2002

Fuzzy control implementation of a home automation system.

Viswanathan. Hariharan
University of Windsor

FUZZY CONTROL IMPLEMENTATION OF A HOME AUTOMATION SYSTEM

By

Viswanathan Hariharan

A thesis
Submitted to the Faculty of Graduate Studies and Research
Through the Department of Electrical Engineering
In Partial Fulfillment of the Requirements for the Degree of Master of Applied Science
At the
University of Windsor

Windsor, Ontario, Canada

2002

0-612-75789-7

ABSTRACT

Traditionally, home automation systems use rudimentary forms of instrumentation and control. However, tremendous potential in this field coupled with research has produced more sophisticated solutions.

This paper presents a project that provides a comprehensive home automation system using fuzzy logical control and 2-way voice communications. The fuzzy logic rules are based on human experience. The project has typical applications in a home (or in office) including;

- Lighting control system that automatically control lights [10]. This is based on fuzzy controller wherein the controller automatically takes into account the occupancy level and the ambient lighting conditions to compute the crisp fuzzy output.
- Temperature control system (heating and air-conditioning) that provides comfortable climate inside the home. Fuzzy logic controller provides automatic temperature control. The controller output is based on the temperature setpoint, current temperature and the rate of change in temperature.
- Safety system to monitor devices like smoke detector and carbon monoxide detector. At alarm condition, the control logic will shut off the heating and air conditioning system.
- Water heater controller that monitors the operation of water heater. Operation of the water heater is based on water temperature, the occupancy of the house and flow of water across the pipes.

The digital communication for all discrete controls is based on X10 protocol, while the analog input for the Temperature control is based on 1-wire technology[1]. A desktop PC (Personal Computer) provides supervisory control. Additionally, the system can be interfaced via the Voice controller that provides interaction with the system through human voice. System commands can be input through the voice commands. Since the human voice pattern is unique, the software provides training to improve voice recognition.

The code is written in Visual Basic 6[2] using Active X controls. Suitable user-friendly graphics are presented to operate the system easily.

The goal of the system is to provide reliable control to the common and important household devices and also to conserve energy. This method of control can be used for homes, business establishments or other areas. Depending on the application, the system has to be fine tuned to suit the requirements.

[1] 1-Wire technology is a trademark of Dallas-Maxim Semiconductor Inc.,
[2] Visual Basic is a product of Microsoft Corporation.

ACKNOWLEDGEMENTS

This thesis has been possible by the support and guidance of my supervisor Dr. H. K. Kwan, in particular, for suggesting me this Home Automation project with fuzzy control and voice interface components.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

CHAPTER – 1

1.0 INTRODUCTION

The conventional methods of using feedback controllers do not produce the required control performance for an application such as home automation system. The control application like lighting control is a classic case of a non-linear system that can be effectively tackled by a Fuzzy controller. Temperature control based on fuzzy methods provides good control over the entire operating range. The challenge of such a system is to provide optimal results under different real life scenarios dealing with the human lifestyle. These issues are addressed in this thesis project.

1.1 What is new about this home automation system?

The home automation industry has revolutionized our living environments by coming up with many smart products. PC based control is a successful strategy that is implemented in many industrial applications and can be extended to home automation also. With the multitude of products available in the arena of home or building automation, it is possible to develop an intelligently controlled system.

In the case of lighting control, the system faces challenge from Mother Nature in the form of varying intensity of light. Typically, no two days are identical, and the control strategy good for one day may not be the best for another. The other uncertainty faced by the system is the inconsistent pattern of human behaviour. One way to handle this problem is to modify the control parameters to suit the requirements. This condition necessitates that the home occupants need software knowledge, and time to make these changes constantly. How about a system that constantly updates itself to the natural surrounding to perform the required tasks? This is the type of control that is being focused on this thesis. The fuzzy logic takes into account all the natural uncertainties to provide optimum control.

Most of the temperature controllers used in homes are On –Off controls. Temperature control is a classic application of fuzzy control.

While in many homes, smoke and carbon monoxide detectors exist as an individual entity, this system provides a totally integrated solution. This approach increases the safety of the occupants.

Finally, the system is integrated with a 2-way voice communication interface for commands and feedback. The voice communication is based on an open platform and can be expanded, or modified by means of software.

1.2 What are the tasks for the system?

The tasks of the system are:

- Fuzzy Controller for Lighting System (Chapter 4).
- Fuzzy Controller for Temperature Control System (Chapter 5).
- Water Heater Controller (Chapter 6).
- Safety (Monitoring) (Chapter 7).
- 2-way voice communication (Chapter 8).

The fuzzy control shall be based on linguistic rules derived from human behaviour

It shall be possible to over-ride the fuzzy control and switch to manual operation if desired.

If the system is operated under "Fuzzy control", then the output is derived from the controller inputs, and as dictated by the Fuzzy rules.

The safety system constantly monitors the various systems inside the house. Any abnormal situation will result in the shutdown of the Temperature Control system to prevent circulation of bad gas in the duct.

Lastly, important controls can be accessed by voice control that is based on a 2 way voice communication system.

The controls are coded using Visual Basic software and reside in the PC (Personal Computer). The controls are supported by user-friendly graphics.

1.3 What are the benefits of fuzzy control?

The fuzzy control provides superior control strategy for non-linear systems. While this is true for both Lighting control and Temperature control that use Fuzzy control in this thesis project, some of the hidden benefits of a Fuzzy lighting control are discussed below.

- **Economy**

The operation of the devices at the near bottom line level results in energy savings. With the energy costs soaring, substantial savings are achieved over a period of time.

- **Life span of Equipment**

The intention of the fuzzy system shall be to operate the equipment on an 'as and when required' basis at the same time absolutely maintaining the comforts of the occupants. This pattern of operation enhances the longevity of the equipment that could translate to dollars over a period of time.

➢ **Reliability**

The system is reliable, rugged and repeatable to provide sustained operation. The X10 devices have been serving the home automation industry for several years, and are a proven one.

➢ **Security**

In a typical home, it is often a practice to leave a few lights ON when the family is away. It certainly gives a feeling to potential burglars that people are around and maybe, in a few cases dissuade them. By choosing the lamp that should be operational, the fuzzy system shall maintain the minimum lighting in the absence of daylight even when the home is unoccupied. This minimum level of lighting can be adjusted to the required level of brightness.

➢ **Ease of installation & Maintenance**

The discrete communication network between the devices and the PC is based on X10 protocol. This means that no additional wiring is necessary. The X10 devices[3] are easy to install and fit on to any standard power outlet socket. Once installed no maintenance is envisaged for the devices. The operation life for these devices are reasonably good and most of them are available off-the shelf.

[3] X10 devices are used in the areas of discrete control. Analog input for the temperature is derived from the sensor that communicates with the PC serially.

CHAPTER - 2

2.0 X10 TECHNOLOGY

What is X10?

X10 [1]is a powerline carrier protocol that allows compatible devices to communicate with each other via the existing 110V AC (Alternating Current) wiring. What this means is no new wiring is necessary to use X10 devices. Additionally, it gives the flexibility to implement X10 communication regardless whether a home or building is new or old.

2.1 X10 Transmission

X10 communication is established by transmitting digital signals on the power line. These signals involve short RF (Radio Frequency) bursts that represent digital information. These RF bursts occur near the zero cross over of AC signals. The presence or absence of the RF bursts represents a digital one or zero respectively. Typically, the presence of a 120kHz burst for 1 millisecond is a digital "one". Figure 1 below represents the AC wave carrying the X10 signal.

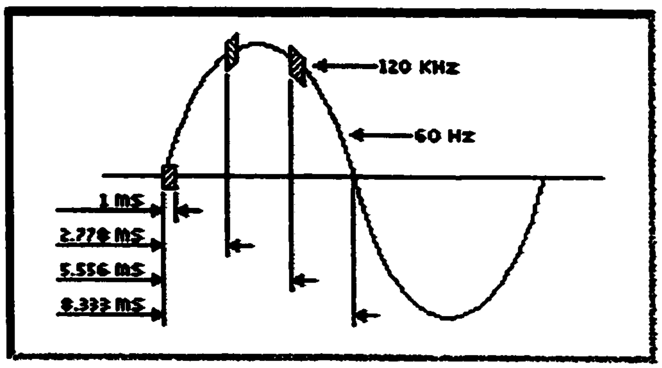


Figure 1 X10 transmission on AC cycle

There are two main components in the X10 communication. One is the transmitter that sends commands over the power line and other is the receiver that receives these commands and executes them. All X10 receivers have an unique address. This address consists of alpha and numeric characters. Example A1, B10, P16 etc. All the transmitting signals are tagged with the device address, and all the devices with the same address respond to the transmitted command.

The complete X10 transmission consists of 11 cycles over the power cycle. This requirement limits the speed of transmission between the moment the command is given to when it is actually executed.

2.2 X10 coding

The 11 cycles of the transmission is divided into;

Start Code – 2 cycles

House Code – 4 cycles

Number Code or Function code – 5 cycles

These codes are shown in Figure 2 below;

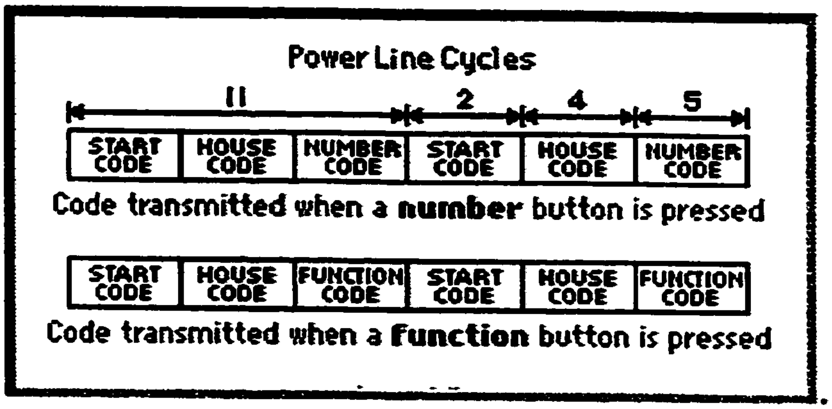


Figure 2 X10 digital codes

For each block of data, the bits are transmitted in a truly compliment form on alternate half cycles on power line. The only exception is the "Start code" that is always unique and is 1110. This is shown in Figure 3 below;

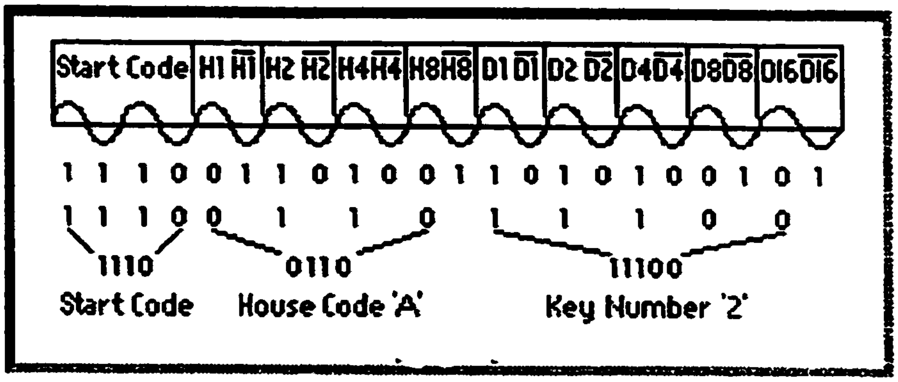


Figure 3 X10 code details

Typically the house code varies from A through P and number code 1 through 16. These codes provides a total combination of 256 unique tags for the X10 devices. The encoding format for these codes is given below in Figure 4;

HOUSE CODES	H1	H2	H4	H8	KEY CODES	D1	D2	D4	D8	D16	
A	0	1	1	0	1	0	1	1	0	0	
B	1	1	1	0	2	1	1	1	0	0	
C	0	0	1	0	3	0	0	1	0	0	
D	1	0	1	0	4	1	0	1	0	0	
E	0	0	0	1	5	0	0	0	1	0	
F	1	0	0	1	6	1	0	0	1	0	
G	0	1	0	1	7	0	1	0	1	0	
H	1	1	0	1	8	1	1	0	1	0	
I	0	1	1	1	9	0	1	1	1	0	
J	1	1	1	1	10	1	1	1	1	0	
K	0	0	1	1	11	0	0	1	1	0	
L	1	0	1	1	12	1	0	1	1	0	
M	0	0	0	0	13	0	0	0	0	0	
N	1	0	0	0	14	1	0	0	0	0	
O	0	1	0	0	15	0	1	0	0	0	
P	1	1	0	0	16	1	1	0	0	0	
					All Units Off	0	0	0	0	1	
					All Lights On	0	0	0	1	1	
					On	0	0	1	0	1	
					Off	0	0	1	1	1	
					Dim	0	1	0	0	1	
					Bright	0	1	0	1	1	
					All Lights Off	0	1	1	0	1	
					Extended Code	0	1	1	1	1	
					Hail Request	1	0	0	0	1	①
					Hail Acknowledge	1	0	0	1	1	
					Pre-Set Dim	1	0	1	X	1	②
					Extended Data (analog)	1	1	0	0	1	③
					Status=on	1	1	0	1	1	
					Status=off	1	1	1	0	1	
					Status Request	1	1	1	1	1	

Figure 4 House & device codes

[1] Hail Request is transmitted to see if there are any X10 transmitters within listening range.

[2] In a Pre-Set Dim instruction, the D8 bit represents the Most Significant Bit of the level and H1, H2, H4 and H8 bits represent the Least Significant Bits.

[3] The Extended Data code is followed by 8 bits, which can represent Analog Data (after A to D conversion).

2.3 X10 Interface communication protocol

The thesis project uses a serial communication [2] between the PC and the X10 devices through the existing house wiring. The 2 way communication device CM11A[4] is used in the thesis project.

There are 2 formats of X10 transmission – Standard and Extended;

2.3.1 Standard Transmission

Any transmission between the PC and CM11A typically will involve a House code, Device code, Function code or their combinations.

The format for the standard transmission consists of

Header & Code – 2 bytes (by PC)

Checksum – 1byte (by interface)

Acknowledge – 1 byte (by PC)

Interface ready – 1 byte (by interface)

The header and code combination is as follows;

HEADER :

7	6	5	4	3	2	1	0
					1	F/A[5]	E/S

<No. of Dims>

Where No. of Dims is a preset Dim command that varies between 0 and 22. 0 corresponds to 0% brightness and 22 corresponds to 100% brightness for shutter control installed in windows of a house. This feature is not used in the thesis project.

[4] CM11A is a product of X10.com Corporation

[5] F/A – Function /Address & E/S – Extended/Transmission

F/A defines that the following byte is a function (1) or address (0).

E/S defines whether standard transmission (1) or extended transmission (0) is used.

CODE:

7	6	5	4	3	2	1	0

Address: < HouseCode > <DeviceCode>

Function: < HouseCode > < Function >

The Function command is transmitted to all the devices having the same house code.

Interface CheckSum : When CM11A interface receives the transmitted signal from the PC a checksum is performed to ensure a valid transmission takes place. All the received bytes are added and checksum byte is returned. If the checksum is incorrect, the PC awaits a new checksum.

Interface ready to receive : Once the X10 transmission has occurred the CM11A will send hexadecimal 55 to the PC to indicate that it is in a "ready" state.

2.3.2 Extended Transmission

In this format, the standard transmission is extended with 2 additional bytes of data. The protocol may be represented as follows;

Header, Code, Data & Command – 4 bytes (by PC)

Checksum – 1 byte (by interface)

Acknowledge – 1 byte (by PC)

Interface ready to receive – 1 byte (by interface).

The Data and Command bytes go hand in hand for an extended transmission. While the Command byte gives the specific instruction to be executed, the Data byte provides numerical reference to the extent the command must be executed.

The header for an extended transmission is always:

7	6	5	4	3	2	1	0
0	0	0	0	0	1	1	1

Bits 7 and 3 are always 0 because the extended transmission does not have dim level.

Bit 2 is set to 1 because extended transmission is always a function.

Bit 0 is set to 1 for extended transmission.

Serial Parameters:

The serial communication between the PC and CM11A has the following specifications;

Baud Rate : 4800 bps

Parity : None

Data bits : 8

Stop bits : 1

This project uses a DB9 serial connector at the PC end, which has the pin out connections as shown in Table 1 below

Signal	DB9 connector
S-IN	Pin 2
S-OUT	Pin 3
GRND	Pin 5
RI	Pin 9

Table 1 DB9 connector details

2.3.3 *X10 Reception*

So far we have discussed the transmission of data from the PC to the X10 devices. We will examine how the CM11A will receive data through the power line and forward to the PC.

Whenever a data is received by CM11A, a serial ring (RI) signal is inserted to initiate the wake-up procedure for the PC. Upon complete reception of data, CM11A will begin polling the PC and upload the data buffer. There is a limitation on the buffer capacity of CM11A, which is usually a maximum of 10 bytes.

The 10 bytes are defined as follows in Table 2;

Byte	Function
0	Upload buffer size
1	Function / Address mask
2	Data Byte #0
3	Data Byte #1
4	Data Byte #2
5	Data Byte #3
6	Data Byte #4
7	Data Byte #5
8	Data Byte #6
9	Data Byte #7

Table 2 CM11A Buffer details

Although the maximum buffer size is 10 bytes, the interface will upload only the specified number of bytes within the buffer. The size of the upload is specified in byte 0. The byte 1 which is the function / address mask indicates whether the following 8 bytes should be interpreted as an address or as a function. Bit value 1 signifies function, while bit 0 signifies an address.

Data bytes employ the same format as the device or house code.

CHAPTER - 3

3.0 FUZZY INFERENCE SYSTEMS

3.1 Conventional Control

In this section, the standard approach for a conventional control [3] shall be discussed. The advantages of fuzzy control over the conventional approach shall also be examined.

The conventional approach to control relies on the concept of feedback. By imparting a negative feedback, the output of a dynamic system is being modified in a way to suit the requirements. Negative feedback is incorporated by constantly measuring the system output and subtracting the value desired from the actual value. Such a control application forms the basis of a closed loop control. This type of control is almost prevalent in most of the automatic control system. Closed loop control is implemented in a majority of control applications, be it industrial or a cruise control inside an automobile.

Outside the realm of closed loop control, there exists an operation that follows a specific sequential pattern to form the sequential control. In automatic operation, the sequential control requires one set of actions to be controlled before starting the next. Sequential control tends to be quite simple in application. A common example is the closure of an elevator door before it can move.

Yet another mechanism of control is called On/Off control. In this type, the output of the system is actuated either on or off by an actuating device. Typical example is a thermostat seen almost everywhere.

Servomechanism type control is used when the controlled variable changes quite frequently due to the application. It is a very popular implementation in robotics.

The closed loop control discussed above can be further extended to one of the most implemented type called the PID (Proportional-Integral-Derivative) control. A PID controller has a more detailed algorithm to determine the output. It looks at the current value of the error, the integral of the error over a recent time interval, and the current derivative of the error to determine not only the quantity of correction that requires to be applied, but also for how long. A tuning constant is multiplied with each of these quantities to produce the controller output CO(t) as;

$CO(t) = Pe(t) + I\,(\int_0^t e(\tau)d\tau + D(d/dte(t))$

Where P,I & D are proportional, integral and derivative tuning constants

e(t) is the error between the setpoint SP(t) and process variable PV(t) given by;

e(t) = SP(t) – PV(t). Refer Figure 5 for details.

Every PID controller needs to be tuned to the application in order to eliminate the error.

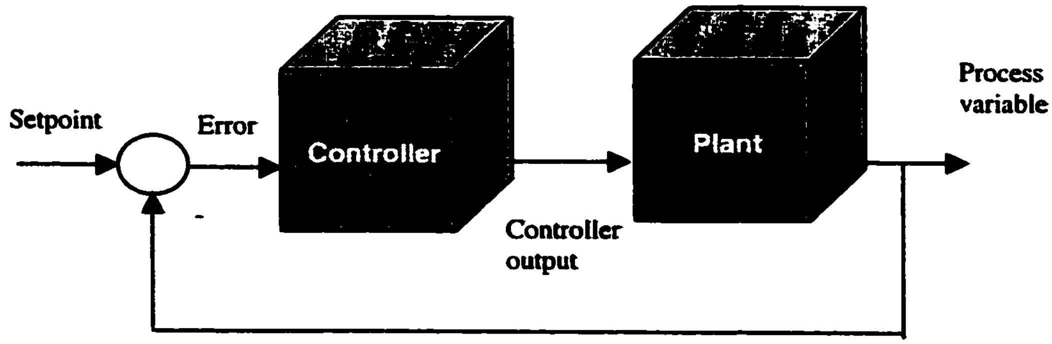


Figure 5 Feedback Control

With these words about conventional control, their drawbacks will be analyzed, which would lead to more appreciation of fuzzy control although fuzzy control is not without disadvantages.

To implement a conventional control successfully, one has to understand the plant dynamics totally. This may be difficult for complex process applications. There is also a strong preference for a linear model. On assuming a linear model, there may be linearity only for a small range, beyond which the model behaves in a non linear way. At these ranges, the control system does not behave accurately.

Fuzzy control applications typically perform well under these extreme conditions. The thesis project also considers a non linear problem which is handled easily by Fuzzy control.

3.2 Fuzzy Control

The fuzzy inference system is developed on the basis of the past behavioral patterns of the system in question. If a human being is to be emulated, which is the case in this thesis project, the fuzzy inference system becomes the fuzzy logical controller.

Fuzzy systems are developed on the basis of fuzzy modeling which comprise of the following main components;

- Rule base consisting of fuzzy if-then rules.
- Membership functions that are used in the fuzzy rules, and
- An inference mechanism that derives the controller output.

Typically, a fuzzy inference system can take either a fuzzy or crisp (discrete) input. The thesis project requires a crisp output that will determine the exact level of either illumination or temperature required. Let us discuss some of the fundamental items pertaining to the fuzzy logic systems.

3.2.1 Fuzzy Sets

As the name signifies, the fuzzy set [4] [5] is a set without any boundaries. Most of the objects commonly encountered do not have clearly defined boundaries. In otherwords, there are no precisely defined criteria of membership. This is contrast to a classical set that has the boundaries clearly defined. The fuzzy sets are characterized by smooth transitions of the set boundaries with continuously changing grades of membership. A fuzzy set represents the logical state of various systems and is the way the human thought process evolves. For example, "temperature is low", or "occupancy is high". These features gives the flexibility to model a system based on simple linguistic statements. Let us look into the fuzzy sets in its fundamental form;

Let X be a space of objects with generic element of X denoted by x

Therefore X = { x }

A fuzzy set A in X is characterized by the membership function $f_A(x)$ which relates each point in X with a real number in the interval [0,1]. $f_A(x)$ defines the grade of membership of x in A. The membership of x in A increases as $f_A(x)$ tends to unity. In a classical set, the membership of x in A can take only two values, either 0 or 1 to indicate if it belongs or does not belong to the set.

As an example, consider A to be a set of real numbers greater than 5. The membership grade of an object x in a fuzzy set A is represented by a number $\mu_A(x)$ in the interval [0,1]. Therefore for the set A defined above, the membership grades could typically be $\mu_A(5) = 0$, $\mu_A(10) = 0.1$, $\mu_A(100) = 0.7$, $\mu_A(300) = 1$ etc. The value of $\mu_A(x)$ is subjective.

3.2.2 Fuzzy Systems

Let S be a system with input, output and state of S at time t denoted by u(t), y(t) and x(t) respectively. S is a fuzzy system if u(t) or y(t) or x(t) or the combination ranges over the fuzzy sets.

Let S be a discrete time system given by the state equations;

$x_{t+1} = f(x_t, u_t)$

$y_t = g(x_t, u_t)$

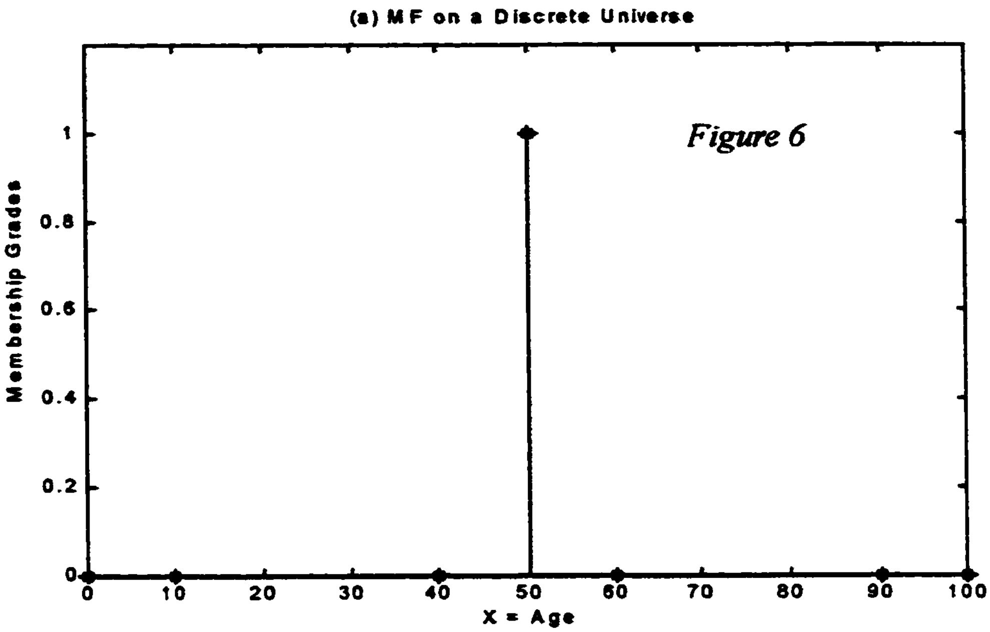
where u_t, y_t and x_t denote the input, output and state of S at time t. f and g are functions. For a fuzzy system, the variables range over spaces where elements are fuzzy sets.

3.2.3 Membership functions

If X is a collection of objects denoted generically by x, then the fuzzy set A in X is the set of ordered pairs.

$A = \{(x, \mu_A(x)) \mid x \text{ belongs } X\}$ where $\mu_A(x)$ is called the membership function. X is called the universe of discourse. The universe of discourse can be either discrete or continuous as seen by the following example.

Let X = { people of age 50 years). Figure 6 shows the membership function on a discrete universe for this set. We can extend this to a membership function for a continuous universe where set X = {people about 50 years age}. Figure 7 plots this relationship.



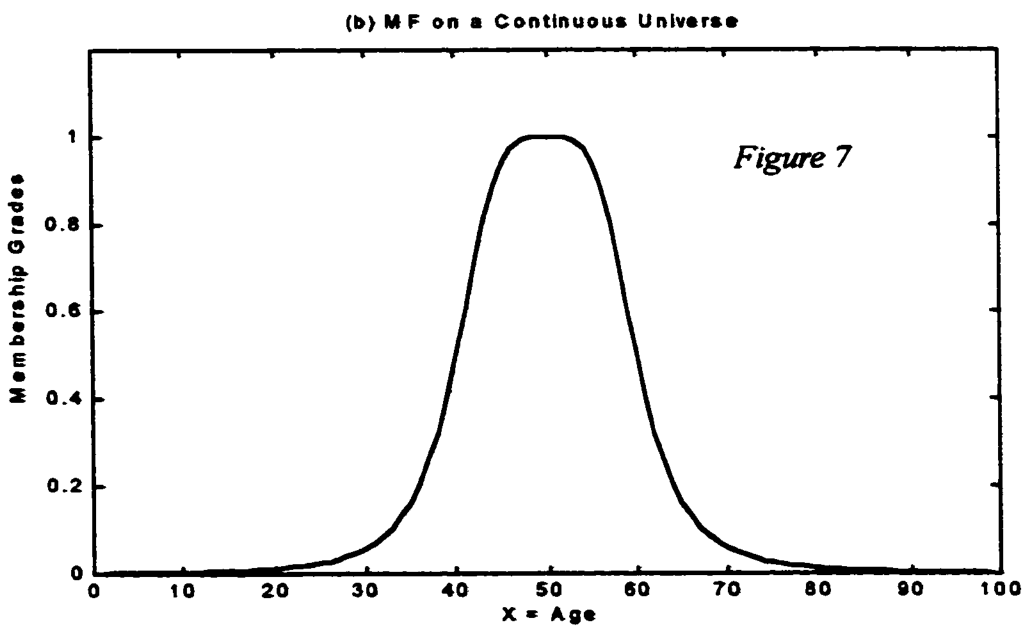


Figure 6 MF on discrete universe

Figure 7 MF on continuous discourse

From the foregoing discussion, the fuzzy set depends on two items – to define a suitable universe of discourse and to define a membership function. The membership function is a matter of subjective choice and is decided best upon the application. For Lighting Control as discussed in Section 4.8, we define low, medium and high as the membership functions for the current illumination level. Similarly, for temperature control as discussed in Section 5.2, five linguistic variables are considered for the fuzzy controller output of the temperature controller system.

We see in a continuous universe of discourse, as shown in Figure 7, there is a smooth variation of the fuzzy set over the desired value. Typically, a continuous universe of discourse is partitioned into several fuzzy sets that have seamless overlap with one another. Of course, we can have any number of membership functions, which will determine the granularity as discussed in Section 3.3.1.

Figure 8 describes the membership functions of linguistic values. The smooth overlap of these membership functions may be noted.

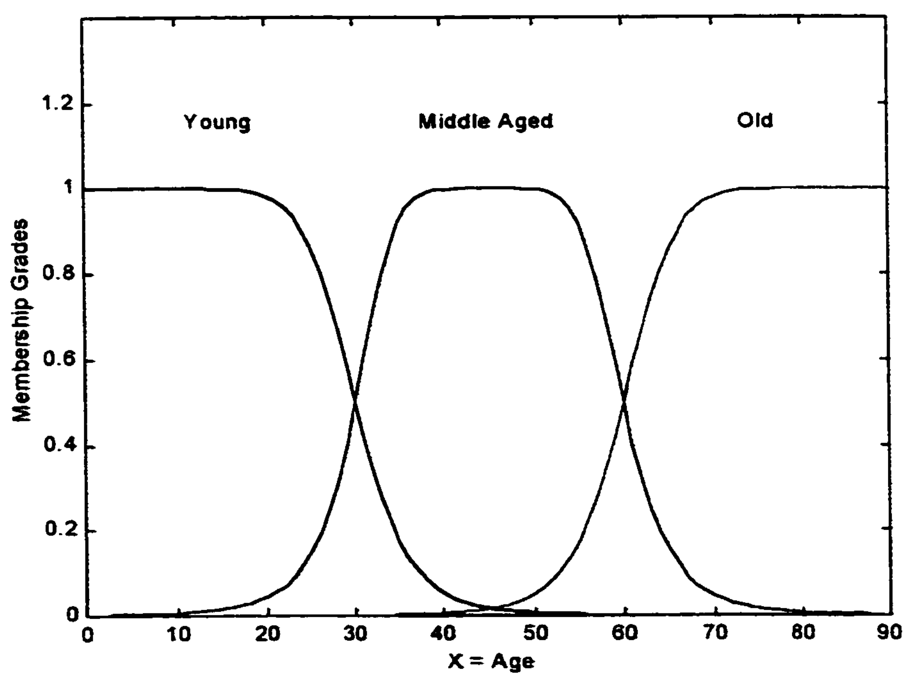


Figure 8 Membership functions

Membership functions (MF) in one dimension belong in several classes. Some of the most common one dimension membership function that is used with a single input are;

- Triangular MF
- Trapezoidal MF
- Gaussian MF
- Bell MF

Due to the computational efficiency, both triangular and trapezoidal MF are used extensively. The thesis project uses the single dimension MF for the inputs.

Triangular MF

A triangular MF is specified by 3 parameters (a,b,c) in the following relationship;

$$\text{Triangle}(a,b,c) = \begin{cases} 0, & x<=a \\ (x-a)/(b-a), & a<=x<=b \\ (c-x)/(c-b), & b<=x<=c \\ 0, & c<=x \end{cases}$$

The values of (a,b,c) determine the x coordinates of the three points of the triangle.

Trapezoidal MF

$$\text{Trapezoid}(a,b,c,d) = \begin{cases} 0, & x<=a \\ (x-a)/(b-a), & a<=x<=b \\ 1, & b<=x<=c \\ (d-x)/(d-c), & c<=x<=d \\ 0, & d<=x \end{cases}$$

Again the parameters a,b,c,d represent the x coordinates of the trapezium. The trapezoidal MF reduces to a triangular MF if the coordinate b=c.

While the triangular and trapezoidal MF are composed of straight lines, the Guassian and Bell MF are smooth curves related by a non linear function.

Gaussian MF

A Gaussian MF is specified by 2 parameters namely (c,σ)

$$\text{Gaussian}(c,\sigma) = e^{-\frac{1}{2}((x-c)/\sigma)2}$$

C represents the center of the Gaussian MF and the σ represents the width.

Bell MF

Bell MF is specified by 3 parameters (a,b,c)

$$\text{Bell}(a,b,c) = \frac{1}{1+\left|(x-c)/a\right|^{2b}}$$

if b is negative, the bell is inverted.

3.2.4 Universe Of Discourse

The horizontal axis of the graph depicting the membership of a number to a fuzzy set is called the *Universe of Discourse* [6]. The universe of discourse is the range of values where the fuzzy set is defined. The vertical axis is the membership value, in the universe of discourse, to the fuzzy set. The membership of a number (x) to a fuzzy set *A* is represented by: $\mu_A(x)$

3.3 Structure of Fuzzy Logic Control system

The typical architecture of a fuzzy logic controller is represented below in Figure 9.

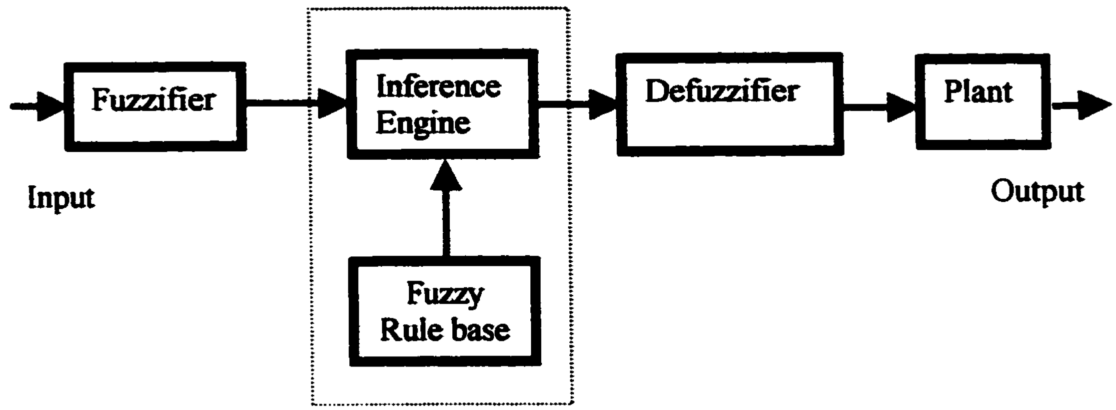


Figure 9 Block diagram of Fuzzy inference system

The Fuzzy logic control systems comprises of;

- Fuzzifier
- Inference engine
- Fuzzy Rule base
- De-fuzzifier

The fuzzifier converts the crisp values of the inputs into appropriate linguistic values. For example if the current level of illumination is 10%, it is fuzzified as "low".

The fuzzy rule base is the domain containing *if-then* rules to imitate the behaviour of the plant. It serves as the expert knowledge base. The inference engine simulates the reasoning or the thought process. In the thesis project, the inference engine imitates the human decision to control the lights to the required illumination, or control the temperature to the required setpoint by performing the appropriate reasoning to achieve the required control strategy. The defuzzifier performs the inverse operation of the fuzzifier. It converts the linguistic statements into discrete values. For example it converts the linguistic output value of the dimmer from "high" to a crisp value of 90%.

3.3.1 Input-Output variables

In order to achieve the desired output from the fuzzy logic controller, it is important to choose the process variables that characterize the system to a maximum extent. Each linguistic input variable in a universe of discourse is characterized by a set of linguistic values. For example, if the input is occupancy of the room, the linguistic values will be "low" and "high". We also define a term "set" which is the set of linguistic values of the input. The size of the term set determines the fuzzy partition of the fuzzy input. For example, the input "occupancy" seen above could have the term set as "low", "medium" and "high" or simply "low" and "high". The fuzzy partition determines the fineness of the control output. Typically, the input space could be divided into a number of overlapping terms. Figure 10 and Figure 11 below presents a representation of fuzzy partitions.

Figure 10 represents the coarse partition, while Figure 11 represents the fine partition. It is essential to choose the correct fuzzy partitioning of the inputs and outputs to achieve a successful controller.

These examples consider the triangular and trapezoidal membership functions that are used in the thesis project.

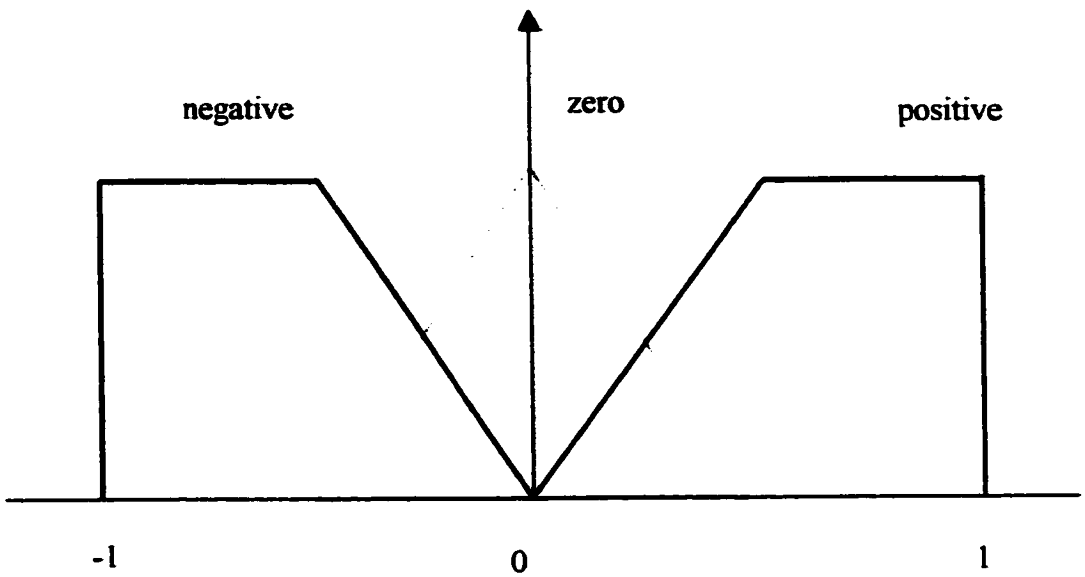
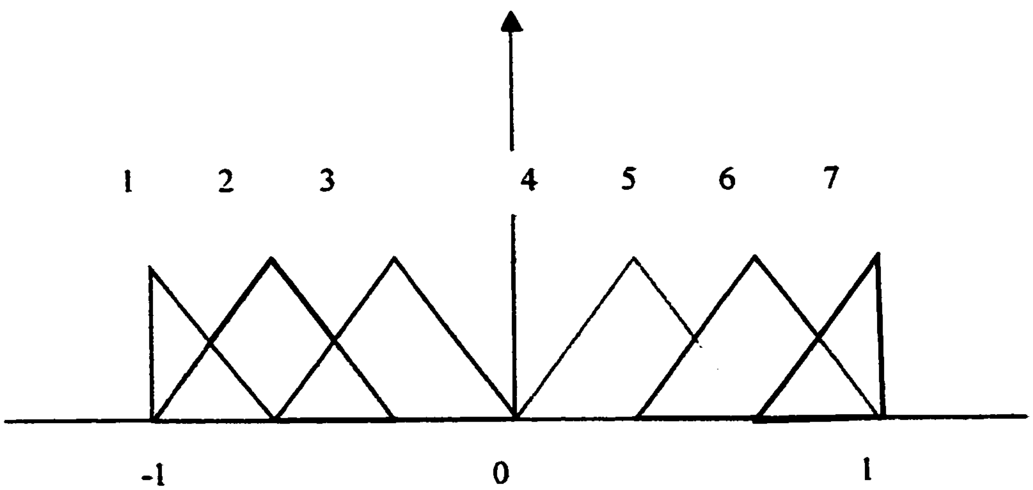


Figure 10 Fuzzy partition - Coarse



1-Negative large, 2-Negative medium, 3-Negative small

4- Zero

5-Positive small, 6-Positive medium, 7-Positive large

Figure 11 Fuzzy partition - Fine

The input and output membership functions are based on a practical decision to suit the kind of application the fuzzy controller is intended to handle.

3.3.2 Fuzzifier

With reference to Figure 9, the Fuzzifier translates a measurable data into a linguistic value. It provides a mapping from a measured input value to the fuzzy singletons in the universe of discourse specified. In designing a Fuzzy controller, the inputs are always available as a discrete measurable data.

3.3.3 Defuzzifier

Defuzzification is the inverse operation of a fuzzifier wherein the fuzzy actions that are defined over an universe of discourse is mapped into non fuzzy crisp or discrete actions. Thus, we may say that the defuzzification process extracts a crisp value to represent a fuzzy set. With reference to Figure 9, the defuzzifier provides the precise output of the system. Any system based on Fuzzy logic controller uses this output to drive the system to a new state based on the input condition received.

The following are the methods used for de-fuzzification. Consider a fuzzy set A of a universe of discourse Z.

Centroid of Area method (COA)

For a continuous universe of discourse, the COA generates an output of

$$Z_{COA} = \frac{\int_z \mu_A(z) z \, dz}{\int_z \mu_A(z) \, dz}$$

where $\mu_A(z)$ is the aggregated output membership function. This method of defuzzification is one of the popular methods adopted. The COA generates the center of gravity of distribution of a control action.

Bisector of Area method (BOA)

Per definition, this strategy generates a bisector line that divides the fuzzy set into equal areas. The BOA scheme satisfies the equation;

$$\int_{\alpha}^{z_{BOA}} \mu_A(z)z\,dz = \int_{z_{BOA}}^{\beta} \mu_A(z)z\,dz$$

where $\alpha = \min\{z\}$ where z belongs to set Z and

$\beta = \max\{z\}$ where z belongs to set Z.

Mean of maximum method (MOM)

The MOM generates an output that represents the mean value of all control actions whose membership function reaches the maximum value. Z_{MOM} is the average of the of the maximum z.

$$Z_{MOM} = \frac{\int_{z'} z\,dz}{\int_{z'} dz}$$

where $Z' = \{z \mid \mu_A(z) = \mu\}$

Smallest of maximum method:

This strategy selects the value having the minimum magnitude of the maximum value.

Largest of maximum method:

This strategy selects the value having the maximum magnitude of the maximum value.

Of all the defuzzification methods discussed above, the COA method yields better results for steady state performance, while the MOM is more suitable for transient performance. The various de-fuzzification methods are shown below in Figure 12, see Reference [8]

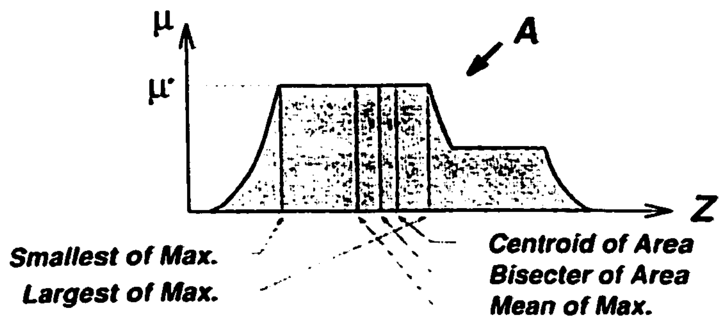


Figure 12 De-fuzzification Methods

3.3.4 Fuzzy Rule Base

The fuzzy logic control is characterized by a set of *if-then* rules which describes the relationship between the input and output. For a multi input – single output system, the general form of the fuzzy rules are given by;

If x is A_k.....and y is B_k, then $z=C_k$, $k=1,2....n$

Where x,...y and z are the linguistic variables representing the process state variable and control variables respectively. A_k....B_k and C_k are the linguistic variables x...y and z in the universe of discourse.

3.3.5 Inference Engine

We shall digress a little to arrive at the concept of inference engine. Any modeling of a process control system can be approached in a number of ways. For example, let us assume a system that has three inputs and two inputs. Figure 13 presents the basic structure of the system.

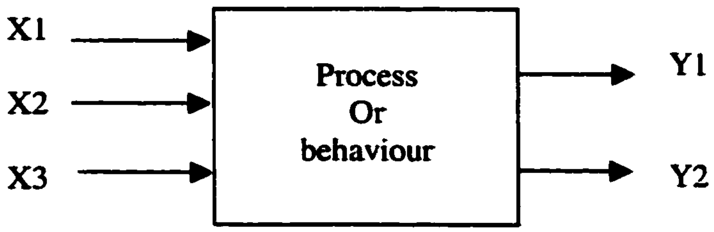


Figure 13 Basic Process Control System

On the other hand, if the process is quantitative and deterministic, then the decision making process may be represented as a series of multivariate functions one for each output. Figure 14 shows a system represented by a series of multivariate functions. Thirdly, if the input to a process system is finite but without any significant vagueness in the concepts of the values, it may be prudent to describe the process as a rule base with an accompanying inference engine of a non fuzzy sort. While it may be possible to take the approach of the multivariate functions as before, the rule based concept is easy to develop and analyze. Figure 15 represents the rule based system.

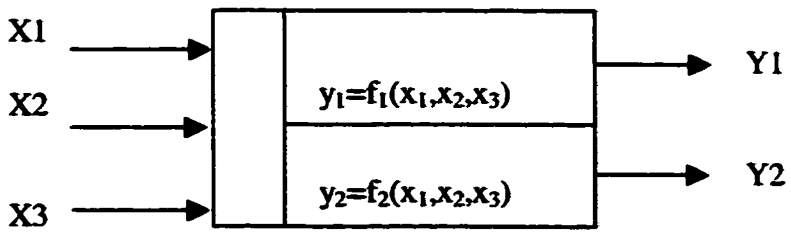


Figure 14 System based on Multivariate functions

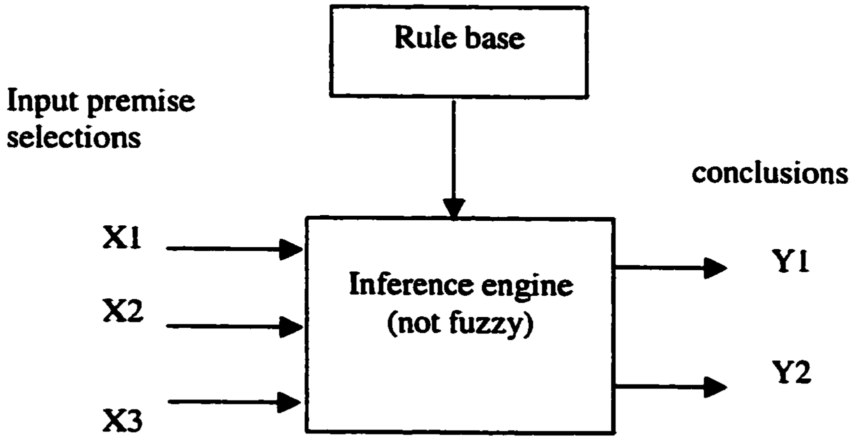


Figure 15 Rule based System

The final approach would be the fuzzy system, which is described in Figure 15. Let us examine the rule base of the fuzzy inference engine. The rule base is a collection of small number to several numbers of rules. Typically the format of these rules is simple and straightforward.

Rule # n

IF premise THEN consequence

The antecedent can consist of one basic proposition or can be built of multiple basic propositions linked by the conjunction AND. The logical negation can also be used by the word NOT.

The overall rule base is linked by disjunction using the word OR, although the use of disjunction is not a preferred approach within each rule. The overall rule base will look like RULE #1 ....OR ....RULE#2.....OR......RULE#3.............OR RULE#n

The rules are based on implication, conjunction, logical negation and disjunction.

3.3.6 Fuzzy Models

Since the inception of fuzzy sets and fuzzy logic, two major applications to the fuzzy logic as a controller was proposed. The first to go was by Mamdani, and second by Sugeno. We shall decribe the following models [8] [9] in a little bit detail especially in the context of the thesis project;

- Mamdani Fuzzy Model and
- Sugeno Fuzzy Model

Mamdani Fuzzy Model:

Mamdani fuzzy controller is based on human expertise obtained from operators. This model first attempted to control a steam engine and boiler operations based on linguistic rules. The human expertise is translated into a set of logical rules, fuzzy logic with appropriate fuzzification and de-fuzzification that are used to generate the real control values from the rules.

The control rules are expressed as logical statements such as;

IF z_1 is S_{1i} and z_2 is S_{2i} and....z_n is S_{ni},

THEN z_0 is S_{0i}

Where $z_1, z_2, \ldots\ldots z_n$ are the input variables

S_{1i}, S_{2i}....S_{ni} and S_{oi} are fuzzy sets and z_0 is an output fuzzy variable.

The controller based on Mamdani rules has the following features;

- The range of the variables is divided into a finite number of real intervals or the variable is allowed to take only finite number of real values.
- The decision rules are based on IF-THEN statements to specify a value of the controller outputs for a given values or intervals of the controller inputs.
- Only the rule that fires at a particular time determines the actual value of the controller output at that time.

The input range would be discontinuous if there is no overlap in their ranges. However, introducing a smooth overlap and membership functions to characterize each interval by using fuzzy logic, converts the discontinuous logical controller into a fuzzy controller with a continuous input-output mapping.

Therefore, the Mamdani controller with the fuzzification and defuzzification interfaces is equivalent to a nonlinear controller with multiple inputs and outputs.

Sugeno Model:

This model was proposed by Takagi,Sugeno and Kang. Known as the TSK model, this aids in the development of fuzzy rules from a given input and output.

A typical fuzzy rule in this model takes the form;

If x is A and y is B then $z = f(x,y)$

Where A & B are fuzzy sets in the antecedent and f(x,y) is a crisp functin in the consequent.

With x and y being the input variables, f(x,y) takes the form of a polynomial. The function f(x,y) can be any function provided it maps the output of the model within the fuzzy region as specified by the antecedent of the rule.

If f(x,y) is a first order polynomial, the resulting fuzzy inference system is called a first order Sugeno fuzzy model. If f is a constant, it is a case of zero order Sugeno fuzzy model. The thesis project is based on the zero order Sugeno model containing two or three inputs and one output. The membership functions in the input have adequate overlap to produce a smooth output.

The Sugeno system can be viewed as a collection of local models each one being valid in a particular region. While each operating region is defined by the conditional part like that of Mamdani model, the difference is that the consequent part of each rule is an analytical expression describing the corresponding local model. Example;

Ri: If z_1 is S_{1i} and z_2 is S_{2i} and......z_n is S_{ni}

Then $y_i=f(p_{1i},p_{2i}.......pni,z_1,z_2,......z_n)$

Where $z_1,z_2,.....z_n$ are real input variables of the collection of models.

$S_{1i},S_{2i}.....S_{ni}$ are fuzzy sets and y_i and p_{ij} are the real output variable and parameters of the i^{th} local model respectively.

The output of a zero order Sugeno function is a function of its input. If the neighbouring membership functions in the input variables have adequate overlap, the output is smooth. We shall describe the reasoning procedure as depicted in Figure 16 [8] for a zero order Sugeno fuzzy model, wherein the overall output is obtained by the weighted average of the crisp outputs of each rule. This method speeds up the computation time of defuzzification process as required in the Mamdani model.

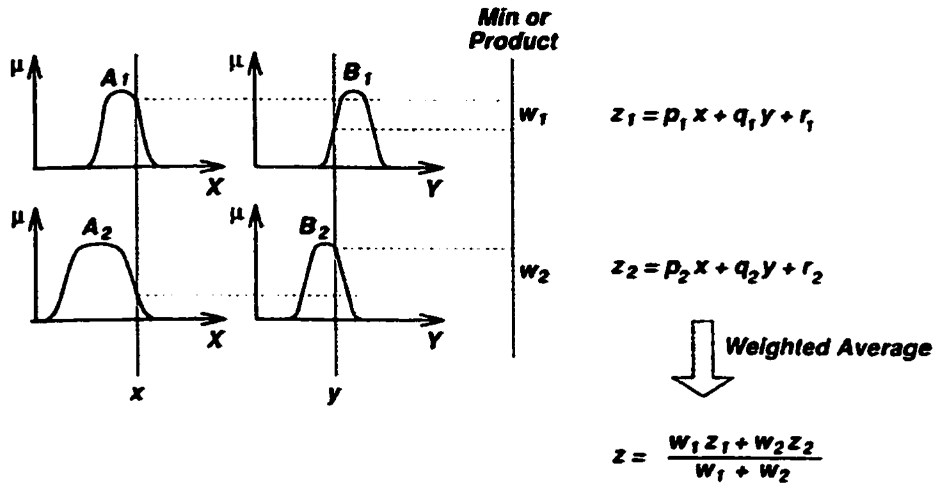


Figure 16 Zero Order Sugeno Model

The thesis project is based on the zero order Sugeno system. The overall output is based on the weighted average of the crisp outputs of each rule. The zero order Sugeno fuzzy model is functionally equivalent to a radial basis function with some minor restrictions. We shall now examine the architecture of a class of adaptive networks that are equivalent to fuzzy inference systems.

3.3.6.1 Adaptive network based fuzzy inference system.

The fuzzy membership functions and the connective layers in an adaptive network can be optimized. At this point, it must be mentioned that an adaptive neural network model is not proposed for the thesis project. We are only presenting a zero order Sugeno fuzzy model representing the architecture of an adaptive system. See Reference[8]. We shall derive the overall output of the system by the weighted average method. Let us start by examining the Architectural model presenting a zero order Sugeno system with m inputs and n rules as shown in Figure 17;

Consider the ith rule of a zero order Sugeno fuzzy system consisting of n rules (i=1,2,....n)

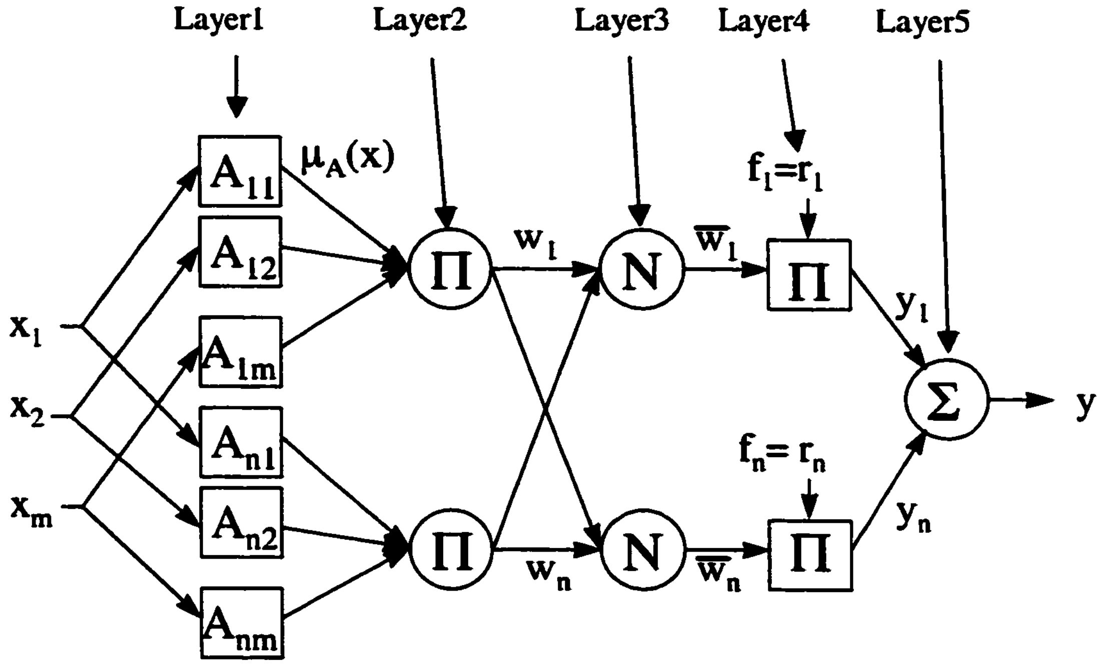


Figure 17 Architectural Model of Fuzzy System

where: x is the input vector.

A is an antecedent membership function (*LHS*).

$\mu_A(x)$ is the membership of x to set A.

w_i is the degree of fulfillment of the i^{th} rule, i=1.2...

$\overline{w}_i$ is the normalized degree of fulfillment of the i^{th} rule.

r_i is a constant singleton membership function of the i^{th} rule (*RHS*).

y_i is the output of the i^{th} rule

The architecture described above illustrates the reasoning mechanism for a Sugeno model shown in figure 16. The different layers are described below;

Layer 1:

$O_{1,i}$, is the membership grade of a fuzzy set $A_1, A_2 \ldots A_n$. The membership function can be any generalized function described earlier. It specifies the degree to which the inputs $x_1, x_2 \ldots x_m$ satisfies A.

Layer 2:

Every node in this layer is a fixed node labeled $\prod$. The product of all the incoming signals gives the output which is the degree of fulfillment of the i^{th} rule. Mathematically, the zero order Sugeno system is given by the following equation;

$$w_i = \prod_{j=1}^{m} \mu_{A_{ji}}(x_j)$$

where $\mu_A(x_j)$ is the membership of x_j to the fuzzy set *A*, and w_i is the degree of fulfillment of the i^{th} rule. The degree of fulfillment also represents the firing strength of a rule. Typically any T-norm operators that are meant to perform the fuzzy AND can be used as the node function in this layer.

Layer 3:

Every node in this layer is a fixed node labeled as N. The outputs of this layer are the normalized firing strength, which is also a measure of the normalized degree of fulfillment given by;

$$\overline{w}_i = \frac{w_i}{\sum_{i=1}^{n} w_i}$$

Layers 4 & 5

The output (y) of a fuzzy system with *n* rules can be calculated as:

$$y = \sum_{i=1}^{n} \overline{w}_i f_i = \sum_{i=1}^{n} y_i$$

In this case, the system is a zero-order Sugeno system and f_i is defined as:

$$f_i = r_i$$

The foregone adaptive network is functionally equivalent to a Sugeno fuzzy model. The layers 3 and 4 can be combined to obtain a network with only 4 layers.

CHAPTER - 4

4.0 LIGHTING CONTROL SYSTEM

Objective : To design and implement a Fuzzy logic controller for lighting applications in a home using X10 technology for communication.

4.1 Introduction

The task of lighting control is a very delicate and balancing task than it appears to be. The Fuzzy Logic controller has to deal with two varying elements constantly – one is human factor, and the other is Mother Nature for ambient light conditions. The controller must deal with these variables at all times of the day.

The fuzzy logic controller computes the dimmer output based on the ambient lighting and occupancy levels generated by the motion sensors. Refer to Section 4.8 for details on Fuzzy Controller design.

The illumination level established by the controller is very subjective, primarily because the brightness of any particular object is not the same for two humans. Simple analogy, some persons are compulsive to wear sunglasses and they can't take the glare of the sun, while some can do with a Polaroid lens that reduces the ultraviolet glare, and lastly people who can take the sunlight without any form of sunglasses. The response of the eye to the illumination or luminance is nearly logarithmic. How about the case when a person walks into the house from a bright outdoors setup, say on a sunny daylight? The interior appears dark for a while until the eye adjusts to the new illumination level.

The second input to the Fuzzy controller is the occupancy level inside the house. Motion detectors are installed to monitor the movement inside the house, which can be tricky at times. During the sleep time, the motion detector must ignore any subtle movement by the bed during sleep time. At the same time, it must consider if someone walks to washroom in the night. The response to the latter situation must have the controller turn the light ON, while it must maintain a status quo of the lighting in the former.

These are the psychological conditions that make it difficult to evaluate the performance of the Fuzzy logical controller. The foregoing situations present different interpretations to the lighting levels detected by a photo cell. One of the objectives of the controller is to maintain an optimum illumination level at any point of time. Neverthless, the controller performs far better than a traditional ON-OFF operation for a lighting would do. It takes into account the variables that typically determine the illumination level required, and meets the project objective stated initially.

In the design of the Fuzzy controller, however, the abstract aspects of the human dispositions are ignored. The illumination levels are based on normal human vision. Any difference in the requirement can be adjusted by fine tuning the system and having the controller drive the illumination to the required level.

4.2 The Conventional Lighting Controller

There are several ways in designing a conventional controller for lighting applications. Refer to Figure 18 for a typical conventional controller that can be used for lighting. Essentially it is a feedback control, which turns the light either ON or OFF making it a simple ON-OFF controller. These controllers are based on the time of the day and the month (seasonal changes) to derive the ambient lighting level. However, the drawback of calculating the output based on time is that it does not account for variation in seasons that affect ambient lighting level at any given time.

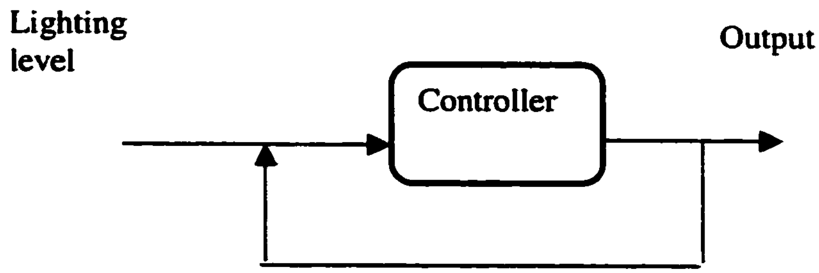


Figure 18 Block Diagram of a Conventional Controller

4.3 Decomposing into Zones

The whole application of lighting control for a home, shall be decomposed into several zones. There are merits and demerits in doing so. The primary advantage is this decomposition helps in breaking down the overall control into smaller areas to deal with. Secondly, it provides an easier and practical way to implement into smaller zones and integrate them together.

On the down side of it, the setting of a light in a particular zone must not affect the lighting level of its neighboring zone. Additionally, the human activity in a particular zone must not affect the neighbouring zone. It is practically difficult to restrict the overlap between zones, but the sensors can be placed in some strategic locations to minimize such an interference.

4.4 Event based Segmentation

The input and output conventions for the Fuzzy controller shall be described in the later chapters. The event based segmentation defines salient events that lead to a control decision. Some of the events within any single zone are;

- Movement into or within a zone.
- Movement out of a zone. (derived by a time out signal).
- Change in existing lighting condition.
- Manual over-ride of the control.

New control decisions are made whenever there is a change in any of the events described above.

Unless there is a change in any of the events listed above, the controller output remains the same although the inputs are scanned continuously.

4.5 Occupancy in a zone

As seen in the previous section, the Fuzzy controller takes into account the occupancy level of the zone being controlled. Measurement of occupancy is highly non linear and there can be no correlation with any standard reference. When motion is detected in any unoccupied zone, a signal is flagged and the zone is tagged as occupied. The system must be able to differentiate between the various movement patterns to conclude the occupancy level after detecting one.

A person could enter a zone and exit the same, never to re-enter at least for a while. Someone could also enter a zone and exit in a sporadic manner or frequently.

The motion sensor scans for movement and flags the zone as occupied when any movement is sensed. But such is signal is inadequate to provide good lighting control simply because the movement pattern is not analyzed. This problem is handled by the system by computing the time elapsed after every detection of motion. This means that whenever any motion is detected, the countdown timer kicks in. The zone is flagged as occupied unless the motion detection is timed out. This will confirm that there has not been any "occupancy" detected. If there is activation at the same place, the timer is reset. A new countdown begins until interrupted by another movement. The countdown time can be adjusted to suit the requirement. Thus, the movement within the zone is constantly monitored. For the thesis project, a time period of 60 seconds is set. This timing can be adjusted to suit requirement.

Typically, the time elapsed between the sensing of the motion and the triggering of the light is around 700 milliseconds.

4.6 The Choice of X10 devices

This section describes the various X10 components that are used for Lighting control. Their individual functions and operations are also described.

All the X10 devices for the project were purchased from X10.com based on California. A wide spectrum of products are available, some are used for PC based control, while others control the devices directly. The devices were selected to simulate the operation of the Fuzzy Lighting controller in a given zone. It must be mentioned that for real life situation, a few more devices of the same type may have to be added to effectively implement the fuzzy logic.

For the thesis project the following X10 devices are required. Refer Figure 25 and Section 4.9 for Hardware integration.

- PC serial interface
- Serial interface cable and connector

- Motion detector
- Motion detector transmitter
- Lighting sensor
- Lamp dimmer

The details of the model numbers and items purchased are given below in Table 3:

ID	Item description	Application	Part Number	Remarks
1	PC to powerline interface	Connects to the serial port on the PC on one end and the power socket at the other. This is the main communication bridge between any X10 device and the computer	CM11A	
2	Hawkeye motion detector	This senses the presence or absence of motion. Has a range of about 40 feet and communicates wireless to a RF receiver base. The motion sensor has to be mounted in a strategic location for optimum sensing, typically say near the entrance.	MS13A	Requires battery power.

ID	Item description	Application	Part Number	Remarks
3	RF Transceiver	This is a receiver base to receive RF signals from any X10 transmitter. Typically Hawkeye motion sensor communicates with this receiver base. The receiver base has an antenna to intercept the RF signals and plugs itself into a wall power socket, thereby communicating to the computer interface.	RR501	
4	Two way lamp module	The lamp module has one end plugged into the wall socket and takes the lamp to be controlled at the other end. This module is suitable for preset dim applications used in the project.	PLM21	
5	Light sensor	This module has a photosensor to monitor the	SD533	

ID	Item description	Application	Part Number	Remarks
		illumination level. It has an extension wire that plugs into the wall socket for X10 communication. Any change in the illumination level will trigger an output. The setpoint for the illumination level is user adjustable through a pot located in the module		

Table 3 Lighting control – Component description

4.6.1 PC Serial Interface

The Computer Interface connects to the PC and sends signals over the existing house wiring to control lights connected to X10 modules.

The CM11A provides the X-10 interface for the System software. It connects to any available serial port on the computer using the supplied interface cable. The CM11A converts the X-10 signals to an ASCII format for use by the computer system.

Brief Specification of CM11A:

- Cable: DB9 female

- 120 VAC 60 Hz only.
- Power consumption: 3 watts.
- UL Listed

The CM11A has a feature to store macros that can be downloaded to execute a control program. The interface even could work with the PC switched off. Once the interface has been programmed, the PC is no longer needed. Two standard AAA batteries retain the time, scheduled events and macros for up to 500 hours even if the power fails. One can even unplug the interface after programming it and plug it in someplace else in the home. However, our application of the thesis project does not use this feature as the Fuzzy logic software cannot be simply downloaded to the interface.

The interface is two-way, capable of both transmitting and receiving X10 powerline commands.

To install CM11A simply plug the base pin into a power socket. Connect the RJ11 connector at the bottom of CM11A and the DB9 pin to the serial port of the computer.

4.6.2 Two Way Lamp Module

The lamp module connects the lights that have to be controlled. It receives X10 commands for various dim levels and drives the lamp accordingly. The lamp module used is a two way module which means the PC can check with the module about the status. This module supports preset dim commands. Some of the main features of this module are;

- Soft Start gently "fades on" lights & extends bulb life.

- "Resume Dim" automatically dims light to preset level.
- Increased signal sensitivity reduces need for bridges.
- Responds to status request from PC.

These modules offer additional features not found on standard versions, including Soft Start, which gradually increases the lighting level over a 2 second time span to create an eye-pleasing fade-in effect that prolongs bulb life. Another feature is Resume Dim, which turns lights on at the dim level at which they were set when they were last turned off. This dimmer is suitable for incandescent lights for a rating up to 300W.

To install the module, simply plug the lamp module into a power socket and connect the lamp to be controlled in the slot provided in the front. There are 2 dials to be set in the front that will determine the address of the module. With a tool adjust the house code and the device code to the required code.

4.6.3 Motion Sensor

The Hawk-eye[6] motion sensor is used to detect any movement in the area it looks into. This is a compact sensor and provides reliable detection of motion. The Hawkeye II Motion Sensor sends wireless Radio Frequency (RF) signals to an X10 Transceiver or any X10 security system base receiver. The Transceiver, which is plugged into the power socket, receives the signals from the Motion Sensor and passes them onto the house wiring. Some of the features of the motion detector are;

- Up to 20' ft range

[6] Trademark of X10.com

- Passive infrared technology
- Motion Sensor is compact, light weight of size 2.5 x 2.5 inches
- Intended for indoor use.
- Requires 2 AAA batteries

The Hawkeye II has a built-in photocell that detects even when it's dark. It can therefore detect motion even when it is dark.

The key to the successful detection of motion is the strategic location of the Hawkeye. With reference to the foregone discussions on the human behavioral pattern, an optimum location has to be decided. Ideally, if the application was a bedroom, a good starting place will be to install it focusing away from the exit point of the room. Focusing the bed must be avoided to prevent nuisance sensing. Place the Hawkeye on a shelf or mount it on a wall at least 6 feet above the ground. Let it settle for a minute and then walk past it. One can observe the LED on the face of the Hawkeye flicker to confirm that the motion has been sensed.

The physical installation part is easy. Being lightweight and compact, the sensor can be installed almost anywhere.

A few adjustments have to be made before the full operation of the Hawkeye motion sensor as described below;

Remove the battery cover on the front of the Hawkeye II and install two AAA alkaline batteries. The unit defaults to Housecode A and Unit Code 1, and it defaults to see motion all the time. If the batteries are being replaced, there is a waiting time of 30 seconds before the Hawkeye II will see any motion. Secondly, after a transmission, there is a waiting time of 10 seconds before the Hawkeye II senses motion again.

To change the Unit Code that the Hawkeye II transmits:

Press and hold the Unit button (under the battery compartment lid) until the red LED flashes first and then blinks the current setting. Release and press the button the desired number of times for the Unit Code (once for Unit Code 1, twice for Unit Code 2, etc.). Hold the button on the last press. 3 seconds later the red LED will blink to confirm the setting.

To change the House Code that the Hawkeye II transmits:

Use the same procedure as above to change the Housecode (pressing the House button instead). One press for Housecode A, two presses for B, etc. Hold the button for 3 seconds on the last press. To check the code that is set: press and hold the House or Unit code button - the red LED blinks back the appropriate number of times for the code that is set.

To set the Hawkeye II to detect motion in night only or all day long:

Press the Unit button once. The red LED flashes. Press and hold the House button. The green LED turns on. 3 seconds later the Hawkeye II will report its dusk/dawn setting as follows: The red LED will blink once if the Hawkeye II is set to detect motion at all times. The red LED will blink twice if the Hawkeye II is set to detect motion only when it's dark. To change this release and press the House button once for operation at all times, or twice for operation only at night. Hold the button for 3 seconds on your last press. The red LED will then report the setting with one or two blinks.

To change the delay after motion is detected before an OFF code is transmitted:

Press the House button once. The red LED flashes. Press and hold the Unit button. The green LED turns on. 3 seconds later the red LED will report its delay setting as follows:

1 blink for 1 minute.

2 blinks for 2 minutes.

3 blinks for 4 minutes.

4 blinks for 8 minutes.

5 blinks for 16 minutes.

6 blinks for 32 minutes.

7 blinks for 64 minutes.

8 blinks for 128 minutes.

9 blinks for 256 minutes.

To change this, release and press the Unit button the appropriate number of times for the delay you want and hold the button pressed for 3 seconds on the last press.

The countdown time for reset has been set at 1 minute for the thesis project.

4.6.4 Motion detector transmitter

The motion detector transmitter also called the Transceiver module receives the radio frequency from Hawkeye and translates them into X10 signals to be relayed over the house wiring.

The module has an antenna and must be located in the vicinity of the Hawkeye sensor to receive the RF signals without interruption.

Before getting this unit operational, set the House code by turning the dial in the front from A thru P. The unit code can be selected by the sliding switch to a position 1 or 9.

Simply plug the module into the power socket and extend the antenna fully. The transceiver module has a plug socket to connect a lamp directly. But since the thesis project is using the Fuzzy controller in the PC, this feature is not being used.

4.6.5 Lighting sensor

The lighting sensor is used to detect the level of ambient lighting conditions at any point of time. This is presented as one of the inputs to the Fuzzy controller. The lighting sensor has a built in photocell to measure the intensity of light. The sensor module comes with an adjustable pot whose setting will determine at what level of light should it get actuated.

To begin operation with this sensor, plug the power cord into the electrical receptacle. Using a small tool, adjust the House code dial located in the front. There is a choice of 16 house codes from A through P. Set each module between the unit code 1-4 or 5-8 by adjusting the sliding switch located in the front.

The Sundowner[7] lighting sensor has got a number of manual features that can be used directly to control a lamp without a PC. But in the thesis project, this is used to provide an input on the lighting level.

The Sundowner lighting sensor can be set to latch only at one level of illumination although this level is user adjustable. This means each controller shall actuate for a unique lighting level. The thesis project considers only one level of switchover to demonstrate the operation. Ideally, there should be bank of these controllers each set at different lighting levels to enable the system constantly monitor this input.

[7] Trademark of X10.com

4.6.6 Installation strategies of devices for Lighting Control

For the purpose of simplicity, we shall consider a room inside a home for which the fuzzy controller is being implemented.

The various components shall be located in this model floor plan. It must be mentioned that these locations are recommended for a start. The system has to be tuned to come up with final location details. The locations will also vary depending on the individual profile. A house having kids needs to be fitted with a different strategy than the one with only adults. Figure 19 describes the floor plan.

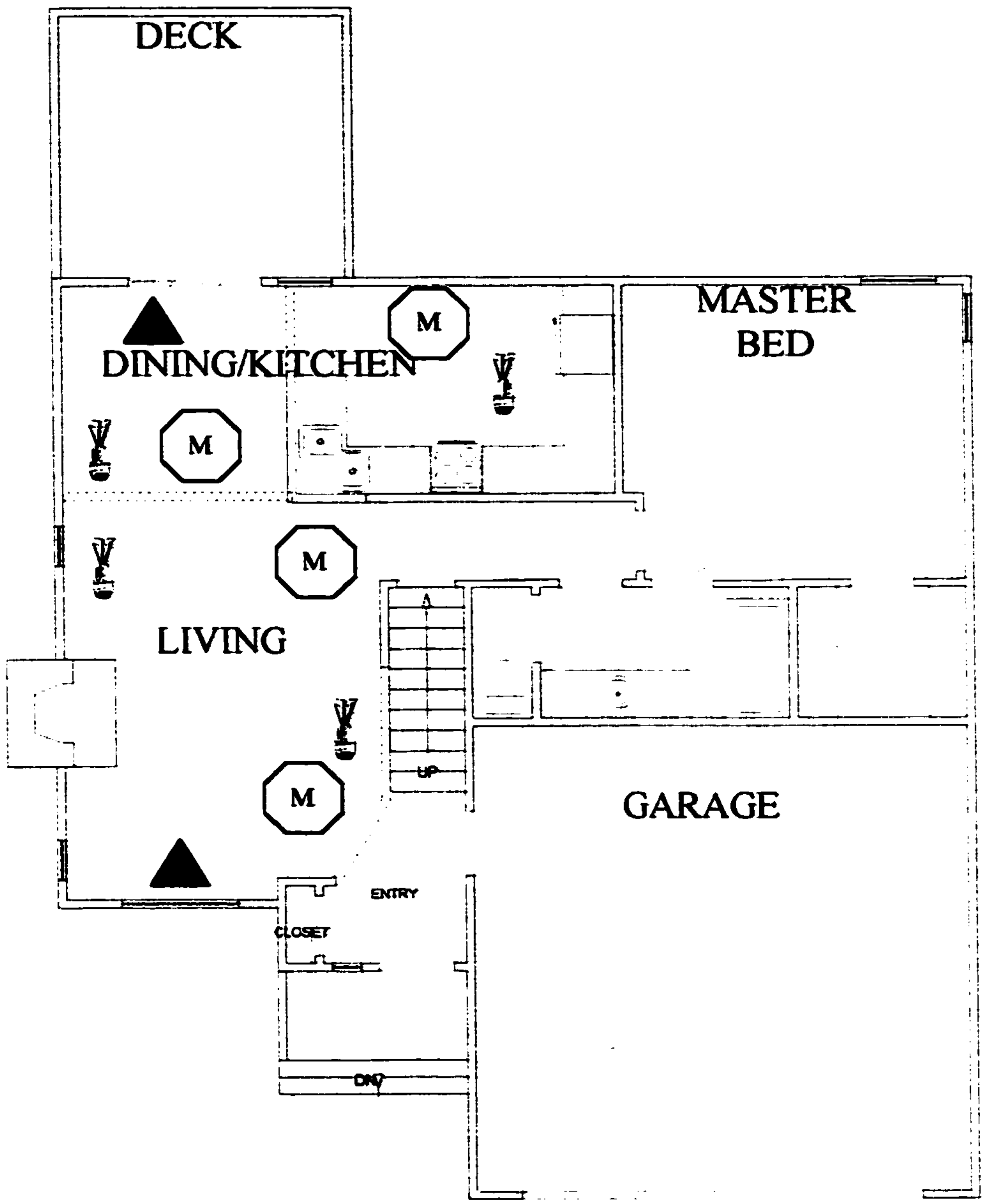


Figure 19 Floor plan example of a home

Lighting sensor

Motion sensor

Lamps

4.7 Logic Narrative for Lighting Control

There are two modes of operation under which the lights can be controlled from the PC;

- Manual
- Automatic

Refer to Section 4.12 for details on computer screens and operations.

When the manual mode is selected, the operation is straightforward. The desired address for the lamp is selected and the required commands are initiated from the screen. In the manual mode, one can use either the percentage dim command or the preset dim command. The preset dim is scaled from 0 to 63, where 0 is the lowest and 63 is the highest illumination that is possible.

Under the automatic mode of operation, the fuzzy controller takes command. The output is generated based on the existing lighting level and the current occupancy level. For ease of operation the following scales are used ;

Occupancy – uses a scale of 0 to 10, where 0 is empty and 10 is maximum occupancy.

Lighting level – uses a scale of 0 to 100%

Dimmer output – uses a scale of 0 to 100%

The inputs are constantly scanned and any change in their levels will enable the controller determine the output again and adjust the dimmer output level.

Once in Automatic mode, no supervision is necessary and the system will adjust to the changes on its own. The display on the PC tells the current level of status for both inputs and the outputs.

Should one desire to interrupt the automatic operation, either it can be switched to the manual mode or exit the program altogether.

4.8 Fuzzy Controller design for Lighting

The Fuzzy controller that is used in the thesis project is designed on the basis of a Sugeno zero order fuzzy model. The objective of the controller is to maintain an optimum lighting level conditions. The controller consists of two inputs namely the occupancy and the ambient lighting level as described below in figure 20.0

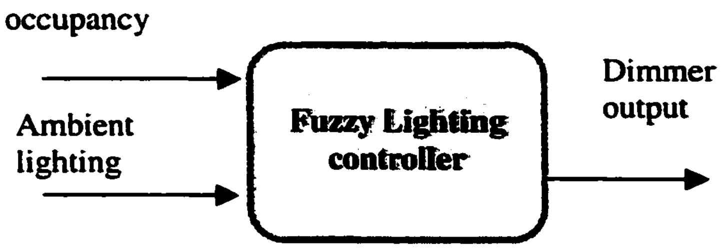


Figure 20 Fuzzy Lighting Controller

The "occupancy" input tells the controller the level of occupation inside the space controlled by the target system. The linguistic variables for occupancy are;

- Low
- High

The "ambient lighting" measures the existing lighting level conditions as seen inside the space of the target system. The linguistic variables for the ambient lighting level are

- Low
- Medium
- High

The fuzzy antecedent and consequent membership functions are defined over the universe of discourse per following;

Occupancy – 0 to 10 (equal to 0 to 100%)

Lighting – 0 to 100%

Dimmer output – 0 to 100%

The dimmer output would be the crisp output based on the set of fuzzy rules that are described below.

if Illum is *low* *AND* Occupancy is *Low* *then* Dimmer is *low medium*

if Illum is *low* *AND* Occupancy is *High* *then* Dimmer is high

if Illum is *Medium* *AND* Occupancy is *Low* *then* Dimmer is *low*

if Illum is *Medium* *AND* Occupancy is *High* *then* Dimmer is *medium high*

if Illum is *High* *AND* Occupancy is *Low* *then* Dimmer is *very low*

if Illum is *High* *AND* Occupancy is *High* *then* Dimmer is medium

For a Sugeno model, the system outputs are crisp singletons.

Where Illum is the ambient lighting level and dimmer is the output from the Fuzzy controller to the lamp dimmer.

Now that the fuzzy if-then rules are established, the following steps takes place to determine the crisp output of the Sugeno system.

1) First we fuzzify the inputs by finding their membership to each antecedent membership function.
2) Next, we apply the Sugeno fuzzy *AND* implication operation to find the degree of fulfillment of each rule.
3) The output is the weighted average of the rule fulfillments.

The plots obtained for the membership function of the input variables and the crisp outputs of the system are shown below.

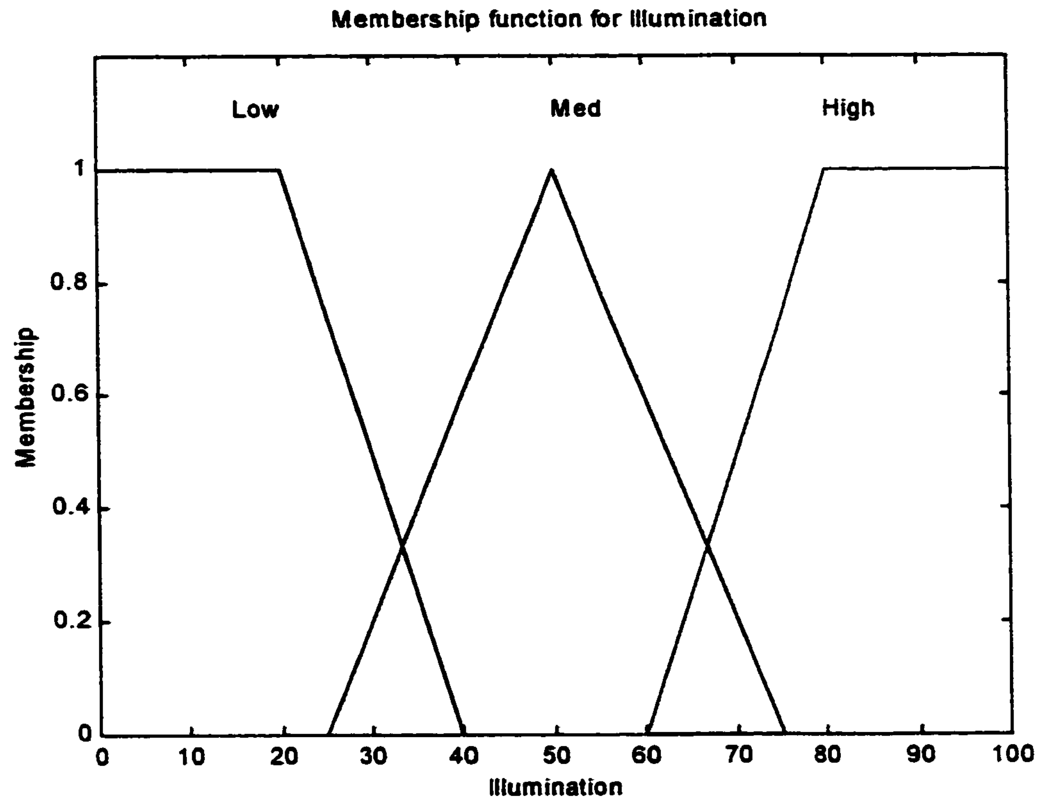


Figure 21 Membership function for Illumination

The low and high are trapezoidal membership functions, while the membership function for medium is triangular.

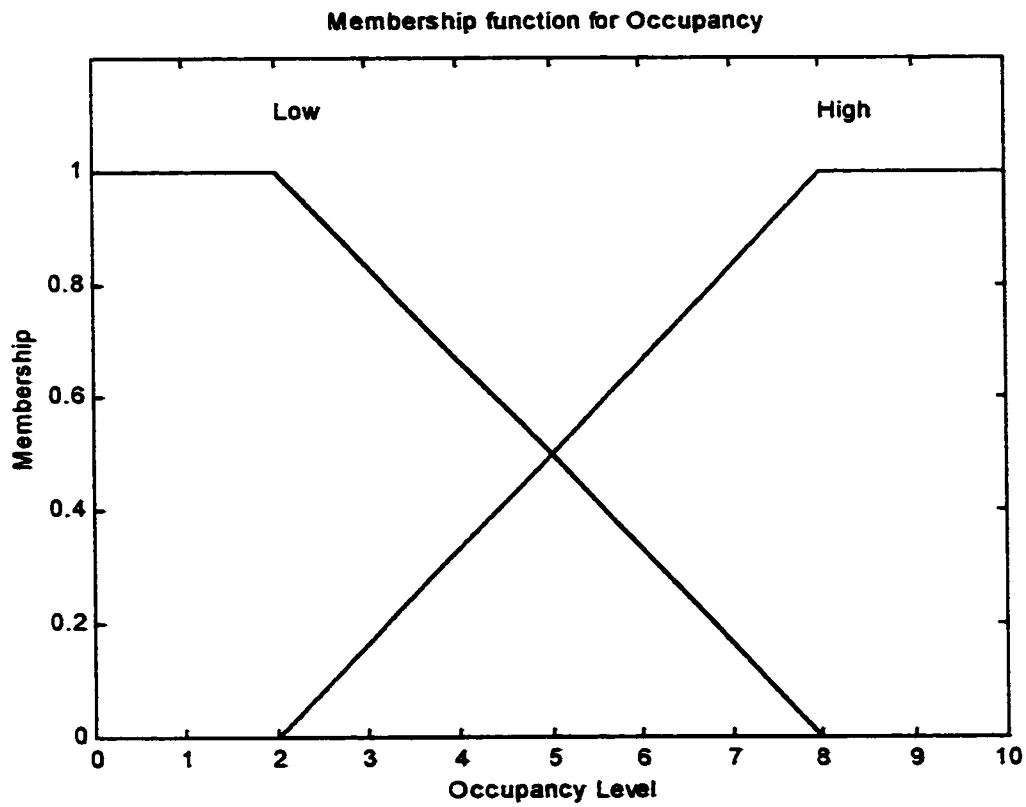


Figure 22 Membership function for Occupancy

Both the low and high membership functions are trapezoidal.

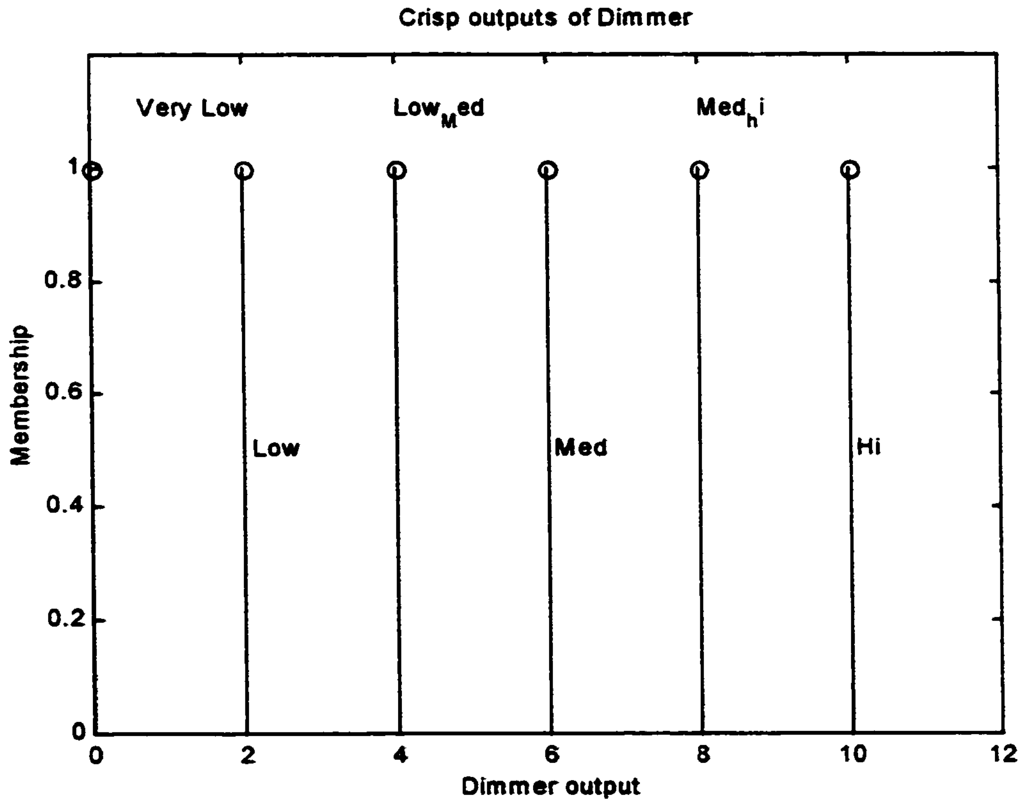


Figure 23 Crisp outputs of Dimmer

The outputs of the controller are the crisp singletons.

4.9 Hardware Integration for Fuzzy Lighting Control

Now that the different X10 components have been discussed in the previous section, we shall examine how these will connect together to operate as one system. Figure 25 shows the overall block diagram of Lighting control.

A note on wiring requirements:

Before that is done, there is prerequisite on how the wiring is to be done whether it is home or other application.

By design, X10 equipment does not need much in the way of special wiring. The manufacturer recommends the following for all homes with X10 installations:

- Run the ground wire to each wall switch location (the ground cable is optional in many light switch wiring schemes and unless it is specified, it may get omitted). Most enhanced X10 wall switches require a 3-wire (hot, neutral and ground) Figure 24 shows the wiring schematic.

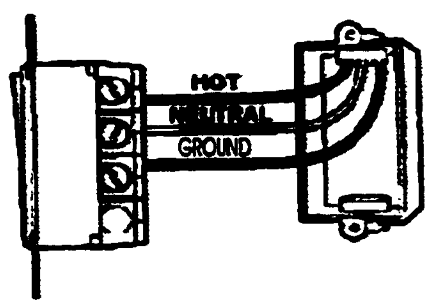


Figure 24 X10 Typical Power wiring

- Plan the locations of X10 control panels and have a hot and neutral wire run to a J-box at these locations.

- Specify the installation of deep J-boxes in all locations that are likely to use X10 switches, receptacles or transmitters. While X10 products fit in the spacing offered by all electrical boxes marketed in North America, the deep models have extra working space and make the installation go a little easier. Deep boxes only costs a few cents more than normal depth models. Look for single gang boxes that are 22cu or higher (cubic inches) and double gang boxes that are 36cu or higher.
- Install a whole house surge arrestor. If a significant number of X10 components are used, adding one of the whole house surge protectors will give ensure against costly damage to both X10 system and other delicate electrical equipment being used.
- Install an X10 bridge at the incoming electrical service. One of the common problems with X10 signals is getting them between the two legs of electricity that service your home. The manufacturer recommends using a Coupler-Repeater (amplifier) if the area to be covered is 3,000 square feet or greater. Smaller homes will work well with a passive signal bridge.

Lighting Control

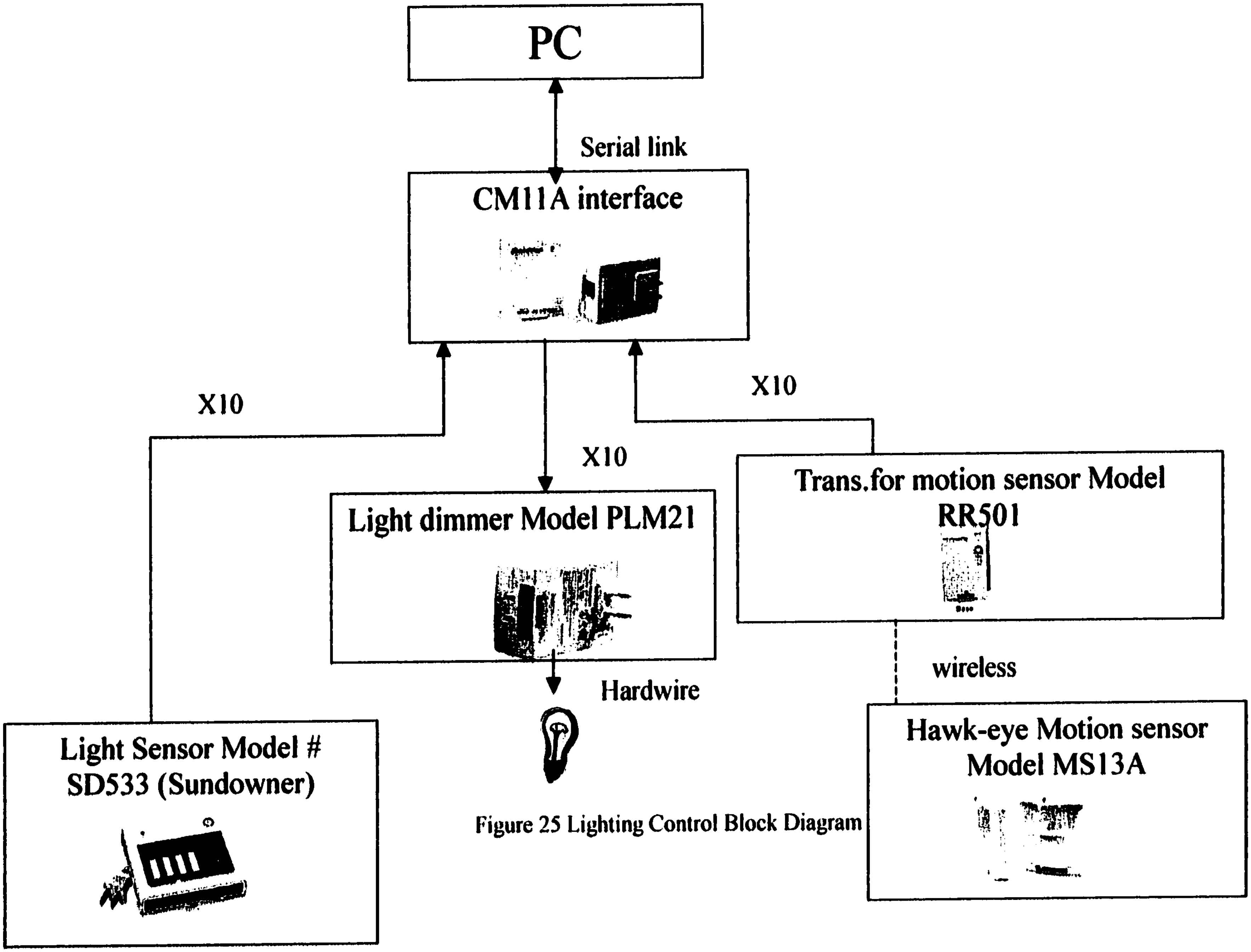
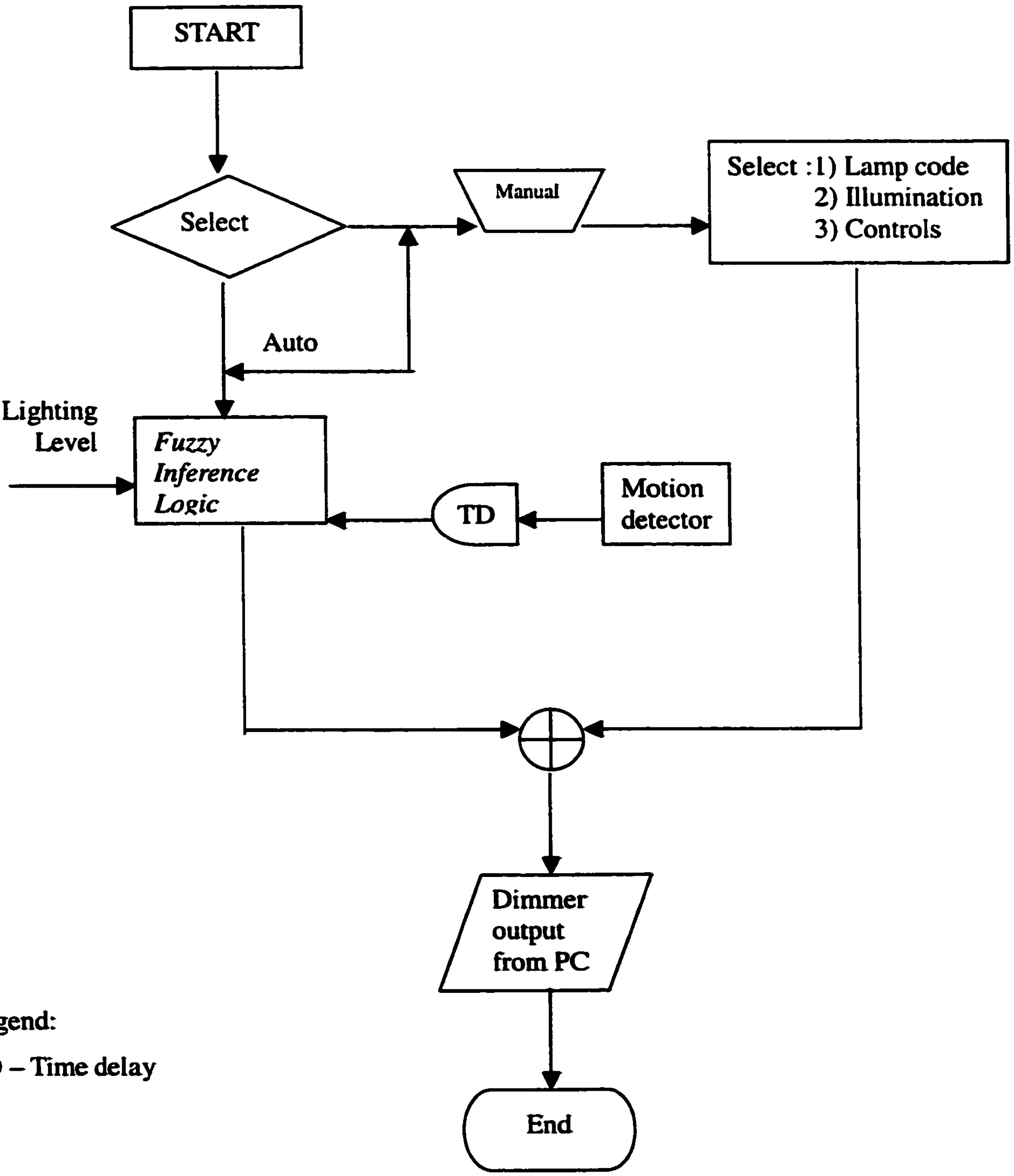


Figure 25 Lighting Control Block Diagram

4.10 Software description for Lighting Controller

The Fuzzy logic controller resides in the PC. The code is written in Microsoft Visual Basic with Active X controls for easy operation. Figure 26 shows the flow diagram for the logic and software that is implemented.



Legend:

TD – Time delay

Figure 26 Flow diagram of logic

4.11 Software Configuration for the Home Automation System

This section describes the software configuration for the entire Home Automation System including Lighting Control, Temperature Control, Safety Control and Water Heater Control. It is assumed that all the hardware components are installed in place and it is ready for automatic operation.

The software runs on Windows 95[8] operating system. The default communication port for the CM11A communications is COM 1. The code can be modified if a different communication port is required.

The software contains user friendly graphics as described in detail in Section 4.11. It is further supported by Voice interface as described in Chapter 8.

This section describes the general software configuration for the entire Home Automation system proposed for the thesis project. Individual operations for different controls are covered in the respective chapters.

4.11.1 Visual Basic and Matlab Client Server arrangement

Another feature of the software is the use of Matlab[9] as a Dynamic Data Exchange server (DDE) [17]. Visual Basic acts like a client in this client-server arrangement. Matlab has features that allow Windows based program to establish API (Application Program Interface) wherein Matlab can be accessed as a server for Dynamic Data Exchange. Conversations are established between Visual Basic and Matlab. When Visual Basic initiates the conversation, it must identify two DDE parameters that are required by Matlab.

[8] Windows 95 is a product of Microsoft Corporation
[9] Matlab is a product of MathWorks Inc., MA

- "Service name" which is the name of the application it intends to have conversation with, and secondly,
- The "topic" as defined by the subject of the conversation.

As soon as Matlab receives a request for conversation from Visual Basic, assuming on a topic it supports, it acknowledges establishing the conversation.

Conversation is established on "items" that are meaningful to both the applications.

Since Matlab acts as a server, it is associated with a unique "service name". The service name is normally derived from the application's executable file, which is usually the application's executable file without the exe extension. For Matlab the service name is "Matlab".

The topic identifies the subject of DDE conversation that is relevant to both client and server. The topics for Matlab are *System* and *Engine*.

There are more than one items supported by each topic. The item also identifies the data being exchanged during the DDE conversation.

4.11.2 Accessing Matlab as a server

Visual Basic application can access Matlab as a DDE server by using the Matlab Engine library directly. The Matlab Engine operates by running in the background as a separate process from the main Visual Basic program. Whenever Matlab server is accessed, the service name, topic and item must be specified. Figure 27 shows the Matlab DDE hierarchy. Reference [17] for details

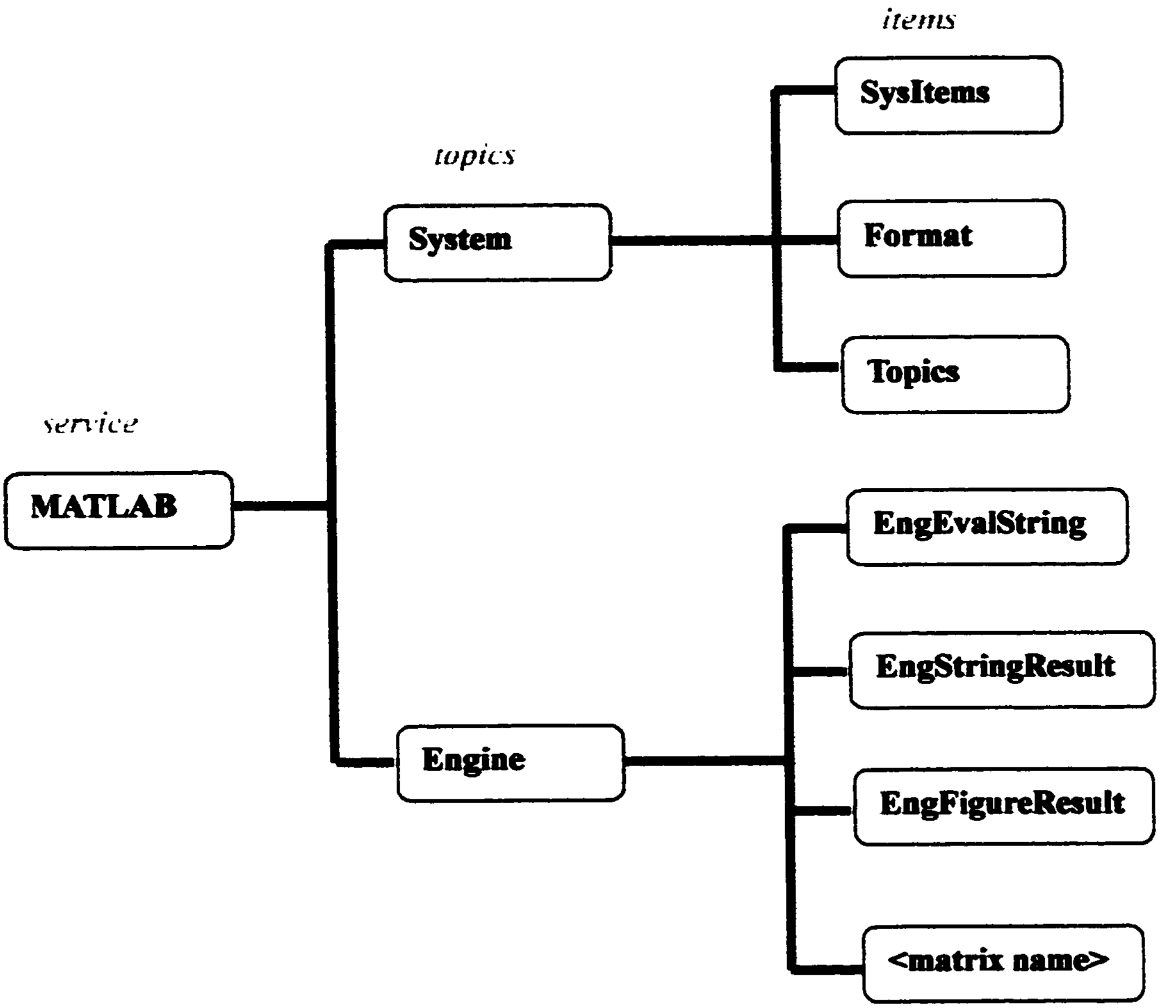


Figure 27 Matlab DDE hierarchy

By using the Matlab engine, one can call a math routine or build an entire system. The Matlab library contains contains the following routines as described in Table 4

Function	Purpose
engOpen	Start Matlab Engine
engClose	Shutdown Matlab engine
engGetArray	Get a Matlab array from Matlab engine

Function	Purpose
engPutArray	Send a Matlab array to the Matlab engine
engEvalString	Execute a Matlab command
engOutputBuffer	Create a buffer to store Matlab text output
engOpenSingleUse	Start a Matlab engine session for single non shared use

Table 4 Matlab runtime routines used in DDE

The block diagram in Figure 28 illustrates the client server arrangement.

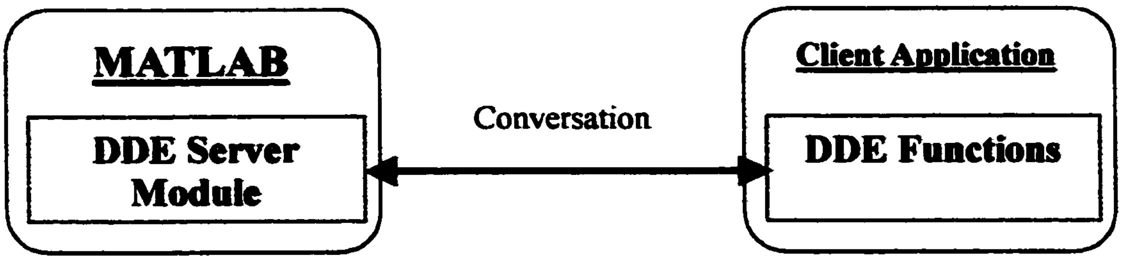


Figure 28 Client-Server arrangement of Matlab & Visual Basic

This feature is used in the thesis project as the fuzzy logic controller resides in Matlab.

4.12 Software Operating Instructions for Lighting Control System

The various screens and sequences are described below.

Step 1:

Matlab must be running in the background. There is no specific version requirement although the Fuzzy toolbox must be available.

Step 2

Start the program "Home Automation" and the screen prompts for the user name and the password. At this point only one user with a password is configured.

Enter the user name as "Windsor" and the password to begin the login process.

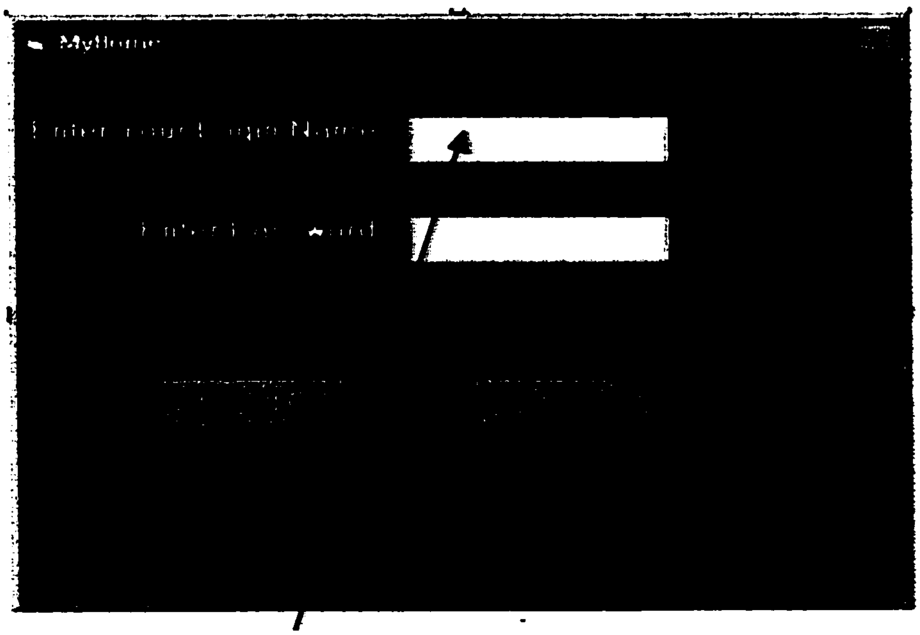
Figure 29 displays this sample screen.

Step 3

Figure 30 screen is displayed on successful logon. This is the Main menu screen from which one can select the controls required. Select "Lighting" to start the Lighting control. This screen also starts the Voice recognition software. Refer Chapter 8.0 for details.

Step 4

Select either Auto or Manual operation as shown in Figure 31



The default login name is "Windsor"

Figure 29 Window – Login Screen

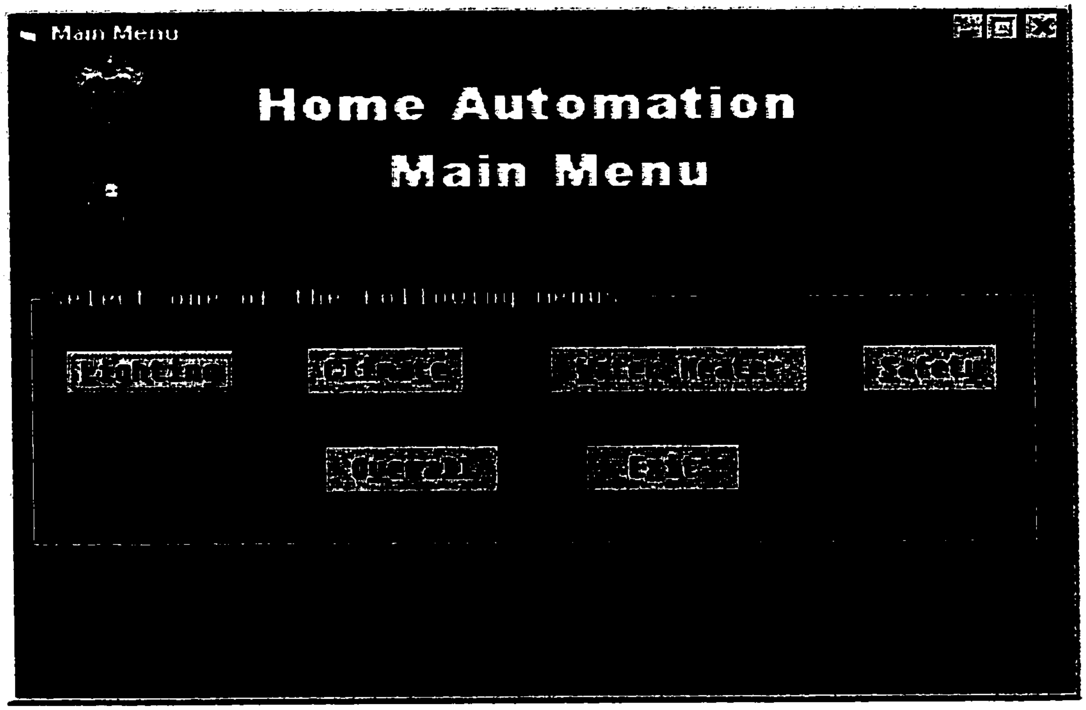


Figure 30 Window – Main Menu

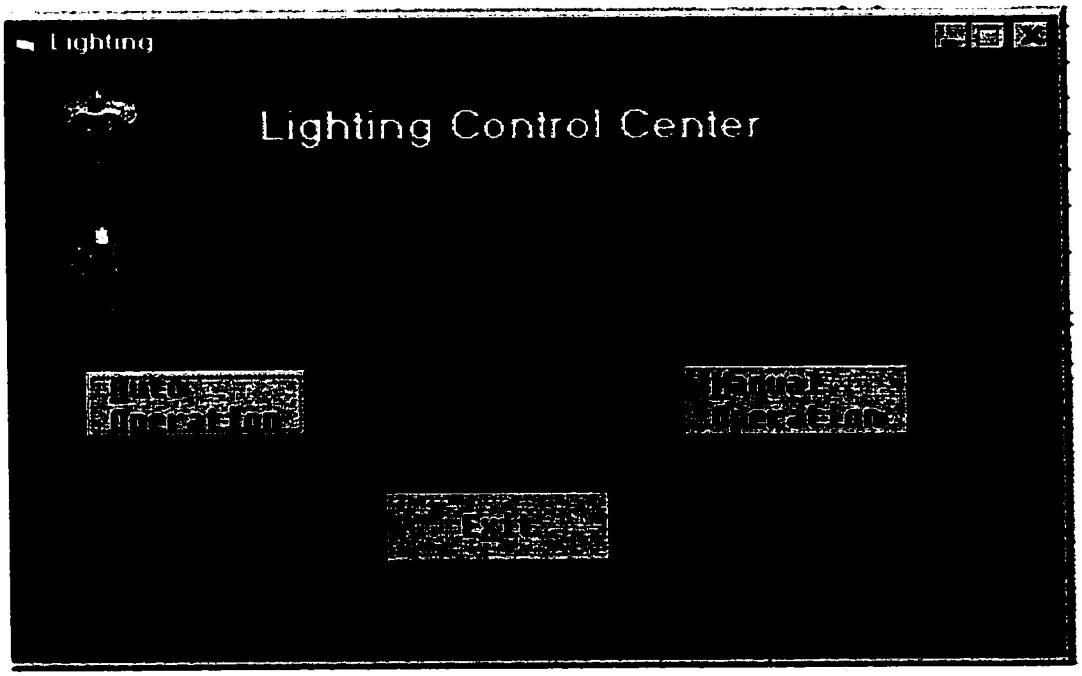


Figure 31 Window- Lighting Control Center

Step 5

The screen in Figure 32 will be displayed if Manual operation is selected.

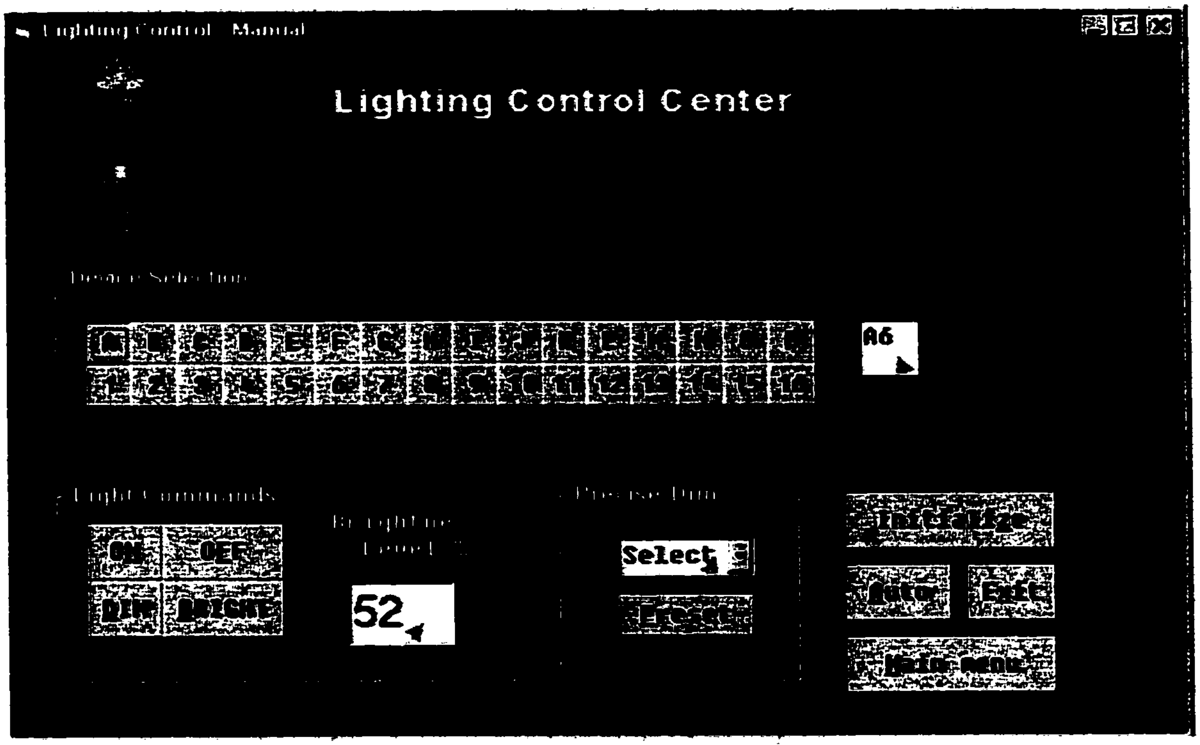


Figure 32 Window – Lighting Control Manual Operation

The first step to be done in manual operation is to "initialize" the CM11A.

Select the house code and device code from the array of buttons. The code selected is displayed on the little window next to the array of buttons.

One can choose either the ON/OFF commands or the %dim/bright commands by entering the required value in percent in the text box. The Precise dim command triggers the pre-determined level of lighting according to the value selected from the pull down menu. It ranges from 0-63 where 0 is the lowest and 63 the highest illumination level. To end Manual operation either switch to Auto or Exit the program by hitting Exit button.

Step 5:

The screen in Figure 33 is displayed if Auto operation is selected from Step 3 above.

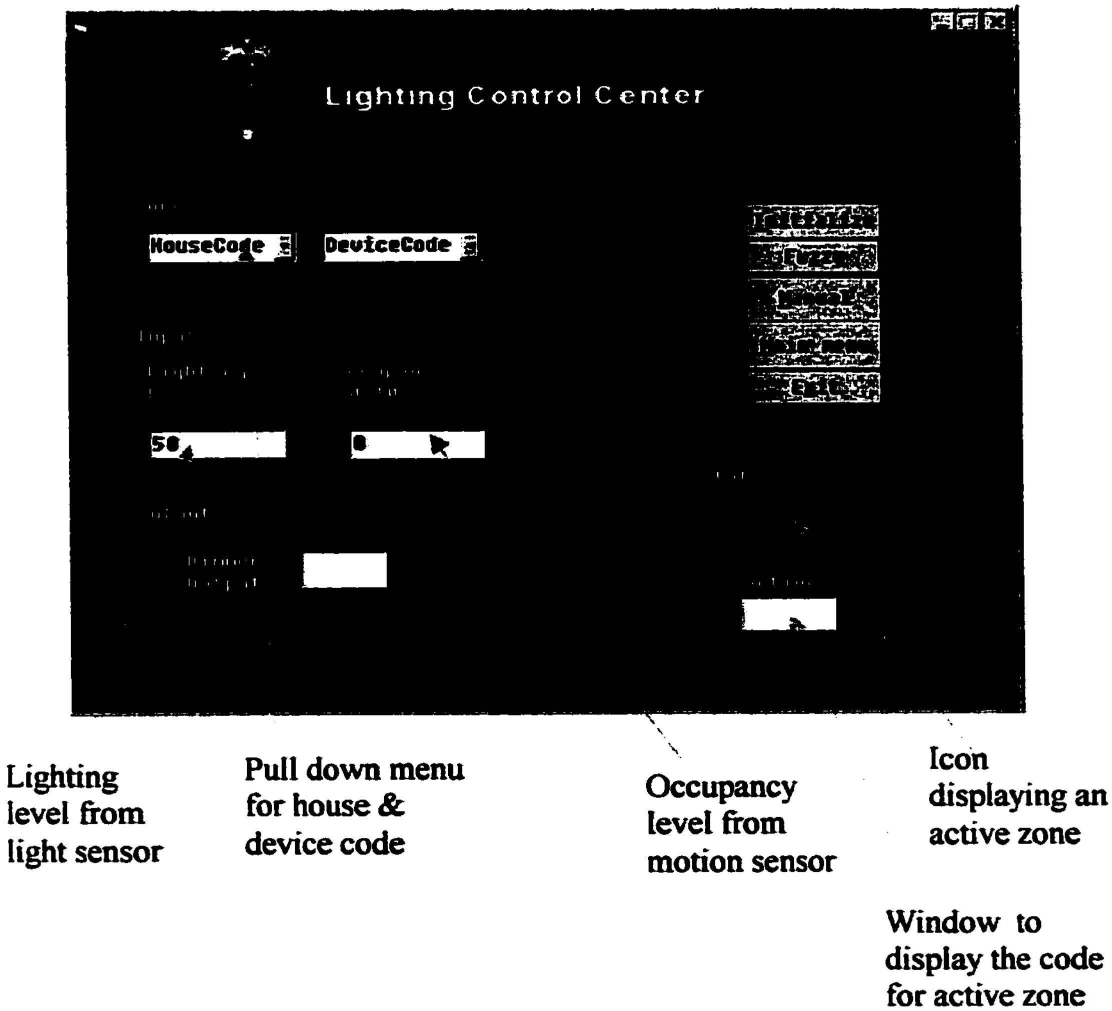


Figure 33 Window- Lighting Control Auto Operation

To start, hit the "initialize" button.

Select the house code and device code of the lamp(s) required in Auto operation. It may be noticed that when this window pops up, the default value for lighting level is 50% and for the occupancy it is 3. As the logic executes, it pulls the current status and updates accordingly. One can start the fuzzy control either by hitting Fuzzy button or by waiting and allowing the scan to start updating.

Depending on the value of the dimmer output window, the lamp brightness varies.

To end Auto operation, either goto manual mode by selecting the Manual button or press Exit button to end the program. However, the program has to be continuously in operation to maintain the auto operation.

CHAPTER 5

5.0 TEMPERATURE CONTROL SYSTEM

Objective : To design and implement a Fuzzy logic controller for maintaining a comfortable temperature inside the house. Reference [11].

5.1 Introduction

Fuzzy logic control is an innovative and efficient approach to control non-linear and unpredictable systems. As discussed before, this method uses rule based reasoning rather than mathematical equations. Fuzzy control systems work more closely with the human behaviour and have certain advantages over the traditional PID controllers.

The Temperature Control System monitors the temperature inside the given enclosed area and controls the Heat Ventilation and Air-Conditioning (HVAC) system accordingly. The output is computed based on existing conditions and the desired temperature, which is entered as the set point. In this thesis project, under simulated conditions, the computed Fuzzy output is shown as a percentage of the HVAC output. A negative value of the output indicates heating is ON, meaning that the temperature set point is greater than the ambient conditions, while a positive value of the output indicates cooling ON. At zero output, the HVAC is turned off. Appropriate software graphics are incorporated to indicate the status of the Temperature Control System.

5.2 Fuzzy Controller design for Temperature Control

A rule based fuzzy system is being implemented. Two inputs are being considered for the fuzzy controller. They are;

- Ambient temperature in °C

- Rate of change of temperature in °C/sec.

The rate of change in temperature is given by;

Rate Input (°C/sec)= (Present temp°C) – (Previous temp°C) / Time(secs) between 2 temp readings.

The rate of change in temperature is an indirect measurement of the occupancy inside a given enclosed area. Any increase in occupancy will increase the rate of change in temperature until the temperature stabilizes at a certain value and vice versa. Reference [12].

The linguistic variables for the input and output of the Fuzzy controller are;

<u>Inputs:</u>

Ambient Temperature input has 5 linguistic variables namely (in °C):

- Temperature far below set point.
- Temperature near below set point.
- Temperature at set point.
- Temperature near above set point.
- Temperature far above set point.

Set point is the desired temperature entered into the system.

The Universe of Discourse for Ambient Temperature is 0°C to 40°C.

These linguistic variables are shown in Figure 34.

The Rate of change in temperature input has 5 linguistic variables namely (°C/sec);

- Rate of temperature change is a fast decrease.
- Rate of temperature change is slow decrease.
- No change in the Temperature rate.

- Rate of temperature change is a slow increase.
- Rate of temperature change is a fast increase.

The Universe of Discourse for the Rate of change in Temperature is –2°C/sec to +2°C/sec. These linguistic variables are shown in Figure 35.

Outputs:

The Heat ventilation and Air conditioning (HVAC) output has 5 linguistic variables namely;

- 100% Heating.
- 50% Heating.
- Zero output.
- 50% Cooling.
- 100% Cooling.

The Universe of discourse for the HVAC output is –100% to +100%, wherein –100% corresponds to maximum heating and +100% corresponds to maximum cooling. These linguistic variables are shown in Figure 36.

The system receives a temperature input from temperature sensor while the software calculates the rate of change in temperature.

Twenty five different combinations are presented based on human experience that form the basis of fuzzy rules. They are;

1) If Temp far above SP AND Temp change is a fast increase THEN 100% cooling
2) If Temp near above SP AND Temp change is a fast increase THEN 100% cooling
3) If Temp at SP AND Temp change is a fast increase THEN 50% cooling
4) If Temp near below SP AND Temp change is a fast increase THEN 50% cooling

5) If Temp far below SP AND Temp change is a fast increase THEN no change
6) If Temp far above SP AND Temp change is a slow increase THEN 100% cooling
7) If Temp near above SP AND Temp change is a slow increase THEN 50% cooling
8) If Temp at SP AND Temp change is a slow increase THEN 50% cooling
9) If Temp near below SP AND Temp change is a slow increase THEN no change
10) If Temp far below SP AND Temp change is a slow increase THEN 50% heating
11) If Temp far above SP AND no Temp change THEN 50% cooling
12) If Temp near above SP AND no Temp change THEN 50% cooling
13) If Temp at SP AND no Temp change THEN no change
14) If Temp near below SP AND no Temp change THEN 50% heating
15) If Temp far below SP AND no Temp change THEN 50% cooling
16) If Temp far above SP AND Temp change is a slow decrease THEN 50% heating
17) If Temp near above SP AND Temp change is a slow decrease THEN no change
18) If Temp at SP AND Temp change is a slow decrease THEN 50% heating
19) If Temp near below SP AND Temp change is a slow decrease THEN 50% heating
20) If Temp far below SP AND Temp change is a slow decrease THEN 100% heating
21) If Temp far above SP AND Temp change is a fast decrease THEN no change
22) If Temp near above SP AND Temp change is a fast decrease THEN 50% heating
23) If Temp is at SP AND Temp change is a fast decrease THEN 50% heating
24) If Temp near below SP AND Temp change is a fast decrease THEN 100% heating
25) If Temp is far below SP AND Temp change is a fast decrease THEN 100%heating

(where SP is the Set point temperature)

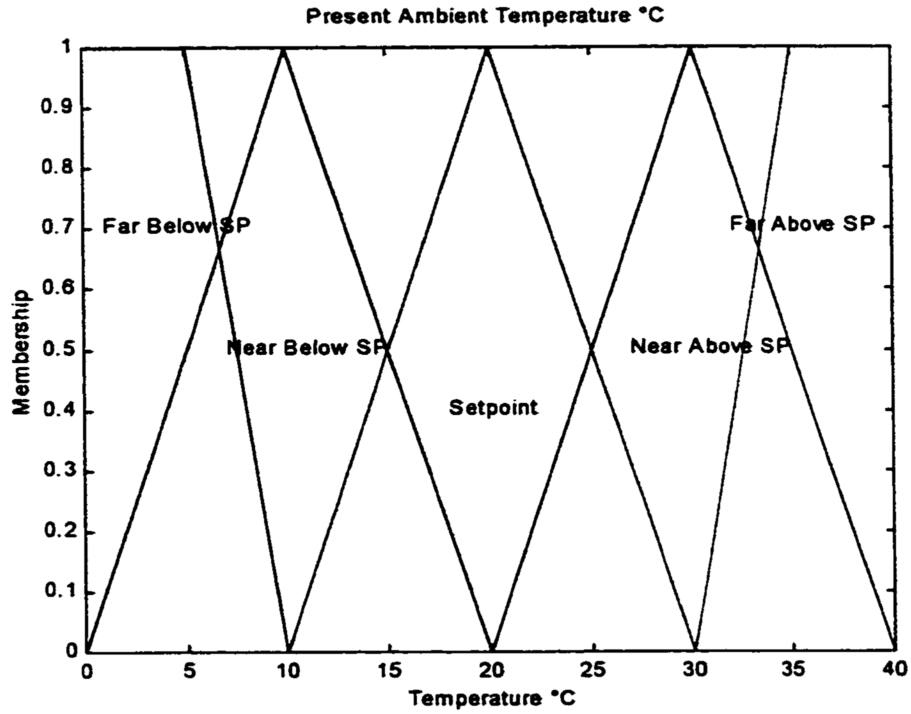


Figure 34 Membership function for Ambient Temperature

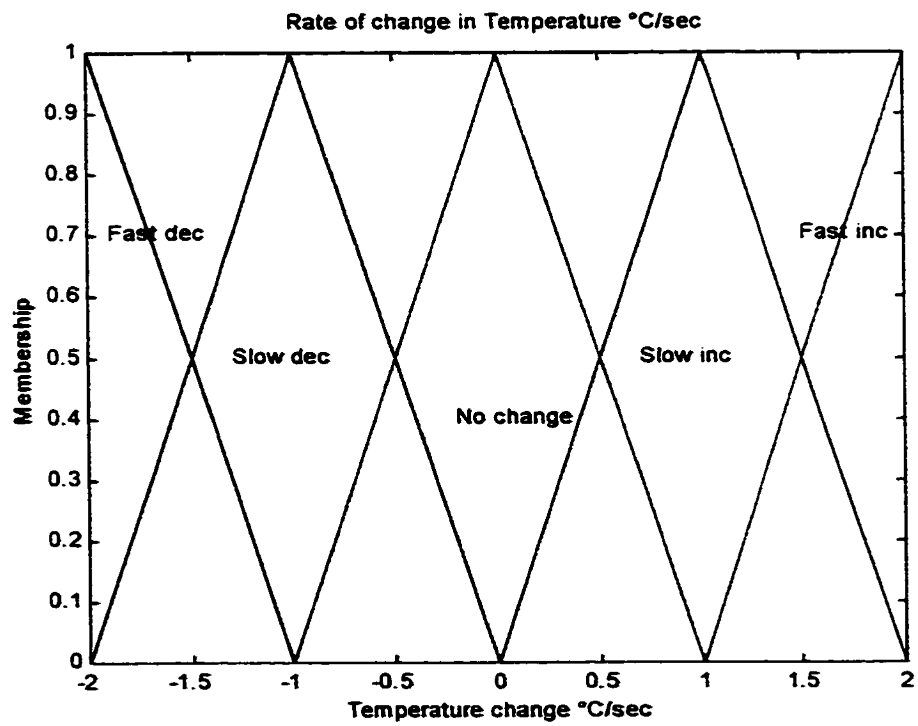


Figure 35 Membership function for Rate of change in Temperature

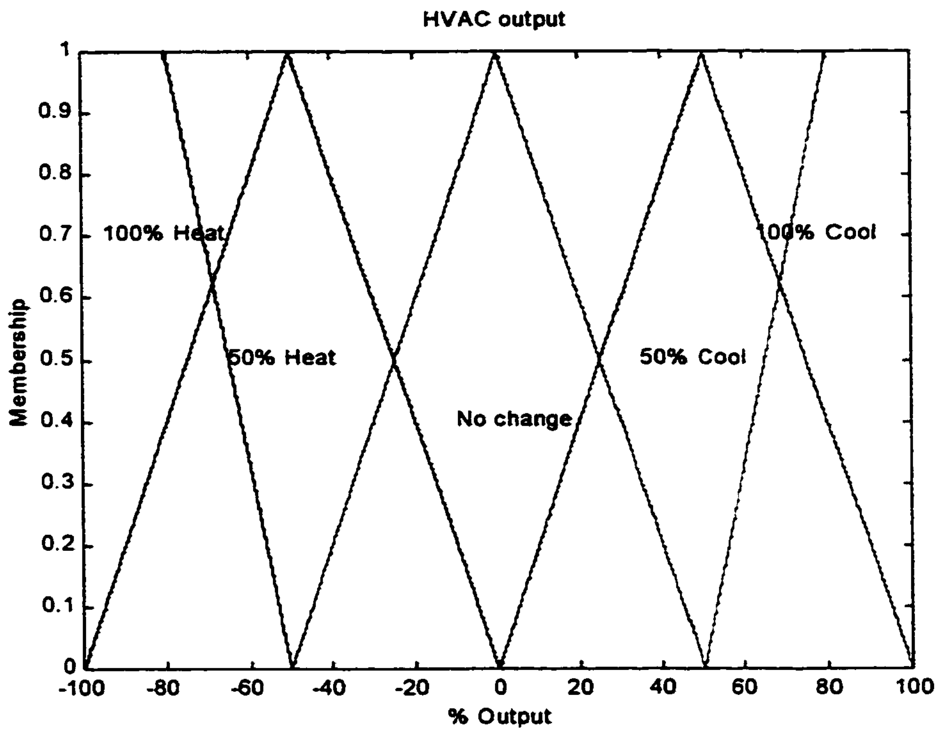


Figure 36 Membership function for Fuzzy Controller output

5.3 Hardware Implementation for Temperature Controller

Hardware implementation of the Temperature Control System is divided into two parts;

- Hardware for input sensors
- System output hardware for Temperature control.

5.3.1 Temperature Sensing

Temperature sensing is done using a high precision 1-wire[10]®Digital Thermometer manufactured by Dallas Semi conductor. See Reference [13]for details.

This project uses a direct analog measurement for temperature input into the system. Model DS18S20 of Dallas Semiconductor was used as a temperature sensor. Some of the features of this sensor are;

- Measures temperature from –55°C to +125°C
- Accuracy of +/- 0.5°C from –10°C to 85°C
- Provides 1-wire interface for serial communication with the PC.
- Thermometer has 9-bit resolution.

The DS18S20 uses the 1-wire bus protocol that implements bus communication using one control signal. The sensor does not require any external power supply. Power is supplied through the 1-wire via the DQ (data) pin when the bus is high. The high bus signal also charges an internal capacitor, which supplies power to the device when the bus is low. This method of deriving power from the 1-wire is also called "parasite

[10] 1-wire is a Registered trademark of Dallas/Maxim Semiconductor.

power". Alternatively, the DS18S20 can be powered by an external power. The pin description for DS18S20 is given in the Table 6 below.

PIN DESIGNATION	DESCRIPTION
GND	Ground
DQ	Data Input / Output pin. Open drain 1-wire interface pin. Also provides power to the device when used in parasite power mode.
V_{DD}	This is an optional pin. This must be grounded for operation in parasite power mode.

Table 6 DS18S20 Pin assignment Details

5.3.1.1 Hardware Implementation of Temperature sensor

In order to connect the 1-wire Digital Thermometer, the following components are required;

- DS18S20 High Precision Thermometer
- Dallas Semiconductor 9-pin serial interface 9097U[12]

The block diagram in Figure 37 explains the hardware connection of the various components for operation under parasite mode.

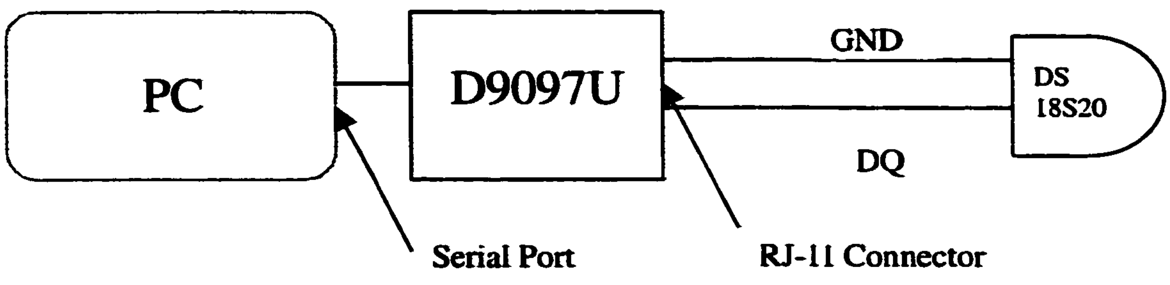


Figure 37 DS18S20 Hardware Implementation

More than one sensor can be connected to the system by paralleling the bus connections. But typically one sensor should be sufficient to cover a reasonable area within the house.

5.3.1.2 Software Implementation of Temperature sensor

The basic driver software controls the data retrieval and data transfer operations. This is coded in Visual Basic. See Reference [14] for details. The driver software communicates with the main program to provide seamless integration.

5.3.2 HVAC Controller

The final control element for the temperature control system is the HVAC unit. The HVAC unit shall function to match the temperature requirements as derived by the output of the Fuzzy Controller.

For the purposes of the project, the output of the Fuzzy controller varies from –100% to +100%. This spectrum includes heating, cooling and "no output". The negative value corresponds to heating and the positive value for cooling. If the output is zero, the system is OFF. The output is shown in the software graphics in Figure 39.

In reality, this can be integrated to the HVAC controls in a number of ways. Firstly, the manipulated variable can be assumed to be the fan speed with either constant Heating or Cooling. At zero output, the fan will stop and there will not be any heating or cooling at this point.

Secondly, the heating or cooling rate can be manipulated with constant fan speed.

---

[12] Alternatively, a 25 pin connector can also be used from the parallel port.

5.4 Software Implementation for Temperature Control System

The software code for the Fuzzy controller is written in Visual Basic with Active X controls. The code resides in the PC and like in the Lighting Control (see Section 4.11) there is a DDE (Dynamic Data Exchange) with Matlab on real time (see Section 4.11.1). This arrangement is very efficient to implement complex coding for the Fuzzy Controller, as well as optimizes the speed.

5.4.1 Software Operation

The various screens and sequences are described below.

Step 1:

Matlab must be running in the background. There is no specific version requirement although the Fuzzy control toolbox must be available in the directory.

Step 2:

Start the program "Home Automation" and the screen prompts for the user name and the password. At this point only one user with a password is configured.

Enter the user name as "Windsor" and the password to begin the login process.

Figure 29 displays this screen.

Step 3

Figure 30 screen is displayed on successful logon. For Temperature control, select "Climate" button. This prompts to display the screen asking for the set-point as shown below in Figure 38.;

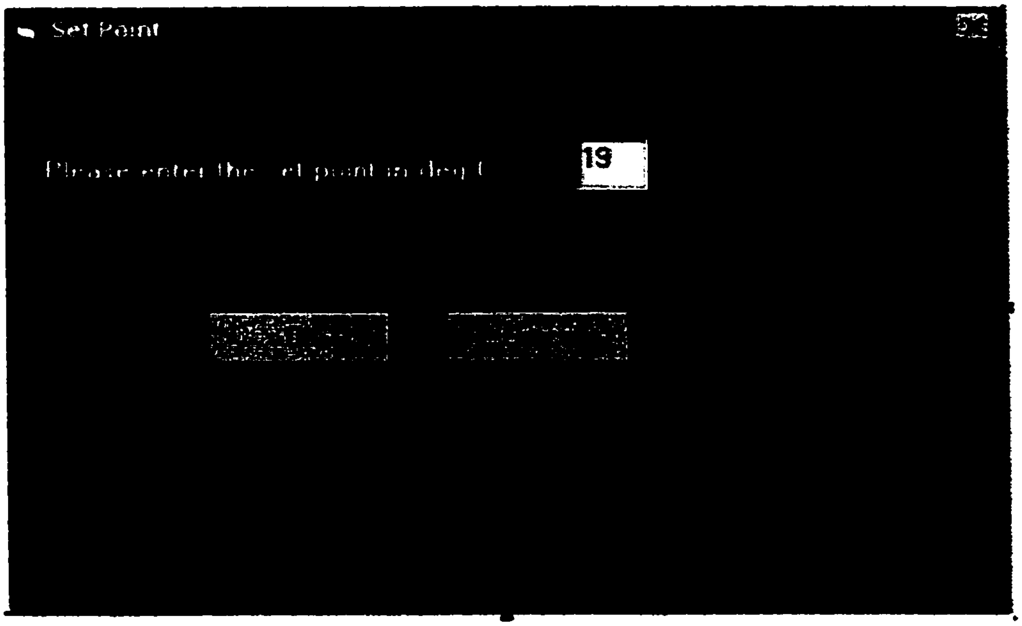


Figure 38 Window - Temperature Setpoint

Step 4

Once the setpoint is entered, the Temperature control window is displayed as shown below in Figure 39. The Fuzzy Controller receives temperature input from the 1-wire sensor and computes the rate of change in temperature, which is the second input to the controller. Considering the setpoint entered in the previous step, the Fuzzy Controller determines the overall output that drives the HVAC to the desired level.

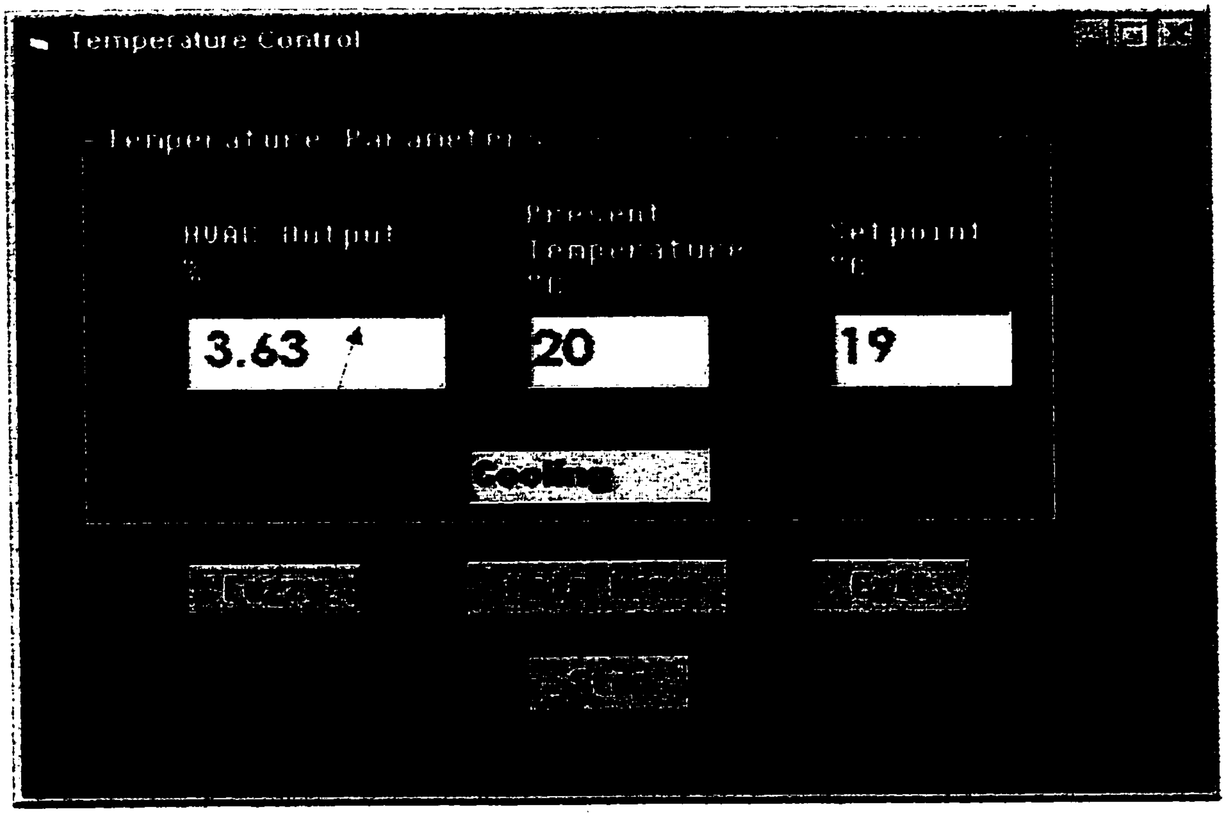


Figure 39 Window -Temperature Controller

HVAC output

STOP command for Shut Down

The Fuzzy controller either produces a Heating or Cooling effect depending on the Setpoint and current temperature conditions.

The setpoint can be changed to any required value just by entering the box.

The "Fuzzy" button invokes the Fuzzy controller operation, while the "Stop" button shuts down the temperature controller system and gives a zero output.

CHAPTER 6

6.0 WATER HEATER CONTROL

Objective: To design and implement an intelligent controller for water heater. Such a controller will maintain the required temperature and also conserve energy based on utilization and requirement.

6.1 Introduction

Yet another application integrated into this Home Automation system is Water Heater controls. This area normally does not get the attention it really deserves, but being a vital system within the home in terms of reliability and energy consumption, it is believed that optimizing the controls would be desirable.

6.2 Logic Rules for Water Heater Controls

The water heater is typically operated by On-off control. Traditionally, if the temperature of the water is below the Set point, the Heater turns ON and vice versa. In this thesis project, we introduce the control of the water heater to be dependent on the occupancy of the house and the flow of hot water in the pipe (consumption). This will ensure that the water heater is not turned ON during prolonged absence of occupants in the house.

The water heater control has 3 inputs namely occupancy of the house, water temperature and flow of hot water. Table 7 below describes the Logic of the water heater operation.

In the table below, "CAUSE" describes the various scenarios of the inputs to the control logic, whereas "EFFECT" is the output of the logic caused by the input.

CAUSE	EFFECT
A1) Temp. below setpoint A2) Flow Switch actuated	Heater is turned ON until setpoint is reached.

CAUSE	EFFECT
A3) Occupancy sensed	
B1) Temp below setpoint B2) Flow switch actuated B3) Occupancy not sensed	Heater stays OFF as house is empty.
C1) Temp. below setpoint C2) Flow Switch not actuated C3) Occupancy sensed	Heater is ON as the temperature is below setpoint and occupancy is sensed.
D1) Temp. below setpoint D2) Flow Switch not actuated D3) Occupancy not sensed	Heater stays OFF as the house is empty and there is no consumption of hot water.
E1) Temp. above setpoint E2) Flow switch actuated E3) Occupancy sensed	Heater stays OFF as the water is hot.
F1) Temp above setpoint F2) Flow switch actuated F3) Occupancy not sensed	Heater stays OFF as water is hot.
G1) Temp. above setpoint G2) Flow switch not actuated G3) Occupancy sensed	Heater stays OFF as water is hot.
H1) Temp. above setpoint H2) Flow switch not actuated H3) Occupancy not sensed	Heater stays OFF as water is hot.

Table 7 Water heater operation – Logic table

6.3 Hardware Implementation for Water Heater Control

X10 offers relay modules that provide potential free contacts for logical switching. If the water heater is an electric device it is easy to connect it to the relay module. If it is gas fired, a solenoid valve (electro-pneumatic valve) may be introduced in the gas line to control the gas flow. It is recommended to consult the manufacturer's instruction in designing a suitable interface. This project however is based on electric furnace and the following block diagram in Figure 40 explains the setup[13].

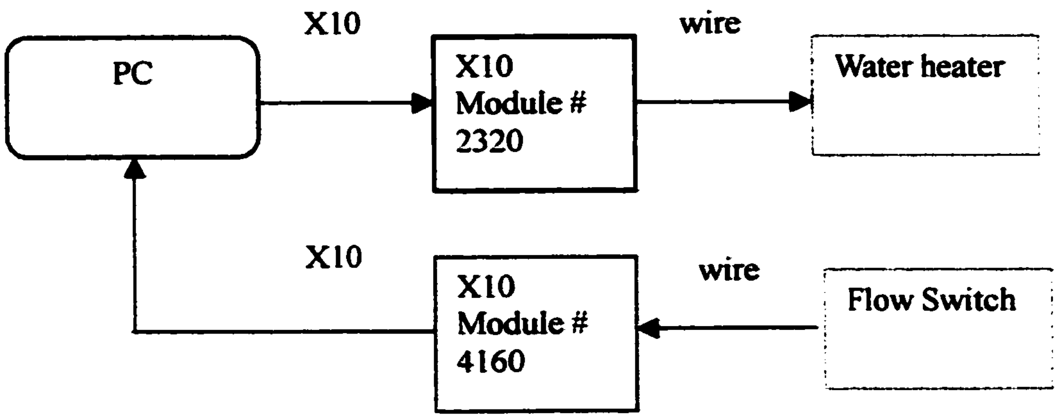


Figure 40 Water heater – Hardware setup

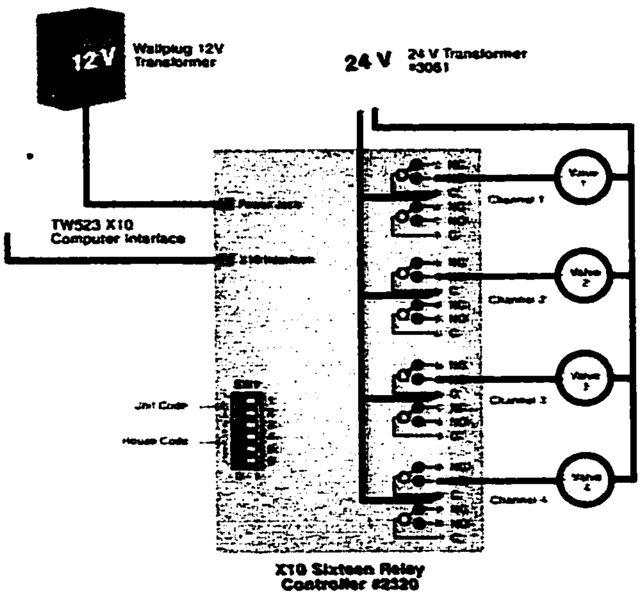


Figure 41 X10 Relay Output Module # 2320

[13] See Figure 25 for Occupancy Sensor connection.

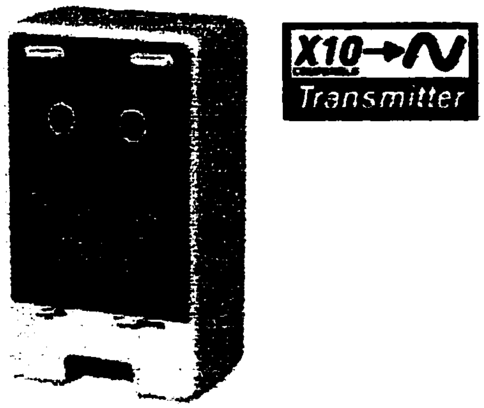


Figure 42 X10 Digital Input Module Model 4160

6.4 Software Implementation for Water Heater Control

The various screens and sequences are described below.

Step 1:

Matlab must be running in the background. There is no specific version requirement although the Fuzzy toolbox must be available. It may be noted that while Matlab is not specifically required for this control application, the software is integrated to control the complete house including Fuzzy controls. Hence absence of Matlab will flag an error window.

Step 2

With the Visual Basic program running, the screen prompts for the user name and the password. At this point only one user with a password is configured.

Enter the user name as "Windsor" and the password to begin the login process.

Figure 29 displays the Login Screen.

Step 3

Figure 30 screen is displayed on successful logon. Select "Water" to display the Water Heater control screen shown below in Figure 43.

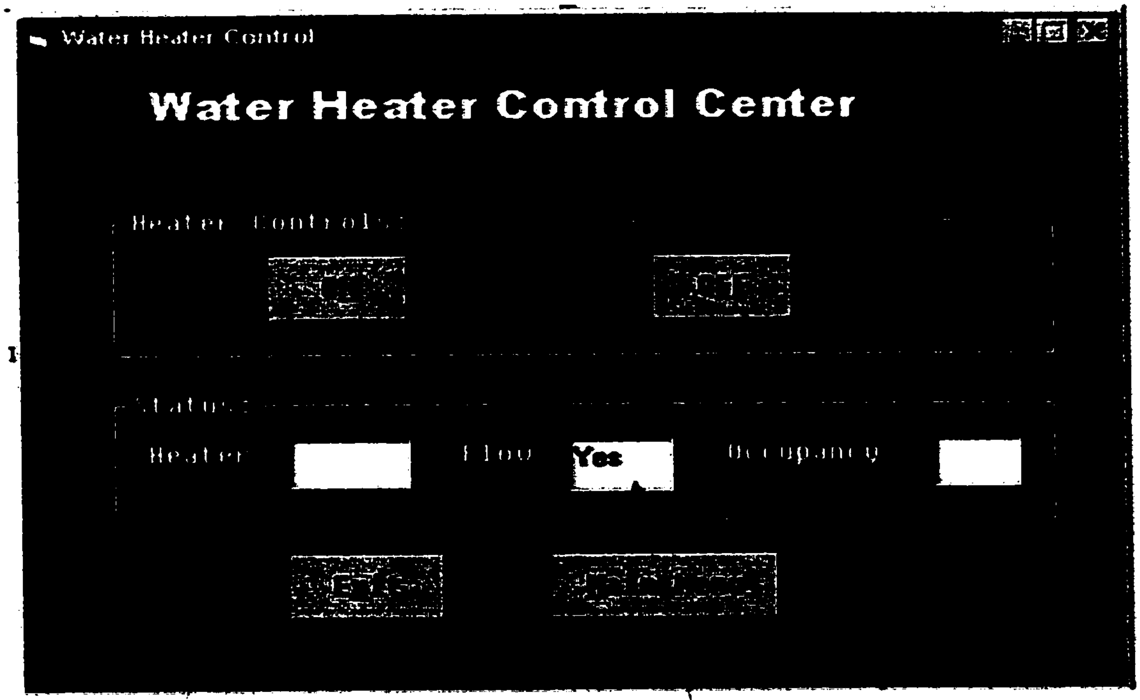


Figure 43 Window - Water Heater Control

Manual Control of
Water Heater

Status of
Flow sensor

Step 4

Two types of controls are possible for the Water Heater – Manual and Auto. The Manual operation overrides the Automatic control.

For the Manual operation, simply turn the Heater ON or OFF by pressing the appropriate window from the screen above.

Under Automatic operation, the status of the Flow of hot water and that of the occupancy is displayed. The Logic Rules of section 6.2 and Table 7 is executed accordingly.

CHAPTER 7

7.0 SAFETY CONTROLS AND OPERATION

Objective: To maintain the Safety of the house by monitoring the equipment continuously for any unsafe conditions.

7.1 Introduction

To ensure that the home is a safe place to live, the Safety controls provides continuous automatic monitoring of potential hazards.

Safety controls monitors the following conditions within the home;

- Smoke detector
- Hydrocarbon detector
- Carbon monoxide detector

In the event of any abnormal situation being detected, suitable alarm is provided to alert the occupants.

7.2 Logic Rules for Safety Controls

The logic operation is very straight-forward. At alarm condition caused by either one condition, the HVAC driven by the Temperature Control system in Chapter 5.0 is turned OFF. Audible alarm is provided to alert the residents.

7.3 Hardware Implementation for Safety Controls and Operation

The PC is the central equipment receiving status from the various detectors and gas monitors installed. As dictated by the local municipal codes, smoke detectors are mandatory to be installed at least one per floor. The smoke detectors shall be dry contact type providing variable switching contacts for normal and alarm states.

Similarly, the carbon monoxide detector shall be located near the bedroom areas considering that major accidents result from carbon monoxide leaks during sleeping hours. The carbon monoxide detector shall also provide variable switching contacts during alarm and normal conditions.

The hydrocarbon detector shall be located near the furnace, if gas heating is used. This shall be located around 1 foot from the floor to detect hydrocarbons that are heavier than air. Figure 44 below shows the Hardware implementation.

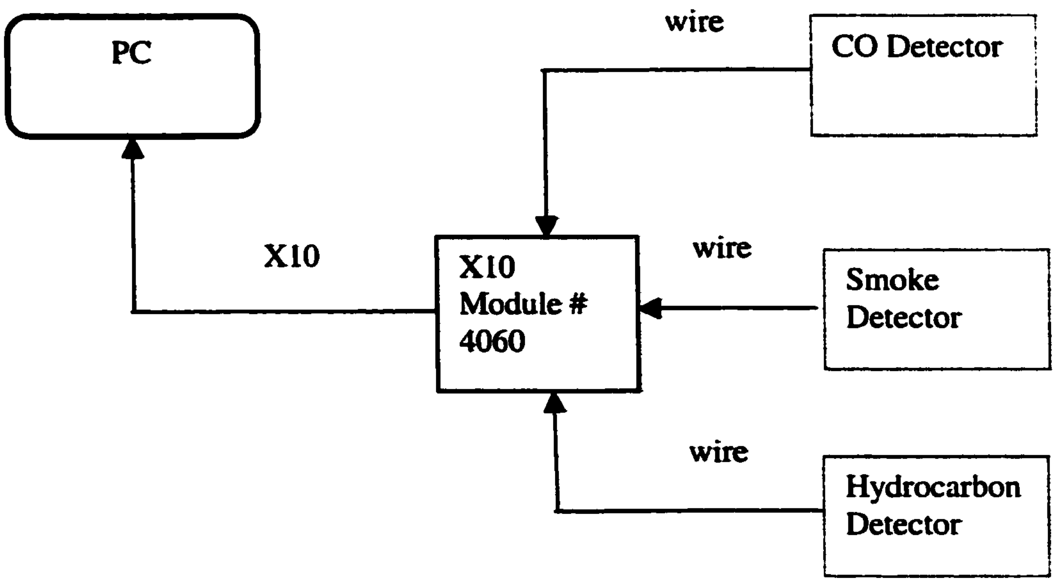


Figure 44 Hardware Implementation for Safety Controls

7.4 Software Implemention for Safety Controls and Operation

The graphics provides selectable location of various rooms in the house to monitor the status. The following steps illustrate the procedure necessary for software operation of Safety controls;

The various screens and sequences are described below.

Step 1:

Matlab must be running in the background. There is no specific version requirement although the Fuzzy toolbox must be available.

Step 2

Start the program "Home Automation" and the screen prompts for the user name and the password. At this point only one user with a password is configured.

Enter the user name as "Windsor" and the password to begin the login process.

Figure 29 displays this sample screen.

Step 3

The Main Menu screen as shown in Figure 30 is displayed on successful logon. Select "Safety" button.

Step 4

Figure 45 will be the screen that is displayed when Safety Control is selected from the main menu.

Each of the detectors can be individually addressed. The pull down menu gives the ability to select the area within the house to be monitored.

If either the Smoke detector or the Carbon monoxide triggers an alarm, an audio visual alarm is generated in the computer screen. Simultaeously, the Temperature Control System is shutdown.

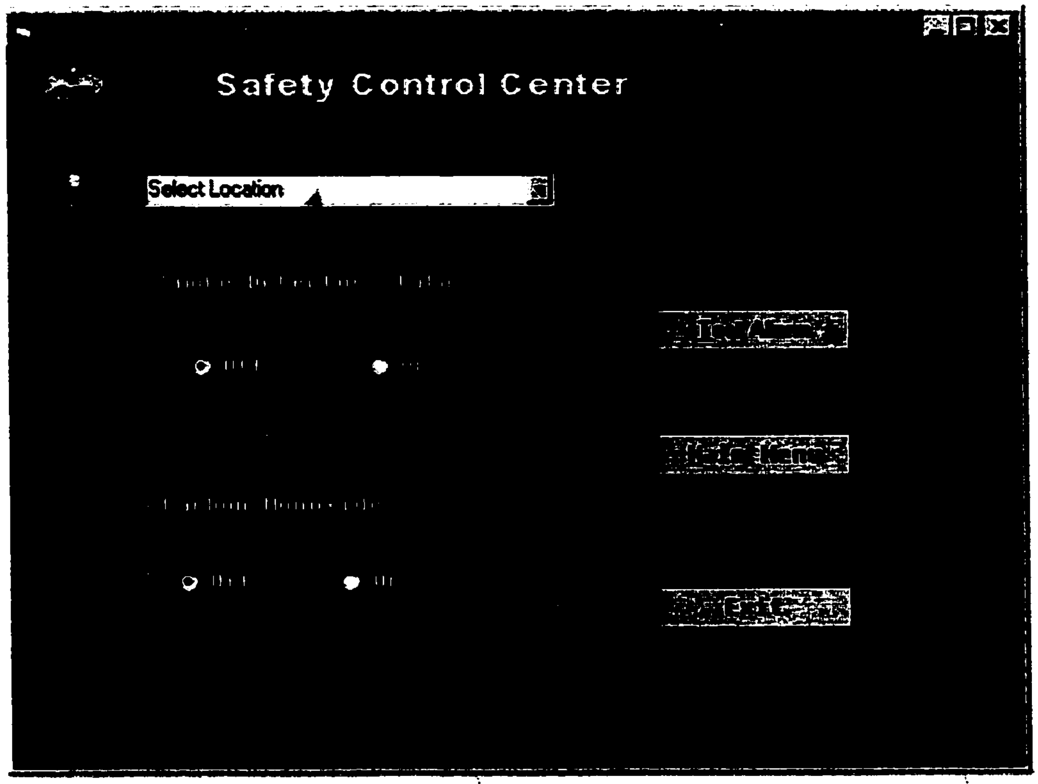


Figure 45 Window - "Safety Control"

Select the room for status monitoring

Status of detectors

Test Alarm button

CHAPTER 8

8.0 VOICE CONTROL AND INTERFACE

Objective: To provide control access to the operation of the Home Automation system via 2-way voice communication.

8.1 Introduction

The Home automation system for the thesis project can be operated through an interactive voice communication method. This interface is a 2 way communication between the user and the PC. The system is "trained" to understand and execute the commands of operation, and in turn provides audio-visual feedback of the event.

The Voice control system is based on Microsoft® Agent[14] Version 2.0 and is developed for the thesis project entirely using Software Development Kit. No additional hardware is used other than the multimedia. See Reference [15]. MS Agent uses a set of programmable software services that supports the presentation of interactive animated characters within the Microsoft Windows® interface. Developers can use creative characters to provide animation and interface with the user. The character is blended to present additional tool for automation by way of voice interface.

MS Agent includes an ActiveX® control that makes its services accessible to programming languages that support ActiveX, including Web scripting languages such as Visual Basic® Scripting Edition (VBScript). This means that character interaction can be programmed even from HTML pages using the <OBJECT> tag. The Visual Basic feature of MS Agent is used to integrate with the rest of the program.

MS Agent is based on Speech recognition engine that is used with the application. The project uses a set of speech input (recognition) engines and speech output (text-to-speech or TTS) engines.

8.2 System Requirements

The following system requirements are needed to run MS Agent. This does not include any software for the development of the application.

- Microsoft Windows® 95, Windows 98, Windows NT® 4.0 (x86) or later
- Internet Explorer version 3.02 or later
- Personal computer with a Pentium 100 MHz or higher processor
- At least 16 MB of memory
- Hard-disk space for core components: 1 MB
- Multimedia kit consisting of sound card, microphone and speakers.

Recommended:

- Hard-disk space for optional components:
- Lernout & Hauspie™ TruVoice® Text-to-Speech engine for speech output: 1.6 MB
- Microsoft Speech Recognition Engine for speech input: 22 MB
- Windows compatible sound card

As mentioned earlier, the conventional multimedia kit is good enough to operate the MS Agent. It is recommended to use a good quality microphone for trouble free operation.

[14] Microsoft Agent and Microsoft Windows are products of Microsoft Corporation.

8.3 Hardware setup

Just to ensure proper operation of the hardware, especially the multimedia, it is important to execute the hardware setup. This will not only check for the proper operation of the hardware, but will also help in adjusting the microphone, volume etc.

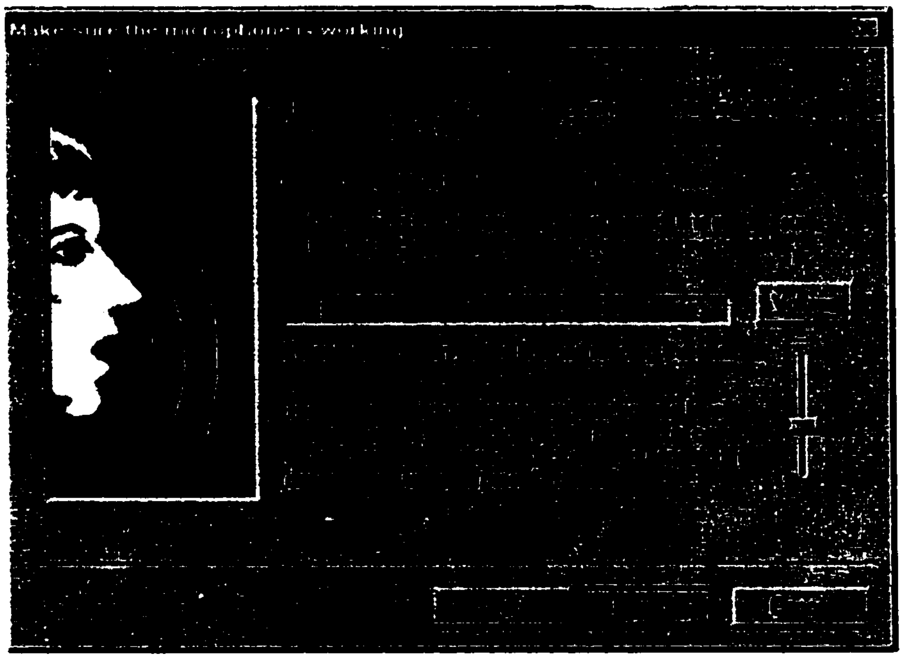


Figure 46 Window - Microphone setup

Figure 46 above shows the Microphone setup window. The Microphone set up wizard goes through the process of setting up the Microphone correctly. It adjusts the volume automatically, and provides different options for Microphone like hand-held, desktop etc before finally configuring it. It also helps in placing the microphone from the speaker at an optimum distance for future permanent operations.

8.4 Software Operation for Voice Control System

The Agent can be configured through the software to perform any interactive tasks required. Creative animation can be developed either using the standard options or by developing custom character. The thesis project uses the character "Genie" out of the standard set of characters available.

Before we get into the voice commands, it is important to verify that the speech engine understands the spoken words correctly. Section 8.3.1 below explains the "Training procedure" during the setup process. Section 8.3.2 gives the voice commands implemented in the thesis project.

8.4.1 Voice Training

As the English accent and the human voice vary from person to person, it is important that the speech engine be made familiar with the user's voice. The voice training procedure is a 10-minute exercise wherein the user reads the displayed text in front of the microphone. As the engine recognizes the speech, the text gets highlighted.

Figure 47 shows the window for Voice training. The instructions to run this training is self-explanatory.

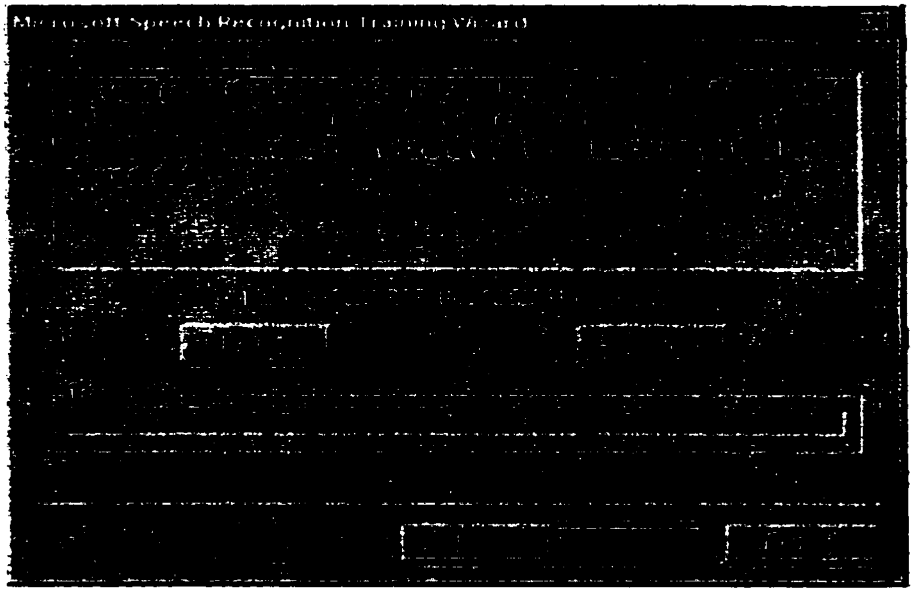


Figure 47 Window - Speech Training

8.3.2 Voice Commands

The system is configured to recognize syllable messages and execute operational commands accordingly. The thesis project presently has setup the voice commands for optimum control. However, there is no end to further addition of commands, which can be done by additional programming. The voice commands that are currently configured are shown in the Table 8 below;

The Current screen is	Voice Command	Action
Main Menu (Figure 30)	*Light*	Display switches to Lighting control system. Genie prompts to choose Auto or Manual control.

The Current screen is	Voice Command	Action
Main Menu (Figure 30)	*Climate*	Display switches to Temperature control system Genie prompts for set point entry.
Main Menu (Figure 30)	*Water*	Display switches to Water Heater Control (Figure 43).
Main Menu (Figure 30)	*Safety*	Display switches to Safety Controls. Genie makes a general comment.
Climate or Lighting or Water or Safety	*Menu*	Display switches to the main menu (Figure 30).
Climate or Lighting or Water or Safety or Overall	*Quit*	Exits the application
Climate or Lighting or Water or Safety or Main menu.	*Overall*	Display switches to Overall Menu. (Figure 48).

Table 8 Voice Commands Menu

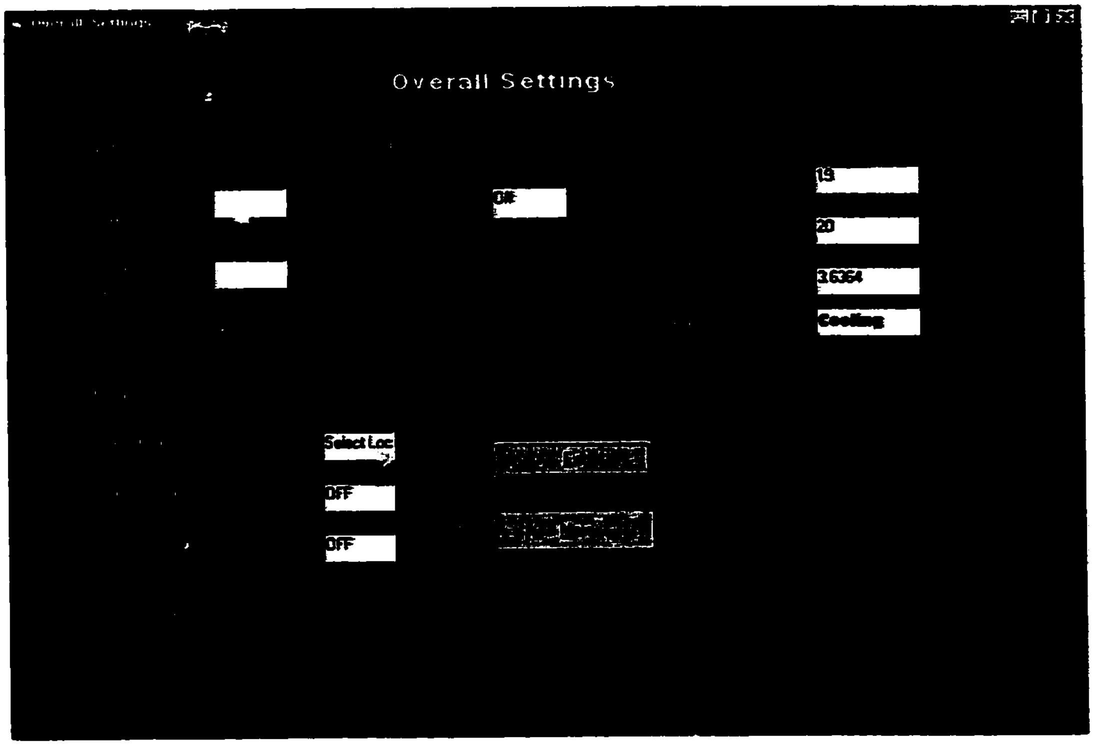


Figure 48 Window - Overall Menu

The overall menu gives the complete status of various parameters of the house. Other menus can be accessed via keyboard or by voice commands.

CHAPTER – 9

9.0 TROUBLE SHOOTING GUIDE

A list of common problems encountered in software and hardware, and the remedial actions recommended is listed in the Table 9 below:

ID	Problem description	Trouble shooting
1	CM11A does not initialize, gives an error message.	Make sure it is connected to COM1. There could be memory overflow and needs to be purged.
2	On manual mode, lamp does not respond to any command.	Check if lamp is turned ON and verify that the address on the screen and the lamp module matches.
3	On Auto mode there is an error message "no response from DDE server"	For auto operation, Matlab has to be running. Secondly, fuzzy logic tool box must be installed.
4	Cannot log on	Passwords and user name are case sensitive. Make sure to use appropriate case.
5	Temperature not being sensed	Make sure the probe is connected via the 9 pin serial port and all connections are proper.

Table 9 Trouble Shooting Guide

CHAPTER - 10

10.0 FUTURE WORKS

The thesis project demonstrates the home automation techniques that can be implemented. Such a system has got endless potential and can be extended to control a variety of other devices within the home.

The future works envisaged beyond the existing scope of work are;

- Web enabling Features

  This would enable accessing the system from the world wide web. This provides facility to remotely control things when one is outside their home.

- Vision Control

  Incorporating cameras into the system has lot of potential for a security system. This particularly suitable when there lot of real estate to be monitored around the house.

- Neural Networks

A network can be trained to operate the systems independently. A feed forward network with 3 layers can be starting point.

BIBLIOGRAPHY

[1] http://www.x10.com/technology1.htm

[2] ftp://ftp.x10.com/pub/manuals/cm11a_protocol.txt

[3] The Foundations of Fuzzy Control; Harold W.Lewis III, International Federation for Systems Research-Volume 10, p9-26.

[4] Fuzzy Sets; Lotfi A Zadeh, 1965, Joint Services Electronic program. p338-353.

[5] Fuzzy Sets and Systems; Lotfi A Zadeh, 1965 "A new View of System theory" presented at the Symposium on System theory, Brooklyn p 29-33.

[6] Fuzzy and Neural Approaches in Engineering; J Wesley Hines,1997, John Wiley & Sons NY, p189

[7] The Foundations of Fuzzy Control; Harold W. Lewis III, International Federation for Systems Research-Volume 10 p77-81

[8] Neuro-Fuzzy and Soft Computing; J.-S.R.Jang, C.-T.Sun & E.Mizutani, 1997, Prentice Hall NJ, p74-82., p335-338.

[9] Fuzzy Control – Synthesis and Analysis; S S. Farinwata, Dimitar Filev & Reza Langari, 2000, John Wiley & Sons England, p114-116

[10] http://www.atip.org/public/atip.reports.91/fuzzy-ds.91.html. Report on Fuzzy system research in Japan – Section 3.18 by Tokyo Electro-Communications University, Department of Communication System.

[11] Decision Making in a Fuzzy Environment; R.E.Bellman & L.A.Zadeh, Management Science vol 17,no 4,1970, p141-164.

[12] http://www.minco.com

[13] http://www.dalsemi.com

[14] ftp://ftp.dalsemi.com/pub/auto_id/softdev/softdev.html

[15] www.msdn.microsoft.com/masagent

[16] A Course in Fuzzy Systems & Control; Li Xin Wang, Prentice Hall PTR, NJ 1997, p 533-534

[17] http://www.mathworks.com

VITA AUCTORIS

Viswanathan Hariharan (Hari) was born in 1964 in India. He is a Chartered Engineer (India) majoring in Electronics & Communication Engineering besides holding a degree in Science from the University of Madras.

Hari has around fifteen years experience on the design of Instrumentation and Control systems for Refinery, Chemical, Power and Nuclear industries. His experience in Control applications focuses on Distributed Control Systems and microprocessor based field instrumentation.