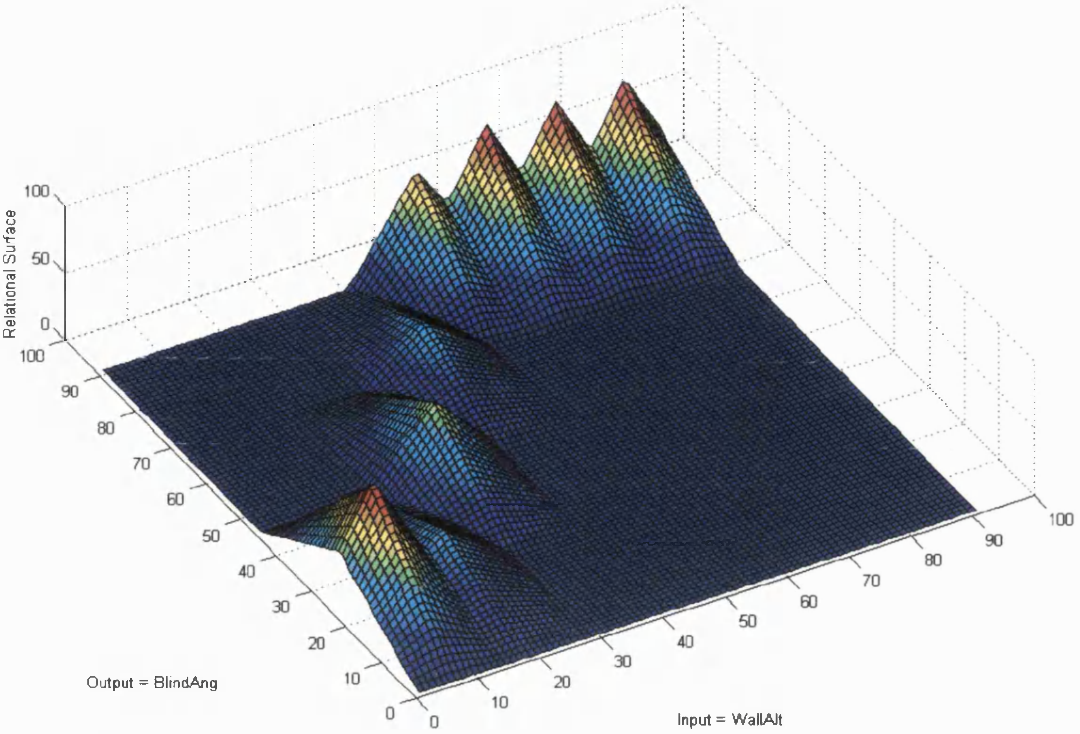


Figure X-X: Relational Surface formed using the Product Implication Method and the Maximum Union Method with variable rule confidences

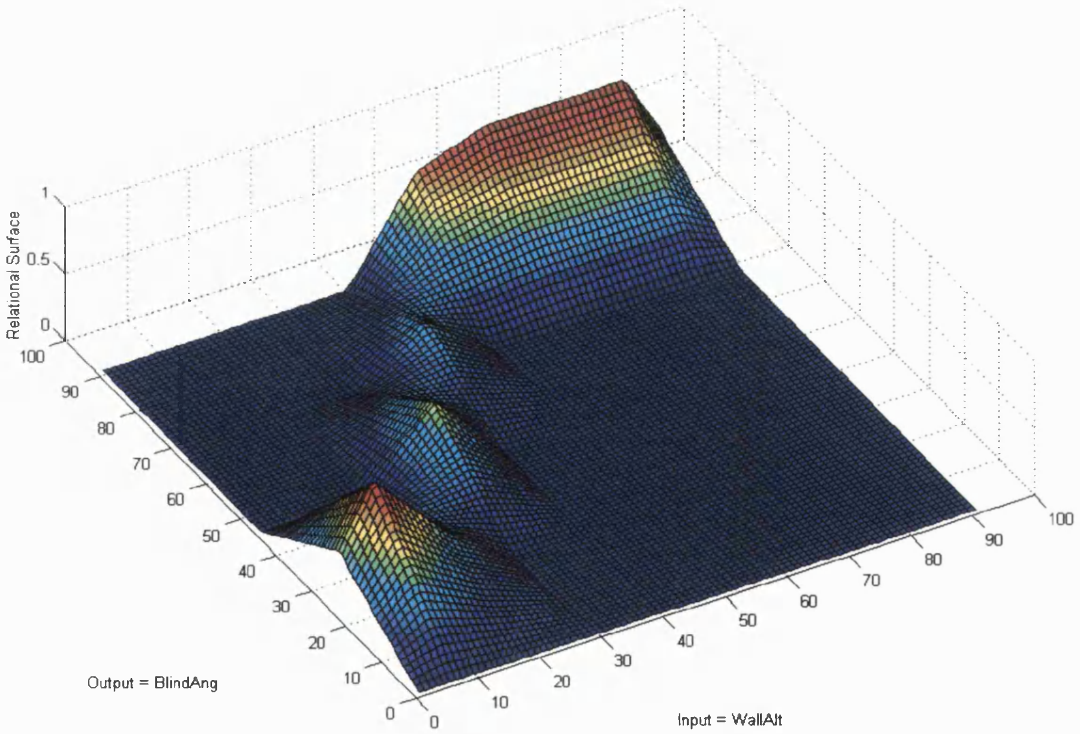


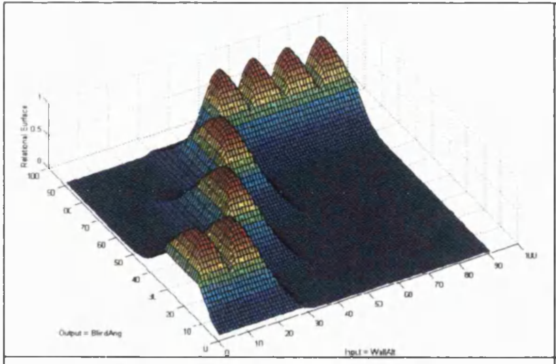
Figure X-XI: Relational Surface formed using the Product Implication Method and the ProbOr Union Method with variable rule confidences

The Effect of Various Operations on other Membership Function Types

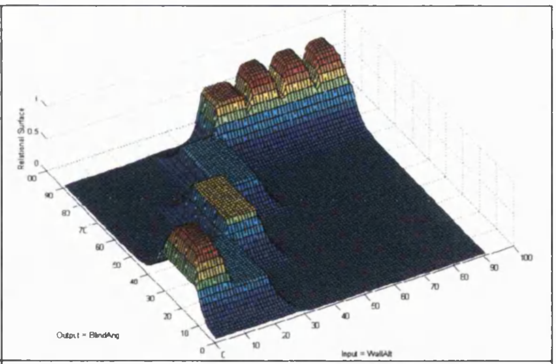
The effects of the operations discussed so far have been illustrated using the triangular membership function. This section gives a brief look at the effect the same operations have on the Generalised Bell, Gaussian and Sigmoidally Shaped membership functions. Figures X-XII to X-XIV summarise these three membership functions respectively.

By looking at these surfaces we can begin to see which combinations of operations will not be suitable for the control framework proposed in Chapter Ten. This will become clearer once the options for the defuzzification operation have been explored.

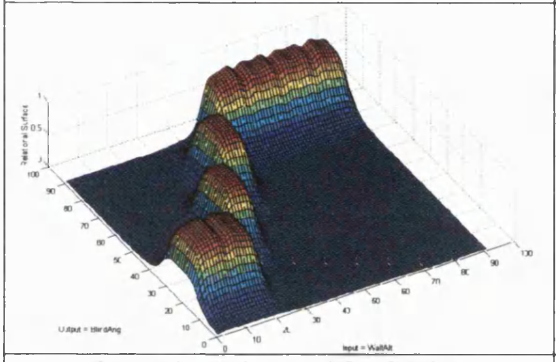
The three dimensional graphs shown in this appendix protray the relationship between an input and output space and the effect of various operations on that input and output space. However, it should be noted that the framework proposed and outlined in Chapter Ten has three inputs and two outputs resulting in a five-dimensional relational surface. This is obviously difficult to visualise and recreate and therefore the three dimensional surfaces are used purely to aid the author's and reader's understanding of what the system is capable of.



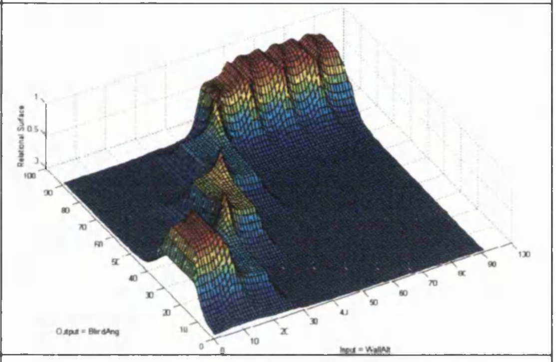
Imp=min, Union=Max, Conf=1



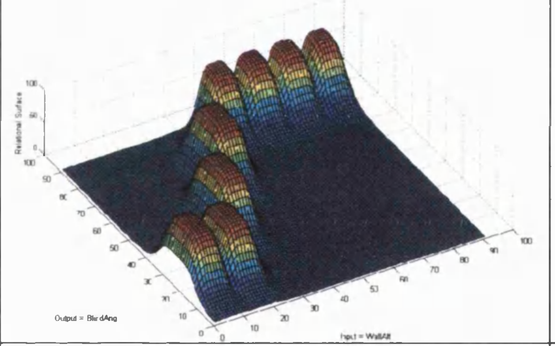
Imp=min, Union=Max, Conf=Vari



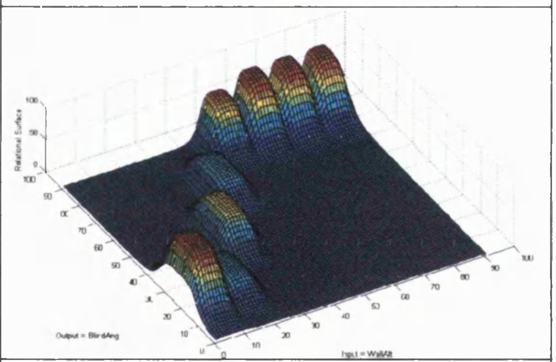
Imp=min, Union=ProbOr, Conf=1



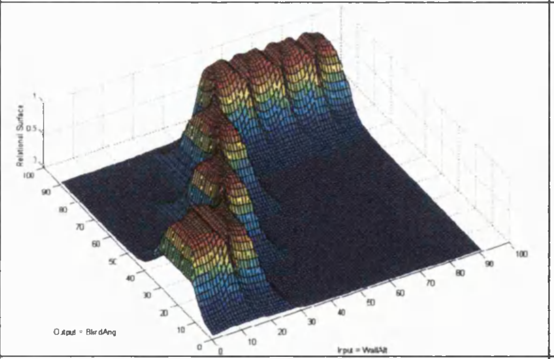
Imp=min, Union=ProbOr, Conf=Vari



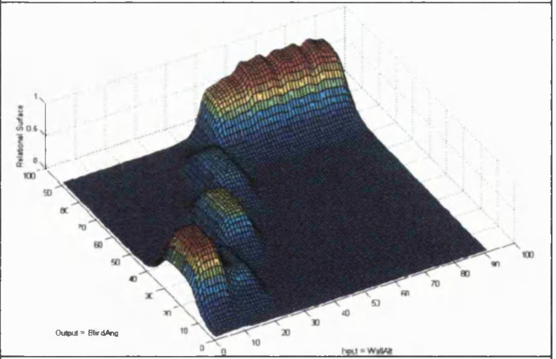
Imp=Prod, Union=Max, Conf=1



Imp=Prod, Union=Max, Conf=Vari

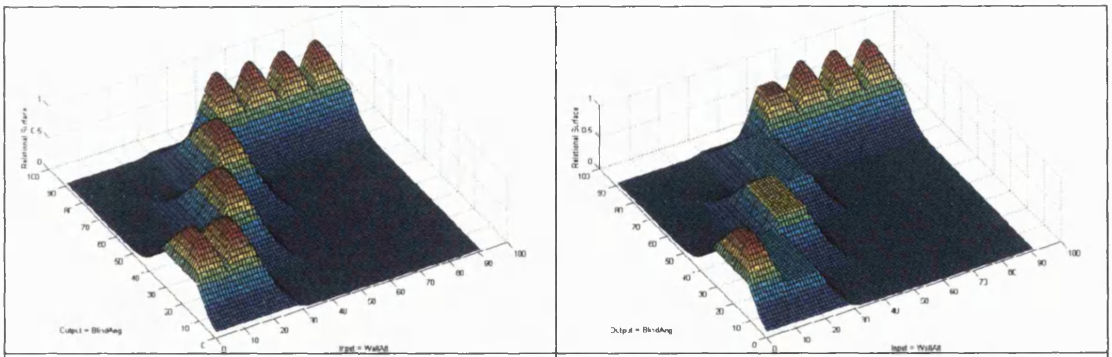


Imp=Prod, Union=ProbOr, Conf=1



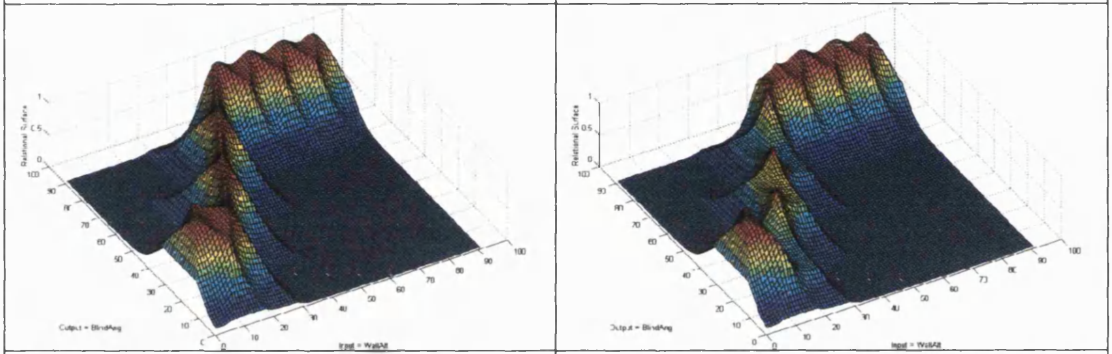
Imp=Prod, Union=ProbOr, Conf=Vari

Figure X.XII: Effect of Various Operations on the Relational Surface of Generalised Bell Shaped Membership Functions



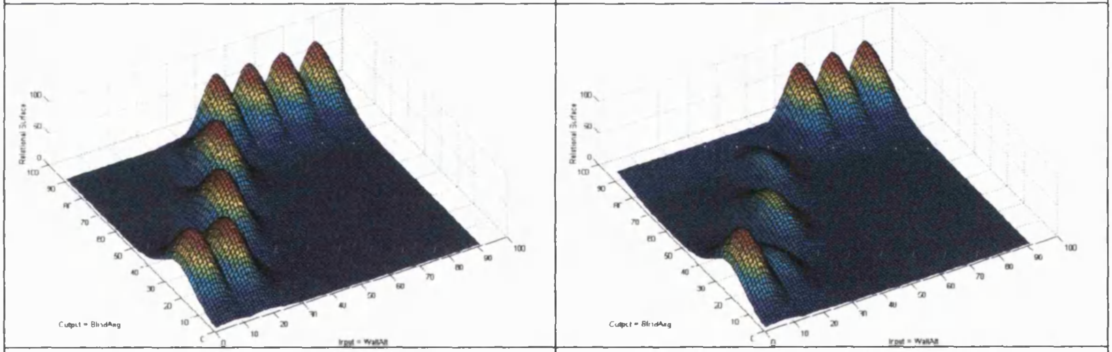
Imp=min, Union=Max, Conf=1

Imp=min, Union=Max, Conf=Vari



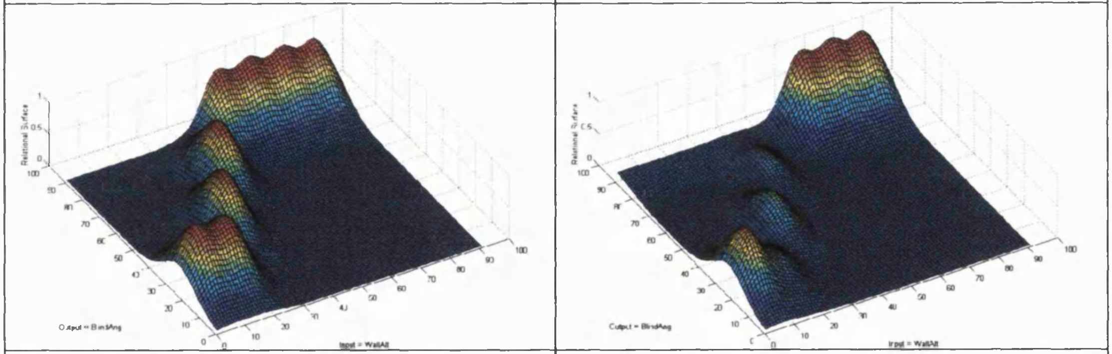
Imp=min, Union=ProbOr, Conf=1

Imp=min, Union=ProbOr, Conf=Vari



Imp=Prod, Union=Max, Conf=1

Imp=Prod, Union=Max, Conf=Vari



Imp=Prod, Union=ProbOr, Conf=1

Imp=Prod, Union=ProbOr, Conf=Vari

Figure X.XIII: Effect of Various Operations on the Relational Surface of Gaussian Curve Membership Functions

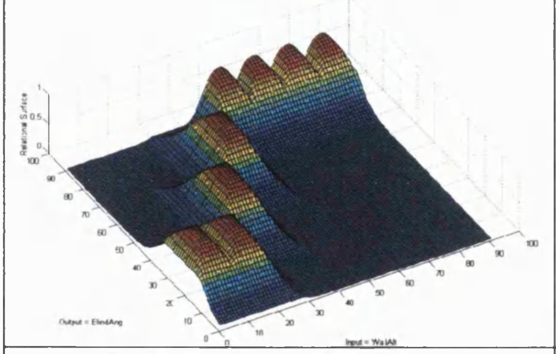
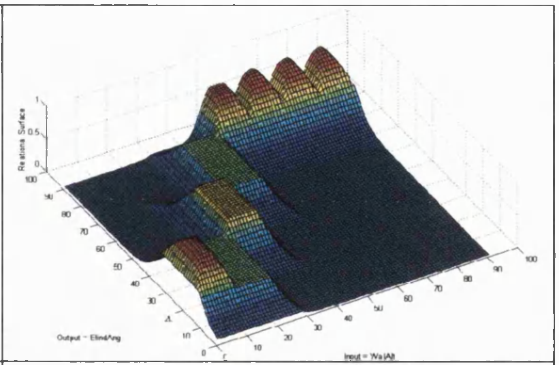
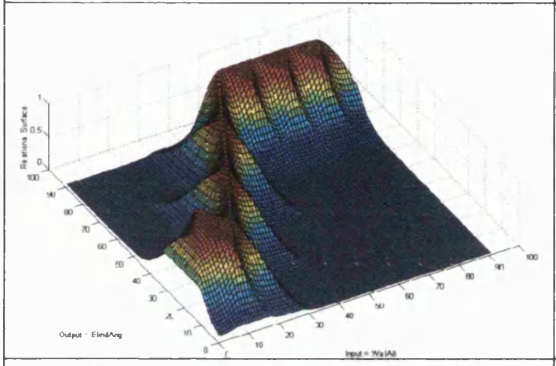
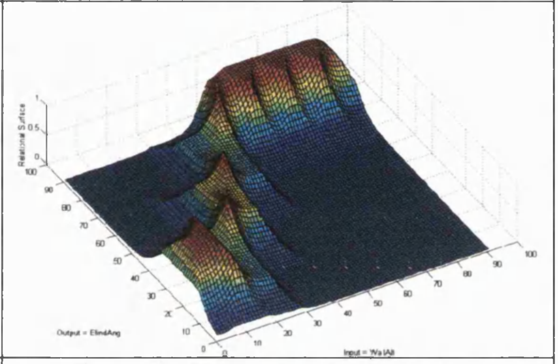
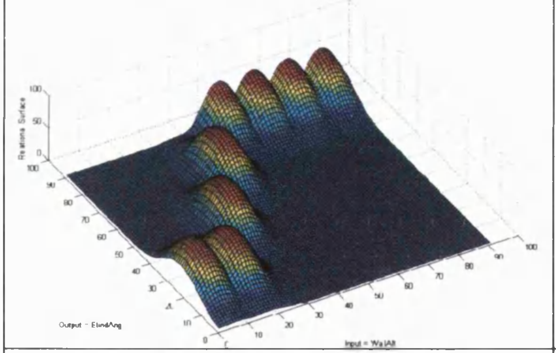
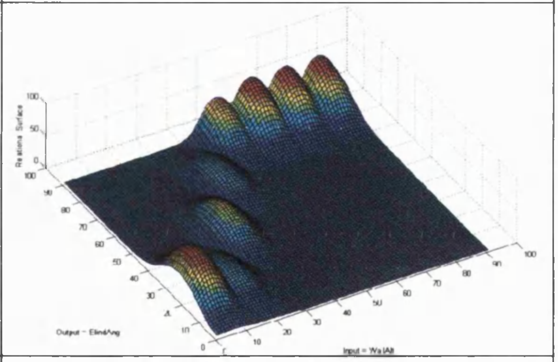
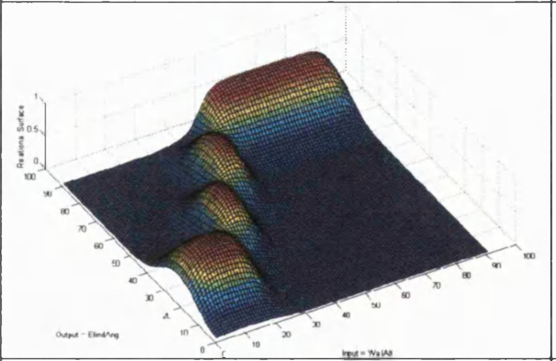
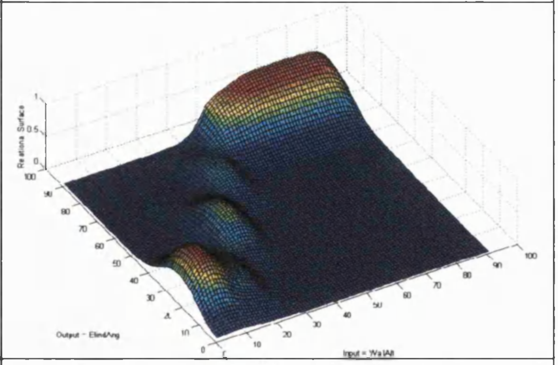
	
Imp=min, Union=Max, Conf=1	Imp=min, Union=Max, Conf=Vari
	
Imp=min, Union=ProbOr, Conf=1	Imp=min, Union=ProbOr, Conf=Vari
	
Imp=Prod, Union=Max, Conf=1	Imp=Prod, Union=Max, Conf=Vari
	
Imp=Prod, Union=ProbOr, Conf=1	Imp=Prod, Union=ProbOr, Conf=Vari

Figure X.XIV: Effect of Various Operations on the Relational Surface of Sigmoidally Shaped Membership Function

Defuzzification

Once a satisfactory relational surface has been established the control system is able to use that surface to determine the correct action from a series of inputs. This non-fuzzy singleton output is obtained by process called defuzzification and a number of different techniques are available.

Mean of Maximum Method

This method simply takes the mean of the output values that have the maximum membership/relational function.

Largest of Maximum Method

This method takes the largest output value that has the maximum membership/relational function.

Smallest of Maximum Method

This method takes the smallest output value that has the maximum membership/relational function.

Centroid Method

This method, also referred to as the centre of area method, determines the point that split the area under the output curve in half. The main disadvantge of this method is that it is slow to compute.

Figure X-XV shows the range of outputs that can be obtained for a particular relational surface by using various methods of defuzzification.

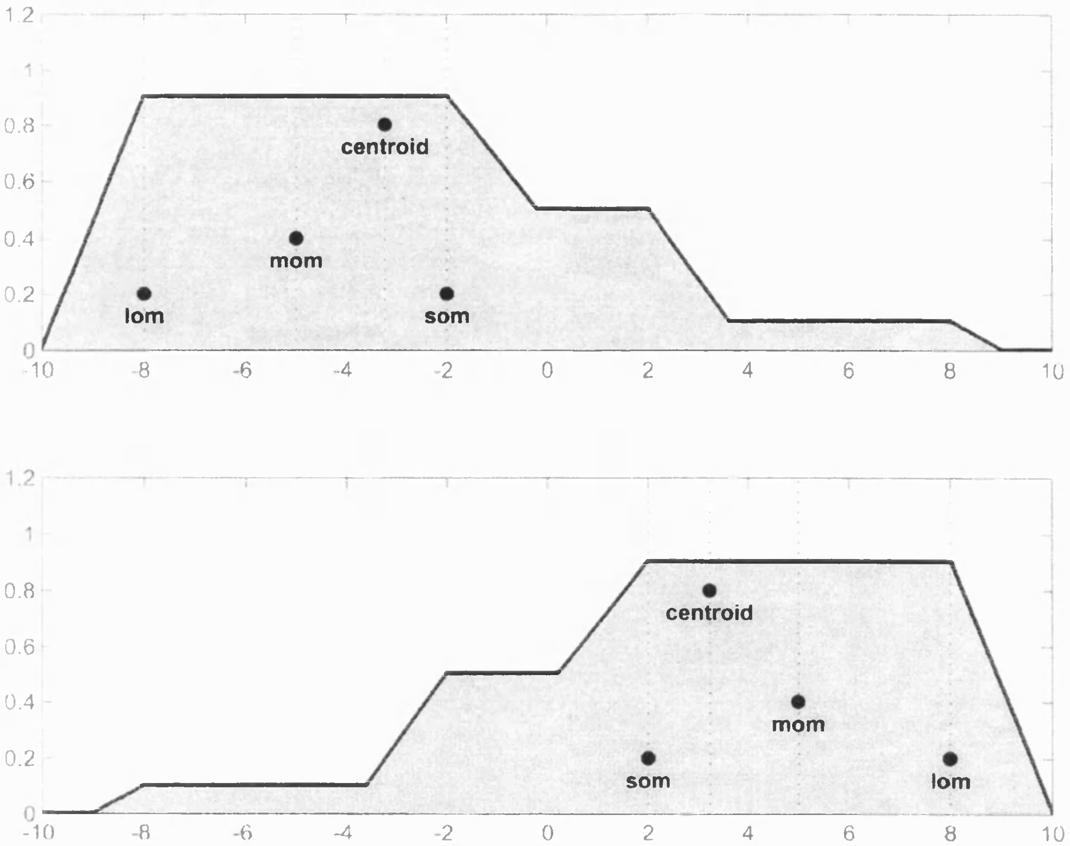


Figure X-XV: Diagrams showing the effect different defuzzification methods can have on an output value.[3]

Selecting a Suitable Method

A number of options are available to the designer when selecting fuzzy logic mechanisms to incorporate within the framework proposed in Chapter Ten. This appendix has shown that each option and its combination with other options can influence the nature of results likely to be obtained. Therefore before commencing with developing and demonstrating a simple version of the framework, a combination of fuzzy operators had to be selected.

The use of a Probability OR union with a Minimum Implication was shown to produce shape blips in the output surface after confidence adjustments had been made to the rules. These blips would confuse most defuzzification methods into selecting the incorrect output values and therefore were considered to be inappropriate for the framework required.

All membership function shapes showed potential for providing an adequate control framework for automated blinds. In particular the Generalised Bell Shaped Membership function showed great potential due it its simple formula and flat capped shape, which enhance the fine-tuning capabilities of the simple systems initially proposed. However, during the initial development stages, the precise shape of the membership curves are not as important as the approximate placement of the curves on the universe of discourse. The number of curves, partitions used and the overlapping character are important factors that should be investigated fully in future work. Therefore it was thought that extending beyond the simplicity of traditional triangular membership functions was not warranted at this stage.

Although the Product Implication method with both the Max and the Probability Or Union methods have potential to be very effective, the more common max-min method of Union and Implication was chosen for its simplicity and computational efficiency.

These conclusions were also reached by Motorola, who until recently were one of the prime producers of Neuron Chips, the heart of the LonWorks system. To aid the use of Fuzzy logic on these eight-bit microcontrollers, Motorola released a Fuzzy Logic Kernel, written in Neuron C, which performed the three basic steps of fuzzification, rule evaluation and defuzzification.[4]

Fuzzification method allowed for triangular and truncated membership functions to be used within the system. The rule evaluation used the Max-Min method which was shown to be quite simple to implement, as it was efficient in timing and code size, and therefore provide powerful and rigorous solution for a Neuron Chip 8-bit embedded controller. The defuzzification process used was the centre of gravity method rather than the mean of maxima method.

The Motorola kernel demonstrates that fuzzy logic operations can be impliment within a LonWorks node. It also shows that a simple fuzzy system, with 2 inputs, each with five membership functions, 20 rules and one output with five membership functions can be processed in about 20ms. Therefore a system the size of the proposed framework should run at under a second, which is more than fast enough for the nature of control required, as network communication is managed by the other two 8-bit processors.

Also if once developed it becomes apparent that the framework is too large to fit on a standard Neuron Chip. The option is available to use the Neuron Chip as a co-processor to a main host processor that runs the application code and provides more non-volatile memory for confidence and weight adjustments. This technique is common amoungst LonWorks Programmable Logic Controllers and Controllers that deal with more complex control tasks and does not have a huge influence on component cost.

---

[1] Brown M. and Harris C., 1994, "Neurofuzzy Adaptive Modelling and Control", Prentice Hall, Hemel Hempstead, UK.

[2] Anon, 1999, "Fuzzy Logic Toolbox: For use with Matlab", User's Guide, Version 2, The Mathworks Inc.

[3] Anon, 1999, "Fuzzy Logic Defuzzification Example", Matlab Software, Version 5, The Mathworks Inc.

[4] Anon, 1997, "Fuzzy Logic and the Neuron Chip", Motorola Semiconductor Technical Data, AN 1225, In the Appendix of "The LonWorks Technology Device Data Book" Rev 4, 1997, AL-338-356.

Appendix XI Full Acknowledgements List

In addition to those mentioned in the acknowledgements, I would like to take this opportunity to thank:

Ray Elliot of Taywood Engineering and **Richard Harris** at Messrs Sandberg
For their support of the CWCT project.

Ken Perry of Taywood Engineering
For his help with setting up the control panel in the test cell and for introducing the author to both the LonWorks technology and many of the manufacturers who eventually leant their support to the project.

Darren Pogson of Somfy
For supplying all the blind control hardware and external sensor equipment.

All at Echelon UK
For providing equipment and training for the project at reduced rates.

Dave Morgan at Wagner UK Ltd. and **Paul Gerrard** at Geze UK Ltd
For supplying the new window frame and all of its operating components.

Andrew Deeming and **John Webb** of Home Automation Ltd
For supplying the lighting control hardware and lighting sensors

Paul Booth and **Tony Wilson** of Granges
For providing the test room window

Owen Howlett of Zumbtobel Lighting
For providing the light fittings and taking such an active interest in the work.

Appendix XII A List of Papers, Posters, Articles Presentations and Prizes Resulting from the Work

A list of papers, posters, articles, presentation and prizes resulting from the work outlined in this thesis is given below:

Date	Title	Type
10/98	'The Individual and The Intelligent Facade', European Intelligent Building Group Conference entitled "Intelligent Buildings: Realising the Benefits", BRE, 6-8 October 1998.	Poster / Paper
11/98	Awarded 1st prize in the Building Research and Information International PhD Essay Writing Competition for my short essay entitled 'The Individual and the Intelligent Façade'.	Prize
02/99	'Automated Shading Devices: Control Issues', CIBSE Natural Ventilation Group seminar entitled "Beyond the Intelligent Façade", Chartered Institution of Building Services Engineers, Balham, 5 February 1999	Presentation
02/99	'The Control of Adaptive Facade Elements', Research Focus – a short article appeared in the Feb 1999 edition	Article
03/99	'The Application of Learning Systems to the Control of Adaptive Facade Devices', CIB conference entitled "Intelligent and Responsive Buildings" Brugge, 29-30 March 1999	Presentation / Paper
04/99	'The Intelligent Facade' presented at the CWCT conference entitled "Glass in Buildings", Bath, 31 March-1 April 1999.	Poster
10/99	'The Integration of Occupant Control within an Automated Blind Control Strategy and the Application of Learning Systems', CIBSE National Conference, 3-5 October 1999, pp327-337.	Poster / Paper
12/99	'Blinded by the Light', CIBSE Journal, December 1999, pp30-34	Article
12/99	'Results from a Study at the University of Bath', Meeting on 'Shading Devices', CIBSE Daylighting Group, CIBSE, Balham, 7th Dec 1999.	Presentation
01/00	'Individual and the Intelligent Facade', *Building Research and Information*, Jan/Feb 2000	Paper

cont'

Date	**Title**	**Type**
02/00	Named as one of four CIBSE Young Lighters of the Year 2000.	Prize
03/00	'Adapting Energy Efficient Automated Venetian Blind Control to Suit Individual Needs using Learning Algorithms', CIBSE Lighting Division Meeting, Institution of Structural Engineers, 21st March 2000,	Paper/ Presentation
04/00	'The Thinking Skin', Presentation at the CWCT Member's Assembly, 10th April 2000.	Presentation
05/00	'Intelligent Controls – Application to Shading Devices' CIBSE Natural Ventilation Group Seminar entitled "Operating Natural Ventilation" 16 June 2000.	Presentation
04/01	'The Evolution of Interactive Facades: Improving Automated blind Control' CWCT conference entitled "The whole-life performance of facades", Bath, 18/19 April 2001.	Paper / Presentation
TBP	Paper to be published in Journal of 'Light and Lighting'	Paper

In addition to these outputs, other outputs include:

- lectures have been given by the author on the subject of automated facades in various courses arranged by CWCT for industry and in the MSc in Façade Engineering run at the University of Bath;
- the CWCT report outlined in Appendix VII;
- demonstrations of the test cell in operation to a large number of visitors during the course of the study. Those visitors were from all aspects of the building industry, architects, engineers, contractors, clients etc.