Development of PVDF Tactile Dynamic Sensing in a Behaviour–based Assembly Robot

Taehee Kim

Ph.D.
University of Edinburgh
1996

Abstract

The research presented in this thesis focuses on the development of tactile event signature sensors and their application, especially in reactive behaviour–based robotic assembly systems.

In pursuit of practical and economic sensors for detecting part contact, the application of PVDF (polyvinylidene fluoride) film, a mechanical vibration sensitive piezo material, is investigated. A *Clunk Sensor* is developed which remotely detects impact vibrations, and a *Push Sensor* is developed which senses small changes in the deformation of a compliant finger surface. The *Push Sensor* is further developed to provide some force direction and force pattern sensing capability.

By being able to detect changes of state in an assembly, such as a change of contact force, an assembly robot can be well informed of current conditions. The complex structure of assembly tasks provides a rich context within which to interpret changes of state, so simple binary sensors can conveniently supply a lot more information than in the domain of mobile robots. Guarded motions, for example, which require sensing a change of state, have long been recognised as very useful in part mating tasks. Guarded motions are particularly well suited to be components of assembly behavioural modules.

In behaviour–based robotic assembly systems, the high level planner is endowed with as little complexity as possible while the low level planning execution agent deals with actual sensing and action. Highly reactive execution agents can provide advantages by encapsulating low level sensing and action, hiding the details of sensori–motor complexity from the higher levels.

Because behaviour–based assembly systems emphasise the utility of this kind of qualitative state–change sensor (as opposed to sensors which measure physical quantities), the robustness and utility of the *Push Sensor* was tested in an experimental behaviour–based system. An experimental task of pushing a ring along a convoluted stiff wire is chosen, in which the tactile sensors developed here are aided by vision. Three different methods of combining these different sensors within the general behaviour–based paradigm are implemented and compared. This exercise confirms the robustness and utility of the PVDF–based tactile sensors. We argue that the comparison suggests that for behaviour–based assembly systems using multiple concurrent sensor systems, bottom-level motor control in terms of force or velocity would be more appropriate than positional control. Behaviour–based systems have traditionally tried to avoid symbolic knowledge. Considering this in the light of the above work, it was found useful to develop a taxonomy of type of knowledge and refine the prohibition.

Acknowledgements

During the past years while I have been staying here at the Department of Artificial Intelligence, many people helped and encouraged me, first to survive and second to obtain appropriate knowledge given the course of study, which eventually turned into a measure likely to govern the rest of my life. I am greatly indebted to them.

First of all, I would like to thank my supervisors Chris Malcolm and John Hallam for their guidance, time, and encouragement, during my research and writing of this thesis.

I would also like to thank my colleagues of the Robot Research Lab. Thanks to Graham Deacon for his valuable comments in solving many experimental problems, and also proofreading of this thesis. Thanks to Prabhas Chongstitvatana for his vision routines and advices in the early stage of my study here. Thanks to Edward Jones for valuable conversations and helping me to cope with X window programming. Thanks also to Giovanni Pettinaro, for accompanying many late nights in the laboratory, at the same time being my flat mate and sharing many cooking times and lots of chat.

I wish to thank the staff of the mechanical and electrical workshops for great help with my experimental equipment. My special thanks to Douglas Howie for building the experimental setup and Sandy Colquhoun for help with electronic devices. Thanks also to the computing officers for their support.

I also want to thank many of the friends that shared some time in the Department, Namseog Park, Joanna Bryson, Ashley Walker, Edvaldo Bispo, Soonae Yang, Dibio Borges, and Elena Perez-Minana, for their help and company. Special thanks to Andrew Fitzgibbon for his help with troubleshooting of vision routines.

My thanks to my parents and my brother Dongheon for their valuable encouragement in studying here. I would also like to thank my uncles and other family members for their support and encouragement.

I wish to thank too friends, Changman Moon, Joobe Song, Youngjae Song for companionship during my hard times. Thanks also to Mungil Hong, Sangho Song, Jooil Kim, Sukju Kim, Yooseong Kim, Jungseog Choi, and Sooleng Lau for their encouragement.

Last, but not least, I wish to thank the professors in the Department of Electronics, Ajou University, for their encouragement.

Declaration

I hereby declare that I composed this thesis entirely myself and that it describes my own research.

Taehee Kim
Edinburgh
October 24, 1996

Contents

List of Figures

Chapter 1

Introduction

Current assembly robotics suffers from problems of sensing and action management, especially when applied to robots with a highly abstract working context where commands are issued in terms of abstract specification of tasks, *i.e.* task level programming. This is a system architectural problem of sensing and action management and their abstraction.

In the field of Artificial Intelligence (AI), what is the right approach to the implementation of intelligence is often discussed. In the behaviour–based approach, one of the methodologies, it is argued that an adaptive intelligent creature ought to be built from the bottom up, through the accumulation of purposeful competences (or behaviours[1]) with local sensing and action management [Brooks 86].

In this thesis, based on the behaviour–based approach, it is stressed that the use of appropriate sensors is important for an assembly robot to increase its functionality in an economic manner. Economical event signature sensors were developed and demonstrated by applying them to assembly problems. Among the applications, sensor fusion[2] problems are addressed. Under the guidelines proposed by the behaviour–based approach[3], an approach to sensor fusion is proposed then tested. In the sensor fusion application, event sensors coupled with appropriate motions to form dynamic sensors made a basis for the system hierarchy.

[1] Matarić [Matarić 94] defines a behaviour as *a control law that achieves and/or maintains some goal.*

[2] Sensor fusion is a way of coordinating information from more than one sensing to provide consistent information about a single subject.

[3] The guidelines are described in detail in Section 2.1.3.

This thesis describes the work on the development of the event signature sensors and their application to sensor fusion for behaviour–based robotic assembly. In this chapter, background and motivation is explained and the outcome summarised.

1.1 Assembly Robots and the Sensor Fusion Problem

With the aim of replacing hard automation, much effort has been devoted towards assembly robots with highly flexible competences. Nevertheless, many robots currently used in industry are limited in their behavioural repertoire. Often the only capability of these robots is to perform a fixed sequence of motions between predefined points in space, which does not meet the initial early expectation for the benefits to be achieved by applying robots to industrial assembly. Although there are systems in research laboratories which accomplish a good level of flexible competence, those systems are often not economic enough to be applied to a real situation. To solve this problem, much effort has been devoted to the development of robot system architectures. Since a sophisticated robotic system would require a high level of information processing capability, methodologies developed in Artificial Intelligence (AI) have often been applied.

For two decades of Artificial Intelligence (AI) research, one approach has been most dominant, the *symbolic approach*, where symbolic representation and manipulation play the central role in the system. This derives from the view that intelligent behaviour is essentially knowledgeable behaviour. These systems tend to be centralised around information processing, with some sort of reasoning engine working with the stored centralised world knowledge. This approach has caused many assembly robot systems to employ centralised symbolic reasoning systems, particularly for those sophisticated systems handling many sensors.

For assembly robots, in order to increase versatility and to cope with uncertainty, employing sensors has been regarded as a sensible choice. One may employ sensors to make the robot more versatile and flexible to external changes in the environment. One field of study in robotics is referred to as *Sensor Fusion*, where the problems of employing different kinds of sensors is investigated. Traditionally, a central representation composed of models of 1) the world; 2) the manipulator; 3) the sensing modality, has often been proposed (*e.g.* [Lozano-Pérez *et al.* 92]). Information from the sensors is referenced to this world knowledge, built in at an abstract symbolic level which is usually positioned top of the system. In this kind of model, this symbolic level is the level at which information from different sensors is combined. At the termination of reasoning, detailed motor commands are sent to the actuators.

However, under this structure of the centralised representation, it is difficult to avoid an ineffective information flow, known as the sense–think–act cycle. For instance, when information comes from a sensor, the central decision system has to reason to issue a motor command. If there is information from two or more sensors, the central system combines the information then make a decision (see Fig 1.1). This sense–think–act cycle has inefficient serial and centralised information processing, whereas in the case of a moving robot, fast reactions may be necessary. As more sensors are employed, the system becomes big and complex, hence inefficient, or even intractable. Failure of part of the system may bring the whole system to fail since the information processing is centralised. Large amounts of knowledge cause a time consuming search problem and difficult maintenance. Modification or upgrading would demand much work throughout the whole system. Lozano-Pérez argues at the end of his ambitious HANDEY project which is supposed to employ all possible modern technology towards a task level assembly robot:

> Task-level robot programming in well-modeled workspaces is eminently practical, especially in view of the dizzying rate of growth in affordable computational power.
>
> ...
>
> Task-level robot programming in workspaces with substantial uncertainty still requires fundamental new research in planning with uncertainty and planning for the use of sensors[4].
>
> ([Lozano-Pérez *et al.* 92], p. 215)

[4] HANDEY uses only range sensors to build the model (depth map) of the workspace including parts, before the system starts planning. The model requires the accuracy of about 1 mm over an area about 1 m in radius (author's footnote).

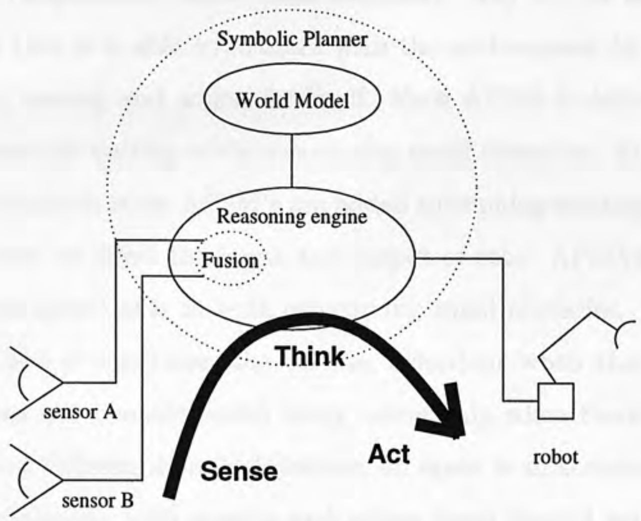


Figure 1.1: Traditional sense–think–act cycle of centralised symbolic planner for sensor mounted assembly robots

Dissatisfaction with the inefficiency and inadequacy of the centralised symbolic approach, and the problems it presents for combining multiple sensing, led Brooks to propose what he calls the **Subsumption Architecture** [Brooks 86]. Brooks argues:

> True intelligence requires a vast repertoire of background capabilities, experience and knowledge (however these terms may be defined). Such a system can not be designed and built as a single amorphous lump. It must have components. ([Brooks 86], p. 5, original parenthesis)

In contrast to the traditional approach, in the Subsumption Architecture, building a centralised symbolic world model is avoided and the information from the sensors is processed by distributed goal–seeking components (*Augmented Finite State Machines*, or AFSM), which in effect distributes the information processing job throughout the system.

1.2 Emergence of the Subsumption Architecture

Rather than attempting a highly abstract intelligent system from the beginning, in the Subsumption Architecture, building a sophisticated system starts from implementing a simple AFSM (Augmented Finite State Machine). Any AFSM is a complete system in the sense that it is able to interact with the environment by itself, with each AFSM managing sensing and action for itself. Each AFSM is defined in terms of a behaviour, for example walking while overcoming small obstacles. As a more sophisticated system is required, more AFSM's are added subsuming existing AFSM's, where AFSM's are allowed to affect the input and output of other AFSM's. Suppose there is an autonomous agent[5] able to walk overcoming small obstacles. A light sensitive AFSM is added and it suppresses the walking behaviour when there is light. Then the agent becomes more sophisticated being active only when there is no light, (see Figure 1.2). In the Subsumption Architecture, an agent is an accumulation of various behavioural competences, with sensing and action loops limited within AFSM's. In other words, in contrast to the conventional centralised symbolic approach, this has a distributed structure of behavioural competences. Modifying the agent is easier, by adding and deleting of any behavioural competence, only requiring local behavioural modification[6]. Failure of one behavioural competence would not result in the whole system failing, but simply the loss of that competence.

From the outset, AFSM's are self contained and defined in terms of their behavioural interaction with the environment. By adding more AFSM's, the creature is developed. This process resembles animal evolution. The criticism is made by Brooks that the human level of symbolic manipulation is not a modular function, but has deep roots in subconscious level cognition and behaviours. Hence, in order to build human level intelligent systems, implementing merely symbolic manipulations mimicking only the *phenomena* of such a sophisticated system is not adequate. Instead, a more promising way, because it is closer to what nature has done, is to build an intelligent system by accumulating behavioural competences from the bottom up. As it becomes sufficiently sophisticated, it may eventually appear to be intelligent although no explicit intelligence is implemented (*Intelligence is emergent, not implemented* [Malcolm *et al.* 89].; *Intelligence is in the eye of the observer* [Brooks 91].).

---

[5] Steels [Steels 95] defines an agent as:— 1) An agent is a system (*i.e.*, a set of elements which have a particular relation among themselves and with the environment). 2) An agent performs a particular function for another agent or system. 3) An agent is a system that is capable of maintaining itself.

[6] However, this is true as long as the AFSM's interact in tightly controlled ways.

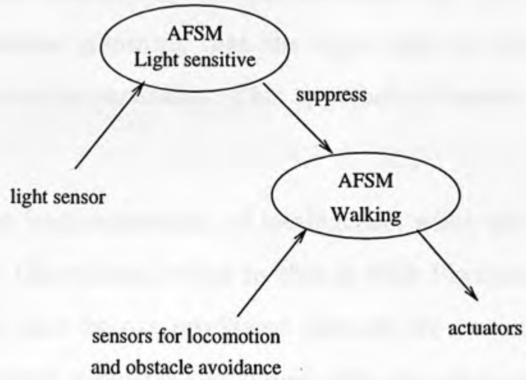


Figure 1.2: An example of an autonomous vehicle using the Subsumption Architecture

(Note that if the characteristic of the connecting node, "suppress" had been changed to "excite", then the vehicle would exhibit a radically different, opposite behaviour. Although they are simple, connections between the modules (AFSM's) have a powerful effect on the behaviours, in other words, it is an economic means to control lower level competences, which implies that it is economic to program, where the program is done in terms of the meaningful modules.)

Not only Brooks, but also others, such as Varela [Varela *et al.* 92] and Malcolm [Malcolm 87, Malcolm *et al.* 89, Malcolm & Smithers 90] proposed similar principles. In the AI community, this kind of approach has been considered a paradigm, often referred to as the **behaviour–based approach**[7].

1.3 AI and Behaviour–Based Assembly Systems

Whereas the Subsumption Architecture proposed a rather practical methodology, the term behaviour–based approach is used in a broader sense to explain the general principles and methodologies as a paradigm of AI. In this section, the position of the behaviour–based approach in the field of AI is looked at and based on this, an approach to behaviour–based assembly systems is introduced.

In AI, it has long been debated what the right way of implementation of intelligence must be. Newell and Simon have proposed the *Physical Symbol System Hypothesis*, which asserts that intelligence can be implemented by the right kind of behaviour pro-

[7] However, Varela *et al.* call it Enactivism.

duced by the right kind of machinery [Newell & Simon 76]. This hypothesis supported knowledge–based systems, assuming that the right kind of machinery is a computer with symbolic manipulation capability. This approach is known as Cognitivism (symbolic approach).

In Cognitivism, in the implementation of intelligence, what we think our intelligence is plays a prime role. Churchland refers to this as Folk Psychology [Churchland 86]. We empirically think that we are intelligent because we can communicate with one another and reason about sophisticated things such as a chess game, which is mainly based on language and utilising symbols. Computers, with explicit knowledge encoded by the designer, have been adopted as the right machinery to implement intelligence, since from a cognitivist's point of view, they have a computation ability similar to that of a human seen as a rational creature. Cognitivism is based on the idea that the *internal* machinery is like the *external* rationalised linguistic descriptions we use in communicating with one another. To adopt computers as machinery for knowledge based systems, Newell proposed:

> **The Knowledge Level Hypothesis.** *There exists a distinct computer systems level, lying immediately above the symbol level, which is characterized by knowledge as the medium and the principle of rationality as the law of behavior*[8].
> ([Newell 82] p. 99, original emphasis)

However, other approaches in AI, *viz.* Connectionism and the behaviour–based approach, suggest that the machinery must be something more intrinsic to the physical and biological features and less purely logical. Connectionism takes the position that intelligence needs to be implemented at the neuronal level [McClelland *et al.* 86], while in the behaviour–based approach, purposeful fragments of behaviours are the ingredients of the logical hardware, as in the Subsumption Architecture.

One important question in the behaviour–based approach is whether a system built under Brooks' Subsumption Architecture can be instructed to do lots of different com-

[8] *i.e.* the symbol level here is the level of computer program code; the knowledge level is the level above in which the knowledge is encoded using programs (author's footnote).

plex detailed tasks which require following a symbolic plan, which may require the interpretation of explicit symbolic knowledge. It is a principle of the Subsumption Architecture to avoid symbolic knowledge, so that if a system appears to be using symbolic knowledge, this will be an appearance implemented by some other mechanism.

As far as practical assembly robots are concerned, which may be used in such factories as we currently have, parts are described in geometric terms often supported by CAD, and an ordered assembly plan is required. Highly flexible task level assembly robots require a symbolic interface, understanding task specification and other symbolic information, in other words, explicit symbolic knowledge. Would it be possible under the Subsumption Architecture to develop a simple agent to the level of an assembly robot with massive sensing capability which plans, understands human plans and geometric information? At present, it is too early to say if it is possible or not.

At Edinburgh, Malcolm suggested a way to build a task level assembly robot, using a behaviour–based approach to implement an executive agent composed of purposeful on-line Behavioural Modules (BM's) which execute details of the plan made by an off-line classical symbolic planner, a hybrid system as illustrated in the Figure 1.3 [Malcolm 87, Malcolm 95]. Since assembly tasks are intimately associated with human symbolic functions, those ingredients could be better described in symbolic terms in an effective and economic way. The symbolic planner is employed as the symbolic interface, simulating human symbolic functions. Detailed sensing and action problems are managed by the low level behaviour–based executive agent composed of Behavioural Modules (BM's) dealing with the real world uncertainties and variations of the environment. As the details of sensing and action are amalgamated in the executive agent, the size of the planner is kept as small as possible as it is mainly responsible for planning of the general order of the assembly. By virtue of the smart executive agent, the planner need not consider too many details. This guarantees a comparatively small amount of world model and plan, which makes the whole system more manageable and economic [Malcolm & Smithers 88].

Observing part assemblies, one may find that sub-tasks can be better described in terms of the generic assembly purpose, *i.e.* putting down a part, fitting faces, peg-in-hole, and so on. Note that animal behaviours too are most conveniently described

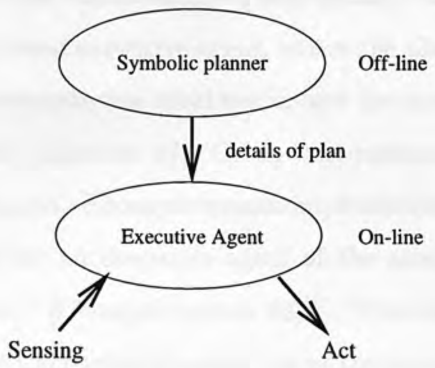


Figure 1.3: Edinburgh's behaviour–based assembly robot: a hybrid system of symbolic planner and an executive agent

(The executive agent is composed of a number of BM's of assembly competences, which have access to sensing and action individually.)

in terms of their purposes. Behaviours have their own management of sensing and action to accomplish their purposes. In such biologically inspired behaviour–based assembly systems, BM's are defined in terms of their purposes in assembly[9], performing sensing and action for themselves. General instructions from the planner only need to parameterise (such as with the sizes of parts) and activate the appropriate BM's, without considering the details of sensing and action (except in so far as sensing is necessary to decide between alternative branches of a plan). Behaviour–based assembly robot programming is well suited for task level assembly programming, since tasks are modularised in terms of the purposes of BM's. Without the high level symbolic planner, a human programmer can play the role of a planner, programming the robot in terms of meaningful BM's in assembly. This programming feature is similar to that of object or task level assembly robot languages such as RAPT [Popplestone *et al.* 78] and AUTOPASS [Lieberman & Wesley 77].

However, the behaviour–based assembly approach is still in its infancy. The main goals of the approach are to know how best to divide the work between the planner and agent, and how to construct the low level behaviour–based execution agent. So far, four major works have been performed at Edinburgh. The sensor-less SOMASS

[9] *i.e.* putting–down–part BM, fitting–face–to–face BM, peg–in–hole BM, and so on.

assembly system showed the effectiveness of segregating task level assembly into a planner and a behaviour–based executive agent, where the planner plans about SOMA cube assemblies in an uncertainty-free ideal world, and the executive agent performs an assembly in the real world [Malcolm 87]. Given this, methodologies to accommodate sensors were next investigated. Chongstitvatana implemented an uncalibrated mobile vision system to be used for an executive agent of the assembly plan with minimal modification of the planner [Chongstitvatana 92][10]. This showed that such a sensor can be incorporated not at the level of planner, but at the lower level of executive agent. Independently of the vision approach, Wilson implemented simple binary touch sensing to detect any failure of BM's attempting a SOMA cube assembly [Wilson 92]. Using this touch sensing, Wilson addressed an architectural problem of BM's, incorporating error handling routines in the BM's. Balch applied force sensing to an electric contact assembly to evaluate the behaviour–based assembly approach, not only demonstrating successful assembly, but also providing a plausible methodology towards task level programming with sensing [Balch 92]. However, each used only one sensor and the use of multiple sensors had still to be investigated.

1.4 Motivation, Approach, and Results

As already mentioned, where there are vision and touch sensing to integrate, conventional sensor fusion systems often choose to do so at the level of detailed geometric models. In order to co-ordinate vision and reach, they may do so with a 3D Cartesian model of the world. However, it has been proposed that those kind of competences can be managed more simply at a lower level of direct mapping without the need of a centralised world model, as animals might do, such as a crab [Churchland 86] for instance. Those animals are very simple in their neuronal structure compared to humans, but their behaviours associated with vision are astonishingly versatile, fast, and robust, seemingly without any symbolic function.

The behaviour–based approach has been proposed to overcome the problems of centralised knowledge–based systems. In contrast to centralised sensor fusion, in the behaviour–based approach, different BM's with possibly different sensors work together

[10] "Uncalibrated" here means that no model of the camera, *i.e.*, its position, *etc.*, was required.

at the lowest possible level, suggesting the way the actuators should operate. For mobile robots, *Action fusion*, or *Behaviour fusion* [Flynn & Brooks 89] is often stressed as a plausible way of fusing different sources of information, where actions proposed by different sensors are directly reflected to the actuators or, if there is conflict, then vector summing, switching between actions, or interlacing may resolve the problem. As a result, the whole system appears to reflect the sum of the sensing modalities. However, although this works for mobile robots, application to assembly robots requires considerations of the difference in characteristics of tasks and in the structure of actuators.

The behaviour–based assembly approach encourages the minimisation of the complexity of robot programming, or planning by a computer planner, when multiple sensors are introduced, while achieving a higher level of versatility by virtue of the sensors. This is achieved by building a robot in a decentralised manner which decomposes the robot structure in terms of modularised purposeful BM's. In distributing the behavioural competences, having lower level BM's deal with situations requiring *reactive* behaviours means that high level BM's can afford to deal with more abstract business. In order to achieve this, BM's of lower level require *appropriate* and *economic* use of sensors, in the first place.

In conventional assembly robotic systems, where central world models play an important role, sensors are typically used for measurement. A measured physical quantity is utilised for comparison with and/or updating of the world model. Hence, researchers have preferred to develop accurate sensors for the measurement of physical quantities. This is a natural consequence of the world model being expressed as a conventional scientific model, *e.g.* solid–geometric. However, under the biologically inspired behaviour–based approach, the sensing modality of robotic assembly systems can be viewed differently. Biological sensors tend to be sensitive to a change of physical quantity: change of smell; change of noise; change of scene of the visual space; change of force, *etc.* In addition, animals tend to introduce motions to obtain more information. For instance, a blind man uses a stick to parallel a normal human in many situations, by constructing a 3D spatial image using tactile sensing and motions[11]. In

[11] Similar robotic application is demonstrated in [Allen 92].

robotic *active vision* [Aloimonos & Bandyopadhyay 87], use of motions simplifies the task of extracting useful information from image sequences, such as using *optical flow* [Marr 82]. Sensors that detect changes in physical quantities, and the introduction of motions to sensing, may improve assembly systems in the sense that less computation is required. This is an appropriate method of sensor use for behaviour–based systems, since they consider active interaction with the world to be of importance.

In an assembly task, which involves moving and fitting parts, one of the most important kinds of information is event signatures — information about something happening or changing. Merely by knowing that an object is making a contact with the gripped part or directly with any finger, the assembly robot can perform many tasks under uncertainties, such as fitting parts face to face, exploring part location. The context of contact and motion supplies a lot of extra information, *e.g.*, in guarded motion [Will & Grossman 75, Beni *et al.* 83], a binary touch gives a 3D position [Allen 92]. Sophisticated sensors designed for precise measurement to be used by conventional assembly robots usually require intensive computation and are often expensive. Instead of quantitative sensing, qualitative sensing incorporated with robot motions can provide appropriate information economically, especially for the behaviour–based systems which put emphasis on active interaction with the world.

In this work, two kinds of cheap event signature sensors for part contact were developed using piezoelectric PVDF (polyvinylidene fluoride) films. PVDF material is known to exhibit good sensitivity to many kinds of physical change of quantity, such as mechanical vibration and change of heat. However, it is not considered as a good choice for a measurement task (*i.e.* the classical approach) unless much effort is exerted to provide constrained conditions, since they have considerable inherent capacitance, which results in differentiation of the signal, and induced noise. But, it can be turned into a suitable sensing device for the more biological behaviour–based approach, where qualitative sensing is much favoured. Because it has good sensitivity to change, sensitive event signature sensors can be built out of it. As event signature sensors, these sensors can be substituted for vision systems, conventional force/torque sensors, and tactile sensors under appropriate circumstances where part contact is involved.

The event signature sensors developed are called **Clunk Sensors** and **Force Sensors**.

The Clunk Sensors respond to the audible sound generated by part contact in an assembly task and propagated through the working table or the rigid robot gripper, such as when the gripped part touches other objects. The sensor used for the table is called **Table Sensor** and the sensor used for the gripper is called **Noise–cancelling Sensor**[12]. The force sensors respond to the change of force applied to the sensor, rather than measuring the absolute level of applied force. The output of the force sensor is proportional to the rate of change of deformation of the sensor body. There are two variations of the force sensor: **Bump Sensor** and **Push Sensor**. The Bump Sensor is used as a touching probe mounted on an outer surface of a finger. The Push Sensor makes direct contact with a gripped object, and is used to stop the robot (guarded motion) when the gripped part touches another object by detecting the change in force applied to the surface of the fingers. Both Clunk Sensors and force sensors can also be used to explore the location and orientation of objects. These sensors were used in an assembly task performed by an ADEPT1 robot to demonstrate their suitability for assembly tasks. Since the Push Sensor demonstrated better performance than the Clunk Sensors in event recognition in general, it was further improved and tested. The improved Push Sensor can provide not only binary signals but also the force pattern applied over time, and partially discriminate the direction of the force applied.

The development of the event sensors were motivated by the need of economic sensors for dynamic purposes. In addition to the sensor development, this thesis is also interested in investigating the methods of using the event sensor for dynamic purposes for a behaviour–based system, particularly for guarded motion. Although guarded motion is not new in assembly, it is seen from a different perspective when included in the behaviour–based approach, and it is suggested that it should form the critical basic behaviour for the system, on which more complicated behaviours are built.

Since in the classical paradigm, sensor fusion is achieved via the central word model, and in behaviour–based systems we avoid such models, sensor fusion is a *useful test case* in the exploration and development of the behaviour–based paradigm. Especially in a distributed system, the problem becomes hard when more than two sensing modalities propose different actions, or they require to be combined. Use of complementary

[12] This sensor uses noise cancelling techniques.

sensors is proposed for the purposes of this test. Since the event signature sensors are restricted to be used locally, where parts are about to make contact, they provide a poor global view of the dimensions of parts and the environment. Hence this kind of sensor can be best complemented by image sensors where global features of the working space can be monitored. Touch and vision are very different kinds of sensing, providing very different kinds of information, so difficult practical problems can be introduced. A vision sensor is employed as the complementary sensor to the enhanced Push Sensor to build a sensor fusion hierarchy using the behaviour–based approach in order to test and criticise this methodology for assembly systems.

It is appropriate to ask what might be the design methodology to build low level BM's with multiple sensors? It seems advisable that sensor fusion should take place at as low a level as possible, in order to allow higher control levels to deal with more abstract global problems. In this thesis we propose the employment of reactive behaviours in order to form a basic set of behaviours. As the basic behaviours, guarded motions with event signature sensing capability are adopted. In building such a sensor fusion system, much attention has been devoted to the active use of action to improve qualitative sensitivity. BM's running in parallel are implemented in order to demonstrate distributed and concurrent sensor fusion. As a result, it is argued that for a reactive robot arm for a behaviour–based assembly system, a servo control mechanism is desirable, where vector information provided by sensors can continuously affect robot motion. It is possible but difficult to achieve this using a position controlled robot, but a force or velocity controlled robot is more appropriate. Some of the principles behind the behaviour–based approach are criticised by assessing an actually built system embodying the principles. The finger–mounted Push Sensors survived all this work without damage or significant wear.

1.5 The Organisation of this Thesis

Chapter 1 is this chapter of introduction.

Chapter 2 reviews related issues.

Chapter 3 reviews tactile sensing for dynamic purposes.

Chapter 4 describes the initial experiments carried out to look at the feasibility and to obtain knowledge of applying PVDF films to sensors for robots.

Chapter 5 describes implementation and an application to an experimental benchmark assembly task of the event signature sensors based on the knowledge obtained from the experiments described in Chapter 4. The performance of the sensors are assessed in the assembly.

Chapter 6 describes the further development of the Push Sensor. The physical construction of the sensor is considered more in depth and the experiments on the sensor performance is described. Various application areas are shown.

Chapter 7 first proposes a distributed and concurrent architecture for a behaviour–based sensor fusion. Then, it describes the implementations, and discusses the performance and problems.

Chapter 8 summarises this thesis, itemises the contributions made, and examines the implication of this thesis.

Chapter 2

Overview of Related Issues

This chapter looks at the issues related to the work presented in this thesis. The position of the behaviour–based approach within the field of AI is first examined. Behaviour–based assembly systems are then described. It is pointed out that further research is necessary for the behaviour–based assembly approach, such as in the choice of architecture for sensing and action coupling. Other work on the sensor fusion area is then reviewed. It is pointed out that the problem of sensor fusion or multisensor integration stems from the method of use of a single sensor. Use of sensors in assembly robots is considered and emphasis is placed on the need of reactive autonomy for such systems. A guarded move using simple but useful touch sensors is proposed as the basis on which more abstract control properties can be built conveniently, particularly for reactive behaviour–based assembly systems.

2.1 Three Paradigms in AI

> ...*complex behaviour may simply be the reflection of a complex environment.* [Simon 81]

AI has been a field of study for over 30 years with two main aims, first producing intelligent machinery in the form of powerful tools using computers (weak AI), such as by simulating biological intelligence. Second, whether and how machines can be given *real* mental functions (strong AI) is controversial[1].

[1] See [Searle 80], for the categorisation of AI into 'weak AI' and 'strong AI'.

In line with the effort to achieve AI, three major suggestions have been made in the course of its history:

- **Cognitivism**[2];
- **Connectionism**[3];
- **Behaviour-based approach.**

2.1.1 Cognitivism

In implementing AI, it may be supposed that intelligence is in the creature, not in the environment, because the creature shows the intelligence. Thus the world is divided into two parts: the creature and the world. The creature is further divided into two parts: body and mind. Mind is believed to be the place where intelligence resides, because intelligence is abstract and seems likely to be a phenomenon of mind, not body. Further, the mind is divided into two aspects: phenomenological mind and computational mind, where the computational mind performs computations, *e.g.* syntactic transformations of propositionally expressed knowledge, while the phenomenological mind has qualitative subjective experiences, such as consciousness (see Figure 2.1 [Malcolm 93]). The computational mind can be implemented by symbolic manipulation alone (*e.g.* symbolic AI). In cognitivism, it is asserted that intelligent behaviour is the result of the operation of the computational mind, and the computational mind is the main place where intelligence is implemented, as claimed in Newell and Simon's *Physical Symbol System Hypothesis* [Newell & Simon 76] and supported by Brian Smith's *Knowledge Representation Hypothesis* [Smith 82]. Cognitivism is based on this distillation of intelligence, and intelligence is thought of as the result of cognitive processes, which can be implemented by manipulating symbols.

[2] A position in psychology and philosophy that intelligent behaviour (only) can be explained by appeal to internal "cognitive processes," that is, rational thought in a very broad sense [Haugeland 78]. Thus, in AI, the central technical problems become a) how to represent knowledge, b) how to reason about it.

[3] An approach to cognitive modelling which focuses on causal processes by which units excite and inhibit each other and does not provide either for stored symbols or rules that govern their manipulations [Bechtel & Abrahamsen 91].

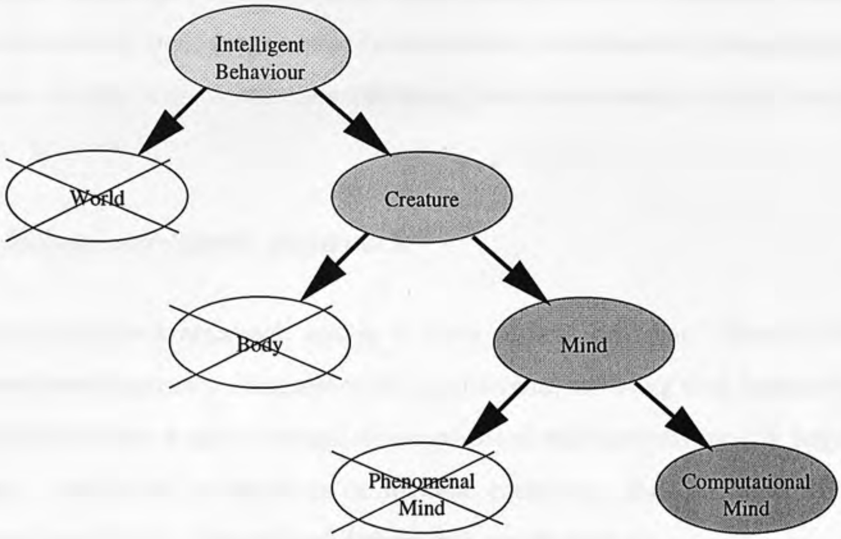


Figure 2.1: The Distillation of Intelligence

2.1.2 Connectionism

An alternative approach to simulating cognitive phenomena has been suggested, known as Connectionism. Since computation is performed on symbols in Cognitivism, loss of any symbols may cause a serious problem to the system. In Connectionism, the representation is distributed throughout the topological connections between the neurons. Damage to individual neurons slightly degrades the resolution of the system, a more graceful degradation than is caused by loss of symbols. This leads connectionist systems to be robust. They also have adaptive learning potential. Connectionists have shown that intelligent behaviour emerges from the network without any explicit centralised symbolic representation having been implemented [McClelland *et al.* 86, Bechtel & Abrahamsen 91]. The topological structure of an artificial neural network plays an important role in the computation while allowing the representation to be decentralised. This provides an alternative model of the body/mind relationship (compared to the program/computer model of symbolic AI).

Connectionism agrees with symbolic AI that representation is the crucial issue in implementing intelligent behaviour, but disagrees about the way in which the knowledge should be represented [Varela *et al.* 92]. Where symbolic AI represents knowledge as propositions which are to be reasoned with, Connectionism represents knowledge as clusters of neuronal weightings whose fundamental operations are association and generalisation. In this way, it criticises the body/mind relationship implied by symbolic AI.

2.1.3 Behaviour-based Approach

The behaviour–based approach makes a more radical criticism [Brooks 91]. The behaviour–based approach disagrees with Cognitivism, asserting that human high level mental functions are a phenomenon of complicated sub-systems mostly beyond self–awareness. Our belief in ourselves as rational creatures plays an important role in justifying Cognitivism. Churchland defines folk psychology as:

> Folk psychology is commonsense psychology — the psychological lore in virtue of which we explain behavior as the outcome of beliefs, desires, perceptions, expectations, goals, sensations, and so forth. It is a theory whose generalizations connect mental states to other mental states, to perceptions, and to actions. ... Folk psychology is "intuitive psychology," and it shapes our conceptions of ourselves.
> ([Churchland 86], p. 299, original emphasis)

In Folk psychology, we *deduce* our behaviour from our beliefs and desires, *i.e.*, our behaviour is *caused* by our beliefs and desires.

In the behaviour–based approach, Cognitivism is criticised as the result of Folk Psychology [Smithers 93, Malcolm 93], because in Cognitivism, empirical mental functions have been formulated as cognition in symbolic terms. It is claimed that, in traditional AI, there is something missing between the creature and the environment as well as between body and mind. Moravec emphasises the significance of the underpinning mechanisms for the mental functions of the human being:

> The deliberate process we call reasoning is, I believe, the thinnest veneer of human thought, effective only because it is supported by this much older and much more powerful, though usually unconscious, sensorimotor

> knowledge. We are all prodigious olympians in perceptual and motor areas, so good that we make the difficult look easy. Abstract thought, though, is a new trick, perhaps less than 100 thousand years old. We have not yet mastered it. It is not all that intrinsically difficult; it just seems so when we do it.
> ([Moravec 88], pp. 15–16)

Dreyfus also notes the importance of the underlying structure:

> Indeed, sensory motor skills underlie perception whose basic figure/ground structure seems to underlie all "higher" rational functions ....
> ([Dreyfus 92], p. 255, original emphasis)

Varela *et al.* refer to the behaviour–based approach as Enactivism, where they explain the enactive approach as 1) perception consists of perceptually guided action and 2) cognitive structures emerge from the recurrent sensorimotor patterns[4] that enable action to be perceptually guided [Varela *et al.* 92].

In the behaviour–based approach, a creature is believed to be a collection of goal-directed behaviours. Every behaviour is independent in that it interacts with the *environment* for its own *purpose*. Because each interacts with the environment purposefully, combining behaviours implies summing the purposefulness of every behaviour, without increasing the complexity of any particular part of the system. The accumulation of behaviours allows the system to grow incrementally, incrementally developing into a more sophisticated structure rather than being redesigned. The potential to exhibit intelligence is included implicitly in each behaviour, the relationship between the behaviours, and the purposefulness of every behaviour in the environment. Intelligence is exhibited while the creature interacts with the environment. Brooks emphasises the importance of actual interaction between the creature and the environment, which results in the emergence of intelligence: "Intelligence is determined by the dynamics of the interaction with the world"; "Intelligence is in the eye of the observer[5] [Brooks 91]."

---

[4] which develop cognitive structures (inserted by author),

[5] *i.e.*, although intelligence has not been implemented, it appears while the creature interacts with the environment.

The following features can be seen to be important in the behaviour-based approach and could be also regarded as guidelines in building behaviour–based systems [Brooks 91, Malcolm *et al.* 89, Flynn & Brooks 89, Beer 90].

- **Intelligence Emergent, not Implemented** Intelligence is not implanted explicitly in the robot, but appears while it interacts with the real world.
- **Low-level Amalgamation of Sensing and Action** Sensing is tied into the action at a low level in order to decrease the complexity and increase the flexibility of the high level.
- **Parallelism** The atomic units (behaviours) are all parallel unless otherwise constrained.
- **Distaste for Symbolic Representations** Knowledge about the world is always incomplete and liable to error, therefore the behaviours wherever possible use the real world as their model (*i.e.* by intimate interaction with the environment).
- **Active Use of the World** Instead of the delegation of control via the procedural hierarchy and parameter passing typical of modern programming practice, behaviours are preferentially activated and controlled by sensed environmental triggers.
- **Minimalism** Through the previous principles (parallelism, distaste for symbolic representation, and active use of the world), comparatively little computational power is required to achieve the desired level of performance.
- **Behavioural not Functional Modularity** The task is segmented in terms of purposeful task-achieving behaviours.

2.2 The Practical Viewpoint of the Behaviour–Based Approach

2.2.1 Active Involvement with the Environment

Brooks' assertion that *Intelligence is in the observers' eyes* implies that the detailed description of intelligence is not necessarily an ingredient in the implementation. This possibly results in an economic implementation, by exploiting the nature of the environment.

This situation is exemplified by one of the vehicles described in [Braitenberg 84]. The mobile robot drawn in Figure 2.2 has two actuators and two photo sensors. They work such that the revolutions of the actuators are proportional to the amount of light that the contralateral photo sensors absorb. Thus a light offset to one side will steer the robot towards it, and only even illumination will steer the robot straight ahead. (If the sensors and the actuators are wired in the opposite way, then the robot will show the behaviour of being repulsed by the light.) As the light is turned on, the robot will approach the light. If the light is moving, the robot will follow. There is no explicit representation of behaviour, environment, or goal. But an observer may argue that the robot has the purpose or will to follow the light. The apparently unified behaviour of the robot is not even produced by a central controller, but by the interaction in the environment of two completely independent controllers for each individual wheel. Exploiting the environment and the physical properties of the world that are related to the task resulted in a simplified implementation since some of the information processing (or computation) is performed implicitly, *i.e.*, non-computationally, by the inherent physics of the entire creature/world system.

A well managed relationship between the environment and the agent (*e.g., structural coupling* [Varela *et al.* 92]) reduces the amount of explicit information processing to be performed by the agent. A well organised structure of the subsystems of the agent also possibly diminishes the amount of computation to be done by a high level information processing part of the agent. In the behaviour–based approach, the implications of the relationship between the environment and the agent, and the importance of the physical and the logical structure of the agent's goal seeking subsystems are emphasised.

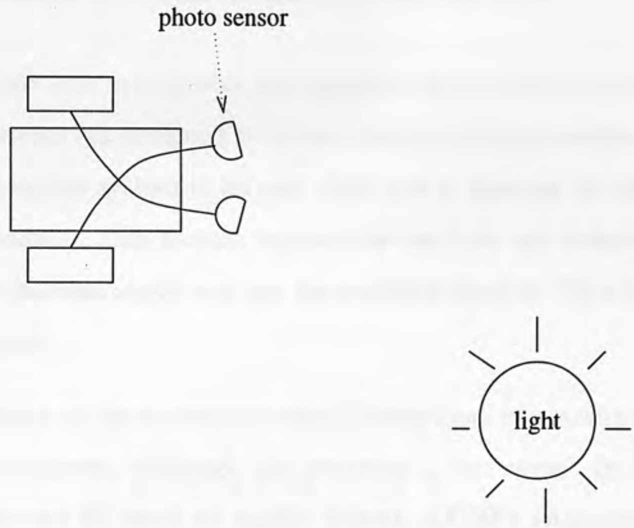


Figure 2.2: One of Braitenberg's vehicles

Both for biological systems and robots, sensors play an important role as channels through which information about the environment is obtained. According to the characteristics of the task performed, appropriate selection of sensors will contribute to the effectiveness of the system. The purpose of **dynamic sensing** [Beni *et al.* 83] is to enhance sensing by exploiting robot motions. Robot motions can reduce the complexity of sensory information processing, at the cost of the extra motions.

2.2.2 Low level amalgamation of sensing and action

The principle of the low level amalgamation of sensing and action often encourages a system to be built without any central control. Investigation and robotic implementation of the cricket's phonotaxis provides an example of how a high-level-look function can be implemented in a relatively simple way without any central control and symbolic representation [Webb 93], hence economic – minimalism. The cricket robot navigates towards the source of a particular sound, while overcoming obstacles. The phototaxis of the Braitenberg's vehicle, explained above, can be also in this category. Structural coupling, *i.e.*, well–managed interaction, is also evident in the cricket auditory system.

2.2.3 Approaches to Behavioural Modularisation

Decomposing a task into manageable and tractable subsystems provides practical benefits. The subsystems are designed to be as independent as possible. A subordinated behaviour is a complete system in its own right and is ignorant of the existence of the subordinating module. This feature implies the modules are independent, so that a system can grow incrementally and can be modified flexibly. This helps a system to be more manageable.

However, approaches to the modularisation of behaviours in practice can vary depending on the circumstances, although the principle is the same. In the Subsumption Architecture [Brooks 86] used on mobile robots, AFSM's (Augmented Finite State Machines) represent behaviours. They can be accumulated in a vertically hierarchical manner. For instance, a wandering behaviour (AFSM) can be subsumed by a light sensitive behaviour (AFSM) by means of the connection, inhibit: the activation of the light sensitive behaviour deactivates the wandering behaviour when there is light. AFSM's can also run in a physically independent manner.

For assembly robots, a batch task is composed of subtasks in a chronological order. That is to say, an assembly is composed of subtasks, such as locating parts, moving parts, and mating parts. Malcolm suggested an approach to modularise behaviours in terms of the purposes of these subtasks [Malcolm 87]. They come one after another as planned by a classical symbolic planner. These modularised subtasks are called Behavioural Modules (BM's). One BM can have other BM's as behavioural ingredients, although how best to do this is still under investigation.

In assembly robots, *task level* programming refers to a high level of description of the task in natural human terms such as "mate part A and part B", "cover part A with a cap in box B", or even "assemble a telephone set". A demanded task involves a human purpose. A behaviour is defined by its purpose. A BM is defined by its behaviour. A subordinating module inherits the behaviours of the subordinated modules. The system grows while increasing its behavioural sophistication, at the same time increasing adaptability to higher level task specification. By its nature, this kind of system abstraction is considered an appropriate approach to task level programming.

In a general sense, these kinds of modules are BM's if defined in terms of behavioural competences based on their purpose. In this sense, Brooks' AFSM's can be referred to as special cases of BM's, both being defined in terms of behavioural competences. While the detailed structure of BM's are left as a general problem, AFSM's are composed of finite state machines with timers built in [Brooks 89].

2.2.4 Micro Abstraction versus Macro Abstraction

In terms of the context of abstraction in practice, artificial neural networks which is practical implementations of Connectionism, involve *micro abstraction*, while the behaviour–based approach involves *macro abstraction.*

In artificial neural networks, functions of real biological neurons and their topological structures are modeled. Individual artificial neurons have very simple information processing capability, such as summing the inputs and thresholding the sum. What makes the real neuronal network powerful is the plastic topological structure of many neurons. However, as a system becomes more sophisticated requiring the coordination of numbers of behaviours, it would demand a large number of artificial neurons, perhaps millions, to be organised. This would give rise to a design and management problems, *i.e.*, 'How to organise them?', 'How to train them?' That this neuronal level of micro abstraction might be inappropriate for investigating the problems of the organisation of behaviours, is a frequent criticism.

On the other hand, in the behaviour–based approach, behaviours are modularised (abstracted) and the problems of relationships between the behaviours are investigated, as well as the implementational problems of individual behaviours. For a reasonably sophisticated robot which exhibits numbers of behaviours adapting to the changing environment flexibly, what each neuron does is not very important, but the organisation of the behaviours is important, since the robot shows its autonomy and purposefulness by means of exhibiting appropriate behaviours under the prevailing circumstances. However, artificial neural networks may well be used to compose a behaviour.

2.2.5 Problems of the Behaviour–Based Approach

Although the behaviour–based approach proposes plausible methods to build autonomous agents, it also has some problems that need to be considered and further investigated.

Defining a Behaviour

Behaviours are defined arbitrarily case by case. For instance, a behaviour may be defined as a wandering behaviour, and another may be an avoiding light behaviour. If they are subsumed by a sound reactive behaviour in a certain manner, the agent will show a behaviour wandering while avoiding light, and reactive to sound where noticed. However, the question naturally arises of what are the defining characteristics of a behaviour, *i.e.* what makes something a behaviour (or behavioural module) rather than an element of some other architecture. Unfortunately, there are no definitive rules, although there are guidelines (Chapter 2.1.3, page 21).

However, as long as behaviours are defined in terms of purposes under given circumstances, the agent could be a purposeful system. It relies on the designer's intuition and experience, *i.e.* expertise, with reference to the design specification and the characteristics of the environment.

Coherence

In the behaviour–based approach, behaviours are built to work as independently as possible of other behaviours to achieve a behaviourally distributed constitution. But, this gives rise to the problems of maintaining overall behavioural coherence. For simple insect-like agents, this is not a big problem. But, as the system becomes more complex, especially if using many sensors, coherence will become a major problem. This problem has been recognised and addressed by Brooks [Brooks 94]. Brooks proposes that coherence can be achieved for complex systems in at least two ways: 1) exploiting natural sources of coherence; 2) exploiting certain mechanisms (design coherence):—

1. **Exploiting natural coherence**
   - The world often integrates things, such as an ant's trail.
   - The structure of the task may impose a natural sequencing of actions.
   - Multiple behaviours may actually be additive even if some of them contribute negatively.
2. **Exploiting certain mechanisms**
   - Using internal parameters: feeling hunger will activate food foraging behaviour, but fear will inhibit this, while exciting other behaviours.
   - Using the environment for communication: the effect on the environment of one behaviour may trigger or inhibit other behaviours.
   - Mutual exclusion: a form of lateral inhibition[6].

Action selection has been proposed for mobile robots, as a method to organise sensing and action modalities in complex dynamic environments, where there are multiple sources of incoming information [Maes 90]. By virtue of the purposefulness of behaviours, action selection where the agent responds to different sensory states, is regarded as *attention*, because the system looks as if it pays attention to certain aspects [Brooks 94]. The method, "using internal parameters", as stated above, is equivalent to action selection.

The problem of coherence is a fundamental problem. It is related to the problem of defining behaviours, how to structure sets of behaviours, and how to exploit direct[7] and indirect[8] communication between them.

[6] Lateral inhibition: a set of conditions established when two or more neural cells are interconnected so that excitation of one produces inhibition in the other [Reber 85].

[7] *i.e.* explicit physical communications between modules, although it is better kept to its minimal form.

[8] *i.e.* making use of the environment.

2.3 Applying Behaviour–Based Approach to Assembly Robot Problems

As briefly introduced in the previous section, a behaviour–based system is a plausible robot structure which allows the system to accommodate multiple sensors. In the behaviour–based approach [Malcolm *et al.* 89, Flynn & Brooks 89, Brooks 89], the whole robot task is decomposed into useful task achieving behaviours, where sensing and robot action are implemented at low levels, thus the higher level of behaviours need not consider the details of the lower level sensing and action. This approach also allows the incremental growth of the system, by appending behaviours to the current configuration in order to increase the versatility of the robot. Also, modification of the system is accomplished by replacing appropriate behaviours, without modifying the whole system. This approach leads to a robust and flexible hierarchically structured robot system.

For an assembly robot, this approach has been specially adopted to reduce the complexity of the planner. In the SOMASS system, the low level executive agent, a set of goal seeking mechanisms (BM's), takes charge of resolving uncertainties, which allows the symbolic planner to reason about an ideal world ignoring the uncertainties [Malcolm & Smithers 88].

Let us take an example. Suppose there is a factory worker and a manager. The factory worker is supposed to perform detailed jobs while the manager sets up general plans and directs the factory worker in a desired way. If the factory worker is smart, able to deal with small batch work under some uncertainties, the plan from the manager does not have to describe all the details. The manager can concentrate on the general plan, while the factory workers deals with the details of any practical problems.

For a traditional assembly robot, the planner, which deals with the real world uncertainties, is like an unlucky manager, having to make a complex plan considering all the associated uncertainties and details. The planner has a very simple factory worker: merely actuators and isolated information channels from the sensors. Given contemporary technologies, it is often impossible to build such a practical planner working reliably in the real world, due to the complexity. On the other hand, if the planner hires a smart and skilled agent dealing with uncertainties by itself, the complexity of the job required for the planner becomes more tractable.

In behaviour–based assembly systems, the whole job is divided into two parts: planning and execution. The planner plans for general assembly problems such as the order of assembly. The behaviour–based executive agent performs tasks specified by the planner, demonstrating such assembly competences as bolting together; peg-in-hole; moving parts; fitting part faces; *etc.* Built under the behaviour–based approach, the executive agent deals with sensing and action for itself. The executive agent is composed of BM's, with each responsible for a particular assembly competence. Sensing and action, and the control are defined locally within a BM, which is modularised in terms of the behavioural competences required in assembly. The behaviour–based assembly system is a hybrid system with a symbolic planner and a behaviour–based executive agent, a smart agent which performs assembly work following the symbolic plan produced by the planner [Malcolm 95]. Compared to the traditional approach with sensing and action problems dealt with by the planner, behaviour–based assembly systems should provide a more appropriate structure for employing sensors, lowering the level of sensing and action management — low level amalgamation of sensing and action.

Behaviour–based assembly systems might provide an economic division between planning and real world execution, but the problem is where should this division be made in order to achieve the minimal complexity of the whole system? In other words, which and how much of the work is better managed by the planner and the agent respectively? This is analogous to the problem of dividing a task for the manager and the factory worker to produce the maximal productivity. It largely depends on the characteristics of the task, situation, and kind of information available.

For instance, for grasping, an executive agent can have a BM with a grasp competence. If the situation is straightforward, this BM can drive the robot to go and pick up the target part. But if the situation is more complex, for instance, when a complicated regrasp plan is required, it may not be economic for the BM to deal with all the problems involved. Further research is required on separating the task specification between the off–line planning job and the on–line reactive job for reduced complexity of the whole system. In addition, behaviour–based assembly systems have the same kind of problems in composing the executive agent, as the behaviour–based approach for mobile robots has. Problems of the individual structure of each BM, the architectural relationship between BM's, and the application of multiple sensors still require further study.

Morrow and Khosla [Morrow & Khosla 95] adopted the approach of employing a low level skilled assembly executive agent with the sensing and action encapsulated. Sensorimotor primitives, such as guarded move and correlation[9], are defined to compose an assembly skill, to achieve D–connector assembly. Generalisation and further development of primitives are left for further investigation. The coordination problem between the primitives and structural problems of the primitives when more than two sensors are employed, have not yet been addressed.

There have been a number of hybrid approaches to constructing an agent from a symbolic planner and a reactive agent or agents, centred around mobile robots. Firby pointed out that conventional planners lack interaction with the world, (he calls this interaction *situation–driven execution*), then implemented RAP (Reactive Action Package) where primitive robot actions are generated and monitored at execution time by the planner [Firby 89, Firby 93]. Georgeff *et al.*, tested a partial hierarchical planning strategy and a reflective reasoning system on an autonomous mobile robot, Flakey, where the robot exhibited a performance effectively combining partial planning and reactivity in an actual world [Georgeff *et al.* 87]. Gat implemented a classical symbolic planner on top of a reactive control mechanism in order to show that completely unmodified classical AI programming methodologies using centralising world models can be usefully incorporated into real–world embedded reactive systems [Gat 92]. Gat stresses that plans should be used to guide, not control, but action. Other approaches to combining reactive ingredients to a conventional symbolic planner can be found in [Payton 86], [Schoppers 87], [Ferguson 92], *etc.*

In general, these approaches were motivated by the fact that a mobile robot can face unpredicted situations which cannot be foreseen by advance planning. Hence these

[9] Sinusoidal Cartesian velocity (dither) added to the commanded velocity on purpose, is correlated with the forces and torques from the force sensor: shaking the endeffector and observing the reaction forces to the perturbation in order to obtain rich, reliable information [Lee & Asada 94].

systems are based on on–line planning, given the criterion of the activity of a mobile robot working in a real world. However, Edinburgh's behaviour–based assembly system approach is based on off–line planning. Since an ordering of the parts of the assembly task can be planned in advance, the complexity of a run–time interface between the planner and the executive agent is avoided by the off–line planning. Nevertheless, more debate on the need of partially on–line planning may arise where there is much uncertainty in local part manipulation.

In practice, in assembly robotics, the computational complexity of off–line pre-programming at a symbolic level for a simple agent (such as the average industrial assembly robot) has been recognised by Lozano-Pérez and Brooks. It was while they were collaborating on the design of TWAIN [Lozano-Pérez & Brooks 85], intended to map out the future assembly research at MIT, that Brooks decided that this centralised symbolic pre-planning was too complex to be practical, and that fundamental principles concerning agent architecture and sensor integration were missing [Brooks 86]. Lozano-Pérez responded to this criticism by moving away from the classical TWAIN design to HANDEY [Lozano-Pérez *et al.* 92], which involves a much more competent on–line agent (although the implementation is still classical in flavour). At Edinburgh, Fleming's attempt to incorporate uncertainty representation and reasoning into the RAPT system failed because of its severe computational complexity [Fleming 87]. Malcolm introduced the behaviour–based assembly systems in order to avoid these problems [Malcolm & Smithers 88].

So far, four major works have been performed at Edinburgh. The sensor-less SOMA assembly system showed the effectiveness of segregating task level assembly into a planner and a behaviour–based executive agent, where the planner plans about the SOMA assembly in an uncertainty–free ideal world, and the executive agent performs an assembly in the real world [Malcolm 87]. Given this, methodologies to accommodate sensors were next investigated. Chongstitvatana added an uncalibrated mobile vision system to the executive agent of the SOMA assembly with minimal modification of the planner [Chongstitvatana 92]. It is uncalibrated in the sense that no model of the camera, *i.e.* its position, *etc.*, was required. This showed that such a sensor can be integrated not at the level of planner, but at the lower level of executive agent, without the need for a global world model.

Independently of the vision approach, Wilson [Wilson 92] employed binary touch sensors at the fingertips as touch probes, and light beam break sensors on the inner surfaces of gripper fingers, to monitor the presence of the held object or to find a good grasp. Using these sensors, any failure of a BM during the SOMA assembly were detected. Using this touch sensing, Wilson addressed the architectural problem of facilitating error handling routines in BM's. In Wilson's system, sensors are mostly used to detect errors. A particular type of BM which yields a number of exit states was formulated. In detection of failure, the current BM exits in order to activate a correcting behaviour.

Applying force sensing has also been investigated by Balch for a different assembly task [Balch 92]. A force/torque sensor mounted on an assembly table was used during the assembly of an electrical contactor to evaluate the behaviour–based assembly approach. This application not only demonstrated successful assembly, but also provided a plausible methodology towards task level programming with sensing capability, modularising sub-tasks in terms of assembly purposes with local sensing capability.

However, each of these behaviour–based assembly applications has used only one sensor. In traditional assembly systems, problems concerning the use of multiple sensors are handled by the planners at the very top of the system hierarchy. This caused problems due to the resulting over–complexity of the planners and the sense–think–act cycle. It is a further research problem to investigate the use of multiple sensors for behaviour–based assembly systems. This is important because the challenge of behaviour–based assembly is motivated by the possible advantages which might be obtained by the low level amalgamation of sensing and action. For instance, in using multiple sensors, the planner must not consider details of sensing and action in order to achieve minimal complexity, while the executive agent manages the problem of real world uncertainties with multiple sensors. In the next section, approaches to the use of multiple sensors are reviewed.

2.4 Multisensor Integration and Fusion

Due to limitation in sensing modality of any one sensor, multiple sensors are required for more versatile and flexible robots. In the field of robotics, the study of organising multiple sensors is referred to as *Multisensor Integration.*

To stress the importance of studying the use of multiple sensors, Durrant-Whyte writes:

> If robot systems are ever to achieve a degree of intelligence and autonomy, they must be capable of using many different sensors in an active and dynamic manner; to resolve single sensor ambiguity, to discover and interpret their environment.
> ([Durrant-Whyte 88], p. 4)

Multisensor systems are defined and classified in [Luo & Kay 89], [Hackett & Shah 90], and [Ishikawa 92], where Ishikawa emphasises that as a robot system becomes more sophisticated, a sensor fusion scheme is necessary, and this can be better achieved through parallel processing and modularisation of the sensors into logical sensors[10].

The scope of the terminology "multisensor integration" and "sensor fusion" can vary depending on its definition. Luo and Kay make a plausible clear definition of multisensor integration and fusion:—

> Multisensor integration ... refers to the synergistic use of the information provided by multiple sensory devices to assist in the accomplishment of a task by a system. ... Multisensor fusion ... refers to any stage in the integration process where there is an actual combination (or fusion) of different sources of sensory information into one representational format.
> ([Luo & Kay 89], p. 903)

However, multisensor fusion (or sensor fusion) can have two scopes: 1) narrow scope: combining of sensors (or sensing) in order to obtain a unified physical or logical explicit

[10] Logical sensor: sensors are defined from the agent's computational point of view in terms of the information content, not by the physical attribute of the sensors. Consequently, a logical sensor can have a number of different possible physical realisations. For instance, for vision and touch sensor fusion, the agent looks at these sensors as one device, *i.e.* a logical sensor [Henderson & Shilcrat 84]. The same is also referred to as a Virtual sensor if seen from the functional point of view.

information in a certain format; 2) wide scope: including the narrow scope, combining sensing in order to obtain a unified effect (*e.g.*, action fusion, or behaviour fusion[11]). Sensor fusion can be a subset of multisensor integration, while multisensor integration is not necessarily used to provide one unified effect, *e.g.*, smart selection of sensors for different jobs. Multisensor integration is assumed to be used for automatic decision making and guiding which is normally associated with mobility, such as of a robot. Sensor fusion can be used for static purposes, such as identifying an object. In assembly robots, shape recognition tasks usually comprise understanding the part shape, and using this to update a world model (*e.g.* [Durrant-White 87], [Hutchinson *et al.* 88], and [Wen & Durrant-Whyte 92]). However, sensor fusion can also be used for dynamic purposes for example, by employing the concept of logical sensors, and it can also be referred to as multisensor integration.

In order to explain the wide scope of sensor fusion, an illustration of possible levels of abstraction of sensory information and actuation (hierarchy of technological domains) is exemplified in Figure 2.3 [Malcolm 96]. On the left hand side of the figure, a sensory signal is processed (abstracted) step by step into symbols which a modern computer can directly deal with — signal to symbol transformation. On the right hand side of the figure, levels of the forms of a motor command is shown — symbol to signal transformation. An input and output binding can be established at any level of abstraction. For instance, a centrifugal speed governor for a steam engine is a physically or mechanically established automatic system. An electric thermostat is at the electric level, and if it is digitised, it then becomes a digital level. Moreover, one may want to computerise a thermostat in order to control it in a general manner, *i.e.* using computer programs.

When there are more than two sensors introduced to be combined (*i.e.* sensor fusion), the computational level is often used. Use of computers would be convenient since the symbolised sensory information could be directly accessed by computation, such as formal reasoning. The majority of modern sensor fusion implementation falls into this category, where sensory information is combined at the computational level as in

[11] Flynn and Brooks [Flynn & Brooks 89] explain behaviour fusion as: "...different sensors trigger different behaviours and arbitration is done at the actuator stage rather than the sensor stage." Action fusion is used to mean the same.

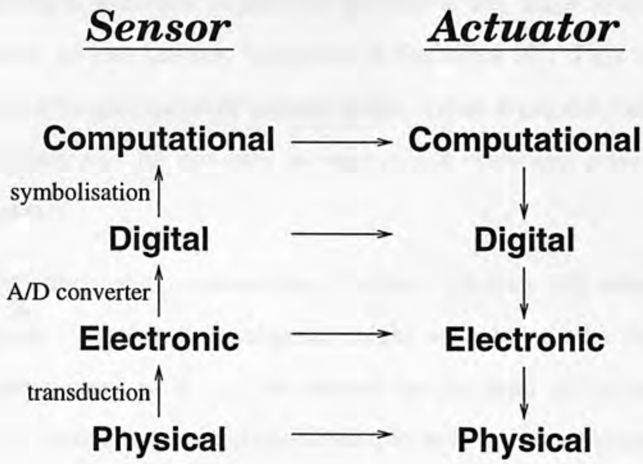


Figure 2.3: Levels of abstraction in sensing and actuation, and their bindings

Figure 2.3. However, it is also possible to combine multiple sensor information into one modality at a different level. For instance, the Braitenberg vehicle described in Section 2.2.1 combines information from two sensors implicitly in order to exhibit a single unified behaviour. They are combined by exploiting physical attributes, such as the fact that wheels are used and the manner of associating sensors and motors. Hence, it would be appropriate to argue that the sensory information is combined at the physical level (Figure 2.3). Action fusion may also fall into this category.

This is also related to the issues of different kinds of knowledge representation. Malcolm and Smithers proposed three kinds of knowledge representation: 1) Explicit; 2) Implicit; and 3) Tacit representations [Malcolm & Smithers 90]. Representation implemented in terms of explicit data structure is *explicit* representation. Indirectly declared representation, which can be derived by a certain manipulation from explicit representation, is referred to as *implicit*[12] representation. *Tacit* representation can be implemented in a procedural form, the crucial point being that it cannot be symbolised and derived explicitly. The flying skill of a bird is an example of tacit representation. The skill is embodied in the bird, but it is not explicitly represented. As a practical example, the SOMASS system has a planner and an executive agent built with a number of BM's. Details managed by the BM's are not accessible to the symbolic planner,

[12] Implicit here is used in a narrow scope to qualify a specific kind of representation as described in the main text, thus *implicit representation* has a special meaning. In this thesis, unless *implicit* is used to qualify the kind of representation, it has its usual meaning.

hence it is argued that although explicit to the BM's, the *ways to do* of the BM's are *tacit* to the planner, as the *knower* [Malcolm & Smithers 90]. This tacit knowledge is compared to the unconsciousness of human skills, where a person may know very well *how* to do something, but be unaware at an explicit conscious level of the details of the skill [Polanyi 67].

Given these three kinds of representation, Hallam [Hallam 96] added *absent* knowledge representation: "...the knowledge we might wish to ascribe to a system is not present in any component of it ..., but rather in the head of the system designer." Hallam uses as an example a collection of simple robots cooperating to collect small objects into a big pile. The collective behaviour results from a system containing no knowledge directly related to the task it performs. The phototaxic Braitenberg vehicle can also fall into this category. The representation cannot be located, but still exists somewhere. To make this distinctive, this thesis proposes the terms *located* and *unlocated* representation (see Figure 2.4). The first three representations (explicit, implicit, and tacit representations) are located representations since it can be pointed to where they are located. Unlocated representations cannot be pointed to, but emerge while the agent acts. The unlocated representations tend to utilise the involvement of the system in the environment. Sensor fusion can be performed in terms of unlocated representations, in such case as sensors are fused at the physical level in the Figure 2.3. Unlocated representation could also be algorithmically distributed, *i.e.*, *not* physical. Maes suggests that, for autonomous agents, internal representations relative to the purposes and circumstances of the agent are preferred [Maes 90]. The internal representation here can be referred to as the Tacit and unlocated representations.

The reason for emphasising these different kinds of knowledge representation is to make clear that sensor fusion can take place in terms of knowledge represented in any of these ways — not just at the explicit symbolic level. Note too that there is a natural tendency (but not a strict correspondence) as one moves down the hierarchy of technological domains to move down a similar hierarchy in terms of type of knowledge representation. Thus the guideline of behaviour–based system implementation to amalgamate sensing and acting at the lowest level will have the effect of this sensor–action amalgamation downwards in terms of both type of knowledge representation and technological domain

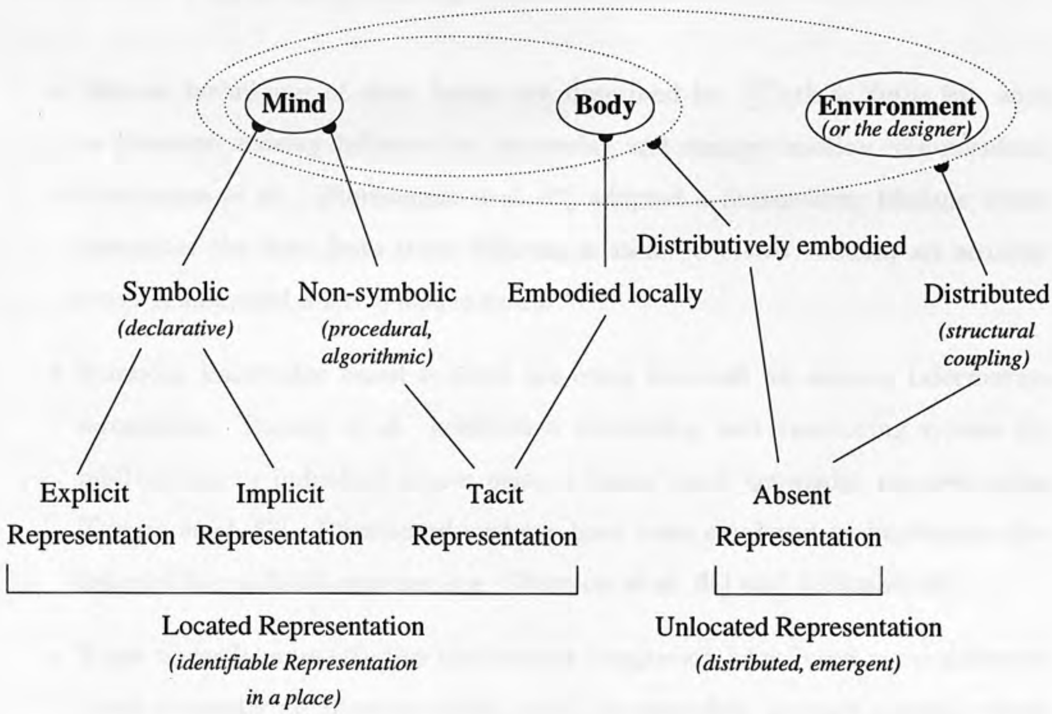


Figure 2.4: Types of representation

of implementation.

The utility of unlocated representation depends on the circumstances, thus is difficult to define in a general term. However, the behaviour–based approach puts emphasis on the active use of the environment by the system. This implies the behaviour–based approach is interested in the use of unlocated representation, which could make the whole system physically simpler than when the representation is expressed more explicitly.

The sense of sensor fusion used in this thesis has a wide scope, which includes fusion by any type of representation, located or unlocated. The rest of this section summarises a number of investigation of multisensor integration and fusion.

Since introducing more sensors demands greater information processing capability of the system, multisensor integration is often regarded as the problem of how to organise the information from various kinds of sensor. Many researchers have adopted a variety of information abstraction schemes in order to accomplish sensor fusion, at both low data levels or a higher symbolic level:—

- Various techniques of data fusion are described in [Clark & Yuille 90], such as Bayesian sensory information processing and energy function minimisation. Roukangas *et al.* [Roukangas *et al.* 86] adopted a Supervisory Module which integrates the data from three different sensors: a stereo camera, an acoustic range sensor, and a force/torque sensor.

- Symbolic knowledge based systems are often favoured for sensory information integration. Barnes *et al.* achieved a controlling and monitoring system for multiple-sensor industrial robots using a frame based knowledge representation [Barnes *et al.* 83]. Blackboard systems have been employed to implement distributed hierarchical systems, *e.g.* [Harmon *et al.* 86] and [Almand 85].

- Trials towards more effective multisensor integration introduced more elaborate robot controller architecture which could accommodate multiple sensors, where the problem becomes not only the matter of sensor fusion, but also a matter of system hierarchy and abstraction.

- In co-ordinating multiple sensors, distributed or decentralised hierarchies are often emphasised. Blackboard systems [Harmon *et al.* 86, Almand 85], Decentralised Sensing Network [Durrant-White 87], Object-oriented (schemas) systems [Lyons & Arbib 85], Subsumption Architecture [Brooks 89], and so on.

- Lyons and Arbib [Lyons & Arbib 85] defined objects within a task context, similar to the behaviour–based approach. The objects are active structures which perform information processing. The objects were incrementally constructed to perform a specified task. This structure simplifies the representation of multisensory object models. These objects are referred to as schemas in [Arbib 81].

The variety of approaches suggests that the principles involved are not yet understood. Lozano-Pérez *et al.* [Lozano-Pérez *et al.* 92] explicitly state this in discussing the architectural problems: "Task-level robot programming in workspaces with substantial uncertainty still requires fundamental new research in planning with uncertainty and planning for the use of sensors."

In looking for principles and methods, some have looked to biological systems for inspiration [Malcolm & Smithers 90, Brooks 86, Beer 90, Albus 81, Powers 73, Arbib 81]. It is claimed that the method of combining sensing and action in a robot structure plays a crucial role in increasing the effectiveness of sensing and hence the competence of the robot overall. All of these focus on the importance of distribution of control and modularisation into independent goal directed subsystems. These subsystems are capable of dealing with sensing and action by themselves. Concurrency is often emphasised as a means to coordinate multiple sensors effectively. This kind of system ought to be useful to manage complicated problems caused by employing disparate multiple sensors. Moreover, the discussions are not limited to technical robotics issues, but are often raised during discussion of the best way to achieve AI.

For instance, a model of a robot control architecture suggested by Albus is formulated such that behaviours (including high level mental behaviours such as expressing a will) can be described in terms of a functional mapping between the input and output. For a sequence of behaviours, a function is formulated, including a temporal variable t, such that:

$$\mathbf{P}_i(t) = H_i[\mathbf{S}_i(t - \Delta t)]$$

where $\mathbf{S}$ is the input vector, $\mathbf{P}$ is the output vector, and H is the functional operator as the behaviour.

The general structure is depicted in Figure 2.5, and an exemplified part is drawn in Figure 2.6, in the case of a bird.

Powers proposed a similar kind of model of the control hierarchy of humans [Powers 73]. In Powers' model, there are different orders (levels) of control, where lower levels perform more primitive functions while higher levels do more abstract and sophisticated functions. Powers' model, the Subsumption Architecture[13], and Albus' model are similar to one another in that:—

- The system is distributed in terms of tasks, meaningful in environmental terms

[13] or the behaviour–based approach. Note that the Subsumption Architecture is one of the implementations under the behaviour–based approach.

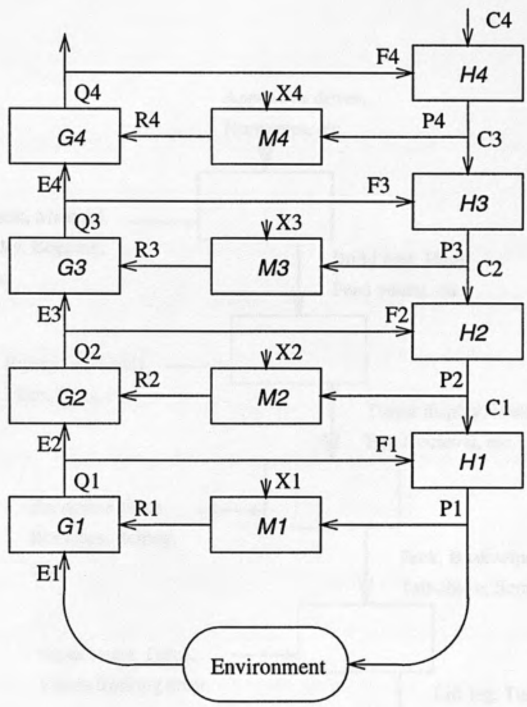


Figure 2.5: Albus' model of a biological control system [Albus 81]

(A cross–coupled processing-generating hierarchy. The H modules decompose input goals C into output subgoals P using feedback F. The M modules recall expected sensory data R which is compared with observed sensory experiences E. The G modules recognise sensory patterns Q and compute feedback errors F. Input to the M modules comes from subgoal information P which indicates what action is being contemplated or executed, as well as from context information X derived from a variety of sources throughout the brain [Albus 81] — note, there is no direct relationship between the labels used here and the variables used in the formulae shown in the page 39.)

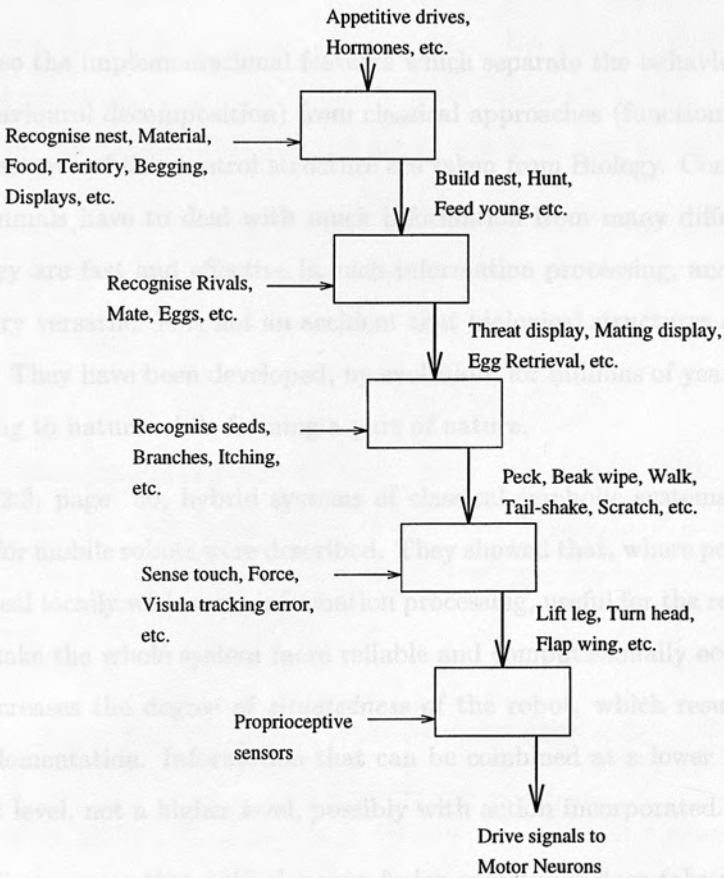


Figure 2.6: An abstracted form of Albus' model for a bird [Albus 81]

(behaviours).

- All control levels have their own access to sensing and action if necessary, and thus are complete in their own scope of control.
- More abstract and sophisticated control levels use (subsume) the lower control levels.
- Sensing and action couplings tend to be implemented at as low a level as possible.

These are also the implementational features which separate the behaviour–based approach (behavioural decomposition) from classical approaches (functional decomposition). The features of this control structure are taken from Biology. Complex animals such as mammals have to deal with much information from many different kinds of sensors. They are fast and effective in such information processing, and they are robust, and very versatile. It is not an accident that biological structures are so reliable and flexible. They have been developed, by evolution, for millions of years. They have been adapting to nature while forming a part of nature.

In Section 2.3, page 30, hybrid systems of classical symbolic systems and reactive subsystems for mobile robots were described. They showed that, where possible, having a low level deal locally with some information processing, useful for the reactive task in hand, can make the whole system more reliable and computationally economic. This, in effect, increases the degree of *situatedness* of the robot, which results in a more reliable implementation. Information that can be combined at a lower level is better done at that level, not a higher level, possibly with action incorporated.

Animal studies suggest that animal sensor fusion sometimes does take place at a low level (*i.e.*, not a cognitive level, as possibly modeled in a centralised symbolic fusion) and locally couples perception and action [Murphy 96]. In the behaviour–based approach, rather than sensor fusion in the form of data, behaviour (or action) fusion [Flynn & Brooks 89], (*i.e.*, fusion at the actuator stage) is often recommended for better variation in behaviour and more effective fusion between two or more modalities of sensing. Making use of the environment for instantiating behaviours is also considered to be an effective method of action selection [Connell 88]. For instance, an action of one BM with a sensor may influence the environment and this influence is detected by another sensor used by another BM, which is then triggered. This kind of behaviour triggering through the environment means that no explicit physical or logical coordination is required in the robot.

As seen so far on the multisensor integration research area, approaches to more effective and adaptive structures in using multiple sensors have been proposed and are still being investigated. The behaviour–based approach is one of the them. In the behaviour–based approach, although research has been performed on the use of multiple sensors, centered around action selection for mobile robots, the use of multiple sensors for behaviour–based assembly systems is still to be investigated. This is also related to the architectural problem of the executive agent, which employs multiple sensors.

Along with the consideration on the appropriate system structure to accommodate multiple sensors, considerations on the individual sensing devices are also important, since the kind of sensors employed affects the system structure. It is pointed out by Hackett and Shah:

> One of the most important areas which will have a significant impact on the research in multi-sensor fusion is in sensor design. The majority of currently available sensors are slow, less robust, and expensive.
> ([Hackett & Shah 90], p. 1328)

2.5 Reactive Sensing for Assembly Robots

When a potential application of a sensor is decided on, the way the sensor is to be used and how the information is to be processed must be clarified. In this section, ways of using sensors for assembly robots are looked at. Reactive dynamic event sensing is stressed as a useful sensing modality for assembly robots, especially for behaviour–based assembly systems. Tactile sensing is seen as an appropriate sensing device for reactive dynamic sensing.

2.5.1 Use of Sensors for Assembly robots

Much of the effort involved in setting up a robot work cell is devoted to reducing uncertainties. As an assembly job is changed, special purpose jigs, tools, or feeders are often changed accordingly. In contrast to hard automation, sensors can be employed in order to make robots more versatile in coping with uncertainties in the environment [Rosen & Nitzan 77]. This means when there is any change of task, reprogramming the robot may be all that is necessary, rather than changing much of the hardware.

Sensors are used in a number of different ways for assembly robots. In terms of applications where the effect of robot motions are important, sensors can be categorised in the following ways [Malcolm & Fothergill 86, Lozano-Pérez 82]:

- to control local motion, such as used for force servoing to achieve a peg-in-hole task, where complying to external constraints is required;
- to determine the precise location of features of objects;
- to determine what to do next (sequence control);
- initiating and terminating motions, as in guarded moves.

There are various types of sensors available, which can be divided into internal sensors and external sensors [Klafter *et al.* 89]. Internal sensors are used in controlling the manipulator, concerning the internal status such as angular position of manipulator joints. External sensors detect occurrences outside the robot and provide information about the world to the robot. Although internal sensors are dedicated to internal use, they can also be utilised indirectly to cope with external uncertainties. From the fact that the robot is capable of locating its end-effector precisely, sweeping motions or snapping motions can be employed, to make sure the object is in a location to within the precision of the robot operation [Malcolm 87]. There are a number of external sensors that can be used to cope with the uncertainties of the environment: cameras; tactile sensors; force/torque sensors; proximity sensors, *etc.*

However, Lozano-Pérez emphasises that in assembly, the reasons for sensors not having been used widely are: 1) The lack of reliable and affordable sensors; 2) Existing techniques for sensory processing have tended to be slow when compared to mechanical means of reducing uncertainty [Lozano-Pérez 82].

Although more than a decade has passed since then, these problems are still present. For example, manipulator level robot program languages such as VAL II are still the predominant type of language used in practice. Programming with sensors in such languages is very cumbersome. It is important to develop both sensors which are appropriate for an assembly robot and a better robot programming scheme for sensors.

2.5.2 Pulling Down the Level of Sensory Coordinations

Sensory–motor coordination using artificial neural networks is investigated in order to equip a robot with a capability to adapt to the changing environment rather than to be rigidly programmed [van der Smagt 95, Ritter *et al.* 92]. For instance, in the work of Rucci and Bajcsy, sensor space and actuator space are mapped at a neuronal level, to provide adaptive sensory motor coordination [Rucci & Bajcsy 95]. Their system exhibits visual attention to touched locations, in two degrees of freedom. Although this system exhibits a primitive behaviour, it shows that vision, touch, and motion can be organised without having to be referenced to a centralised world model. Other similar examples are MURPHY [Mel 88] and INFANT [Kuperstein 91]. Churchland illustrates how crabs might manage vision and reaching economically, managed by a simple mapping between vision space and motor space, at a low neuronal level [Churchland 86].

Although in these examples only the problems of feedforward control obtained by mapping are highlighted on, they show that it is possible to organise sensing and action at a low level, in a manner which animals might use, in particular, as Churchland emphasises, avoiding the use of a Cartesian geometric model. Churchland suggests that animal nervous systems are more likely to be organised in this way, rather than using a geometric model, because it is computationally simpler. The low level amalgamation of sensing and action in the behaviour–based approach adopts this feature of the low level management of sensing and action for the same reason — avoiding the complexity of a global geometric representation.

2.5.3 Reactive Event Sensing in Assembly

Canny and Goldberg introduced RISC (Reduced Intricacy in Sensing and Control) robotics [Canny & Goldberg 93]. Their RISC robots decompose complex operations into simple elements, to achieve reduced intricacy. One of the guiding principles is: "Sensors and actuators can be combined to yield very flexible active sensors."

The importance of event sensing became recognised in practice by virtue of its simplicity and versatility (*e.g.* in the InFACT system [Hardy *et al.* 92]), particularly in conjunction with a qualitative robot controller. Košecká and Bogoni [Košecká & Bogoni 94] define autonomous agents as Discrete Event Systems (DES). They define **states** and **events**, where the states correspond to some continua in the task evolution, and the transitions between states are caused by events, representing the qualitative changes in environment or task evolution. Tasks, such as piercing and part picking, are described by sequencing states and events. DES complements the linear mathematical representation of complex computer-controlled robotics and automation systems, in modeling system state changes within a process [Sobh & Valavanis 94].

In assembly tasks, where parts are mated, event sensing is one of the most important sensing modalities. Figure 2.7 shows the frequency of occurrence of specific tasks in assembly [Pettinaro 96]. Many assembly problems can be solved by just knowing if parts are making contact, such as peg-in-hole and screwing [Pettinaro 96]. The effects of robot motions on the environment and external changes in the environment lead to events. By associating robot motions and sensing, the robot can be provided with more information than continuous sensing. For instance, using guarded moves, a robot can identify the location of an object economically with comparatively simple sensing and computation [Will & Grossman 75]. Dynamic sensing [Beni *et al.* 83] focuses on the use of motions to enhance sensing activity.

It can be argued that guarded moves are well suited to behaviour–based assembly systems in the following senses:—

- It is useful to solve simple contact problems in assembly tasks.
- It can be conveniently modularised because it consists of a complete control loop

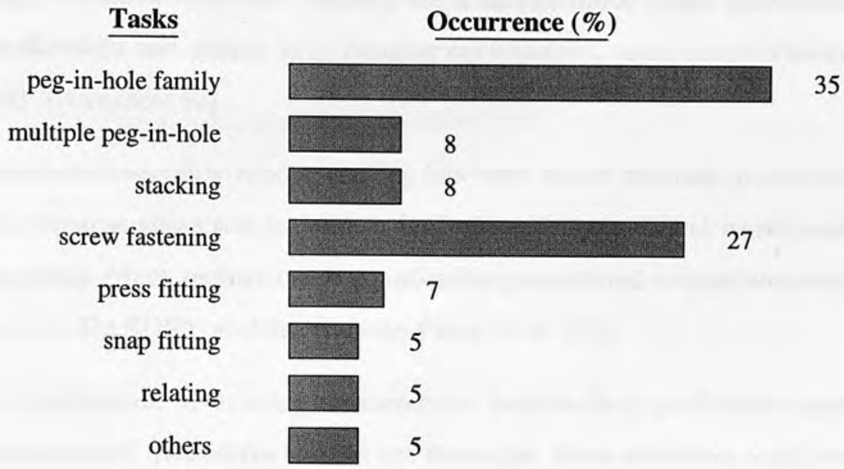


Figure 2.7: Frequency of occurrence of specific tasks in assembly [Pettinaro 96]

with sensing and motion.

- The sensors in guarded moves can be used to provide feedback to behavioural competences which act on the environment.

For behaviour–based assembly systems, guarded moves might provide a useful basis as part of the basic reactive behaviours of a system hierarchy, on which more sophisticated behaviours can be built.

2.5.4 Quantitative Sensing versus Qualitative Sensing

Quantitative sensing involves measurement. Measurement is favoured in many robotic applications, such as measuring accurate distance. For example, a mobile robot which knows exactly where it was, and can measure time, speed, and direction of travel, can compute by dead reckoning where it now should be — in an ideal world. However, the world is not ideal. The wheels may be slightly different in size, wear slightly differently, and the friction coefficient of the floor may be different at different locations [Nilsson 84]. The fact is that the world is full of uncertainty and the agent is supposed to interact dynamically with the environment. Smithers [Smithers 94] proposed: "..., I want to suggest that sensors on robots are not best understood as measuring devices, though this is how they are normally thought of." Rather than measuring, Nehmzow adopted a strategy for a mobile robot which qualitatively identifies the situation and adapts to a changing environment, using simple binary sensors (whiskers) [Nehmzow 92].

For conventional assembly robots, sensing has been biased towards quantitative measurement, because these robots tend to be built with geometrical world models in a Cartesian space which require much quantitative geometrical comparison and calculation (*e.g.*, the HANDEY system [Lozano-Pérez *et al.* 92]).

Both the application of existing measurement sensors in a qualitative manner and the development of qualitative sensors are desirable. More attention could be paid to qualitative sensors which can be used in a more reactive manner. Reactive sub-systems and appropriate sensors would provide an economic base on which more abstract parts of a system can be built. This would enable a robot system architecture distributed in terms of information processing, particularly in terms of behavioural competence.

2.6 Summary

There has been very little work developing the principles of the behaviour–based approaches as a paradigm in AI. Brooks seriously introduced the approach to AI with a practical robotic example [Brooks 86], and reinforced the principles in [Brooks 91]. Varela *et al.* [Varela *et al.* 92] discussed the approach from slightly different perspectives which mainly originated from Autopoiesis [Maturana & Varela 80] and some Eastern philosophy. There has not been other recognised major work on the basic principles, but there do exist some partial discussions, applications, and supportive materials which are dealt with elsewhere in this thesis. The general review and extraction of basic principles on which the behaviour–based work in this thesis rests is [Malcolm *et al.* 89][14]. Very recently (1996), Pfeifer summarised the principles and features behind the approach, by proposing a number of design principles of autonomous agents [Pfeifer 96]. Pfeifer reviews much the same as [Malcolm *et al.* 89], plus some Artificial Life (AL) work[15]. The general principles he derives from this survey are very

[14] See the guidelines in Section 2.1.3, page 21.

[15] AL research encompasses both real autonomous robots and simulated agents in simulated worlds, *e.g.*, see [Maes *et al.* 96]. An extra feature here is the requirement of self–sufficiency, *i.e.*, foraging

close to those in [Malcolm *et al.* 89], plus again some specifically related to AL. Since Pfeifer seems unaware of the [Malcolm *et al.* 89] paper, this is a nice confirmation.

This chapter looked at the background issues related to the work presented in this thesis. The position of the behaviour–based approach in the field of AI was first looked at. The behaviour–based approach suggests a plausible way to build a robust and reactive autonomous agent with sensors in order to cope with uncertainties. Decomposition is in terms of purposeful behaviours hence it eliminates the inefficient bottleneck sense–think–act control loops of traditional robotic systems.

A behaviour–based assembly system is a hybrid system with a minimised symbolic planner, and a behaviour–based executive agent skilled for assembly tasks. The planner plans for general assembly problems, and the low level executive agent performs the required tasks in the real world, coping with uncertainties by localised sensing and action coupling. Previous work showed that sensing and action problems can be managed by the low level executive agent without any intervention by the planner. Architectural problems of executive agents using multiple sensors are left for further study.

Multi–sensor integration and fusion were defined and explained with respect to the architecture of technological domains and the different kinds of representations. Other approaches using multiple sensors for assembly robots were reviewed. Distributed hierarchies to accommodate multiple sensors were described, which are based on the anatomical and behavioural observation of animals. Including the behaviour–based approach, these suggestions provide a number of general features:—

- The system is distributed in terms of tasks, meaningful in environmental terms (behaviours).
- All control levels have their own access to sensing and action if necessary, and thus are complete in their own scope of control.
- More abstract and sophisticated control levels use (subsume) the lower control levels.

for food — not a consideration in Assembly Robotics!

- Sensing and action couplings tend to be implemented at as low a level as possible.

Criticisms were made with respect to the lack of appropriate sensors for assembly robots, especially used in a reactive and qualitative manner. The reactive guarded move is recognised as a sound basis of sensing and action coupling for assembly robots, especially for behaviour–based assembly systems. Guarded moves can be accomplished either by a force torque sensor or a tactile sensor. Force torque sensors are expensive and often noisy when mounted on the wrist of a robot. Most tactile sensors are expensive, fragile, and insensitive to the torque or shear force although there are a few exceptions.

This thesis next reviews the issues related to tactile sensing for dynamic purposes. This thesis then describes the development of event touch sensors which are appropriate for guarded moves and other touch applications, yet are economic, fast (reactive), and robust. The application of the developed touch sensors to behaviour–based sensor fusion problems is then described.

Chapter 3

Tactile Sensing for Dynamic Purposes

Human tactile sensing is so versatile and delicate that fine manipulation greatly relies on it, for instance in the assembly of a clockwork watch. Note that the clearances of assembly, *e.g.* 1/100 mm, exceed the dead-reckoning accuracy of almost all assembly arms, therefore some extra assistance is needed. Hence, it is useful to aim for something like human tactile sensing for assembly robots that are equipped with a fine part manipulation capability.

It might be possible that confusion might occur between the terms and classifications regarding tactile sensors. Tactile sense literally means the sense of touch. Robotic tactile sensors range from a matrix tactile sensor to a whisker. Suppose there is an infrared sensor mounted on a fingertip of a robot gripper, which is used to locate an object. This can also be regarded as a tactile sensor, where the infrared beam acts as the touch medium. The terms "active" and "dynamic" sensing can cause confusion. Although both can be used to mean sensing involving motion, this thesis distinguishes: 1) active sensing, as opposed to passive sensing, to mean sensing which actively emits energy with which to sense, such as infrared or sonar sensors; 2) dynamic sensing, as opposed to static sensing, to mean sensing which involves motion[1].

Robotic tactile sensing can be broadly classified into static and dynamic in terms

[1] However, "active" can be used to mean both senses. Active Vision uses motion in order to help vision sensing [Aloimonos & Bandyopadhyay 87]. Hillis [Hillis 81] referred to tactile sensing incorporated with motion as "Active Tactile Sensing". But, Active Sonar is commonly used to mean self–emitting ultrasonic sensing [Akbarally & Kleeman 95].

of the manner in which the robot is controlled. This thesis is interested in tactile sensing for dynamic purposes, *i.e.*, involving motions. In this chapter, tactile sensing for dynamic purposes in part manipulation is reviewed. Firstly, the functional scope of the transducers of human tactile sensing is described. Secondly, technologies for robot tactile sensing are reviewed. Lastly, robot tactile sensing for dynamic purposes during part manipulation is discussed.

3.1 Human Tactile Sensing Transducers

There are various types of sensory receptors in the skin, which are non–uniformly distributed throughout the human tissue. Their known functions are briefly described as follows [Albus 81, Bridgeman 88, Moss-Salentijn 92]:—

Free nerve endings can detect very slight pressure and are extremely sensitive. They are also responsible for temperature sensing.

Pacinian corpuscles are the largest of the encapsulated endings. Due to the protection of its large capsule from steady mechanical pressure, the nerve ending is sensitive only to changes in pressure. They also serve the kinesthetic sense.

Meissner's corpuscles are found in hairless skins, responsible for the localised pressure sensing. These provide a high degree of spatial localisation.

Ruffini end-organs detect continuous deformation of the skin and deep tissues.

Hair end-organs detect mechanical deflection of the hairs to which they are attached.

The tactile sensing of human beings provides important information for assembling parts, especially when objects cannot be seen or more accuracy is required than other sensors can provide [Russell 90].

3.2 Technologies for Robot Tactile Sensing

Human tactile sensing comprises touching, force, temperature, vibration, and feeling texture by slipping. While it is difficult to condense all these competences into a small package, or even to implement one competence well, current tactile sensors mimic some of these competences. Replicated human sensing competences for robots can be itemised as: simple contact; magnitude of force; 3-dimensional shape; slip; thermal properties; and so on [Nicholls 92a].

Tactile sensing can be divided into two modes based on the properties of the object sensed: *extrinsic object properties* and *intrinsic object properties* [Harmon 92]. Extrinsic properties comprise shape (edges, corners, faces...), texture, and hardness. Intrinsic factors are force, moment, and displacement, in addition to their time derivatives. Extrinsic properties are retained by the object, and hence are mostly static, although motions are required to sense texture for instance. On the other hand, intrinsic factors only appear dynamically in response to the environment.

3.2.1 Extrinsic Tactile Sensors

Research on tactile sensing has been biased towards the extrinsic approach, where tactile sensing cells are spread over the contact surface, focusing on the problems of processing the projected images (*e.g.*, continuous force or binary) of the gripped object. Various technologies have been investigated for use as the sensitive cells of tactile array sensors. Functionally, these cells can be either binary or force sensors.

For example, Hillis [Hillis 81], in the early 80's, developed a tactile array sensor with each sensitive cell being a force sensor. Anisotropically conductive rubber (ACS) was used. Conductive rubber presses through a meshed separator on a printed circuit board so that the area of contact, hence the contact resistance, varies with the applied pressure. Dario and De Rossi [Dario & De Rossi 85] reported their work on the development of a human-skin-like tactile sensor. Their sensor comprises deep ("dermal") and shallow ("epidermal") sensing layers, based on the technology of ferroelectric polymers using PVDF transducers. The dermal layer was intended to mimic the role of the slowly adapting receptors of the human skin, which are sensitive to the spatial features of the indenting object, while the epidermal layer was implemented to cover a few sensing sites and particularly sensitive to dynamic contact stimuli, like the quickly adapting skin receptors [Dario & Buttazzo 87].

Other techniques, such as capacitive, magnetic, and optical transduction, are well reviewed in [Nicholls 92b], [Russell 90], and [Howe & Cutkosky 92].

3.2.2 Intrinsic Tactile Sensors

Compared to Extrinsic tactile sensors, research on intrinsic tactile sensors is scarce. Salisbury [Salisbury 84] analysed contact geometries in order to obtain high quality control of the force and motion states of the grasped object. Bicchi and Dario [Bicchi & Dario 87] reported their work on an intrinsic tactile sensor using seven strain gauges mounted on a finger bone in order to measure the force exerted on the finger during part manipulation. In their work, an extrinsic sensor was implemented on top of the intrinsic tactile sensor used in a complementary manner. While strain gauges are widely used for force measurement, Okada and Rembold [Okada & Rembold 92] point out the difficulties in using strain gauges due to their fragility, sensitivity to temperature, and possible crosstalk for multiple-axis load cells. They proposed an optical technology to measure[2] force for an intrinsic tactile sensor.

Bicchi and Dario [Bicchi & Dario 87] identify advantages and disadvantages of the extrinsic and intrinsic approaches to tactile sensing, as reproduced in Table 3.1. However, there are a few points to be questioned. First, Bicchi and Dario describe extrinsic tactile sensors as unsuitable for slippage detection. But it would be possible for a force array tactile sensor to detect slippage by comprehending the change of the contact images. Furthermore, apart from array tactile sensors, a vibration tactile sensor, although still an extrinsic tactile sensor, can be made to detect slippage (*e.g.* [Son *et al.* 94] and [Howe & Cutkosky 89]). It is possible for an intrinsic tactile sensor to sense slippage by interpreting the vibration detected by a force transducer (*e.g.* [Eberman & Salisbury, Jr. 94]). Second, on the encumbrance feature, extrinsic tactile sensors could have many wires if they are tactile arrays, while those with few transducers would not. Intrinsic tactile sensors can be bulky depending on the technology and the transducer type employed.

[2] Note that event sensing would only require the sensitivity to change, which is technologically less demanding than accurate measurement which would be required for traditional knowledge-based systems, as discussed in Section 2.5.4, page 47.

Features	Type	
	Extrinsic	*Intrinsic*
Spatial resolution	Inherently finite	Theoretically infinite
Bandwidth	Limited	High
Contact force measurement	Generally Inaccurate	Fast, linear, nonhysteretic
Frictional effects	At present, not sensed	Measured
Slippage detection	None	Possible
Sensor surface shape	Free	Only simple shapes
Sensor cover compliance	Allowed	It produces errors
Paratactile sensitivity	Possible	Impossible
Encumbrance	Many wires	Rather bulky, few wires

Table 3.1: A comparison between extrinsic and intrinsic tactile sensors [Bicchi & Dario 87]

3.3 Use of Tactile Sensing with Motion

Beni *et al.*, proposed *Dynamic sensing* [Beni *et al.* 83], for the situation where robot motion increases the functionality of sensors, obtaining more information than when used in a static manner. For instance, a single photo cell can be used with planar motion of a robot in order to obtain a full 2D image. Guarded moves [Will & Grossman 75] by their nature involve motions in sensing. For instance, when a robot places an object on an unknown part, exploratory moves would be required with appropriate sensing incorporated until any expected contact is met. A guarded move is a form of dynamic sensing which combines sensing and motion.

Since tactile sensors retrieve only information about the contacted part of the gripped object, and the active area of the sensor is often small compared to the size of the gripped object, so the information received may not be sufficient to recognise the object. Examination of multiple contact images involving motions is referred to as *Active Tactile Sensing*[3] [Nicholls & Lee 89, Hillis 81, Allen 92]. Sensing of texture of an object would also require robot motion. One of the features of the PVDF tactile sensor developed by Dario and De Rossi [Dario & De Rossi 85], is to obtain the surface texture of an object using some control method shown in [Dario & Buttazzo 87].

In mating parts, detecting the force acting on the gripped object is important, as Dario

[3] The term, *Active Tactile Sensing* here should be distinguished from active sensing where the sensor actively applies emitted energy such as a light beam as a part of the sensing activity, as identified at the beginning of this chapter.

[Dario 89] points out: "Sensing and controlling forces and moments generated at the end effector is necessary in order to control compliant motion, a fundamental aspect of dextrous behaviour."

Tactile sensors, useful and suitable for part manipulation in motion, are torque or shear stress sensitive tactile sensors. This type of tactile sensor has an advantage over conventional force/torque sensors mounted on the wrist or other joints, since there is less physical medium between the part and the sensor.

Intrinsic tactile sensors are used for dynamic purposes. Depending on the reacting force from the contact, the robot may be programmed to operate accordingly. This competence is involved in the study on manipulation in contact [Salisbury 84, Bicchi *et al.* 90].

Apart from conventional intrinsic tactile sensors, there are approaches using tactile array sensors, or other techniques which measure the shear force between object and finger. Hackwood *et al.* used magnetoresistive sensors for tactile array sensors which can detect torque and tangential force as well as normal force [Hackwood *et al.* 83]. Each cell of the tactile array sensor is composed of magnetic dipoles in an elastic medium whose position and orientation are detected by magnetoresistive sensors.

Howe *et al.* [Howe *et al.* 90] argue that for manipulation by a robot, touch sensors which detect phase changes (events), such as making or breaking contact, are important since control schemes must change to match the varying task requirements, which is analogous to human tactile sensing. They use the term *Dynamic Tactile Sensing*, since the events occur as the consequence of robot motion.

One of the simplest tactile sensors is a binary switch which detects the existence of an object between the fingers or the existence of an object at a predetermined position [Wilson 92]. Binary event touch sensors are used in a distributed manner by Nicholls and Hardy [Nicholls & Hardy 92], in the InFACT project. Shinoda *et al.* proposed the tensor cell concept [Shinoda *et al.* 95]. Tensor cells are sparsely embedded in a compliant material which forms a tactile surface. Each tensor cell can be made of a rigid cube with etched PVDF films pasted on each side which detect force patterns normal to them. They implemented one experimental tensor cell to carry out an experiment, with a maximum force of 0.8 N. It detected the location of the normal force applied with an accuracy of 1 to 2 mm. However, the computational requirements and their complex physical structure make these sensors expensive. Sensitivity is also absorbed by the soft covering material in the case of the tensor cell sensor.

Force sensitive resistors on a soft finger surface are used to detect the shear force applied to the gripped part during part manipulation by Borovac *et al.* [Borovac *et al.* 94]. Force sensitive resistors are compact and economical force measuring devices [Int95].

In dynamic tactile sensing, an understanding of the information that can be obtained from the impact of the object is often necessary, where impulsive forces may dominate all other forces [Wang & Mason 87]. For instance, Söderquist and Wernersson used measured acceleration in order to find the point of application of the impacting force, as well as its line of action, which had a positional accuracy of roughly 6 % of the dimension of a body [Söderquist & Wernersson 92].

Tactile sensing for dynamic purposes has the potential to be easily exploited in the context of behaviour–based assembly. Guarded moves can be made to have simple but complete control loops. This feature eases the low-level amalgamation of sensing and action and behavioural modularisation (see the guidelines for the behaviour–based approach in Section 2.1.3, page 21). Active use of the world can also be achieved rather naturally. Since guarded moves use robot motions, the robot can affect the environment actively with a sensing capability incorporated.

3.4 Summary

In this chapter, issues related to tactile sensors, particularly for dynamic purposes, were reviewed. Tactile sensors can be divided into *extrinsic* and *intrinsic* tactile sensors. Tactile sensors that are able to detect the extrinsic properties of the object such as shape, are extrinsic tactile sensors. Practical intrinsic tactile sensors measure the force and/or torque applied to the gripped object during manipulation. Intrinsic tactile sensors are useful in control of the manipulator in contact. *Dynamic Tactile Sensing* refers to tactile sensing that involves robot motion which utilises contact force measured or change of state event.

The work described in this thesis is concerned with tactile event sensing, to detect change of state. By detecting changes of state, an assembly robot can be informed about events such as when the gripped part makes contact with something else. In part mating, contact sensing is important. Intrinsic tactile sensors and some other tactile sensors are able to detect change of state. This thesis describes a practical implementation of economic tactile event signature sensors using PVDF films that are suitable for assembly tasks, followed by a description of and discussion about the behaviour–based robot control that uses the touch sensors developed.

In the next chapter, the characteristics of the PVDF films, which are known to be very sensitive and versatile, are first described. Experiments with the PVDF films, that were carried out in order to extract useful characteristics of the transducer for tactile event sensors is then described.

Chapter 4

Investigating PVDF Films

This chapter describes the characteristics of PVDF films and general experiments performed with PVDF films to evaluate the feasibility of using the film as touch event signature sensors exploiting vibration. These experiments are not only aimed at those sensors which detect sound, but also for exploring the potential of PVDF films as sensors for assembly robots in general. The outcome of the experiments confirmed the feasibility of firstly, event sensors interested in sound propagated through the assembly space, secondly, event sensors which are sensitive to force change occurring on the sensor body. The latter type of sensor detects part contact events by detecting force changes occurring on the gripping surfaces of the fingers.

This chapter contains a description of the motivation for developing the event signature sensors, the rationale behind the experiments, the characteristics of the PVDF film as a sensory transducer, and the experimental setup. There were two kinds of experiment carried out: Indirect response and Direct response. In the indirect response experiment, the PVDF transducer is fixed to different kinds of robot working tables: a wooden and a metal table for an RTX robot and a different metal table for an ADEPT1 robot. Various kinds of impacts are applied to the tables, then the signal patterns are collected and analysed. In the direct response experiment, impacts are applied directly on the PVDF transducer then the signal patterns are again collected and analysed. These experiments are described and discussed in the rest of this chapter.

4.1 Why Event Signature Sensors?

For robots, sensors are the means of interaction with the surrounding environment. Robots with sensors can take advantage of the interpreted sensory information, an abstraction of the world, so as to make a decision about the world. The quality of the abstraction depends on what is detected and how relevant it is to the task of the robot. Hence, the versatility of robots much relies on the type of sensors used, how they interpret the incoming information, and how they are exploited, *with respect to the task.*

For conventional robots where an explicit world model plays a major role, sensors are often required to make measurements in order to compare a geometrical world model to the real situation. However, reactive and adaptive behaviour–based robots would require a different kind of sensor: sensors localised in behaviours, providing quick response, but first of all, providing the feedback for the behavioural competences just performed or being performed (see the comparison between qualitative and quantitative sensing, Section 2.5.4, page 47). Event sensing, as qualitative sensing, would be useful for such control systems as the behaviour–based systems.

Employing low level reactive and adaptive behaviours has advantages over the conventional measure–think–act methods as explained in Chapter 2. Event sensing could provide localised information for a modularised part of the system, *e.g.*, a BM. Modularised behaviours include self-contained information processing capabilities with the localised sensing, hence the encapsulation and distribution of sensing–and–act information processing would become easier and more natural.

It is often economical to make use of any information sources available, such as robot motions, as noted by Beni *et al.* on Dynamic sensing [Beni *et al.* 83]. Low level reactive behaviours are desired to make full use of their motions incorporated with sensing in order to disambiguate uncertainties. One of the most appropriate type of sensors for this kind of purpose is an event signature sensor, which notifies the occurrence of an event, for example notifying when a gripped part knocks something else while the robot moves.

Some biological sensors respond to *changes* of physical quantities. For example, the muscle spindles monitor changes in the length of a skeletal muscle by responding to the rate and degree of change in length [Tortora & Anagnostakos 90]. Another example is the Pacinian corpuscles in the somatosensory system [Bridgeman 88]. They are located deep inside hands and feet in abundance, and are only sensitive to changes in pressure. As an example of a more sophisticated sensor, the eyes of frogs are sensitive to the movement of objects [Arbib 87].

Our aim is to equip a robot, particularly a reactive assembly robot, with an economical yet useful sensor which is able to notify events involving the gripped object during manipulation. Machine Vision systems may measure the relative location of the objects and the end of the manipulator in order to notify any possible part contact, and force sensors identify the force applied to the gripper or the gripped object. In practice, however, the spatial resolution of Vision systems is often limited to a millimetre due to the confined image resolution[1]. Force sensors or intrinsic tactile sensors are normally massive, big and expensive. Conventional extrinsic array tactile sensors are in general unsuitable for sensing of events happening to the gripped object, such as contacts[2].

Developing of economic (in both response time and resources) event signature sensors is important in that detecting events can introduce reactive and adaptive robotic assembly relatively economically, especially when supported by the methodology of the behaviour–based approach.

Although air-borne sound is weak, the clunk sound which is generated during robotic assembly and can propagate through a hard material such as the robot working table, can be strong enough to be reliably detected as an event. Inspired by this idea, making use of PVDF films as the transducer, the feasibility from various perspectives of event signature sensors is explored, by carrying out, first of all, general experiments on the vibration properties of PVDF films. These experiments are described in the rest of this chapter.

[1] Although some elaborate estimation techniques can provide a higher accuracy where features are described by many pixels (*e.g.*, a long straight edge) [Naidu & Fisher 91], in general, and especially for small features, the accuracy of normal Vision systems is limited to one pixel. Usual Vision systems have 512×512 pixels in one image. Suppose a 10×10 cm part appears on the image occupying 100×100 pixels. These are typical values for an industrial part viewed in close up by a camera. Then, one pixel corresponds to 1 mm in the space near the part.

[2] Contact force measurement of an extrinsic tactile sensor is considered to be generally inaccurate as mentioned in Table 3.1.

4.2 Strategy for the Development

In developing a vibration sensor, the issues are:—

- What are the vibration characteristics of the sensor, table, robot and objects.
- How much noise degrades the effectiveness of the sensor.
- What kind of signals can be taken into account in order to determine the kind of sensor we could have.
- Where to fit the sensors.

In order to answer these questions, first data and noise sampling was carried out under various conditions making use of the film as a microphone affixed to three different kinds of robot working table. Then the data were analysed using the Fast Fourier Transform routine available in the software package Pro-Matlab. These three different tables are wooden and metal tables as an RTX robot working table, and a massive metal surface table for an ADEPT1 robot. In these experiments, the RTX and ADEPT1 robots were driven to generate mechanical noise.

The experiments are divided into two sets:—

Indirect response – Experiments to establish the feasibility of the table vibration sensor, which is fixed to the table and detects sound propagating through the working table. Analysis of the noise of the robots sampled from the working tables, and analysis of the impulse signals generated by various materials are performed.

Direct response – Experiments to establish feasibility of the gripper vibration sensor, which will be fixed to the gripper and detects sound propagating through the gripped part. Analysis of the noise blocking effect using various compliant materials underneath the PVDF film, and analysis of the signal from direct touching onto the film are performed.

As a result, the frequency characteristics of both noise and signal were obtained, and by signal processing simulation, appropriate methods of filtering were determined for

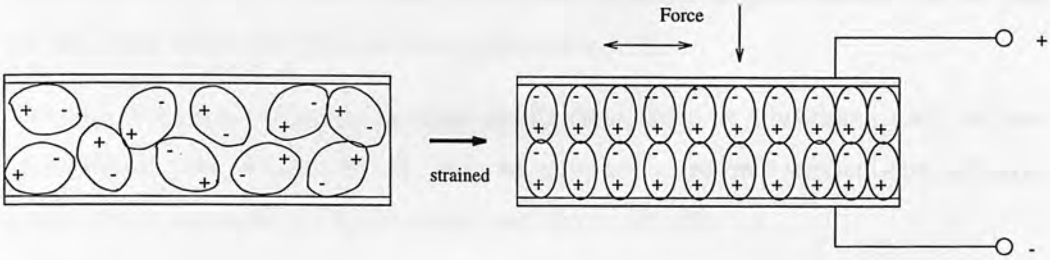


Figure 4.1: Polarisation

acoustic vibration event signature sensors. From the experiences obtained from these experiments, an interesting sensor which detects the change of force applied has been developed. This is described in the next chapter.

4.3 Characteristics of the PVDF Film As a Sensory Transducer

PVDF has been recognised as a potential material for robotic sensory transducers, particularly for tactile purposes [Dario *et al.* 83, McClelland 89]. The piezoelectric effect is electric polarisation produced by mechanical strain in certain crystals, the polarisation being proportional to the amount of mechanical strain. Conversely, an electrical polarisation will induce a mechanical strain in piezoelectric crystals. As piezo material, PVDF films are used throughout the event signature sensor implementation.

The PVDF film is a highly polarising material, which is a long chain of semi-crystalline polymer of repeated units. PVDF remains unpolarised as long as no force is applied. Once an external force has been applied to the film resulting in compressive or tensile strain, the film develops a proportional open circuit voltage (see Figure 4.1). Exposure to a reciprocating force results in a corresponding alternating electrical signal. The frequency response ranges widely from 0.005 Hz to gigahertz. The film is sensitive to vibration, at least 50 times more than common microphones.

The piezo film also acts as a pyroelectric transducer, it can be used to detect thermal radiation. When thermal energy is absorbed, the film expands with increasing temperature. This results in a detectable deformation and a corresponding charge is output.

The reverse effect occurs on cooling of the film. Suitably designed sensors can be used for detecting heat radiation including infrared radiation.

The film has been successful in many applications, such as vibration sensors in general, force sensors, accelerometers, compact switches, ultrasonic applications, infra-red applications, pyro-electric applications, and so on [Pie87].

PVDF transducers are often used as vibration detectors in robotics applications. Son *et al.* employed four PVDF films in one finger tip with different frequency component amplification parameters for the films to detect the instant when the gripped part is just about to slip [Son *et al.* 94]. Shinoda and Ando used a PVDF transducer matrix to characterise and localise any touch directly on an elastic hemisphere body with transducers built in, by detecting the ultrasonic waves produced by touching [Shinoda & Ando 94]. Patterson and Nevill, Jr. used PVDF film to detect object texture by employing exploratory sliding motions [Patterson & Nevill, Jr. 86]. The PVDF extrinsic tactile sensor developed by Dario and De Rossi [Dario & De Rossi 85] could detect object shape, texture, hardness, and temperature [Dario & Buttazzo 87].

PVDF transducers have also been used for matrix tactile array sensors. For instance, Grahn and Astle have built 12 PVDF–based tactile sensor cells where each cell measures the normal force exerted on it [Grahn & Astle 84]. By means of ultrasonic pulse–echo ranging, each sensor cell measures the change in thickness of a compliant, elastic pad whose surface is deformed by the gripped object with a spatial resolution of 0.5 mm.

4.4 Experimental Setup

This section describes the experimental set up. The purpose of the experimental facilities were:—

1. To convert the mechanical vibration to an electrical signal (the electronic interface circuit).
2. To convert this electrical signal to a digital signal (A/D converter, the PC, and the sampling program).
3. To analyse the frequency properties of the signals by simulating signal processing.

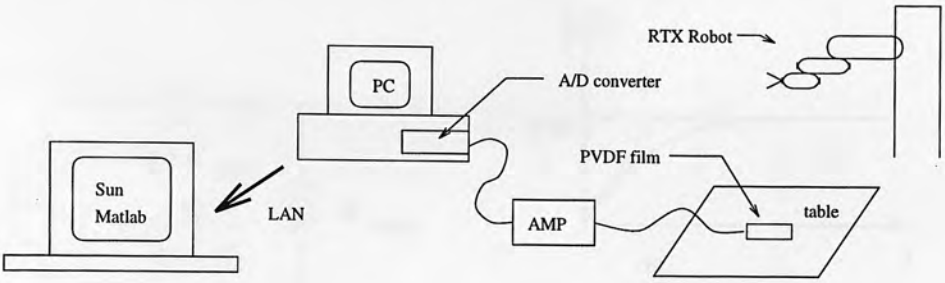


Figure 4.2: Facilities and interconnections

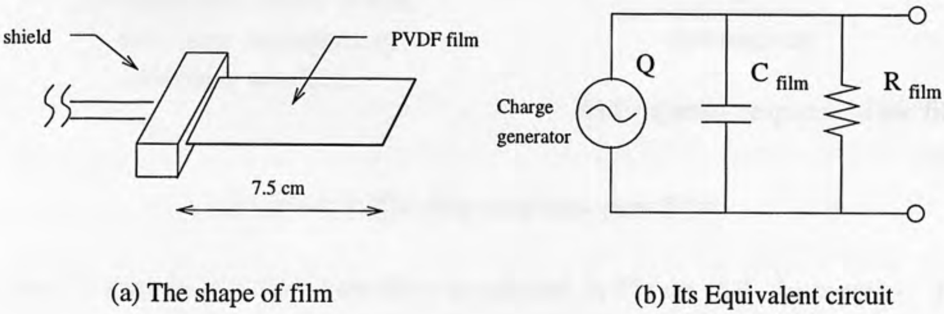


Figure 4.3: The PVDF film and its equivalent circuit of the film

The facilities used were an operational amplifier [Horowitz & Hill 89], an XT PC, a 16 channel A/D converter with one channel D/A converter[3], and the Pro-Matlab software package. The sampling program was written in Borland Turbo Pascal.

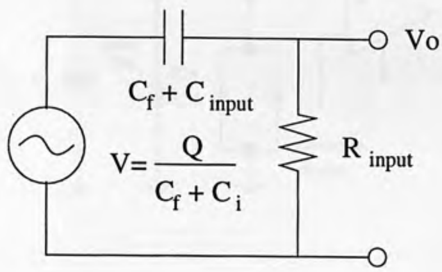
The schematic diagram of the set up is shown in Figure 4.2.

4.4.1 Interfacing PVDF film

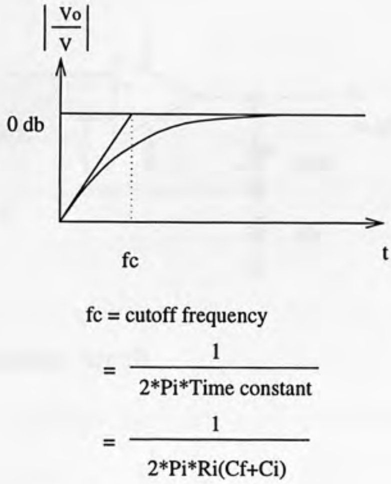
The shape of the film component used and the electronic equivalence of the film is shown in Figure 4.3. The capacitance of the film is proportional to the surface area and inversely proportional to the thickness, while the internal resistance is so high that it can be ignored. The typical capacitance is known to be 379 pF/cm^2 at 10 kHz at a thickness of 28 micro-metres [Pie87].

Together with the input resistance of the interfacing amplifier and the capacitance of

[3] PC ADDA-12 Card, Chipboards Ltd., Almac House, Church Lane, Bisley, Working, Surrey, U.K.



(a) Equivalent circuit of film with input impedence of electronic interface.



(b) Frequency response of the film.

Figure 4.4: The film as a high pass filter

the wire, it comprises a high pass filter as plotted in Figure 4.4. In practice, if the input resistance of the amplifier is too high, the charge generated by the film remains too long, while if the input resistance is too low, much of the signal is lost. Experiment showed an input resistance of 1 MΩ to be a suitable compromise, and this was adopted for the subsequent tests.

The interfacing circuit was designed using LMC660NC quad CMOS op-amps. It was protected by two diodes [Horowitz & Hill 89] against extremely high voltage, produced such as when the film is broken, where the voltage rises up to thousands of volts. Even though the current might be small, this high voltage will break down the high impedance CMOS input amplifier. The input voltage varies always within -0.7 to +0.7 volts even though the voltage from the film is higher because a diode acts as an open circuit only when the voltage applied is less than approximately 0.7 volts[4]. The interface circuit diagram is shown in Figure 4.5. The output swings between -5V to +5V. The amplifiers were built on prototyping bread boards.

[4] However, in practice, the output seldom increases higher than 1V, hence the diodes were removed later.

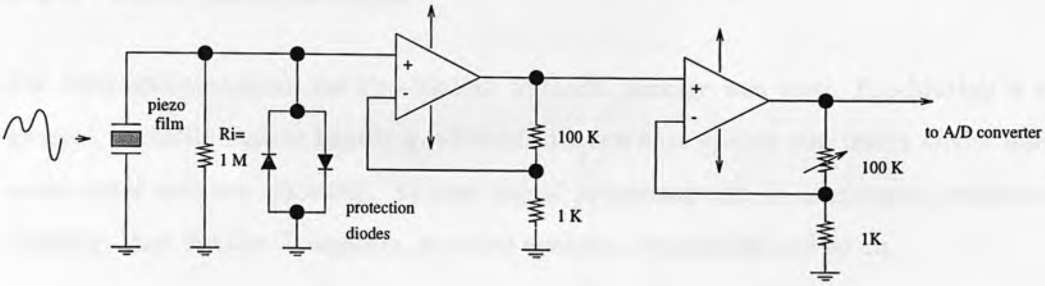


Figure 4.5: The interface circuit

4.4.2 Sampling Data

A 16 channel A/D and one channel D/A converter was mounted on the PC. The signal which comes from the output of the amplifier, is sampled via one of the A/D converters. The A/D converter accepts voltages from 0 to 9 with a resolution of 0.22 mV. The neutral of the amplifier output is regarded as 5V by the A/D converter, hence a voltage band of from -4 to +4V is sampled. The conversion time is 60 microseconds (*i.e.*, the maximum sampling rate for a channel is 16.67 kHz).

The sampling program was mostly written in Borland Turbo Pascal. In order to increase the sampling rate and ensure the regularity of sampling, the bottom level iterating sampling routine was written in Assembler.

The program provides pulse outputs via the D/A converter just before it samples each datum, which allows monitoring of sampling performance. The sampling frequency is 8.33 kHz, which would likely cover the frequencies of acoustic vibration we are interested in[5], although higher frequency vibration can possibly be generated. The sampling is performed for three seconds, hence 24999 integer numbers are collected during one session of sampling. The program is shown in Appendix A.

[5] Shannon's Sampling Theorem: $f_s \geq 2f_h$, states that a signal must be sampled at a rate at least as high as twice the highest frequency in the spectrum to be recovered. The minimum sampling rate required, $2f_h$, at which the signal could theoretically be recovered is called the **Nyquist rate** [Stanley *et al.* 84].

4.4.3 Data Interpretation

For data interpretation, the Pro-Matlab software package was used. Pro-Matlab is a general, versatile matrix handling mathematics software system run under UNIX and some other systems [Mat90]. Various signal processing can be performed including filtering, Fast Fourier Transform, spectral analysis, correlation and so on.

The interpretation of the data was performed by Power Spectral Density (PSD) analysis in order to determine proper filtering parameters. A signal for a certain duration of time can be decomposed into numbers of frequency components. The signal is characterised by the magnitude of the contribution of each frequency component. A PSD shows the magnitude of the contributions of the frequency components of the signal. By analysing the PSD, the frequency characteristics of a signal and noise can be extracted so that a proper signal processing technique can be determined. The Pro-Matlab program, which displays the time domain signal and its PSD is shown in Appendix B.

In this section, the facilities built for picking up and interpreting the signals were described. They are the interfacing electronic circuit, the PC with A/D converter and the sampling program, and the Pro-Matlab code for PSD (Power Spectral Density) analyses.

4.5 Indirect Response

In this section, experiments on determining the indirect response of PVDF films are described. There were three kinds of table used as a robot working table: a polished wooden table ($90 \times 90 \times 2$ cm), a metal table ($90 \times 90 \times 0.5$ cm), and the ADEPT working table which is a massive cast-iron metal table (order of tons). The RTX robot used sits on a metal framed working table and the working surface is covered by a styrofoam board. The experimental working tables were placed on the board. A PVDF film was affixed on each table in turn by double sided tape, and signals, generated and propagated through the tables under various conditions, were sampled and analysed.

The questions we want to answer with these experiments were:

- What are the characteristics of the noise?

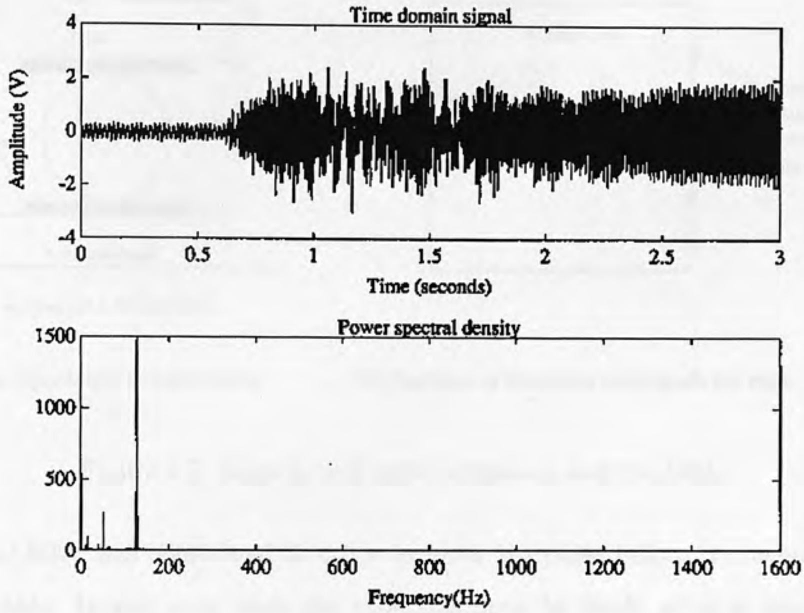


Figure 4.6: Noise from shoulder of the RTX robot

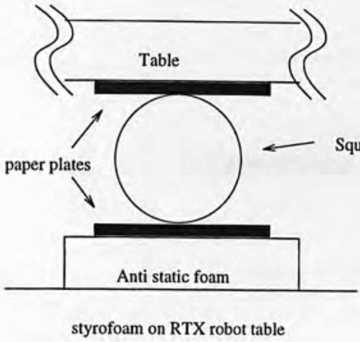
- What is the frequency response of the tables?
- What kind of events can be detected in the signal?

4.5.1 Wooden Table

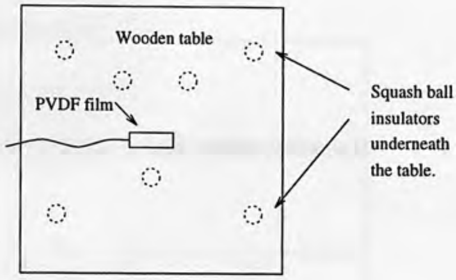
Since both the wooden table and the RTX robot itself sit on the RTX working table, when the robot moves, mechanical noise from the robot is easily propagated through the table to the film. The various kinds of noise which arose from the moving robot with each actuator at different speeds, were sampled and analysed.

The RTX robot is seen to be a very noisy robot. Not only does it give a loud noise when it moves, but it is often mechanically unstable (due to its servo instability), which causes vibration when it tries to stop after moving, or during a slow motion (at speeds approximately less than 0.5 cm/sec). Strong vibration is generated from the shoulder actuator regardless of the speed, especially when it tries to stop. Figure 4.6 shows the noise from the shoulder of the robot when the robot moves up and down.

The vibration is so severe that it could not be easily reduced electrically. Use of a



(a) The squash ball noise insulator.



(b) Positions of insulators underneath the table.

Figure 4.7: Squash ball noise insulators and the table

mechanical filter was considered in order to block the noise before it reaches the new working table. In this case, since the table has mass by itself, what is needed is the addition of appropriate damping and elasticity. A squash ball has these properties.

Seven squash balls with anti-static foam were used to support the wooden table (see Figure 4.7), which seems the optimum number required for effectively protecting the wooden table from noise propagation, while not causing the table to have low resonance frequencies which tends to make the sound persist along the table long after an impact on the table[6]. This will help the sensor to be ready for re-use shortly after an activation.

The result of using the squash balls is encouraging as shown in Figure 4.8[7]. As can be seen in the figure, almost all noise is blocked except frequencies less than 100 Hz that possibly includes electrical noise from the mains power of 50 Hz. This configuration was adopted for experiments in the RTX environment. Another possible good material for use as a mechanical filter could be hi-fi phonograph turntable insulators.

In investigating the general frequency properties of the table, four kinds of mechanical impulses were generated by dropping four different balls. These balls are three different sizes of metal bearings, and a rubber ball. These are selected because metal and rubber

[6] Imagine a string of a guitar, which vibrates at its lowest frequency, and the vibration lasts longest when it is struck without pushing on the string, *i.e.*, undamped

[7] Note that the scaling of y-axis of the PSD graph has changed.

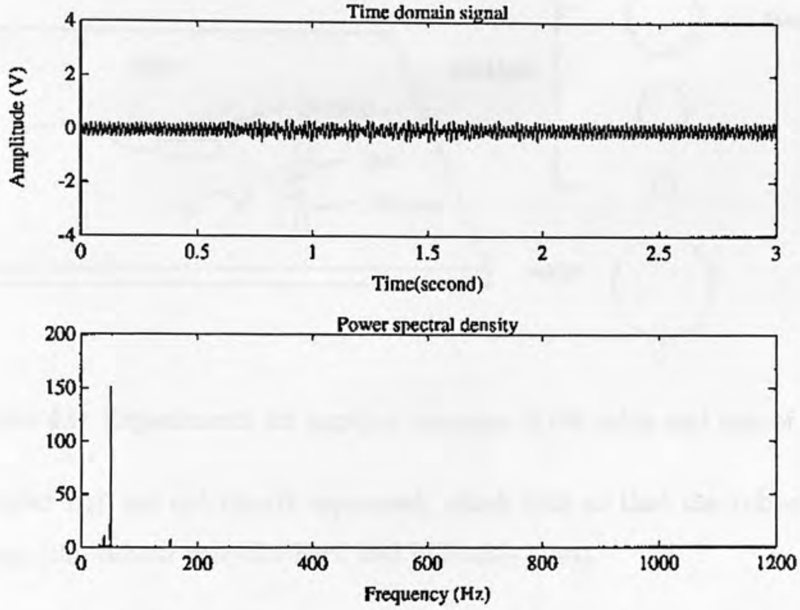


Figure 4.8: Noise reduction results (noise from the shoulder attenuated by the squash ball insulators)

are different materials at both extremes in their stiffness. By evaluating vibration from impact with these materials, the general frequency properties of the table are expected to be evaluated.

The balls were dropped at appropriate heights from a platform. Both single bouncing and free multiple bouncing of each ball were tried on the table. Single bouncing develops one impulse on the table while free bouncing results in multiple impulses on the table until the bouncing terminates. The diagram of the experiment and size of balls are shown in Figure 4.9.

Since a multiple impulse is a set of repeating single impulses, the density of the signal during a sampling session (3 sec.) is higher than a single impulse, which results in a clearer look at the frequency distribution of the signal. Multiple impulse responses and their PSD distributions from the medium size steel ball and rubber ball[8] are shown in Figure 4.10. In the top graph of the figure, each bouncing of the metal ball is clearly separated as a peak followed by vanishing vibration. However, the bouncings

[8] Since they are of different kinds.

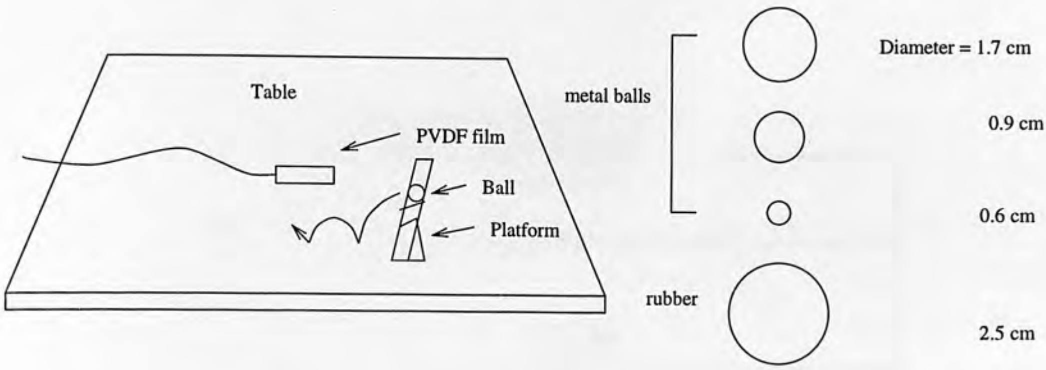


Figure 4.9: Experiments for impulse response of the table and size of balls

of the rubber ball are not clearly separated, which tells us that the rubber balls are not bouncy (the rubber was of a hard and inflexible kind).

Although the two balls are different in their kinds of material (one is metal and the other is rubber), the frequency response is similar except that the metal ball emphasises rather higher frequency components around 1000 Hz. These plots of frequency response provide a good indication of the vibration properties on impact of the wooden table.

In order to observe the waveform and its frequency distribution of signals in general, experiments were performed with various materials and in various situations, such as dropping a 2.5 cm side wooden cube, knocking over a tower of five wooden cubes, knocking over a pen at various positions on the table, and so on. All these experiments proved that the frequency response of the table is invariant with respect to the material impacting and the position of the impact, and strong responses can be seen from 50 to 1000 Hz. Typical examples of these are shown in Appendix C.

It is important to observe how the sensor can be sensitive to an event signal while being subjected to noise. Although the table is insulated by squash balls, the noise from the moving robot is still detected at low frequencies at around 50 to 100 Hz. This limits the sensitivity of the sensor to the signal.

Software simulation of signal processing, *i.e.* filtering, was carried out in order to find the best way of processing when building actual sensors. A simulation of a Butterworth 4^{th} order high pass filter with variable cut-off frequency written in Pro-Matlab (see

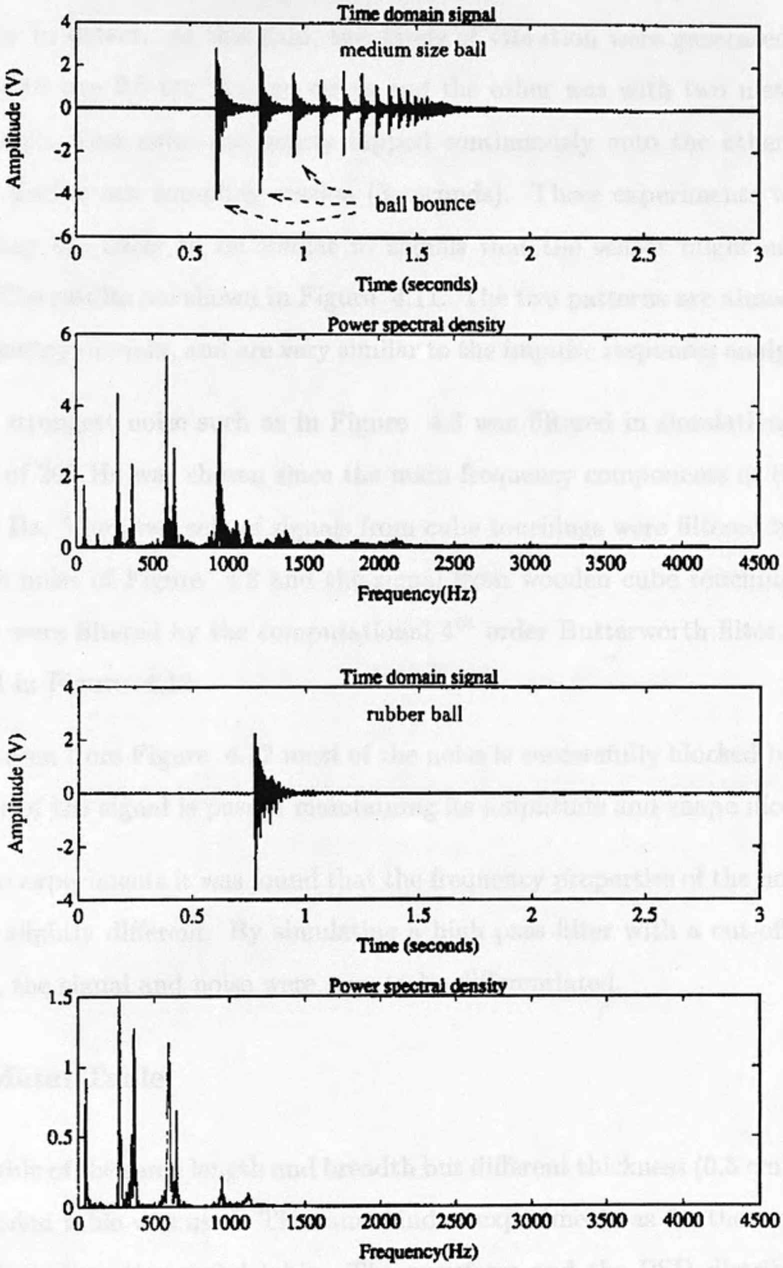


Figure 4.10: Multiple impulse responses and their PSD distributions of two different kinds of balls

Appendix B) was used.

The gain of the amplifier was set such that a signal such as dropping a 5 mm piece of plain solder wire (less than 1 gram) from a height of 1 cm, appears strong enough for a sensor to detect. At this gain, two kinds of vibration were generated manually. One was with two 2.5 cm wooden cubes and the other was with two metal cubes of the same size. One cube was gently tapped continuously onto the other sitting on the table, during one sampling session (3 seconds). These experiments were chosen because they are likely to be similar to signals that the sensor might encounter in practice. The results are shown in Figure 4.11. The two patterns are almost the same in the frequency domain, and are very similar to the impulse responses analysed before.

First, the strongest noise such as in Figure 4.8 was filtered in simulation. A cut-off frequency of 200 Hz was chosen since the main frequency components of the noise lie below 130 Hz. Then two sets of signals from cube touchings were filtered by the same filter. The noise of Figure 4.8 and the signal from wooden cube touching in Figure 4.11 (top) were filtered by the computational 4^{th} order Butterworth filter. These are contrasted in Figure 4.12.

As can be seen from Figure 4.12 most of the noise is successfully blocked by the filter, while most of the signal is passed maintaining its amplitude and shape nicely.

From these experiments it was found that the frequency properties of the noise and the signal are slightly different. By simulating a high pass filter with a cut-off frequency of 200 Hz, the signal and noise were seen to be differentiated.

4.5.2 Metal Table

A metal table of the same length and breadth but different thickness (0.5 cm) compared to the wooden table was used. The same kind of experiments as for the wooden table were conducted on this metal table. The waveform and the PSD distribution from multiple bouncing of the medium size metal ball is shown in Figure 4.13 (compare this with Figure 4.10). The table is also insulated and supported by squash balls. As the results of the experiments show, the metal table is different to the wooden table in:—

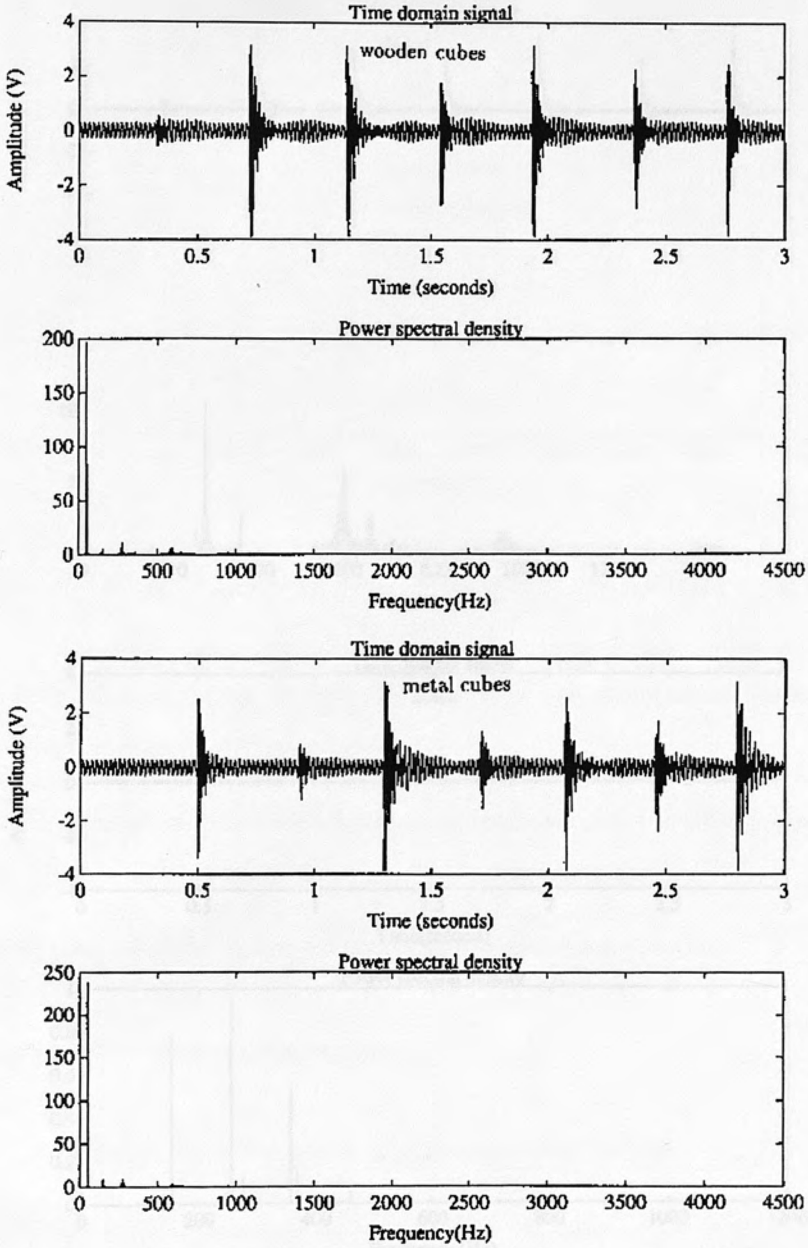


Figure 4.11: Signal from cube touchings

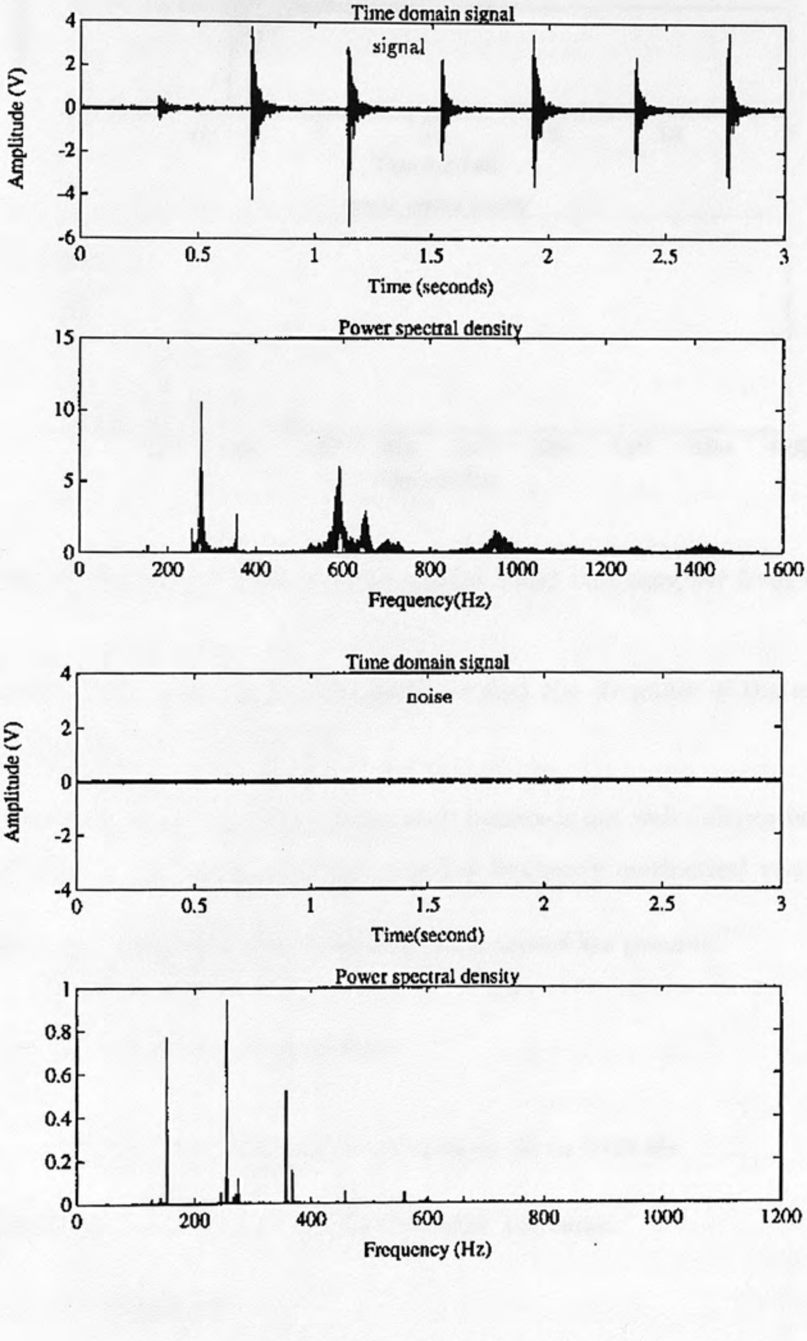


Figure 4.12: Filtered signal and noise

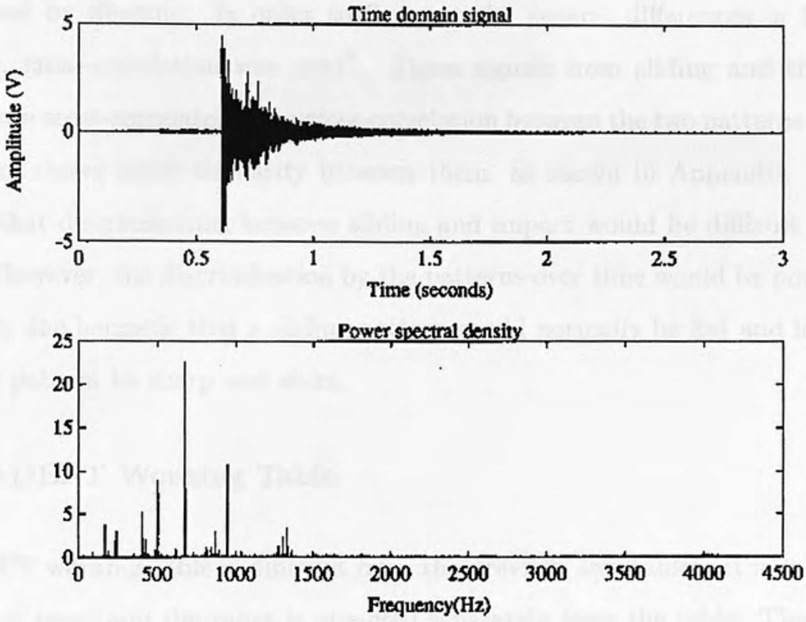


Figure 4.13: Waveform and PSD of medium size metal ball sampled from the metal table

- Narrower PSD. A reason for this might be that the structure of the material is more regular than that of wood.
- Smoothed shape in signal wave form; each bounce is not well differentiated. This metal table tends to respond slightly to low frequency mechanical vibration.
- Nevertheless, slightly higher frequency components are present.

and the same as the wooden table in that:—

- The major frequency components lie between 50 to 1000 Hz.
- The frequency distribution of noise is almost the same.
- Sensitivity is almost the same.

Hence, it is expected that the facilities used for the wooden table, including the filter with a cut-off frequency of 200 Hz, can also be used for the metal table.

In addition to the above, experiments have been performed to discriminate sliding from impact by filtering. In order to find out the general differences in frequency properties, cross-correlation was used[9]. Three signals from sliding and three from impacts were cross-correlated. The cross-correlation between the two patterns of sliding and impact shows much similarity between them, as shown in Appendix D. This indicates that discriminating between sliding and impact would be difficult by linear filtering. However, the discrimination by the patterns over time would be possible, for example by the heuristic that a sliding pattern would normally be flat and long while an impact pattern be sharp and short.

4.5.3 ADEPT Working Table

The ADEPT working table is different from the previous two tables. It is massive (in the order of tons) and the robot is mounted separately from the table. The ADEPT robot is much quieter than the RTX robot, and this noise can barely vibrate the massive metal table. However, the robot working space is surrounded by much electrical equipment such as workstations, the robot controller and an air compressor. All these generate both significant mechanical and/or electrical noise.

The noise in the ADEPT working table was sampled in the worst case when all the noise sources were running simultaneously (See Figure 4.14). It is significant that first, on the time domain signal side, the amplitude of noise does not exceed 1 volt, second on the PSD side, 50 Hz is the dominant frequency and almost all others are under 1000 Hz.

More signals were collected. Among data from various experiments, continuous gentle touching of the ADEPT table by a wooden part is shown in Figure 4.15. This gentle touch is the dropping of one cube (12 gram) from 1 mm height onto another sitting on the table. The major frequency components are from 1300 to 2000 Hz, which is different to the noise in Figure 4.14, and the signal is much stronger than the noise. This shows that the table tends to vibrate at higher frequencies than the wooden and the metal table. It seems possible to extract a useful signal by filtering out noise using a high pass filter with its cut-off frequency around 1200 Hz.

---

[9] Cross-correlation between two signal shows their similarity in terms of frequency characteristic.

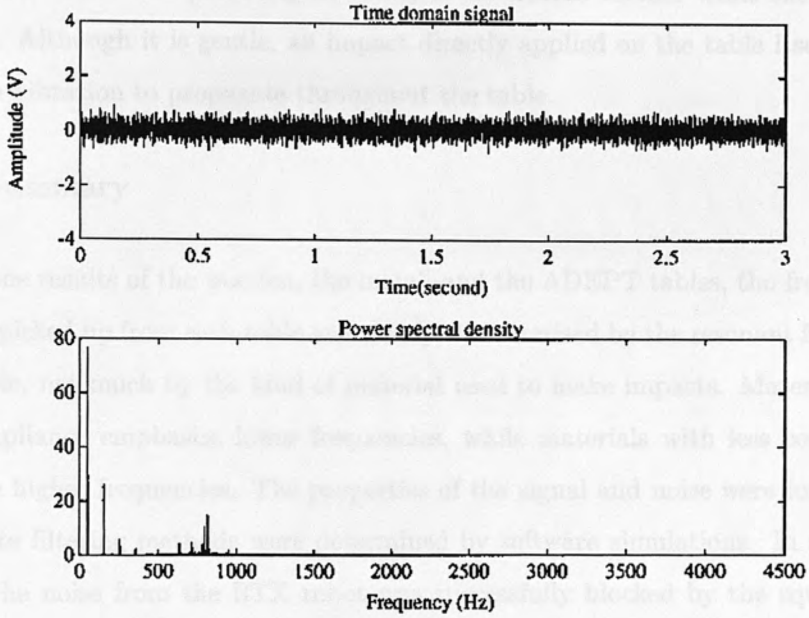


Figure 4.14: Noise from ADEPT robot working table

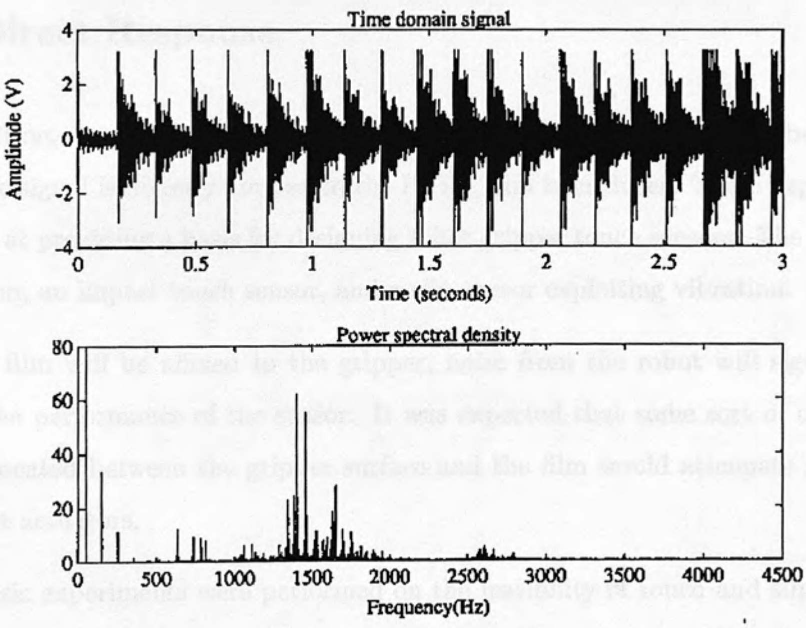


Figure 4.15: Continuous wooden cube touching on the ADEPT working table

The result of the simulation using a computational filter in Pro-Matlab is shown in Figure 4.16. The noise (from Figure 4.14) is attenuated further while the signal is preserved. Although it is gentle, an impact directly applied on the table itself causes detectable vibration to propagate throughout the table.

4.5.4 Summary

From all the results of the wooden, the metal, and the ADEPT tables, the frequencies of signals picked up from each table are mainly characterised by the resonant frequency of the table, not much by the kind of material used to make impacts. Materials with more compliance emphasise lower frequencies, while materials with less compliance emphasise higher frequencies. The properties of the signal and noise were found, and appropriate filtering methods were determined by software simulations. In addition, much of the noise from the RTX robot was successfully blocked by the squash ball noise insulators.

These experiments show that by using appropriate filters, table vibration sensors for each kind of table can be made to retrieve useful event information in assembly.

4.6 Direct Response

In this section, experiments on the response of PVDF to contact are described, where an impulse signal is directly applied to the PVDF film transducer. These experiments are aimed at providing a basis for designing robot gripper touch sensors. The expected sensors were an impact touch sensor, and a slip sensor exploiting vibration.

Since the film will be affixed to the gripper, noise from the robot will significantly degrade the performance of the sensor. It was expected that some sort of compliant material located between the gripper surface and the film would attenuate the noise from robot actuators.

Hence, basic experiments were performed on the feasibility of touch and slip sensors. These consisted of observing the vibration propagating characteristics of compliant materials, and the characteristics of vibration from directly touching the film.

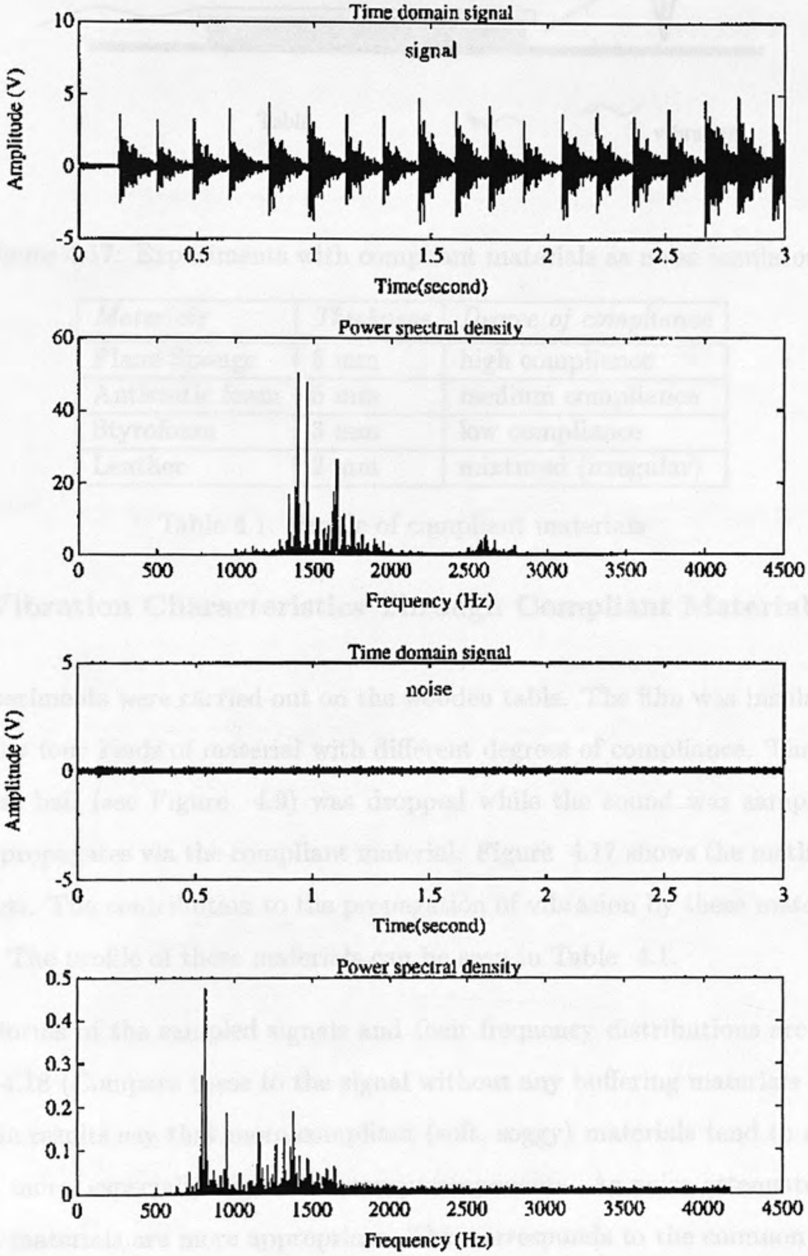


Figure 4.16: Result of filtering simulation of signal (top) and noise (bottom) from the ADEPT working table

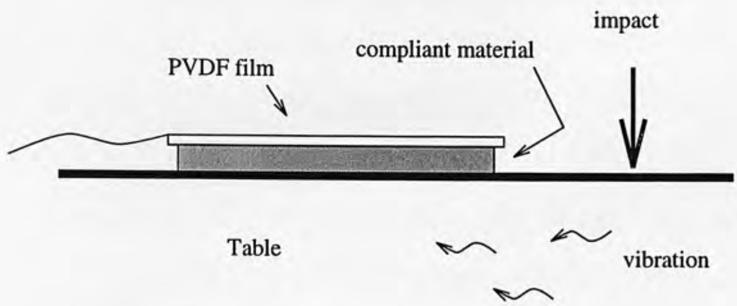


Figure 4.17: Experiments with compliant materials as noise insulators

Materials	*Thickness*	*Degree of compliance*
Plane Sponge	5 mm	high compliance
Antistatic foam	5 mm	medium compliance
Styrofoam	3 mm	low compliance
Leather	2 mm	mixtured (irregular)

Table 4.1: Profile of compliant materials

4.6.1 Vibration Characteristics Through Compliant Materials

These experiments were carried out on the wooden table. The film was insulated from the table by four kinds of material with different degrees of compliance. The medium sized metal ball (see Figure 4.9) was dropped while the sound was sampled. The vibration propagates via the compliant material. Figure 4.17 shows the method of the experiments. The contribution to the propagation of vibration by these materials was observed. The profile of these materials can be seen in Table 4.1.

The waveforms of the sampled signals and their frequency distributions are included in Figure 4.18 (Compare these to the signal without any buffering materials in Figure 4.10.). The results say that more compliant (soft, soggy) materials tend to attenuate the signal more, especially higher frequency components. As noise attenuators, more compliant materials are more appropriate. This corresponds to the common intuition of the relationship between property of materials and the propagation of vibration.

Since the force of the gripper jaws is strong enough to squeeze soft material such as plain sponge, this sort of material is impractical. Leather is a material which has mixed properties of stiffness and softness. The irregular structure of material such as

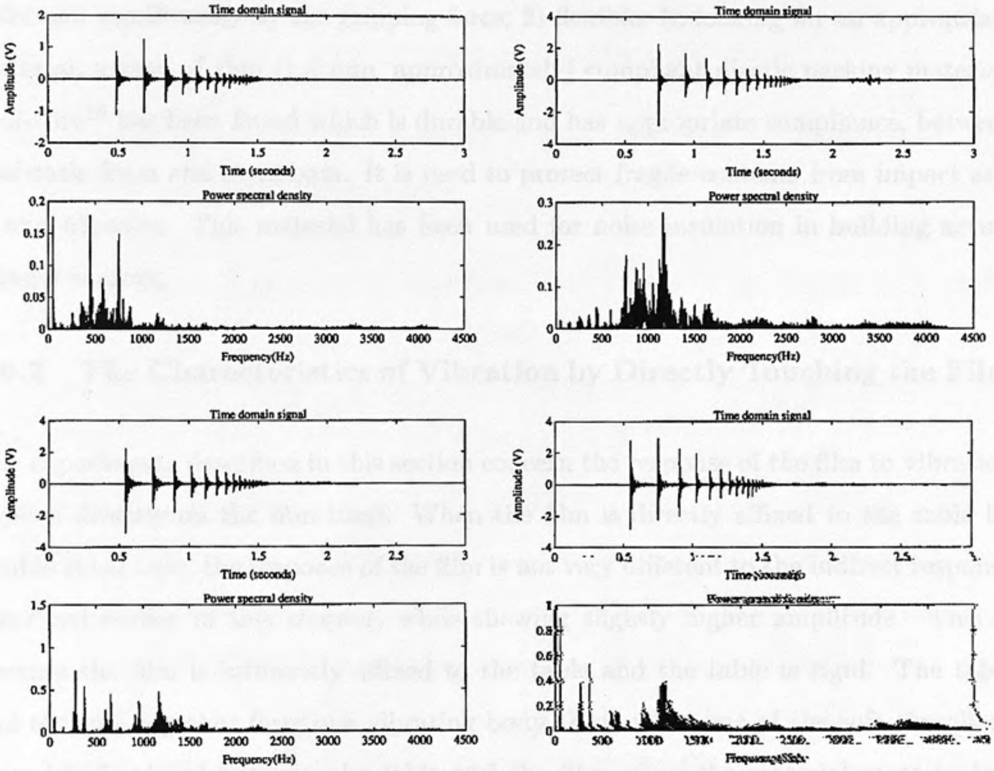


Figure 4.18: Signals collected via compliant materials from the bouncing of the medium size metal ball on the wooden table

(Compliant material used are: top left: plane sponge; top right: antistatic foam; bottom left: styrofoam; bottom right: leather.)

leather was expected to show good properties as an insulator: *not to be deformed while blocking noise.* However, as can be seen from the results, leather passes more noise than antistatic foam and passes slightly less noise than styrofoam, which is not as good as expected. This suggests that leather would be too hard for our application.

Based on observations from the experiments using various kinds of materials, the desirable characteristics of an insulating material were empirically outlined as: 1) not to be deformed significantly by the gripping force; 2) flexible. In looking for an appropriate material, a kind of thin (1.5 mm, approximately) compliant plastic packing material, Cell-Aire[10] has been found which is durable and has appropriate compliance, between antistatic foam and styrofoam. It is used to protect fragile contents from impact and is non–abrasive. This material has been used for noise insulation in building actual gripper sensors.

4.6.2 The Characteristics of Vibration by Directly Touching the Film

The experiments described in this section concern the response of the film to vibration applied directly on the film itself. When the film is directly affixed to the table by double sided tape, the response of the film is not very different to the indirect response described earlier in this chapter, while showing slightly higher amplitude. This is because the film is intimately affixed to the table and the table is rigid. The table and the film together form one vibrating body. But, when one of the soft compliant materials is placed between the table and the film, since the material more or less allows the film to vibrate by itself, the response largely depends on the material of the film itself and the surface finish.

Vibrations generated under various situations are sampled and analysed:

- vibrations from dropping balls, knocking over pens on the film, sliding, etc., using different compliant materials, as in Table 4.1;
- using the plain sponge as a compliant material, vibrations from sliding of wooden and metal cubes, with sand paper[11] as the surface of the film;

[10] Sealed Air Ltd., Telford Way, Ketting, Northants NN16 8UN, U.K.

[11] Sand paper was used to generate intensive noise.

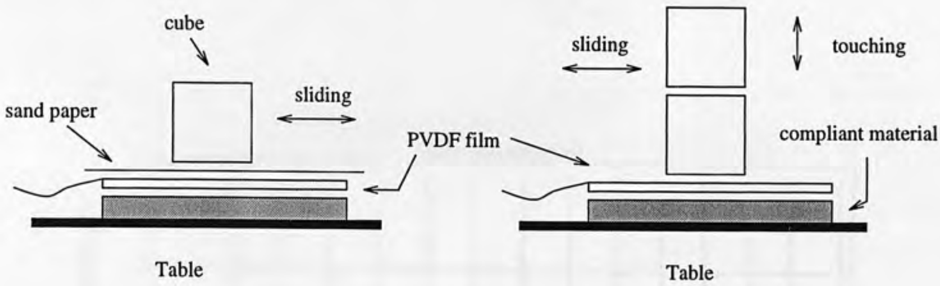


Figure 4.19: Experiments for touching and sliding

- vibrations from touching and sliding of pairs of wooden and metal cubes, with one of a pair sitting on the sensor.

The method of these experiments is illustrated in Figure 4.19. Figure 4.20 shows the signals and their calculated PSD distributions from touching of two wooden cubes sitting on the film (top) and a wooden cube sliding on the sand paper surface (bottom). Some of the other results are included in Appendix E.

From the experimental results, first it was found that the signals are not normally dependent on the kinds of the buffering material. The compliant materials have the effect of mechanically isolating the film. Secondly, it can be seen that the frequencies of the major components in the frequency domain of all these signals are lower than those signals from the indirect response experiments. However, although it cannot be seen from the PSD distributions, very high frequency components can exist for both cases of impact and sliding, which are not dealt with in this experiment.

Sand paper provides much friction, but it wears parts. Sliding experiments have been performed to find out a good material as the surface of the film, which does not wear parts much but still creates significant friction. The materials tested were some kinds of paper, vinyl tape, some kinds of sponge and a pot scourer. Among these, a thin cut pot scourer was chosen to be the skin of the sensors because it provides good friction while not wearing parts much.

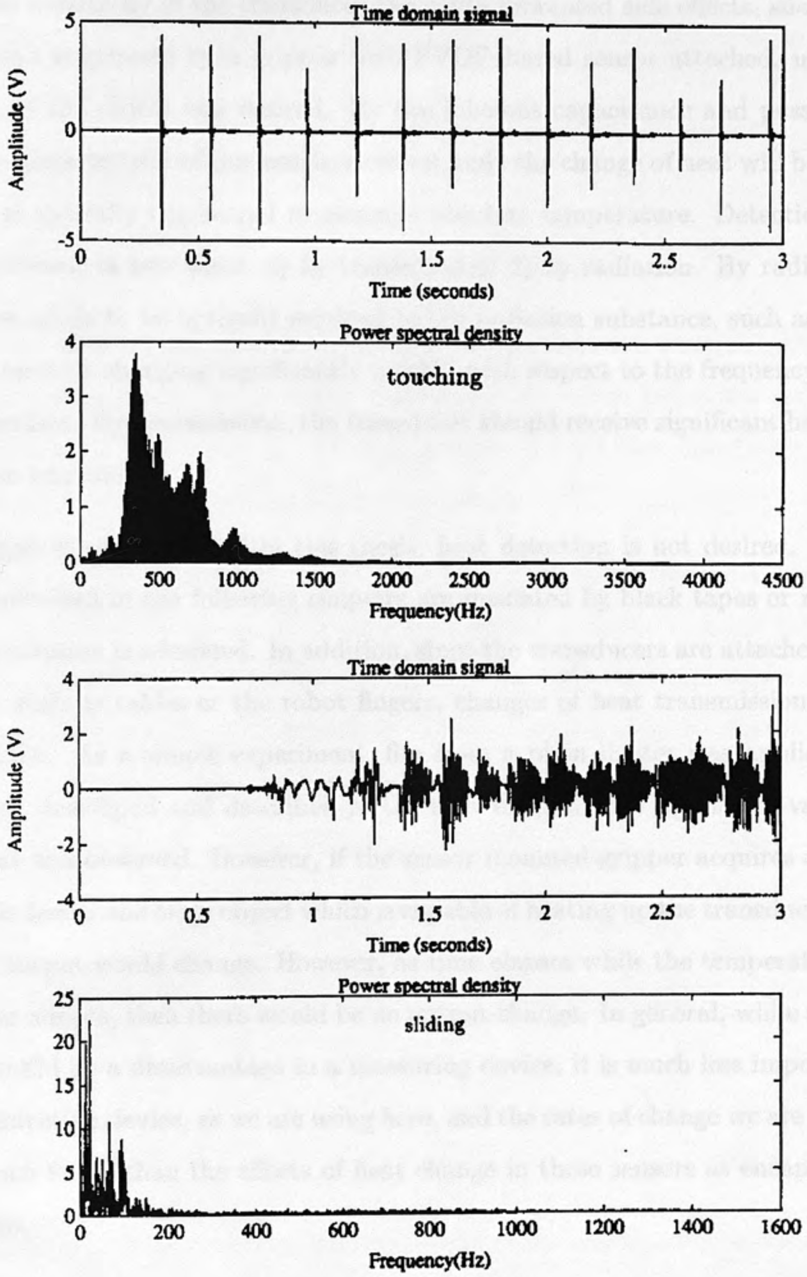


Figure 4.20: Cube touching and sliding on the film

4.7 A Note on Pyroelectric Effect

As mentioned earlier in this chapter, PVDF is sensitive to heat. For robotic applications, heat sensitivity of the transducer can cause unwanted side effects, such as when a hot object is gripped by a gripper with PVDF–based sensor attached, unless heat detection of the object was desired. By the inherent capacitance and possibly some high pass characteristic of the interface circuit, only the change of heat will be detected unless it is specially engineered to measure absolute temperature. Detection of heat can be achieved in two ways: 1) by transmission; 2) by radiation. By radiation, the transducer needs to be optically exposed to the radiation substance, such as infrared, with the amount changing significantly quickly with respect to the frequency response of the interface. By transmission, the transducer should receive significant heat change in order to respond.

In the applications described in this thesis, heat detection is not desired. Since the sensors described in the following chapters are insulated by black tapes or rubber, no infrared radiation is admitted. In addition, since the transducers are attached to other materials such as tables or the robot fingers, changes of heat transmission would be slowed down. As a simple experiment, fire from a plain lighter was applied around the sensors developed and described in the next chapter. No significant variation of the output was observed. However, if the sensor mounted gripper acquires a hot, and reasonably heavy and large object which is capable of heating up the transducer quickly, then the output would change. However, as time elapses while the temperature of the transducer adapts, then there would be no output change. In general, while sensitivity to heat would be a disadvantage in a measuring device, it is much less important in a change–detection device, as we are using here, and the rates of change we are interested in are much faster than the effects of heat change in these sensors as encapsulated in our fingers.

4.8 Summary of the Experimental Results

The knowledge obtained from the experiments is summarised below. These facts turned out to contribute to the implementations of the sensors described in the next chapter.

Interfacing

- The input resistance of the interface amplifier and the capacitance of the film compose a high pass filter. 1 MΩ is appropriate for detecting vibrations from impact and sliding. If a higher resistance is used, low frequency components are more emphasised[12].

Indirect Response

- Noise from the RTX robot is strong at 50 Hz and 120 Hz.
- For the wooden and metal tables, although the noise is filtered by mechanical filters, there is still noise under around 130 Hz which limits the sensitivity of the sensor. By Pro-Matlab filtering analysis, a high pass filter with its cut-off frequency at 200 Hz is seen to be appropriate.
- Sliding can be thought of as an infinite number of continuous impacts. This makes the discrimination of impact from sliding difficult. It is the same case as for indirect response. Sliding and impact occurring on the table are not different in terms of frequency distribution. They could possibly be discriminated by pattern in time: *impact is short while sliding is long.*
- In the case of the ADEPT table, it was found that since the table is massive and mechanical noise comes indirectly *i.e.* via air, the mechanical noise is not a crucial problem, although electrical noise is considerable. Direct touching on the table can still be detected, for which purpose a filter with a cutoff frequency of 1200 Hz is desirable.

Direct Response

- For material to be used as noise insulator, a compliant material is desirable. It is preferable that the material does not deform considerably with typical jaw force.
- Both sliding and impact tend to stimulate the film at comparatively high frequencies. Although noise from the RTX robot is expected to be severe, since

[12] This is used to interface the force sensors which are described in the next chapter.

the dominant frequency components of this noise are of low frequency, it seems possible to build a gripper touch sensor that works.

In this chapter, the **Direct response** and **Indirect response** of three kinds of table, and the characteristics of the noise were investigated. The facilities built were utilised to pick up and analyse the signal. Many of characteristics of the noise and signal were found, which is useful in designing vibration event signature sensors. The actual implementation of the sensors is described in the next chapter.

Chapter 5

Implementation and Test of Event Sensors

This chapter first describes the implementation of event signature sensors based on the experiments described in the previous chapter. This chapter then describes the application of the sensors developed to a benchmark assembly as a test criterion.

The sensors implemented are:

- A table touch vibration sensor;
- A gripper touch and slip sensor;
- A force sensor exploiting deformation.

All these sensors are made of PVDF films and exploit the vibration characteristics of the film. The Table Sensor, and the gripper touch and slip sensor concern audible higher frequency signals, while the force sensor, which exploits deformation, concerns lower frequency signals (less than 10 Hz).

5.1 Table Sensor

The Table Sensor detects sound propagated through the table it is attached to. In the filtering simulations for the wooden and metal tables, the software 4th order Butterworth filters with a cut-off frequency of 200 Hz to 300 Hz both blocked noise and

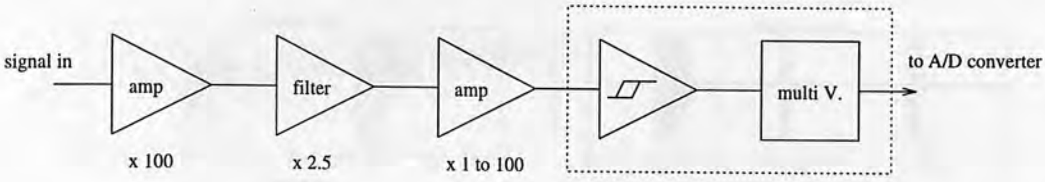


Figure 5.1: Signal processing hardware for the Table Sensor

preserved the signal of interest. For the ADEPT metal table a filter with a cut-off frequency of 1200 Hz was successful. According to the results of the filtering simulations, electronic 4^{th} order Butterworth high pass filters were designed and built. A filter with a cut-off frequency of 220 Hz was built for the wooden table and the metal table, while another filter with a cut-off frequency of 1200 Hz was built for the ADEPT's working table. The circuit diagram is shown in Appendix F.

The signal is amplified, filtered, and then it is further amplified. This output signal is then fed to the A/D converter of the PC. The PC was programmed to threshold the incoming signal in order to detect events.

However, since any event which occurs and finishes between any two consecutive samples is neglected, before the signal is fed to the A/D converter, a facility is needed to hold the output for a while. For this purpose, a monostable multivibrator with Schmitt triggered input was used with a time constant of approximately 1 second. Schmitt triggers electrically threshold the signal, and multivibrators, once excited, hold any high state for the period of time specified by the time constant. In this case, when the input exceeds 1.2 V, the Schmitt trigger is triggered, then the multivibrator holds this state for one second. The PC next thresholds the output of the multivibrator which stays high for one second when there is any event. Hence, the PC will not neglect any event between samples. This is illustrated in Figure 5.1.

The typical competence of the filtering is shown in Figure 5.2 in the case of the wooden table: 1) top left: unfiltered signal; 2) top right: filtered signal; 3) bottom left: unfiltered noise from the robot shoulder; 4) bottom right: filtered noise. The signals were obtained by dropping the smallest ball from a height of 1 cm onto the table. The filter works as expected. It filters out significant noise components while it passes important

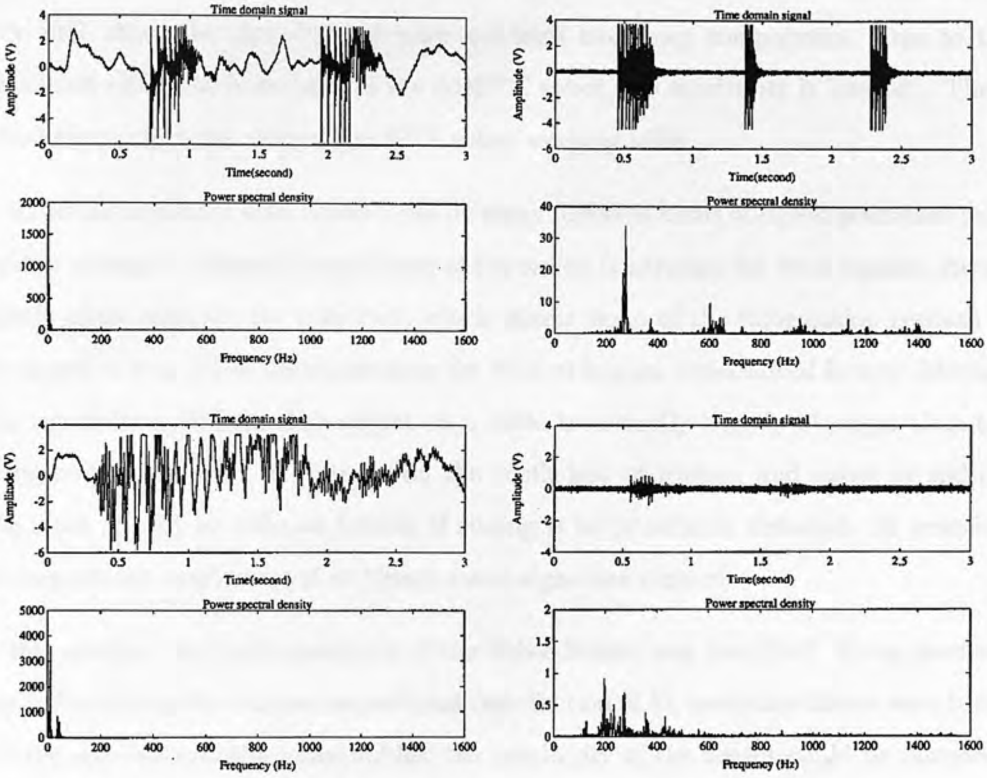


Figure 5.2: The competence of the electronic filter used for the Table Sensor for the wooden table

signal components. In practice a very gentle touch like dropping a small piece of solder wire (1 cm, less than 1 gram) from 2 mm above the table is detected.

The metal table needs some mass, like three or four books (870 grams each) on it, in order to help the noise insulator to work and not to vibrate at frequencies lower than about 50 Hz. Mass slightly deteriorates the sensitivity of the sensor and changes the frequency response of the table, but it attenuates lower frequency noise more than higher frequency signals. Hence, some mass helps the performance of the sensor, although it needs a slightly higher gain of the amplifier. The sensor for the metal table shows almost the same behaviours as for the wooden table.

For the working table of the ADEPT robot, an electronic high pass filter with a cut-off frequency at 1200 Hz was used. A higher amplifier gain is required due to its heavy mass. Because the signal is passed through the filter with a higher cut-off frequency, a soft touching generated by a slightly less rigid material does not activate the sensor very well, since the signal would have less high frequency components. Due to the significant electrical noise around the ADEPT robot, the sensitivity is limited[1]. There is less electrical noise around the RTX robot working table.

In an actual assembly task, there could be many different kinds of signal generated with various strengths. Since the sensitivity of the sensor is adjusted for weak signals, strong signals often saturate the amplifier, which means some of the information content of the signal is lost. Thus discriminating the kind of impact experienced is very difficult. The signal from sliding of an object on a table is normally slightly stronger than the background noise, but it depends on the coefficient of friction and speed of sliding. The noise should be reduced further if sliding is to be reliably detected. In practice, the sensors are implemented as binary event signature sensors.

In this section, the implementation of the Table Sensor was described. Using parameters indicated by the indirect experiment (see Section 4.5), hardware filters were built. For the wooden and the metal tables, the sensitivity of the sensor could be increased sufficiently to detect significant events, but for the ADEPT robot working table, due to the electrical noise, the sensor performance is limited.

5.2 Gripper Touch Sensor (Noise–Cancelling Sensor)

In this section, the implementation of a gripper touch sensor is described. This gripper touch sensor (later called a **Noise–cancelling Sensor**) was implemented from the knowledge obtained in the basic experiments on the direct response of PVDF films (see Section 4.6).

The mechanical noise which comes directly from an RTX robot is significant. Although filters are used, it is hard to implement a vibration sensor on the robot gripper which exploits audible vibration, even though some compliant materials are used to attenuate the RTX noise (refer to Section 4.6.1). Hence, a kind of noise cancelling technique was adopted to reduce the effect of this noise.

---

[1] Dropping of a metal cube (43 grams) from approximately 5 mm height is close to the smallest detectable event.

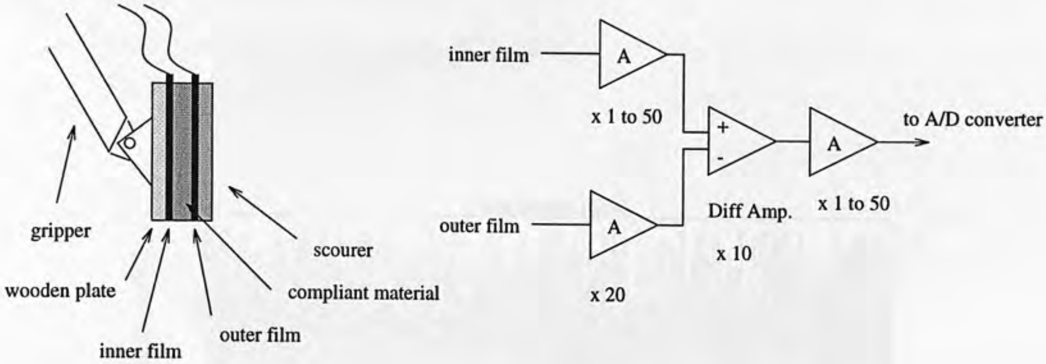


Figure 5.3: The Noise–cancelling Sensor and the simplified circuit diagram of the differential amplifier

A differential amplifier is used for this purpose. Two films are isolated by the compliant material described in Section 4.6.1, on page 82. One film is fixed to the end of the gripper and the other touches an object as can be seen in Figure 5.3. The idea is that the inner film detects the noise more strongly than the outer film, and the outer film detects the signal more strongly than the inner film. Hence, when the amplitude of the noise from the inner film is matched with the noise from the outer film and these are subtracted, a clearer signal can be obtained. This sensor is named **Noise–cancelling Sensor** because it cancels the noise out in order to obtain a clearer signal.

Figure 5.4 shows how the noise is cancelled. Figure 5.4(a) shows the noise from the shoulder of the robot which is sampled by the outer film only. This is significantly reduced by subtracting noise sampled by the inner film as shown in Figure 5.4(b)[2]. The filtering further attenuates the noise as can be seen in Figure 5.4(c). However, the signal which is generated by touching gently the gripped wooden cube with another one is dominant in contrast to the noise as shown in Figure 5.4(d). Even though the noise from the robot shoulder is still strong, it is successfully attenuated by a differential amplifier and a high pass filter, a 4th order Butterworth filter with cut-off frequency at 1200 Hz[3].

[2] Note the different scales of amplitude in the graphs.

[3] Although the noise is mostly less than a few hundred Hz, this filter was used because the low frequency noise was expected to be attenuated as much as possible, but since the signal from touching and slipping would contain higher frequency components, they can still be detected.

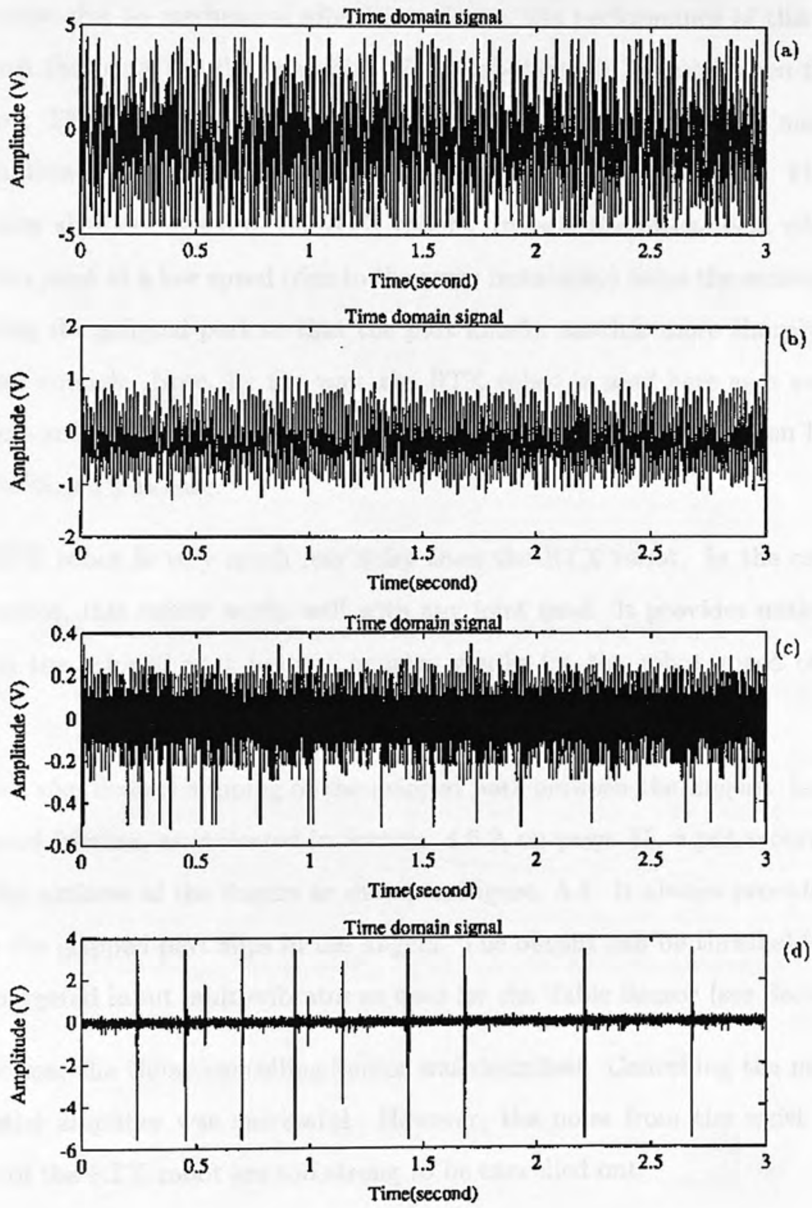


Figure 5.4: The verification of the Noise–cancelling Sensor

However, since the main noise sources, the motors for the wrist and the grippers, are close to the sensor, this noise still greatly influences the sensor performance. The noise is strong and of such high frequency due to the motors that it cannot be attenuated to a satisfactory level using these facilities. Even when they move slowly, they still generate noise due to mechanical vibration. Hence, the performance of this sensor is limited such that only the shoulder joint of the robot should be used when the sensor is activated. This is a fair restriction, since the shoulder provides vertical motion, and vertical motion is the most common part–placing motion in assembly. This sensor works at any shoulder speed of the RTX robots. In fact, the mechanical vibration of the shoulder joint at a low speed (due to the servo instability) helps the sensing process by vibrating the gripped part so that the part knocks another more sharply when it comes close enough. Note, by the way, the RTX robot is used here as a worst case: it is a cheap robot which is notorious for its low–frequency vibration, often likened to robotic Parkinson's disease.

The ADEPT robot is very much less noisy than the RTX robot. In the case of the ADEPT robot, this sensor works well with any joint used. It provides usable signals even when the gripped part touches another gently (at the robot speed of 0.7~0.8 cm/s).

This sensor also detects slipping of the gripped part between the fingers. In order to provide good friction, as indicated in Section 4.6.2, on page 85, a pot scourer is used to form the surfaces of the fingers as shown in Figure 5.3. It always provides signals whenever the gripped part slips in the fingers. The output can be thresholded by the Schmitt triggered input multi-vibrator as used for the Table Sensor (see Section 5.1).

In this section, the Noise–cancelling Sensor was described. Cancelling the noise using a differential amplifier was successful. However, the noise from the wrist and yaw actuators of the RTX robot are too strong to be cancelled out.

5.3 Force Sensor Exploiting Deformation

In this section, a force sensor which exploits deformation of its body is described. This sensor has a low cost but performs well at detecting the change of force applied to

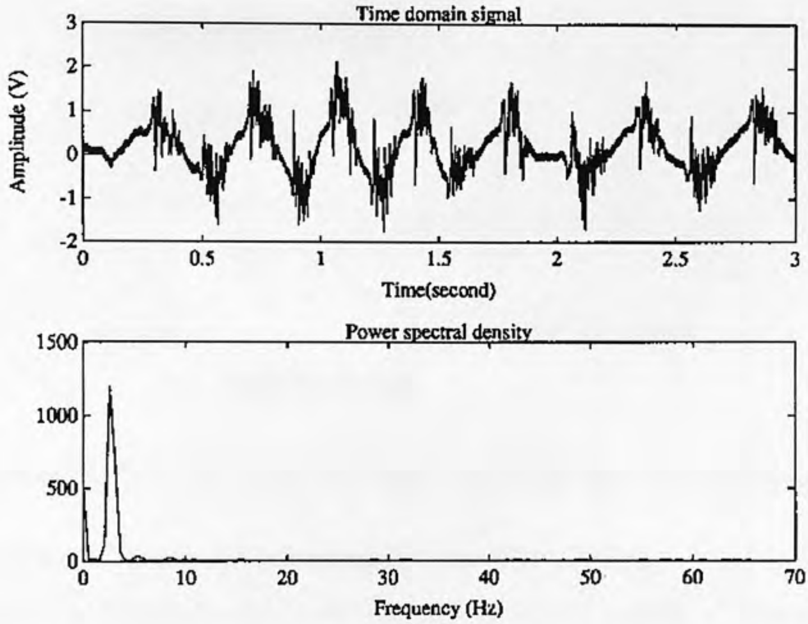


Figure 5.5: The result of a sliding experiment

its body. It is a vibration sensor but is robust to noise, since it takes low frequency components of a signal into account.

During the sliding experiments with the Noise-cancelling Sensor mounted on an RTX robot finger, a signal was sampled when a wooden cube was slipping up and down continuously between the fingers. The robot was gripping the cube at its maximum force. The sampled signal and its PSD for low frequencies are shown in Figure 5.5. In the time domain signal, some cycles of global change (low frequency profile) in amplitude can be seen. Either from a valley to the next mountain, or from a mountain to the next valley, corresponds to slipping from one end to the other of the surface of the sensor, hence there are 14 slippings in the plot. At each beginning of a slip, significant high frequency signals are observed. These vibrations are believed to be caused by the stiction of the cube and the sensor surface.

The global change in this plot is caused by bending of the film as shown in Figure 5.6. Since there is a compliant material underneath the outer film, during a sliding motion, it is continuously bent. This gave an important clue about how to build a kind of force

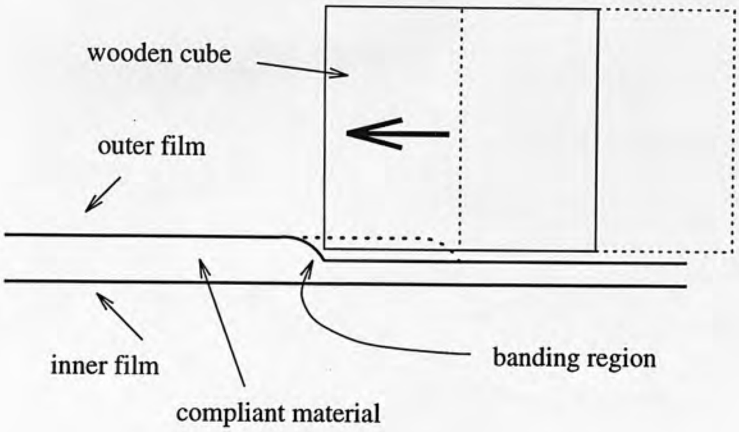


Figure 5.6: Sliding on the film with a compliant material underneath

sensor exploiting deformation of the sensor body. The PSD distribution in Figure 5.5 shows there is a significant signal component of about 3 Hz which corresponds to the deformation of the sensor surface (which is in fact dependent on the speed of sliding movement).

The PVDF film works such that once an external force has been applied to the film, which results in compressive or tensile strain, the film develops a proportionate open circuit voltage. For the Table Sensor and the Noise–cancelling Sensor, lower frequency components of the signal were attenuated and the higher frequency signals from vibration were taken into account. The reason was that first, they were designed to detect audible signals, and secondly, strong noise components are mostly at lower frequencies from 50 to hundreds of Hz.

Since the film responds to strain applied on it, bending of the film causes the film to generate a voltage which corresponds to the bending action as shown in Figure 5.7. Charge developed vanishes at a rate determined by the electrical time constant of the interface circuit. An input resistance of 10 MΩ was used to increase the time constant.

This characteristic is exploited in designing a new type of touch sensor. In order to achieve more effective bending, the film is rolled around a compliant material. Because deformation of this body results in a voltage change at the output, this sensor is named a **Deformation Sensor**.

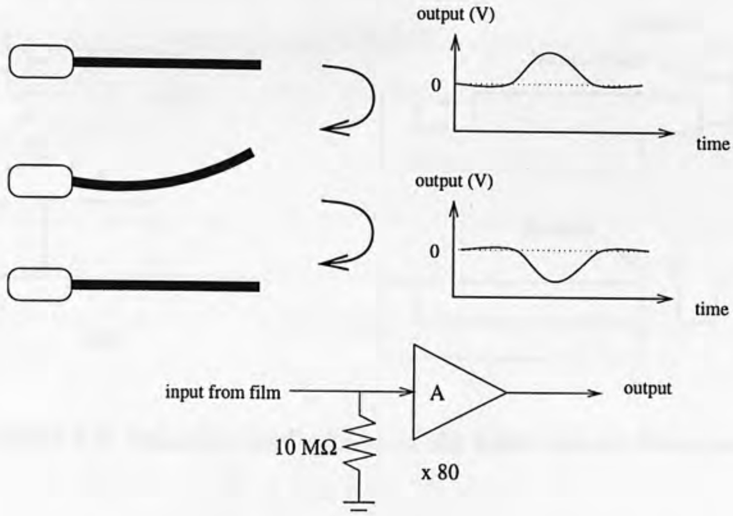


Figure 5.7: Bending experiment of the film

The Deformation Sensor can be applied in two possible ways involving two different locations: one on the touching surface of the gripper and the other elsewhere for exploratory touching. The first type is mounted on the surface of the finger and keeps contact with the gripped part. When there is a sufficiently strong touching of the gripped part, it is detected, because it causes a detectable deformation of the sensor. This variant of the Deformation Sensor is named a **Push Sensor**, since it is used for part manipulation such as pushing. The second type is to be mounted on one of the outer surfaces of a gripper's fingers. This is used for exploratory touching, hence it is named a **Bump Sensor**. It can be used to find out the position of a part. Moreover, it might be useful when a part is supposed to be pushed into place. The robot pushes the part whilst in contact with the sensor. If this part touches another, because it ceases to move, it might cause the sensor to be deformed and to produce a significant change in output voltage[4]. This is illustrated in Figure 5.8. The Push Sensor and the Bump Sensor are also force sensors since they respond to change of applied force.

These force sensors were built for the ADEPT robot. For the Push Sensor, since the gripping force is strong (to parallel a normal human adult's gripping force), a rubber

[4] This particular application of the Bump Sensor with the current condition of the sensor body and the amplifier was not successful. The sensor seemed to be too sensitive, detecting subtle changes of force occurring while pushing the part caused by the irregularity of surface friction. In Chapter 6 is described how to achieve this kind of sensing, using different conditions.

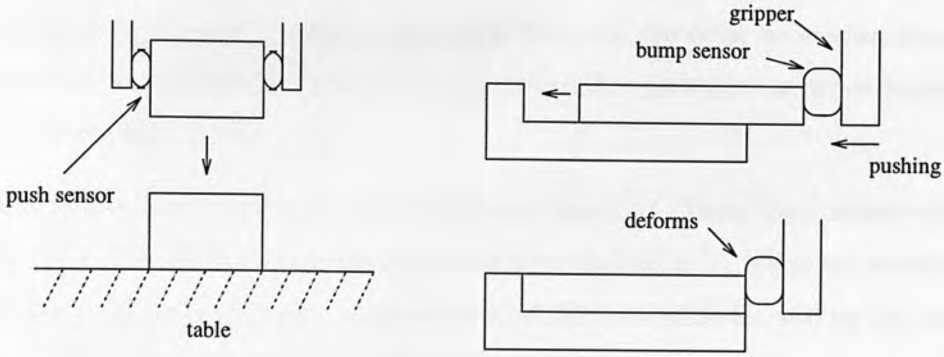


Figure 5.8: Example applications of the force sensors developed

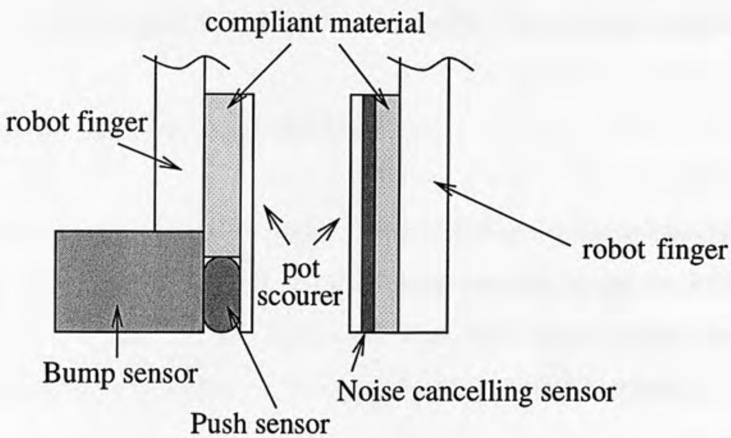


Figure 5.9: The designed fingers and sensors on them

(eraser) was used as compliant material and a PVDF film was wrapped around it. For the Bump Sensor, the sensor is desired to be sensitive and soft so as to affect the position of light parts as little as possible during bumping. For this reason, soft antistatic foam used to protect IC chips was used as a compliant material. Dedicated fingers were designed and built for these force sensors and the Noise–cancelling Sensor. The gripper and the layout of the sensors are depicted in Figure 5.9.

An amplifier with adjustable gain from 1 to 100 with a very high input resistance is suitable for the Deformation Sensor, the high input resistance increasing the sensitivity to the low frequency signals of interest[5]. The signal is fed into the PC. The PC was

[5] Employing a low pass filter would increase the low frequency signal components relative to the high,

programmed to first sample data from the corresponding port of the A/D converter, then signal an event if the value was bigger than the threshold or smaller than the negative of the threshold, because the output goes either positive or negative depending on the bending direction.

In this section, the force sensor developed was described. These force sensors detect when there is a low frequency change in the force applied to it. They are sensitive in detecting very small amounts of deformation which can hardly be seen by the human eye. Softer compliant materials yield higher sensitivity than harder materials. They are very robust to noise, both electrical and mechanical, since the low frequency signal components are highlighted. Since this type of sensor is simple and inexpensive but useful, further development is expected to be useful. This is dealt with in Chapter 6.

5.4 Sensor Interconnections

There are four sensors considered here: a Table Sensor, a Noise–cancelling Sensor, a Push Sensor, and a Bump Sensor. These sensors were set up for an ADEPT robot to perform an actual assembly task for a test case. The Table Sensor was fixed to the table and others were mounted on the gripper of the ADEPT robot.

Both the Table Sensor and the Noise–cancelling Sensors have the same attributes with respect to acoustic high frequency signals. Their role is to detect sounds generated by impacts during an assembly, hence they are called **Clunk Sensors**. Noise limits their performance. But the two sensors (table and gripper) may hear different mechanical noise (*e.g.*, the robot noise is not detected by the Table Sensor). They can be combined in a simple way such as summation. By summing both outputs, the signal might be reinforced while noise might not, since the frequency characteristic of the noise is slightly different. Summing can be performed by an operational amplifier. The output is thresholded by a Schmitt trigger, and a high state is held for a second by a monostable multivibrator. This is a binary sensor (Clunk Sensor) which goes high when vibration due to parts touching is detected.

On the other hand, both force sensors exploiting deformations deal with different

---

which is used and described in Chapter 6 as a noise suppression measure.

events. They are not supposed to be used at the same time. Even if these outputs are combined and are regarded as one sensor, the robot will be able to discriminate with respect to the purpose of the robot motion. Outputs from both sensors are fed into the PC and thresholded. If an event from any one of the sensors is detected, the PC turns on its D/A converter output, which is connected to an ADEPT signal input port [Ade85a].

Two sensor input ports of the ADEPT robot are used. Although the robot receives two inputs, each input is a virtual sensor[6] [Henderson & Shilcrat 84], which has more implications than one sensor although it is represented as one sensor. One input port of the ADEPT is connected to the output of the summed Table and Noise–cancelling Sensors, and the other is connected to the thresholded output of the PC D/A converter which represents two force sensors. The output of the Clunk Sensor is connected to the PC only for displaying of the status of the sensor, although it is a complete sensor in itself without computation. The interconnections of sensors and the ADEPT robot are illustrated in Figure 5.10. The ADEPT can be programmed in VAL II, to use these single bit binary signal inputs for the purpose of interrupting motions, for instance, guarded moves. Detailed circuit diagrams of the sensors and their connections are shown in Appendix G. Interface circuits for the gripper mounted sensors were mounted on the arm of the ADEPT near the sensors in order to reduce the effect of noise[7].

5.5 Application of the Sensors to a Benchmark Assembly

The event sensors developed were applied to an assembly task in order to demonstrate and evaluate how event detection increases the degree of flexibility allowed in an assembly. A set of benchmark assembly parts was used for this assembly task. The parts are base, plate, and peg. The parts and the finished assembly are drawn in Figure 5.11.

---

[6] These can also be called logical sensors. From the functional point of view, they are virtual sensors, and from the computational point of view, they are logical sensors.

[7] The length of the cable from the gripper to the main board required is about 5 metres. Although coaxial cable is used, noise can be induced, and the line capacitance and resistance can deteriorate the quality of the signal. Hence, it is better to transmit a stronger signal through the cable.

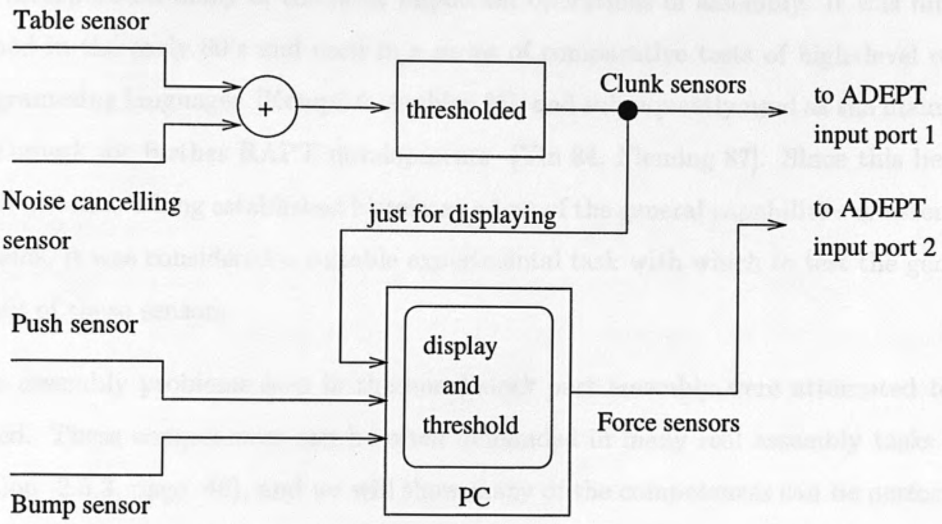


Figure 5.10: Interconnections of sensors and ADEPT robot

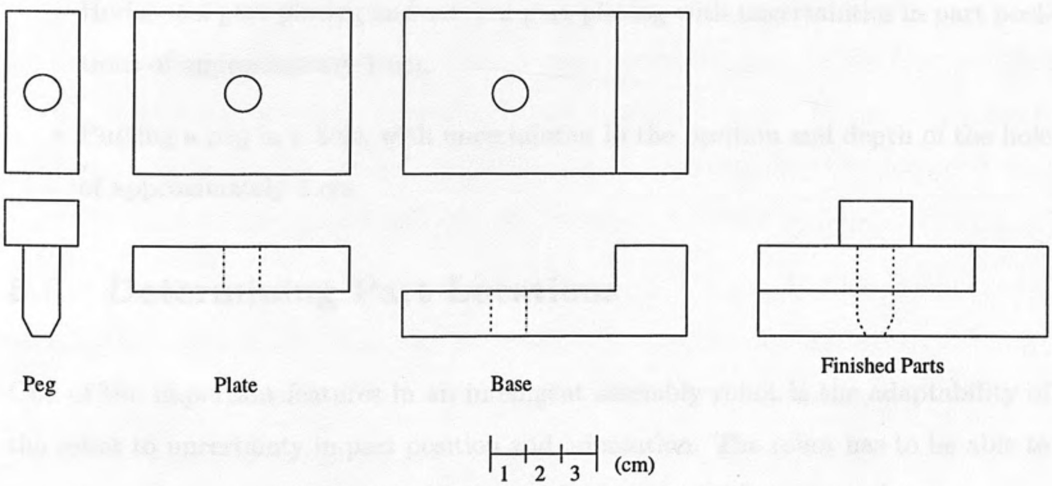


Figure 5.11: Benchmark parts and their finished configuration

The so-called "RAPT benchmark" (see Figure 5.11) was used as a typical assembly task. This was originally devised in the 1970's as a simple exercise which nevertheless incorporated many of the most important operations of assembly. It was further refined in the early 80's and used in a series of comparative tests of high-level robot programming languages [Kempf & Ambler 83], and subsequently used as the main test benchmark for further RAPT developments [Yin 84, Fleming 87]. Since this benchmark has such a long established history as a test of the general capabilities of assembly systems, it was considered a suitable experimental task with which to test the general utility of these sensors.

Four assembly problems seen in the benchmark part assembly were attempted to be solved. These competences can be often demanded in many real assembly tasks (see Section 2.5.3, page 46), and we will show many of the competences can be performed in a flexible manner by employing simple touch sensors. A later part of this chapter describes the strategies to solve them using the sensors developed, then the actual assembly is explained.

The chosen assembly problems to be solved are:—

- Acquiring parts under positional and orientational uncertainty.
- Horizontal part placing and vertical part placing with uncertainties in part positions of approximately 1 cm.
- Putting a peg in a hole, with uncertainties in the position and depth of the hole of approximately 1 cm.

5.6 Determining Part Locations

One of the important features in an intelligent assembly robot is the adaptability of the robot to uncertainty in part position and orientation. The robot has to be able to vary its motion or grasping orientation according to the configuration of parts.

In locating the base and the plate for this task, sweeping motions of the robot with a sweeping tool are used. These sweeping motions are integrated with sensory information, hence the robot is informed by events which signify touching of the sweeping tool by the parts, and it takes corresponding actions, such as stopping the sweeping motion or measuring the width of parts. For these purposes, the Clunk Sensor[8] and the Push Sensor (see Section 5.3) are used. The Clunk Sensor listens to the sound the bumping makes and the Push Sensor detects force changes on the sweeper. In order to demonstrate the application of the Bump Sensor, the peg is located by using the Bump Sensor[9] (see Section 5.3).

Sweeping is a means to orient and locate parts in an assembly (*e.g.*, [Deacon *et al.* 93], for part orientation). The idea is that everything on earth stands stably provided that it sits on the ground gravitationally stably. In using sweeping, the touching surface of the sweeping tool acts as the ground and both the **velocity** of the sweeping tool and the **friction** between the part and the table provide a force like that produced when gravitational acceleration is resisted. Hence, if the sweeping motion is long enough[10], any part would lie on the edge of the sweeping tool "gravitationally stable" with respect to the edge of the sweeper.

In this application, the sweeper is used in two ways, which are the aligning of edges of parts, and measuring the width of parts. The Clunk Sensor and the Push Sensor are able to respond to bumping of the sweeper and part during sweeping. This provides an important clue in an assembly to the **location** of parts. By virtue of this sensing activity, first the robot can have an idea about whether the target part is being pushed or not during a sweeping action. This is more informed than in the case of blind sweeping, resulting in more economic sweeping in terms of space. Second, the robot is able to measure the distance between two locations in space, thus the dimension and the current configuration of a part can be inferred by a certain number of sweeps.

The problem is how sensitive the sensors are to the bumping caused by sweeping, and how accurate they are in measuring the width of a part. Experiments were carried out to establish the limitations. To determine the sensitivity, three parts were used; the

[8] This is a virtual sensor which is the sum of the Table Sensor and the Noise–cancelling Sensor (see Section 5.4).

[9] In this assembly task, the robot was instructed to discriminate by its motion, the Push Sensor from the Bump Sensor although they are combined and fed to the robot controller using one bit, but not to discriminate between the Table Sensor and the Noise–cancelling Sensor. Hence, from the view point of the robot task, there are three virtual sensors. They are a Clunk Sensor, a Push Sensor, and a Bump Sensor.

[10] Peshkin *et al.* worked out on the problem of how long [Peshkin & Sanderson 88]

Parts	*Size(cm)*	*Weight(g)*	Min. Speed(cm/sec.)
Metal Plate	$6.0 \times 4.5 \times 1.3$	93	4.5
Metal Cube	$2.5 \times 2.5 \times 2.5$	43	6.3
Wooden Cube	$2.5 \times 2.5 \times 2.5$	10	9.1

Table 5.1: Profile of parts and minimal required speed for detecting bumping with the sweeper

plate of the benchmark, a wooden cube, and a metal cube. Whether any sensor signals occurred during the bumping of the wooden sweeper and the target object was observed under various speeds of the robot across the ADEPT working table (unpolished metal). The Push Sensor is less sensitive to this kind of bumping than the Clunk Sensor. The Push Sensor was designed to be at its maximal sensitivity to the force coming from the direction from the end of a finger to the gripper body, with the aim of detecting the putting down of a part. For sweeping, the sound that occurs is taken into account as events. The profile of the parts and the minimum speed required[11] for reliable sensing (at least 10 consecutive successes) in sweeping are shown in Table 5.1. The speed increment was approximately 10 per cent. The minimum required speed depends on the weight of a part and the friction between a part and the table.

The speed for sweeping used in the practical assembly was 10 cm/sec. When bumping occurs, the robot stops immediately[12], and the part moves further due to its inertia after being hit. This is normally 3 to 5 millimetres. It depends on the application whether this error should be taken into account or not. During sweeping, the robot was instructed to travel further by 0.5 cm after bumping with the target object, which is a blind sweep in order to reduce the error. When this is completed the robot reads the current location [Ade85b].

The plate is a rectangular prism, as is the base, but with the addition of a step. Orienting this kind of shape needs at least two sweeps for orienting two adjacent sides. An L-shaped sweeper (see Figure 5.12) was used in order to reduce the number of robot motions. When the friction cone of the part on the sweeper, the angle between the

[11] There is also a maximum allowed speed beyond which the quasi-static assumptions, usually applied to predict sliding behaviour, no longer hold [Mason 85].

[12] Strictly speaking, it takes milliseconds of time for the robot (robot stopping problem) to stop after the sensor signals each bump. The length of this positional overshoot depends on the speed of the robot, signal processing time, *etc.*, which is considered more in depth in Section 6.3.2. The robot locates the final position after sweeping not the position where the bumping occurred.

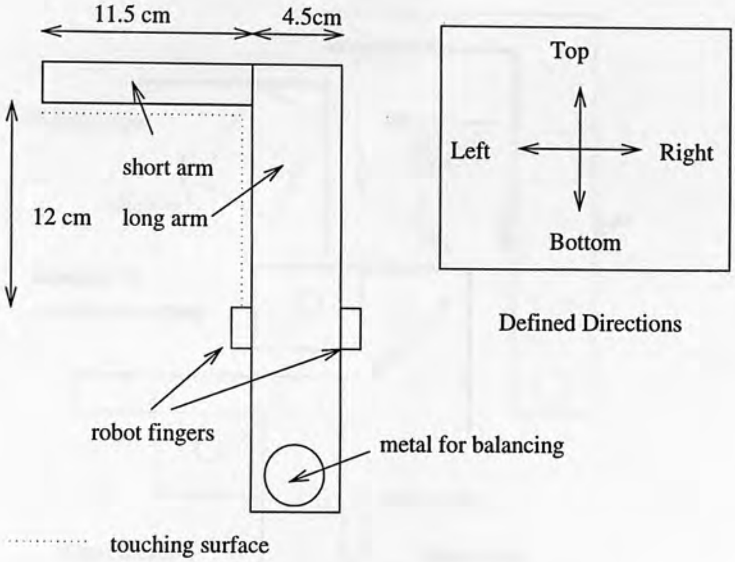


Figure 5.12: The configuration of the sweeping tool

part and the sweeper, and the direction of the pushing are critically aligned, the part would not rotate enough. In order to reduce the possibility of this, slanted sweeping (see Figure 5.13) was used. Slanted sweeping is a way of sweeping with the the edge of the sweeper not orthogonal to the direction of the sweep, in effect the object stands on the edge of the sweeper as if it stands on an incline, which is less stable than on flat ground. However, in order to make the slanted sweeping work, the sweeping angle should be big, and the length of sweeping should be long enough[13] with respect to the part size. In this assembly task, the size of the sweeper and the length of sweeps were determined according to the size of the parts, so as to orient the parts almost every time. The sweeping was repeated 100 times in orienting both the base (161 grams) and the plate as a verification. All the trials were successful. But, orienting a part using this sweeping can still fail when the centre of mass of the part is critically aligned so that the sweep length would not be long enough.

Since the sensors react to bumps between the sweeper and a part, sweeping can start when the part is touching one end of the sweeper in order to reduce the work space needed. To do this, first the robot was instructed to move from the right to the left until the sweeper bumped into the part, then to move from the top to the bottom until the

[13] However, how big and how long remain as problems.

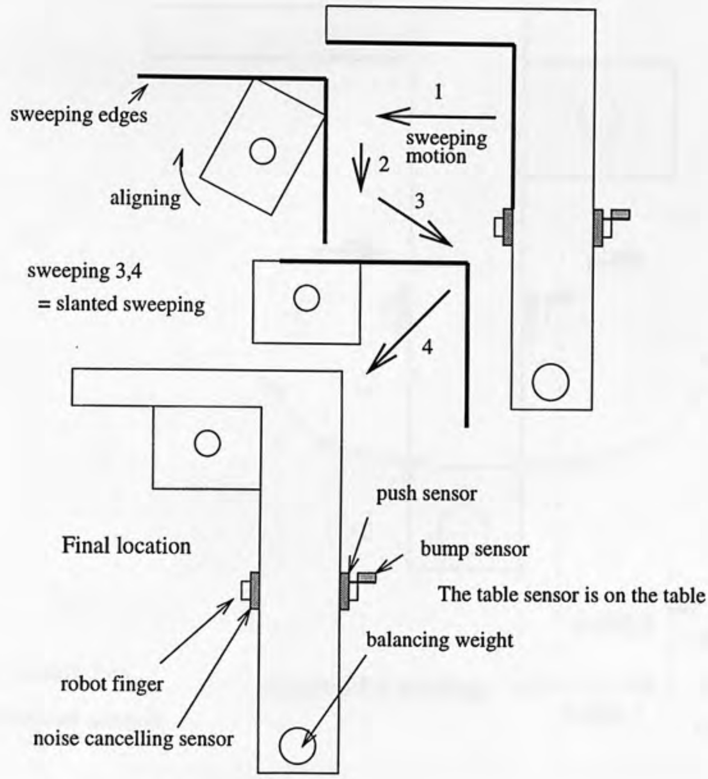


Figure 5.13: Locating a rectangular part

top side of the sweeper bumped into the part. This results in the part being at the top right hand corner of the sweeper as shown in Figure 5.13, with arbitrary orientation. From this point, the sweeping starts. Sweeping is divided into two sessions, one from top left to middle right, and then from this point to bottom left. The final sweeping will align two adjacent sides of the rectangular part to the corner of the sweeper.

There are two alignment cases after the sweeping. Either a longer side or a shorter side of the rectangle is aligned with the long arm of the sweeper. If there is no sensor, further sweeping motions are required in order to adjust the orientation of the part. But, since the sensor has the capability to detect the event of a bump between the sweeper and a part, and the robot allows external signals to come in which interrupt the motion, and it locates itself in space, the current configuration of the parts are discernible. The robot was instructed to find out the geometry of the parts by measuring the width of the parts. Then, the robot flexibly adapts to the orientation found by varying its

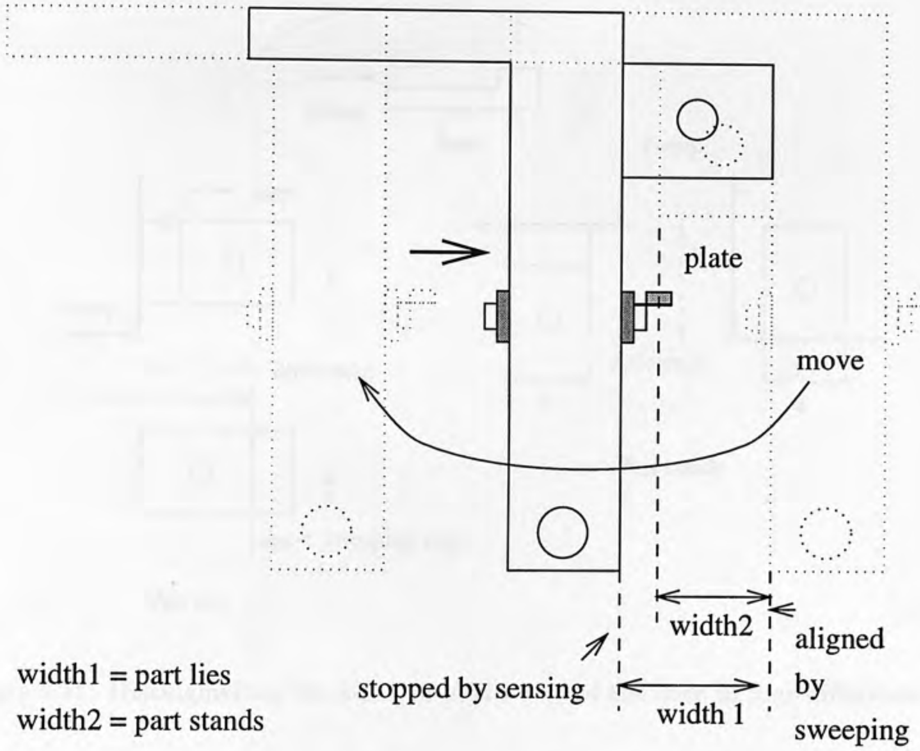


Figure 5.14: Measuring the width of the part

grasping rotation, rather than blindly aligning the part using motions which leads the part to be always in a predefined location. This is illustrated in Figure 5.14. The robot is able to measure the width of parts to within 1 mm error.

However, since the base has a step on one shorter side, it needs an extra trial in finding the location of the step on the substrate of the base. In orienting the base, the robot was instructed first, to orient the base like the rectangular plate then, from the known location and the orientation to find out the location of the step with respect to the shorter side which is known to be nearer to the start point of the sweeping. This sweeping is performed 1.3 cm (the thickness of a part) higher than the previous sweepings, so that only the step can be touched. This procedure is illustrated in Figure 5.15.

As a result of locating the base, there are four possible orientations. The robot will change its gripping orientation according to the orientation of the located base.

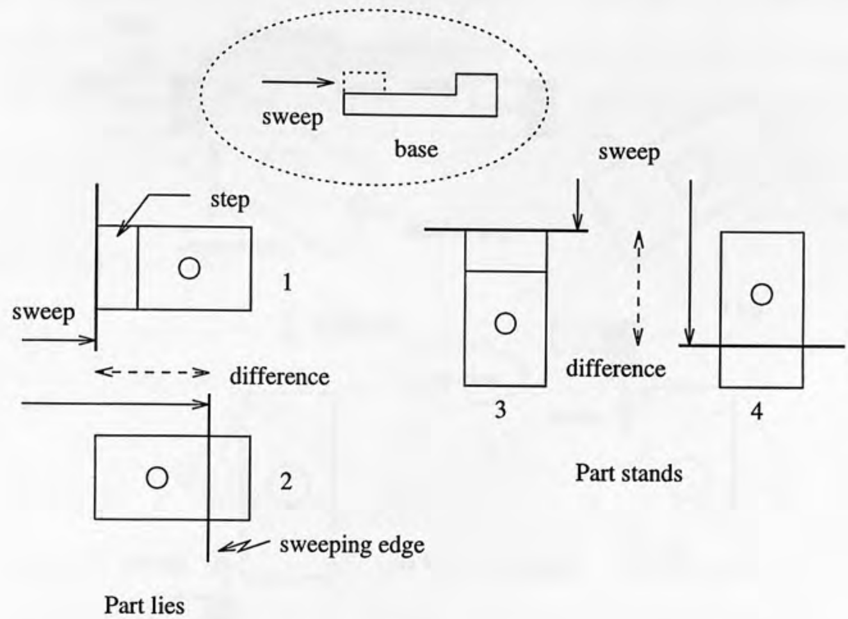


Figure 5.15: Distinguishing the location of the step of the base in four difference cases

It is also possible to locate parts using the Bump Sensor. As described in Section 5.3, on page 100, the deformation touch sensor is mounted on the front surface of the finger, hence it is free from the gripping action. It can be used for pushing parts or exploring part locations and orientations. In this application, it is used for exploring the orientation of the head of a peg.

The peg is kept in a hole on the working table. The robot knows the position of the hole and knows how to pick up the peg after aligning with the orientation. The goal of locating the head of peg is to align the sides of the head of the peg to a known orientation. The peg can be rotated freely about the centre of the hole of the table in which the peg is sited.

The goal is accomplished by detecting one of two distinct orientations of the head of the peg, then sweeping with the surface of the Bump Sensor in the direction which is determined from the discovered orientation of the head. The procedure of finding out the orientation and locating a known position is illustrated in Figure 5.16.

The Bump Sensor is able to detect an extremely gentle touch between parts. The minimum required speed for detecting is less than 0.1 cm/sec. The speed can be much

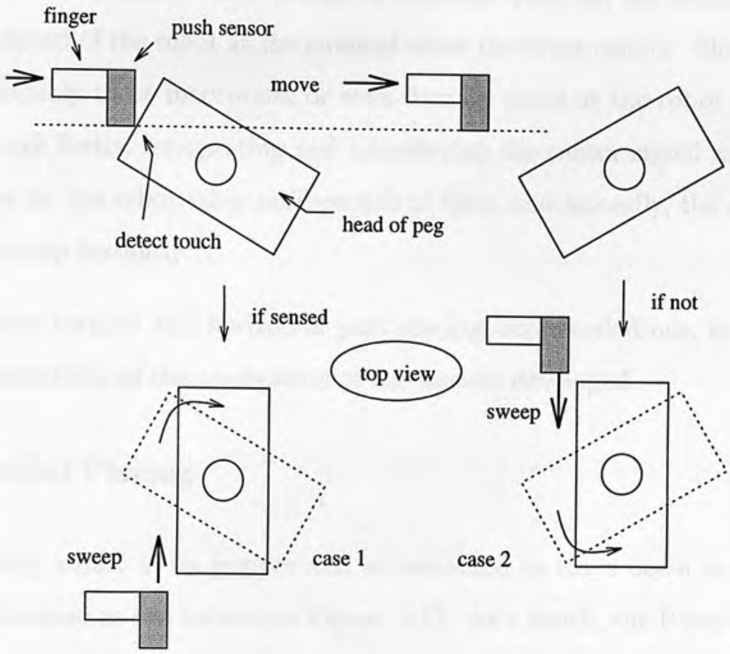


Figure 5.16: Locating the head of the peg

higher because the sensor has much intrinsic compliance. In practice the speed is set at 0.7 cm/sec.

5.7 Vertical Placing and Horizontal Placing

Vertical placing and horizontal placing are basic part fitting competences in assembly. When any positional uncertainty is introduced, these competences can be better performed using sensors. They are considered to be important test cases for the event sensors developed, since the event sensors developed could demonstrate their applicability to assembly problems. These cannot be performed well by sensorless robots where there is uncertainty in part location, although the robot may be accurate in position. They are also difficult to achieve by vision systems, due to their limited locating capability within the length which corresponds to one pixel in the image. A force/torque sensor can solve this problem alternatively.

Both the Clunk Sensor and the Push Sensor can be used for these purposes since they are able to detect sound generated, or force changing applied on the gripped part.

Since these Sensors respond to a change of physical quantity, the sensitivity mainly relies on the speed of the robot at the moment when the event occurs. The speed might cause the assembly to be inaccurate, or even damage parts or the robot manipulator. The reasons are firstly, interpreting and transferring the sensor signal and relating it to the motion by the robot takes milliseconds of time, and secondly, the robot, a rigid body, cannot stop instantly[14].

Experiments on vertical and horizontal part placing were carried out, in order to examine the limitations of the application of the sensors developed.

5.7.1 Vertical Placing

The robot grips a part in its gripper and is instructed to move down in order to put the part on the base as can be seen in Figure 5.17. As a result, the Push Sensor shows its sensitivity to the force change applied on the gripped part in the direction from the end of the fingers to the body of the gripper. Even at a speed of 0.5 mm/sec., the sensor is triggered reliably. The result is the same for other objects of different rigid material. At this speed the effect of the signal propagation delay and the robot stopping problem can be almost completely neglected for such a task (which handles relatively huge parts in order of centimetres). In practice, this speed is too slow, thus causing the task to be time consuming, so the put-down speed is set at 0.7 cm/sec. An analysis of the robot stopping problem will follow in Section 6.3.2.

On the other hand, the Clunk Sensor is less sensitive than the Push Sensor. It reacts reliably to touching from a speed of 0.8 cm/sec. It is the same for both metal parts and wooden parts. If the part is not rigid, such as soft rubber, this sensor will not be able to respond to events. However, even in such situations, the Push Sensor will respond.

5.7.2 Horizontal Placing

The robot moves horizontally in order to mate parts side to side as shown in Figure 5.17. Due to the design of the sensor, the Push Sensor does not show good sensitivity

[14] These problems are addressed more in depth in Section 6.3.2 in the case of a Push Sensor.

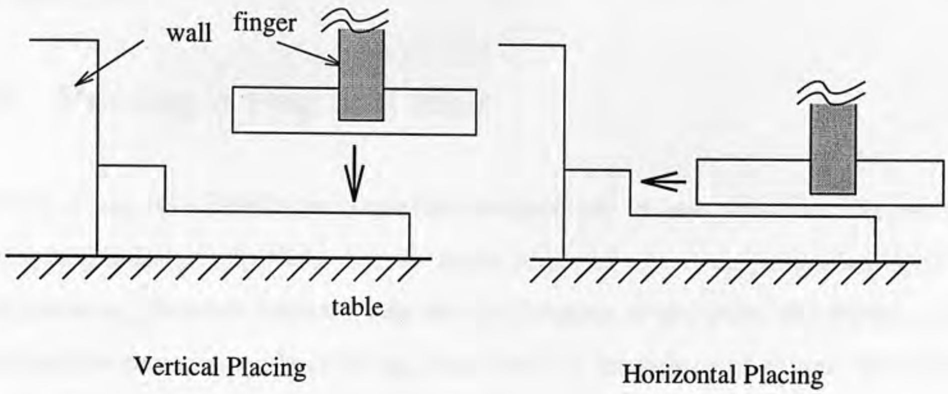


Figure 5.17: Experiments on putting parts and fitting parts side to side

to force changes in this direction (see Section 5.6). The Clunk Sensor has the same sensitivity as in putting down parts. Both sensors ignore events approximately one out of ten times at a speed of 0.7 cm/sec. If the speed is increased, the sensor will show a better sensitivity, but this is not desirable since it could introduce more positional inaccuracy.

A solution is to detect sliding between the fingers and the part. Whenever the deformation or the impact fails in detecting events, sliding will be detected by the Clunk Sensor. In practice, even though detecting either the deformation or impact fails, sliding within 1 mm is detected. Since the speed of the robot is relatively slow (0.7 cm/sec.), the signal propagation delay can be neglected.

One of the shorter sides of the plate is to mate with the inner side of the step of the base. Since the speed is relatively slow (*i.e.* less momentum), and the friction between the table and the base is comparatively low, a means to stop the base is required so that during mating, the base will not be pushed by the plate. A wall, which is a sufficiently big and heavy metal cube for stopping the base from moving, was used as a jig. First, the base is moved to the place near the wall, then mated to the wall and the table. Then, the plate is mated to the base. This is shown in Figure 5.17. Since the sensors provide a binary information at one time, first the bottom surface is mated, then the part is lifted by 1 mm clear of friction, after that its side is mated by moving horizontally, finally the part is lowered by 1 mm to finish. It is assumed that there is no uncertainty locally such as in lifting, and lowering the moving part by 1 mm onto the target part.

5.8 Putting a Peg in a Hole

Putting a peg in a hole is an important competence in assembly (see Figure 2.7). In the peg-in-hole task, there can be many kinds of uncertainties according to the configuration. Possible uncertainties are the location of the hole, the friction of the hole and the peg, the angle of fitting, the depth of the hole, and so on. In this task, among these uncertainties, uncertainties in the location of the hole and the depth of the hole were chosen to be tackled. This section does not attempt to deal with the detailed strategic problems of a peg-in-hole task, but attempts to show an applicability of the sensor developed to the problem.

5.8.1 Finding Hole

The hopping–and–trying method [Balch 92] is used in locating the hole (see Figure 5.18). It was assumed that the robot knows that the hole lies along a certain line through the cross section of the shorter sides of the mated parts, but the robot is ignorant of the exact location of the hole along the line. This is a one dimensional search. This competence can be extended for the equivalent two dimensional problem by spiral or square spiral search.

First, the robot locates and picks up the peg as described in Section 5.6, on page 110. The robot starts from a point on the line which is definitely not in the hole, putting down the peg until a sensor signals the touching of the peg on the plate. It then obtains the current location. The robot then lifts a few millimetres and moves the peg further on the line for one third of the width of the peg, and then lowers the peg for the distance which it was raised just before, plus the height of the chamfer of the peg (see Figure 5.18). If in the mean while, touching is detected, it is not regarded as the hole, but if nothing is detected until the end of lowering, it is regarded as the hole. The robot repeats this along a line until the hole is found. If the robot cannot find the hole in 30 iterations, it fails. Any sliding of the gripped part in the fingers is not desirable, thus the speed is set to half of put-down speed (0.35 cm/sec), which means

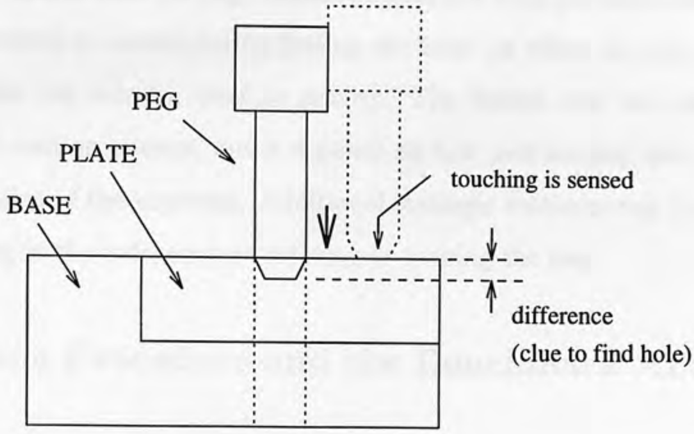


Figure 5.18: Finding the hole by hopping with the peg gripped

the Clunk Sensor can barely detect the events, but still the Push Sensor detects these events every time before the gripped part slides.

Although the hole is found most of times, since there is no guarantee that the peg approaches in alignment with the centre of the hole, it is possible that the peg is tilted by the chamfer sliding into the entrance of the hole. This causes a problem in pushing the peg into the hole because the direction of the peg and the hole is not well aligned.

5.8.2 Fitting the Peg in the Hole

Provided that the robot finds the hole, it pushes the peg down into the hole. It is just a straight push until the end of the hole is encountered. From time to time, friction and misalignment cause the Push Sensor to be activated, although it is not the end of the hole. When the event occurs, the robot introduces vertical jiggling motions in an attempt to clear possible friction and misalignment. If the robot is able to travel further after the jiggling motions, it continues, otherwise it regards this as the end of the hole.

This peg-in-hole strategy can fail due to the friction encountered midway, or due to the irregularity of the inner hole surface where the plate meets the base, the robot regards the current point as the end of the hole and finishes the task leaving the location. This means the jiggling motions could not overcome the problems encountered. This is mainly caused by the fact that the peg sometimes does not keep parallel to the inner surface of the hole, which is caused during finding the hole. In effect the peg remains half done or falls into the hole by itself by gravity. The failure rate was estimated empirically as 20 per cent on average, but it depends on how well the peg and the hole aligned at the beginning of the insertion. Additional strategic motions may be helpful during fitting the peg in the hole, such as rotating or turning the peg.

5.9 The Whole Procedure and the Benchmark Analysis

The whole procedure of the assembly is illustrated in Figure 5.19. The robot was programmed to complete the whole task at one execution.

As mentioned earlier in this chapter, the benchmark assembly is a useful test case for robot assembly. It has a number of assembly problems, such as fitting parts face to face, and peg-in-hole. The sensitivity of the sensors was good enough for the assembly, where the assembly was performed with a robot speed over the minimum speed required for reliable sensing. In result, fitting parts face to face was successful every time in 20 tries, even when there was uncertainty introduced manually, such as by raising the height of the base during fitting of the plate on the base. All the failures of assembly were caused by peg-in-hole, showing approximately 20 per cent of failure as mentioned in the previous section, which could be improved by improving the insertion strategy. However, the performance of the sensors in this benchmark assembly suggests that the sensors can be made to suit ordinary robot assembly tasks.

5.10 The Programming Methodologies

The program is modularised into competences in assembly, which could be referred to as BM's (behavioural modules) in the sense of the behaviour–based approach [Malcolm 87]. Figure 5.20 shows the segmentation of the task into behaviours. This is an example of composing an executive agent as proposed by the behaviour–based assembly systems.

Segmenting the whole task problem into meaningful modules (BM's), can provide an important benefit in planning an assembly task. It enables the robot to be programmed

1. pick up the sweeper.
2. go and sweep the PLATE.
3. locate the PLATE.
4. go and sweep the BASE.
5. locate the BASE.
6. park the sweeper.
7. get the BASE.
8. fit the BASE to the wall.
9. get the PLATE and fit onto the BASE.
10. locate the PEG.
11. get the PEG and fit into the hole.

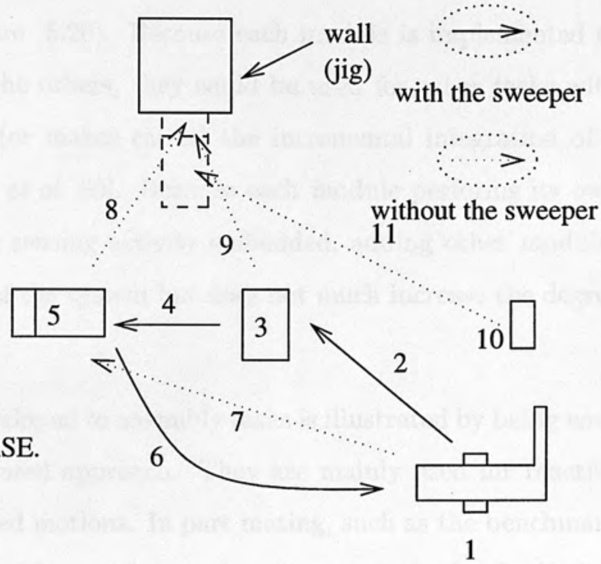


Figure 5.19: The whole procedure of the assembly

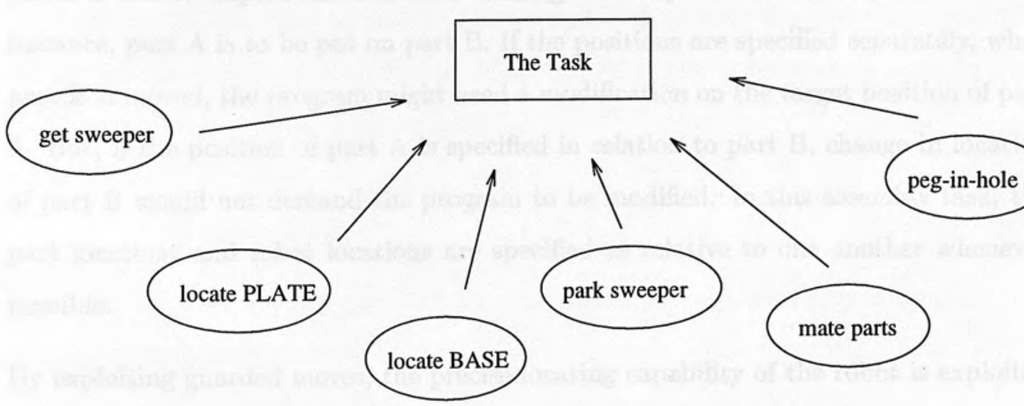


Figure 5.20: Segmenting the task into Behavioural Modules

in terms of meaningful sub-tasks[15]. The commands for the robot are specified in terms of meaningful robot competences. For instance, in this assembly, the main task is divided into such the behavioural competences as **"locate BASE"**, **"mate parts"**, and **"peg-in-hole"** (refer to Figure 5.20). Because each module is implemented to be as independent as possible of the others, they could be used for other tasks with little modification. This enables (or makes easier) the incremental integration of a large system [Brooks 86, Malcolm *et al.* 89]. Because each module performs its own specific task individually with the sensing activity embedded, adding other modules increases the degree of versatility of the system but does not much increase the degree of complication of the system.

The applicability of the sensors developed to assembly tasks is illustrated by being used in the context of the behaviour–based approach. They are mainly used for reactive low level behaviours such as guarded motions. In part mating, such as the benchmark part, use of event detection by simple guarded motions increases the level of robot competence under positional uncertainties. Much of the uncertainty is absorbed by the low level competences. This means that it is easier to build higher level competences such as a planner. The event signature sensors developed play a significant role in this context.

In an assembly task, specifying the position of an object in relation to other objects yields a neater implementation than dealing with separate absolute positions. For instance, part A is to be put on part B. If the positions are specified separately, when part B is moved, the program might need a modification on the target position of part A. But, if the position of part A is specified in relation to part B, change in location of part B would not demand the program to be modified. In this assembly task, the part locations and robot locations are specified as relative to one another whenever possible.

By exploiting guarded moves, the precise locating capability of the robot is exploited to measure the distances in the task and determining the grasp position.

Detailed technical descriptions on the assembly program can be found in [Kim 92].

---

[15] However, the sub-tasks were actually implemented in sub-routines in VAL II.

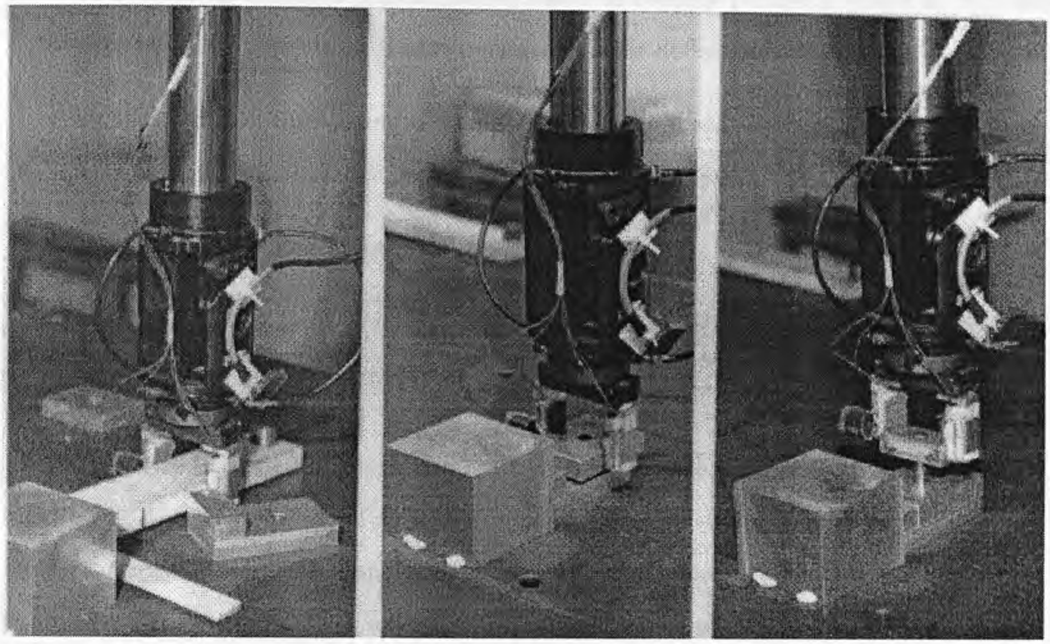


Figure 5.21: The robot assembly

5.11 Summary and Discussion

In this chapter, on the basis of the early experiments described in the previous chapter, the implementation of two kinds of event signature sensors is described. The Clunk Sensors take high frequency signal components into account, while the force sensors (or Deformation Sensors) emphasise the use of low frequency signal components. The Clunk Sensors can be susceptible to noise, but the force sensors are robust to noise and adequately sensitive to part contact.

Sensors developed were applied to a benchmark assembly task, in which many assembly problems arise. How the sensors were used in an assembly task has been described, and how this task was planned, and its implication in assembly tasks using sensors in general, have been explained.

Figure 5.21 shows the robot assembly. From left to right, first the robot locates the base by sweeping. In the second picture, the robot places the plate on the base, then it puts the peg in the hole as shown in the last figure.

The sensors developed proved to be capable of performing such tasks as:

- horizontal and vertical placing, where the Push sensor performs better at vertical placing while the Clunk sensor performs better at horizontal placing;
- locating parts either using a sweeping tool or a touch sensor, where the Clunk sensor performs better at sweeping than the Push sensor[16];
- measuring part dimension;
- peg-in-hole in a limited scope, while this competence greatly relies on the high sensitivity of the Push sensor.

The sensitivity of the Clunk sensor is limited to being just good enough for the benchmark assembly. It may need further refinement for more delicate assembly. The Push sensor has good sensitivity on vertical placing, which may suffice for normal robot assembly. However, the Push sensor requires more development on improving the sensitivity on horizontal placing. Because the Bump sensor has a soft body, it is very sensitive to bumps, more sensitive than the Push sensor. For the peg-in-hole task, the sensitivity of the Push sensor was good enough, but development of the strategy would be required in order to reduce the failure rate.

Since, in practice, a large portion of an assembly task would demand the competences itemised above, sensors capable of these tasks could provide much benefit, because they are relatively economical, versatile, and simple compared to many other sensors for assembly robots.

The Bump Sensor was used for exploring the locations of parts by touching. The Push Sensor and the Clunk Sensor were used for detecting events occurring when the gripped part touches something else. When the robot puts down the gripped part, the Push Sensor notifies the robot to stop as soon as the gripped part touches the table or the target part. This was carried out at a sufficiently low speed not to cause any significant slip of the gripped part between the fingers. Strong impact due to high robot speed can also cause damage to the parts or the robot. The Clunk Sensor is sensitive to the contact of parts and the contact between the sweeper and the part to be aligned.

[16] If the speed of the sweeping motion is too high, then the part will be bounced off further back, which may result in inaccuracy in locating parts and measuring the part dimension. In the experiment, the speed was set to 10 cm/sec which caused the part to bounce back by 3 to 5 mm's. Uncertainty of more than this is not desired.

This sensor is also sensitive to slipping of the gripped part on the surfaces of the robot fingers. This competence is used when the desired event has failed to be detected as soon as it occurs. Since the sensor is incapable of discriminating between touching and slipping, a slipping event is regarded as the initial part contact by the robot, which results in inaccuracy of a few millimeters. However, this degree of inaccuracy is mostly absorbed by the following steps of an assembly. For example, suppose there occurred inaccuracy in placing the BASE, since as a following step, putting the PLATE on the BASE is also guarded by the sensors, the inaccuracy problem is eliminated, in the direction of motion. It is part of designing reliable assembly strategies to make use of the directional uncertainty reductions provided by guarded motions and gripping.

Sweeping with a touch sensing capability is also useful. Because the robot can make sure that the part is being pushed during a sweeping motion, it can save on working space, and possibly reduce the complication of the planner when employed.

The robot program was written in a modularised manner. Each module represents a meaningful behaviour in the assembly task, incorporating the sensory information within itself. Hence, they are as independent as possible from one another. This assembly is an example of behaviour–based assembly, where the assembly plan or program is performed in terms of behaviours (or purposeful competences).

In the context of the behaviour–based approach, the application of the event signature sensors to reactive robot competences such as guarded motions provides much benefit. Most of the reactive robot competences in the assembly are incorporated with the event sensing capability: locating parts by sweeping, placing the BASE, placing the PLATE, locating the PEG, finding the HOLE and peg-in-hole with limited capability. By doing so, flexibility in part location could have been allowed. This shows that employing a relatively simple sensing capability, such as event detection, can add much flexibility to robot assembly. Because joining the sensing and motions like this can be achieved at a low level in the behavioural hierarchy, it eases the construction of a more sophisticated system[17]. This benefit comes from the notion that the low level competences take as large a share as possible of coping with the uncertainties that are necessary for the

[17] Whereas this chapter showed an application using virtually one sensor at one time, Chapter 7 discusses the problems of amalgamating multiple sensing and associated action.

whole system, and they are modularised in terms of meaningful behaviour. Leaving the detailed practical computation involved in motions to the low levels, the high level parameterised by sensing, such as more sophisticated BM's or a planner, can reason in a simplified world. If this encapsulation is applied to an assembly system programmed by a human programmer, the assembly could be programmed in terms of abstract BM's, *i.e.*, behavioural competences in assembly. This implies task level programming.

Assembly of the benchmark part assisted by the sensors developed shows that they could be applied to many other assembly problems.

Comparing the Clunk Sensors and the Push Sensor, it has been found that:—

- The Push Sensor is more sensitive to part contact.
- The Clunk Sensor is more sensitive to part contact with the sweeper.
- The Clunk Sensor requires more sophisticated electrical resources.
- The Clunk Sensor is more noise susceptible.

The results show that the Push Sensor performs better than the Clunk Sensor except for sweeping, as long as the event detecting competence is concerned. However, the Clunk Sensor can still be useful as a slip sensor, and it could be further investigated to suit a situation where the sound pattern is taken into account.

As described in Section 5.7.1, page 112 in this chapter, the Push Sensor shows a good sensitivity as an event sensor, although in the vertical direction only. More consideration on the physical construction of the sensor body may bring a better Push Sensor which is sensitive to other directions as well. One more attractive feature of the Push Sensor is that a pattern of change of force exerted to the finger surfaces can be obtained. Since this potential has been found in the Push Sensor, further refinement has been carried out. This is described in the next chapter, together with the various application areas.

Chapter 6

Further Development of the Push Sensor

Based on the performances observed, the Push Sensor proved to be more sensitive, noise-resistant, and versatile than the Clunk Sensor. Further generalisation and refinement of the Push Sensor has been performed in order to examine the capabilities and implications of this newly developed sensor.

This chapter describes the development of the interface electronics, and the construction of the physical sensor body. It then examines the performance of the further refined Push Sensor and looks at possible application areas. Among the expected application areas, part exploration, part pushing, part placing, snap-fit monitoring, and the discrimination of the force direction onto the part gripped are demonstrated. This chapter also evaluates the performance of the Push Sensor.

6.1 Building an Interface

This section explains an implementation of a desirable interface to the PVDF film based on the known properties of the film and practical experience.

Like other piezo materials, PVDF films require very high input resistance (mega Ω's) because they cannot afford much current (see Section 4.3). This fact leads to a problem in the development of the Push Sensor (which uses 10 MΩ input resistance, see Section 5.3). The use of a high input resistance allowed a technically difficult side effect, the *triboelectric effect* (*i.e.* electricity generated by friction) to cause interference. A wire

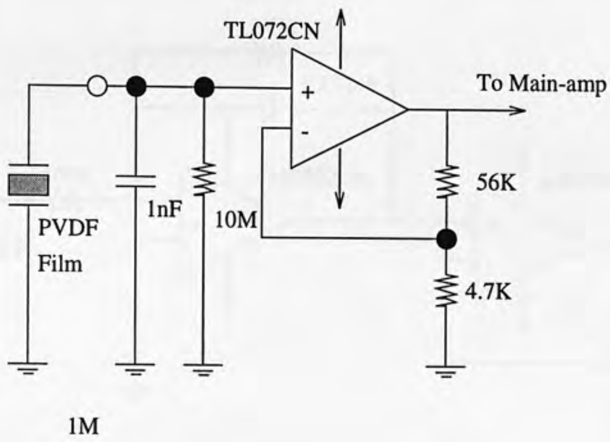


Figure 6.1: Circuit diagram for the preamplifier

between the transducer and the input resistance can generate some electric charge by friction which results from knocking or bending the wire. Although this charge is tiny, due to the high input resistance, it can be captured by the interface[1]. The pattern and the strength of this false signal is similar to one generated by normal sensing action.

There are two possible solutions to eliminate the triboelectric effect. One is to use a non-triboelectric cable, which is available commercially, but is expensive. The other solution is to keep the length of the wire as short as possible and ensure no significant friction is introduced to the wire. The latter is adopted.

A preamplifier is employed for each robot finger with plain twisted wires of length at most 5 to 10 cm. The circuit diagram of the preamplifier is depicted in Figure 6.1. A 1 nF capacitor is placed in parallel with the 10 MΩ resistor to form a low pass filter. This low pass filter crudely filters out any strong relatively high frequency components (higher than a frequency ≈ 16 Hz), in order to prevent the main amplifier from saturation, which will result in loss of information. The amplifier gain is about 10.

The outputs of the preamplifier are fed to the main amplifier, through 5 meter long microphone coaxial wires. The much lower input resistance here eliminates any possi-

[1] For the Clunk Sensor, the triboelectric effect could be neglected. 1 MΩ input resistance is not high enough to induce a significant triboelectric effect, and the high pass filter incorporated with the Clunk Sensor eliminates the effect further down.

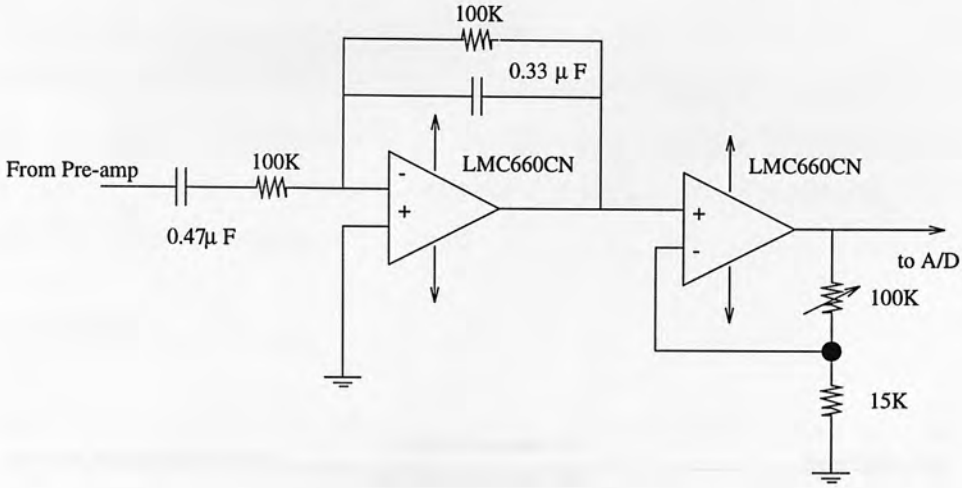


Figure 6.2: Circuit diagram for the main amplifier

bility of the triboelectric effect. The main amplifier has a band pass filter, which passes the signal frequency components around 0.4 Hz. The cut-off frequency has been chosen such that given the simplicity of the circuit as a first order filter, maximal sensitivity could be obtained, and the output settles at zero as soon as possible after a detection of an event so as to be ready for the next event. The circuit diagram of the main amplifier can be seen in the Figure 6.2, and its simulated frequency response [Colquhoun 95] follows in the Figure 6.3.

The analog signal from the electronic interface is processed by a PC (IBM compatible 386). The PC samples data via two channels of an A/D converter[2], at a rate of 20 Hz, using hardware interrupts. Signals from the two channels can vary between ±5 volts and are thresholded at ±0.5 volt to provide binary signals for the robot, via binary ports of a parallel I/O interface card[3] based on an Intel 8255. Typical graphs of these signals can be seen in Figure 6.9, later in this chapter.

[2] PC26AT, Amplicon Liveline Limited, Centenary Industrial Estate, Brighton, U.K.

[3] Chipboard Ltd., Almac House, Church Lane, Bisley, Woking, Surrey, U.K.

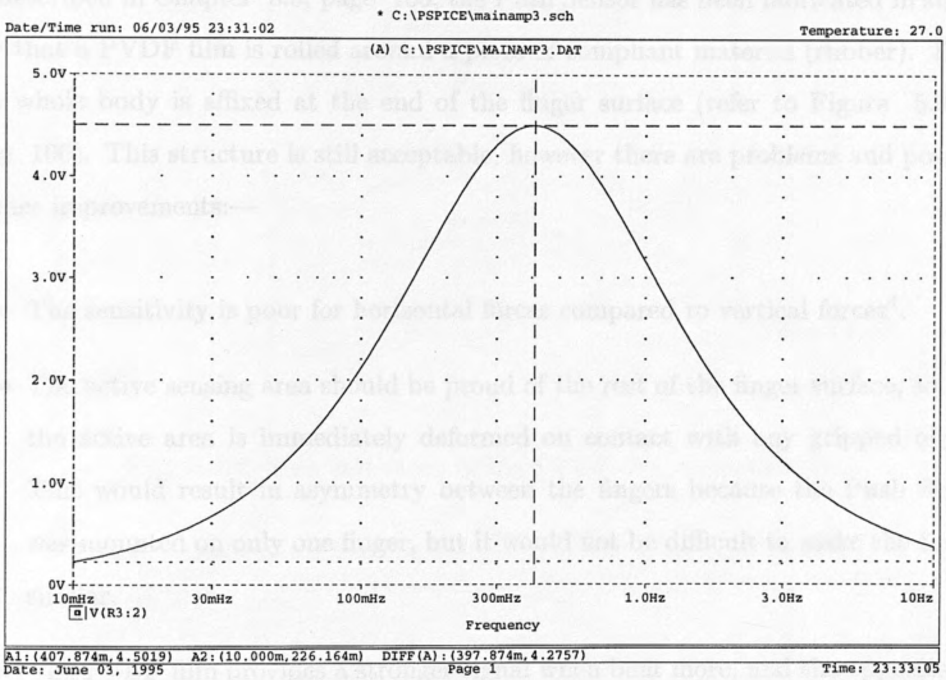


Figure 6.3: The frequency response of the main amplifier

6.2 Physical Construction

This section provides descriptions of the physical construction of the fingers with PVDF films. Two types of film placement are described and the use of compliant material is discussed. Film placement and the compliant materials including the skin used are closely related to the sensing performance.

6.2.1 Film Placement

As described in Chapter 5.3, page 100, the Push Sensor has been fabricated in such a way that a PVDF film is rolled around a piece of compliant material (rubber). Then, this whole body is affixed at the end of the finger surface (refer to Figure 5.9, on page 100). This structure is still acceptable, however there are problems and possible further improvements:—

- The sensitivity is poor for horizontal forces compared to vertical forces[4].
- The active sensing area should be proud of the rest of the finger surface, so that the active area is immediately deformed on contact with any gripped object. This would result in asymmetry between the fingers because the Push Sensor was mounted on only one finger, but it would not be difficult to make the fingers similar.
- The PVDF film provides a stronger signal when bent more, and the opposite polarity when bent in the opposite direction. This characteristic can be exploited to discriminate the direction of force applied to the gripped object, thus providing more than just on–off binary information. It would provide even more information if both fingers were mounted with sensors.

Other configurations of finger placement have been tried in order to overcome the problems the prototype had, and to explore a better sensing ability and robustness. As a result, two types of variations in film configuration are proposed so far: flat type

[4] Vertical force is the force exerted on the gripped part from the bottom to the fingers such as the force applied when a robot performs a vertical placing. Horizontal force is the force exerted on the fingers when the robot performs a horizontal placing, as explained in Section 5.7, page 111.

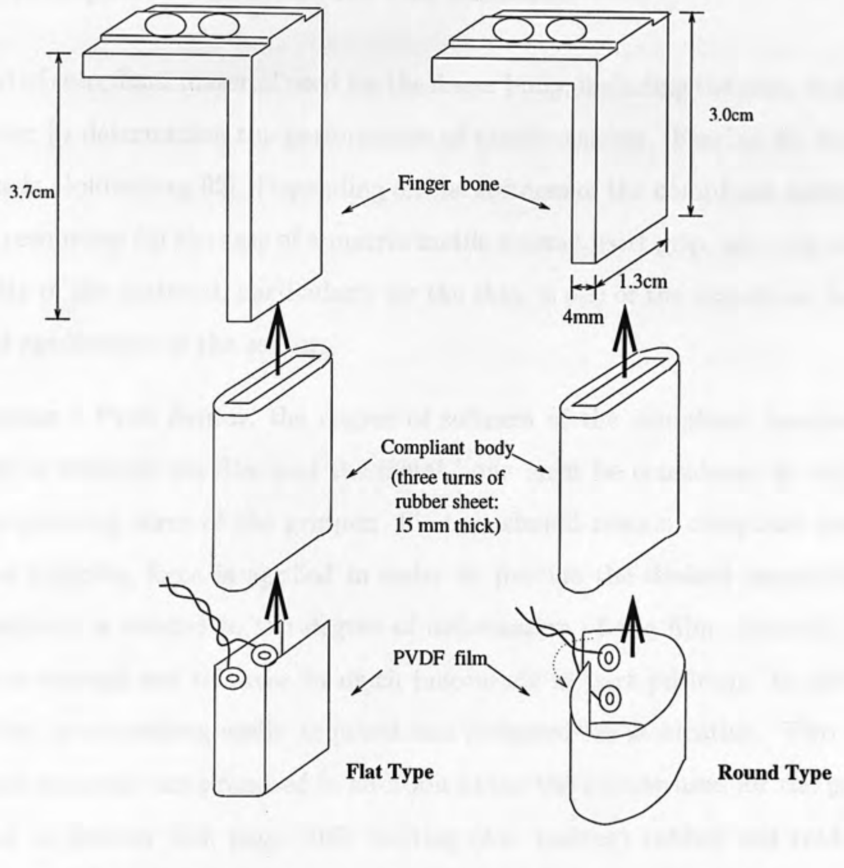


Figure 6.4: Two variations in the PVDF film configuration

and round type, as shown in Figure 6.4[5]. The films used are LDT1-028K's[6], with the dimension of 40mm×15mm.

These two types are more sensitive and provide larger sensitive surfaces than the prototype one. It is also easier to obtain a regular surface during the fabrication. The round type has a good sensitivity to both vertical force and horizontal force. The flat type has poorer sensitivity than the round type and better than the prototype one to horizontal force, but better sensitivity than the round type on vertical force. Detailed experimental results of the experiment of the sensing capability will follow in later sections.

[5] Not shown in the figure are the skins finally mounted on top of the PVDF films.

[6] Piezo Film Sensors – Europe, Merrion Avenue, Stanmore, Middlesex HA7 4RS, U.K.

6.2.2 Compliant Materials for the Sensors

The kind of compliant material used for the finger body, including the skin, is an important factor in determining the performance of tactile sensors [Fearing 92, Russell 90, Shimoga & Goldenberg 92]. Depending on the softness of the compliant material, sensitivity, resolution (in the case of a matrix tactile sensor), part grip, *etc.*, are restricted. Durability of the material, particularly for the skin, is one of the important factors for practical application of the sensors.

In designing a Push Sensor, the degree of softness of the compliant material which is placed in between the film and the metal bone must be considered in conjunction with the gripping force of the gripper. First, it should remain compliant even when a desired gripping force is applied in order to provide the desired sensitivity, since the sensitivity is related to the degree of deformation of the film. Second, it needs to be firm enough not to cause to much inaccuracy in part position. In addition, it had better be something easily acquired and prepared for fabrication. Two types of compliant materials are proposed in addition to the the rubber used for the prototype described in Section 5.3, page 100: melting (*i.e.* casting) rubber and rubber sheet cut from rubber kitchen gloves.

The possibility of using melting rubber, which is used for art casting purposes, was considered and tested because it can be easily melted and shaped into a desired form. It melts at a temperature of 100°C. First, the molten rubber is applied to a finger bone then a PVDF film is wrapped around it. Then, more molten rubber is applied on it to form a skin. In test, this proved not to be suitable for heavy parts (over a hundred grams), since the material is too soft, resulting in significant inaccuracy in part position with unnecessarily high sensitivity. When the robot moves while gripping a part, the part tends to move between the fingers due to the applied forces and the sensor detects the movement. This construction method with melting rubber is best used for light and delicate parts with a mild gripping force.

On the other hand rubber sheet cut from rubber kitchen gloves, is a good material in many respects. It can be wrapped onto a finger bone as many times as required to provide a desired compliance. It is a good material for artificial finger skin since it has appropriate friction and durability, after all it is intended to be used in human grasping. This material has been used for extensive experiments, since it is good enough for experimental assembly work such as that described in Section 5.5. This material is used to build the fingers illustrated in Figure 6.4. As the inner compliant material, a patch of rubber sheet was wrapped around the finger bone to 1.5 mm thickness (three turns), to which a PVDF film was fixed. The width of fingers are approximately 1.7 cm after they are finished with the rubber skin. Unnecessary gaps were filled with silicon rubber. Silicon rubber was also used as an adhesive in addition to instant super glue.

6.3 Sensor Evaluation Experiments

This section describes the procedures and results of the experiments to evaluate the performance of two types of Push Sensor implemented (the round and the flat types) implemented when used for guarded moves. The purpose of the sensitivity experiments is two–fold: an investigation of the limitations due to the sensitivity (*i.e.* how fast the robot should move in order to guarantee reliable sensing), and measurement of how quickly the robot stops after the contact is made (positional overshoot). First, the sensitivity is tested at various robot speeds for both horizontal placing and vertical placing. In addition, the sensitivity to a torque on gripping a contact surface is also tested. Second, positional overshoot is assessed, since this determines the accuracy when the sensors are used for the purpose of object localisation.

6.3.1 Sensitivity

The sensitivity of the Push Sensors is specified in terms of the minimum speed of the robot necessary to generate a strong enough signal to reliably detect the contact of rigid objects. An electric parallel gripper [Pettinaro & Malcolm 94] with adjustable gripping force was mounted on the ADEPT1 robot and used for the experiments. Each finger has one film built in. A pentagonal 1 mm thick polished aluminium plate is used as the gripped object. Vertical placing is illustrated in Figure 6.5 (left). Horizontal placing is performed with the wrist bent by 90 degrees moving the robot also downwards, as illustrated in Figure 6.5 (right).

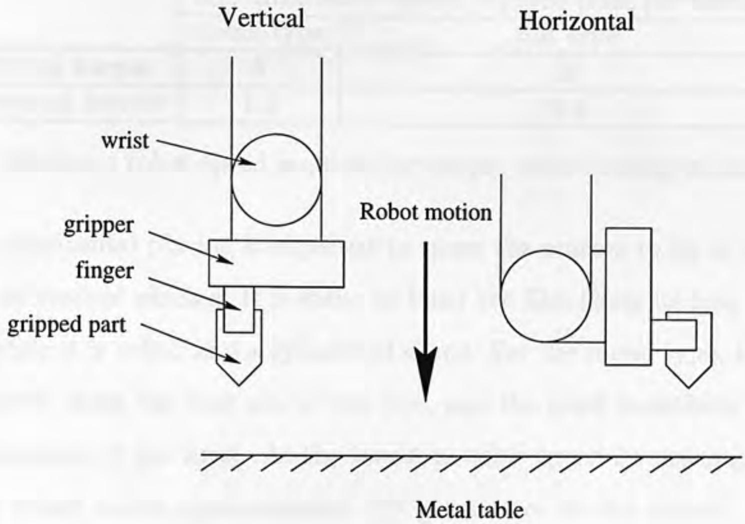


Figure 6.5: Experiment on vertical placing and horizontal placing

	Minimum robot speed required (mm per second)	
	Round type	Flat type
vertical placing	1.2	0.1
horizontal placing	1.3	2

Table 6.1: Minimum robot speed required for reliable sensing in mm per second

Table 6.1 summarises the sensitivity, *i.e.*, the minimum robot speed required (in mm per second) to cause a strong enough sensor signal to be reliably detected. The increment was approximately 10 per cent of the speed, and each speed specified in the table was the minimum speed where at least 5 consecutive tries were successful. The gripping force is approximately 2500 gram force [Pettinaro 96]. As the robot speeds in the table decrease, and the energy of impact decreases, there comes a point where the sensors quite abruptly cease to detect the impacts.

The sensitivity of the round type fingers is more uniform in response to forces from different directions than that of the flat type fingers. However, given the simplicity of the interface electronics, the flat type fingers show a high sensitivity on vertical placing at the cost of a relatively poorer sensitivity on horizontal placing. The ADEPT robot has some degree of compliance at its wrist joint compared to the other extremely stiff joints. This compliance diminishes the sensitivity on horizontal placing[7]. Without this

[7] Compliance will add to the distance travelled to build up the sensing force, and thus the time of

	Minimum robot speed required (mm per second)	
	round type	flat type
lateral torque	8	10
vertical torque	1.3	0.4

Table 6.2: Minimum robot speed required for torque event sensing in mm per second

compliance, horizontal placing is expected to cause the sensors to be as responsive as in the case of vertical placing. It is easier to bend the film along its long side than its short side when it is rolled into a cylindrical shape. For the round type, the tangential force is exerted along the long side of the film, and the good sensitivity compensates for the compliance of the wrist. At the lowest possible speed for sensing, 0.1 mm per second, the robot exerts approximately 102 gram force to the object[8]. This is the highest possible delicacy in using the sensor given its sensitivity.

The sensitivity to the torque exerted on the fingers was also investigated. Figure 6.6 shows the two different kinds of torque applied. Torque parallel to the sensitive surfaces of the finger is named *lateral torque* (left of the figure), and torque vertical to the sensitive finger surfaces is named *vertical torque* (right of the figure). The minimum robot speeds required in mm per second at which the sensors respond reliably, for the round type fingers and flat type fingers are summarised in the Table 6.2. The procedure followed in arriving at the results in the table was for the speed for each trial be raised until in a least five successive trials, a response always occurred. The object was gripped 3.5 cm away from its collision location. The lateral torque causes much less deformation of the sensor body than the vertical torque, due to the finger construction. This limits the scope of the general application of the sensors developed. More research is required on finger construction. For example, a gripper with three fingers could reduce the sensitivity variation over different torque directions.

build up. Since sensitivity is here defined in terms of *minimum* speed, extra compliance will require the speed to be raised in proportion. At typical high working speeds of sensor operation (*e.g.*, 10 mm/s compared to ~1 mm/s here), an extra compliance of even as much as 1 mm (at typically 2 Kg force) will only have a small effect on sensor output.

[8] The force was measured using a 6-axis force/torque sensor sitting on a stable table while the robot issues a guarded move on the sensor.

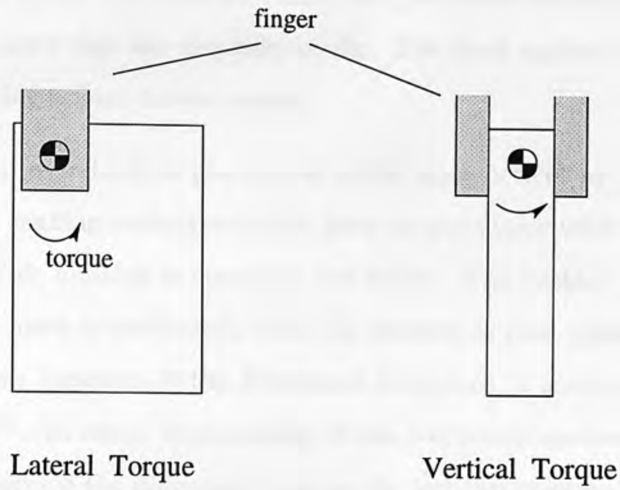


Figure 6.6: Lateral torque and vertical torque

6.3.2 Experiments on Positional Overshoot

When the sensors are used for locating an object, the positional overshoot determines the accuracy obtainable. Positional overshoot depends on: 1) processing and communication time of the equipment: electronics, PC, and robot controller; 2) robot deceleration; 3) the minimum deformation of the finger surface to stimulate the transducer. The 20 Hz sampling rate of the PC can also introduce an error of at most 0.05 second. Noise, the time delays produced by the sampling error[9] plus other constant delays, and local friction variations, will affect the repeatability of overshoot. The overshoot itself will be a combination of deformations (of fingers, robot, *etc.*) plus any slip. Since the errors depend crucially on the material and the construction of the fingers, and the particular robot used, these experiments were not intended to provide accurate numerical data. Rather, by means of this illustration, they were intended to provide the kind of sensor behaviour to be expected.

When contact occurs, first the fingers will deform. If the robot cannot stop by the limit of compliance, the gripped object will slip. The amount of slip depends on the degree of finger body compliance amongst a whole host of other things *e.g.*, surface friction, angular misalignment, mass of object, *etc.* If a guarded move contact involves slip, then

[9] Any physical event which occurred after the last sample is detected at the next sample.

multiple guarded moves will adversely affect the positional accuracy. Repeatability[10] was also tested, since this also depends on slip. The force applied after each contact was measured using a force torque sensor.

To test positional overshoot, a pentagonal metal plate is held by the robot gripper with the tip just making contact with the force torque sensor with an accuracy of at least 0.1 mm. This location is stored by the robot. The contact is then broken, a vertical guarded move is performed, then the location is read again. The difference between these two locations is the *Positional Overshoot*, a combination of any slip and deformation[11]. To assess *Repeatability* of this overshoot, another guarded move is performed, to compare the difference between the last two position readings, and the previous two.

The experimental results are summarised in Table 6.3 for the flat type fingers and in Table 6.4 for the round type fingers. Three repeatability trials were conducted at each speed and tabled to give an idea of the variation. Irregular repeatability values are thought to be caused by erratic stiction of the finger compliant surfaces. Whereas the repeatability is irregular, the positional overshoot was observed as fairly regular[12], thus as an example, two typical samples are averaged and shown in the tables for each speed. The numbers in the table can have a maximum error within 0.1 mm which may occur when the point makes initial contact with the force/torque sensor at the beginning of each experiment. Errors from the robot accuracy are smaller than this, being typically ±0.013 mm according to the manufacturer. It seems probable that positional overshoot is mainly determined by the robot deceleration time, information

[10] The term repeatability is used here to mean the capability of the robot to locate the contact position when applied to a sequence of consecutive guarded moves, whereas traditionally, repeatability is used to mean the accuracy of a robot to locate its end effector to a taught position.

[11] Note that the ADEPT1 robot minimum positional increment is ±0.013 mm, ability to return to a taught point is ±0.051 mm, and ability to position to a point taught off–line is ±0.127 mm.

[12] Since any event between two consecutive samplings will be deemed to have happened at the later sample point, the maximum positional error caused by the quantum sampling rate can be characterised as: sampling rate × robot speed. For example, at a robot speed of 20 mm per second, the maximum error would be 0.05 (sec.) × 20 (mm/sec.) = 1 mm. Hence, samples of the positional overshoot can have such error depending on the robot speed. However, in practice, no such error was found. This suggests that the proposed sampling error might be compensated for by other errors or absorbed by some mechanism, although the culprit has not yet been found. Since discovering the reason for this slightly better than expected performance would require considerable experiment, could well turn out to be due to some idiosyncrasy of the ADEPT and our work cell, and is not of much theoretical interest, it has been considered outside the proper scope of this thesis.

Speed (mm per sec.)	Mode	Distance (mm)	Force (kg force)
2	Positional Overshoot	0.121	0.640
	Repeatability	0.017	0.674
		0	0.690
		0.033	0.809
5	Positional Overshoot	0.301	1.542
	Repeatability	0.020	1.420
		0.060	1.721
		-0.060	1.353
10	Positional Overshoot	0.481	2.210
	Repeatability	0.079	2.270
		0	2.213
		0	2.264
20	Positional Overshoot	0.739	3.552
	Repeatability	0.119	3.826
		0.281	4.522
		0.040	4.075

Table 6.3: Positional overshoot and repeatability using the flat type fingers

processing time, and the minimum deformation to cause sensing. The force values show the degree of compliance of the fingers, if the robot is assumed to be stiff. The ADEPT1 robot is a very stiff robot and contacts were made with the wrist vertical (aligned with the force) to minimise robot compliance, *i.e.*, our results largely show the effects of finger compliance. As can be seen in the tables, the errors measured are relatively small given the speed of the robot, which would not cause much inaccuracy in handling relatively larger parts. Since assembly robots differ very considerably in the parameters which affect this aspect of sensor performance, each particular type of robot will require similar experiments in order to discover the sensor's limits in conjunction with that robot.

6.4 Other Variations of Fingers

Finger With Fingernail

For additional functionality, a fingernail was mounted on a round type finger. A nail cut from a thumb tack is pegged at the end of the inner compliant material close to the film. Any touch on this nail causes the compliant material to deform, and thus the film. This finger nail is used for exploration tasks. Since the flat type fingers have the

Speed (mm per sec.)	Mode	Distance (mm)	Force (kg force)
2	Positional Overshoot	0.595	1.503
	Repeatability	0.046	1.738
		0.017	1.828
		0	1.828
5	Positional Overshoot	0.499	2.516
	Repeatability	0.179	2.972
		0.020	2.830
		0.020	2.672
10	Positional Overshoot	0.840	3.933
	Repeatability	0.159	4.528
		0.079	4.726
		0	4.698
20	Positional Overshoot	1.321	3.566
	Repeatability	0.599	3.622
		0.284	3.736
		0.599	3.679

Table 6.4: Positional overshoot and repeatability using the round type fingers

films around the tip of the fingers with compliant material underneath (silicon rubber to fill the gap between the bent part of the film and the flat bone surface), they can be used for exploring tasks using the sensitive tips.

Sensored Fingers for a Left Hand

In order to overcome the limitations of static jigs, Pettinaro has employed a "left hand" for the robot, as a more versatile jig [Pettinaro & Malcolm 95]. This left hand has only one rotational joint about the Z axis (vertical to the ground), with another electric parallel gripper, identical to the one used for the experiments described so far. Sensored fingers for the left hand have been built, as shown in the Figure 6.7, which shows the assembly of an electric torch. The gripper shown at the bottom is the left hand gripper.

Since the fingers are designed to be used for various sizes of cylindrical parts such as electric torch bodies, they have V-shaped gripping surfaces wider than the fingers used for the right hand. Two PVDF films (LDT2-028K, 70mm×15mm, which is longer than those used for the round and flat type fingers) cover the whole surface of each finger. The inner compliant material is made from 1 mm thick rubber plate which is produced

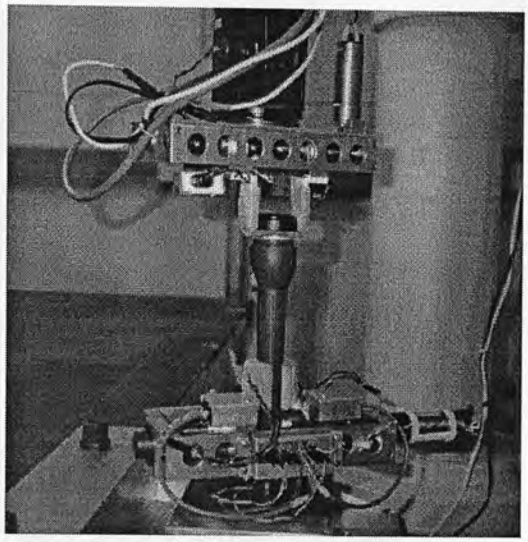


Figure 6.7: Electric torch assembly using two grippers

using melting rubber in a mould. This rubber plate is cut into size and pasted on the finger bone.

Although it has been reported in a previous section that the melting rubber is too soft for relatively heavy parts, since it is made into relatively thin plate, and the relatively tougher rubber sheet cut from the rubber glove is used as the skin, the problems noted before did not arise[13].

PVDF films are pasted on the inner compliant material in a manner similar to the round type fingers used by the right hand, but the contact area is larger. The output from the two films was summed and fed to the pre-amplifier. Rubber sheet cut from a rubber glove is pasted on it as a skin. This combination of compliant materials, given the wider dimension of the finger bone compared to other fingers for the right hand, provides an appropriate toughness for a torch assembly application [Pettinaro 96]. They have been used in practice without any notable difficulty, demonstrating performance similar to

[13] Although the melting rubber was too soft for reasonably heavy parts to be gripped, the compliance of the fingers here is given by employing thin sheets of cast melting rubber. This sheet is fixed to a solid finger bone and wrapped by a PVDF film then rubber sheet cut from a rubber glove. Because the melting rubber sheet is comparatively thin, the displacement caused by exerting force is limited. Hence, in this application, the melting rubber can be successfully used. Furthermore, when there is any application where a compact sensor–mounted finger is needed, use of thin relatively compliant material is recommended.

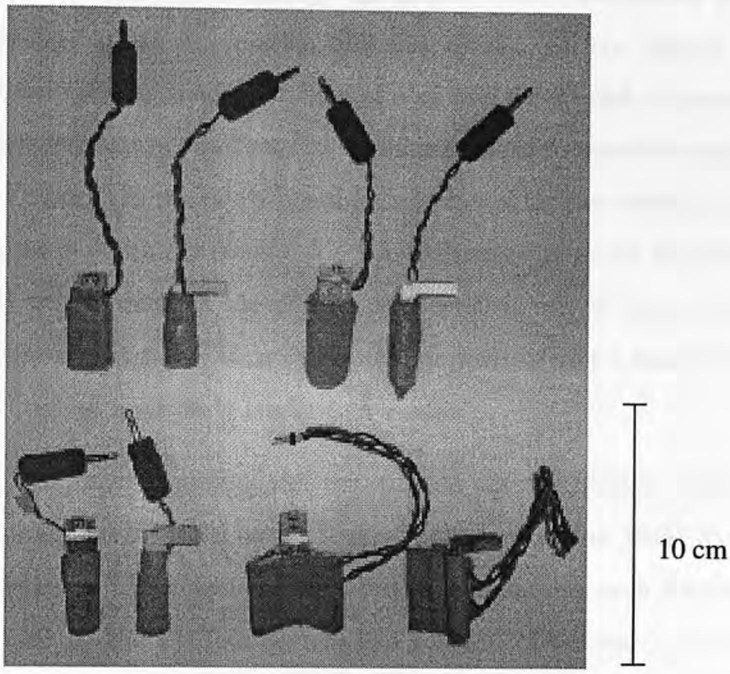


Figure 6.8: Four sets of fingers equipped with the Push Sensors

the round type fingers. The minimal robot speed required for the left hand's fingers are 3 and 5 mm per second respectively in the case of the lateral and vertical torque, whilst grasping a cylindrical part of 2.5 cm in diameter.

The push Sensors developed are shown in the Figure 6.8 (top–left: round type; top–right: flat type; bottom–left: round type with a fingernail; bottom–right: fingers for the left hand).

6.5 More Than Binary: Responding to the Force Directions

In addition to binary event recognition, the PC also extracts information about the polarity and size of the sensed forces and torques. For instance, once a touch in one direction is applied, a touch in the opposite direction, *i.e.*, moving away to break the contact, will result in the opposite polarity in the sensed force signal. In addition, the strength of a signal is proportional to the rate at which a force is applied. These are extra sources of information which could be exploited.

Parameterised event signature sensing is proposed. The Push Sensor provides three kinds of information: event, polarity, and magnitude. So far, simple binary event sensing has been used by the lowest level of a system for a quick response as an alert condition. Here the binary event sensing is parameterised to provide more information. This section attempts to provide a possible direction of further research of the sensors by showing some preliminary results. More experiments would be required in order to generalise the performance of the sensors. However, it can be seen, in general, that the results suggest that a very clear distinction of polarity over a small change of angle could be achieved by a properly engineered sensor.

The PC is programmed to provide this information via an RS232 serial line, while still providing a binary event signal to the binary input port of the ADEPT controller for a tight control loop. The magnitude is obtained by summing each internal value (*i.e.* the integers used by the PC to represent the voltage) of the sampled data while the signal remains beyond a threshold. In effect, the magnitude is an integration of the valid signal lobe, *i.e.* a value proportional to force. By having a serial link with the PC, other computers can get access to more detailed information for further processing.

The test domain selected was the placing of a rectangular part onto a table of unknown slant. The contact of one edge of the object would cause an uneven force distribution between the fingers, and as the slant changes, so does the force discrimination. The experiment showed a successful performance of the round type fingers with the slant uncertainty restricted to one rotation of the wrist axis.

The two types of fingers were tested for the task described above. Rather than employing an unknown slanted table, the wrist is tilted to an unknown angle and the existing flat horizontal table is used. With the round type fingers the output waveforms obtained at various wrist angles are shown in Figure 6.9. A wooden part of 4.5×4.5×9 cm was used as the gripped part.

At zero degrees of tilt, both fingers provide signals of the same polarity. The gain of the amplifier was adjusted such that these two signals are a best match. From one degree of tilt, in a positive sense, the signal from the finger B becomes negative, while leaving the signal from the finger A remaining positive with its magnitude growing slightly. At a negative angle of tilt, the fingers exchange their roles. There is clear distinction in the

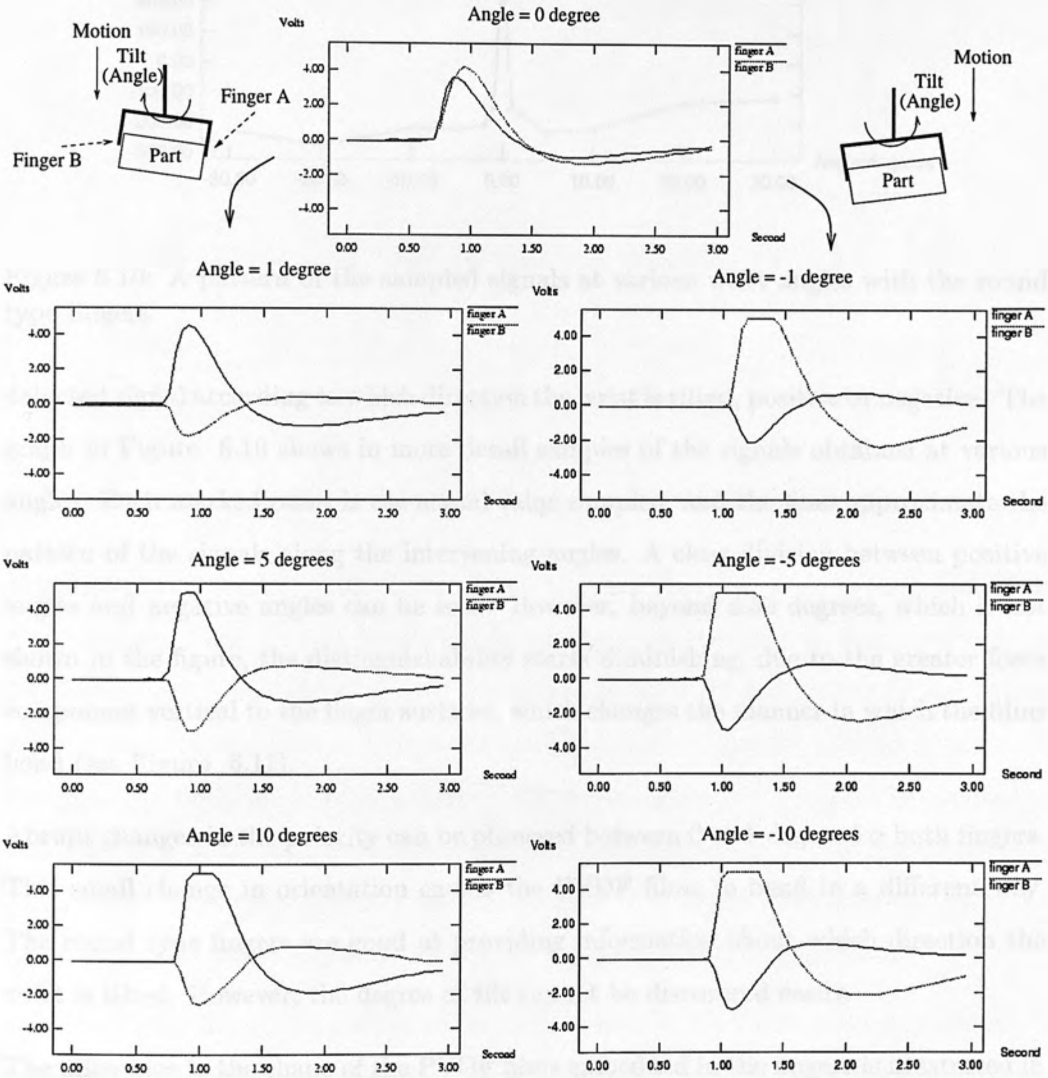


Figure 6.9: The output patterns from vertical guarded moves with various angles of the wrist

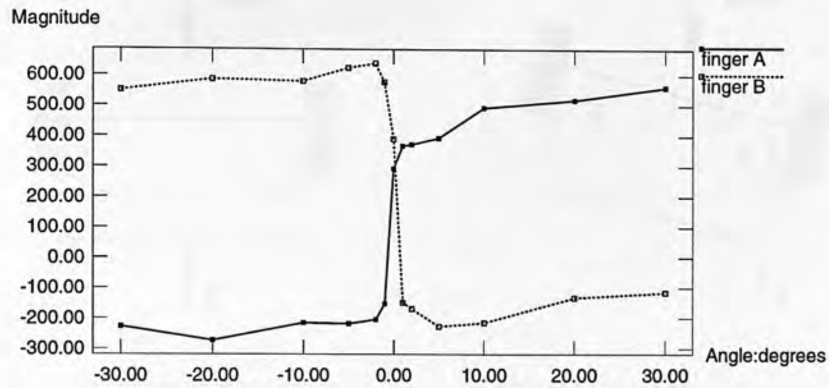


Figure 6.10: A pattern of the sampled signals at various wrist angles with the round type fingers

detected signal according to which direction the wrist is tilted, positive or negative. The graph in Figure 6.10 shows in more detail samples of the signals obtained at various angles. Each marked point is the actual value sampled, and the lines approximate the pattern of the signals along the intervening angles. A clear division between positive angles and negative angles can be seen. However, beyond ±30 degrees, which is not shown in the figure, the distinguishability starts diminishing, due to the greater force component vertical to the finger surfaces, which changes the manner in which the films bend (see Figure 6.11).

Abrupt changes in the polarity can be observed between 0 to 1 degree for both fingers. This small change in orientation causes the PVDF films to bend in a different way. The round type fingers are good at providing information about which direction the wrist is tilted. However, the degree of tilt cannot be discovered easily.

The difference in the shape of the PVDF films embedded in the fingers is illustrated in Figure 6.11, for the horizontal and tilted cases. How the films are deformed from their approximate cylindrical shape is shown in the dotted boxes. Different force patterns applied to the gripped part result in different types of force or torque applied on the films, which the round type of fingers are sensitive to due to their film placement.

It is interesting to see what happens during the abrupt transition, between 0 and ±1 degrees. The sensor outputs at the transition are shown in Figure 6.12, with samples

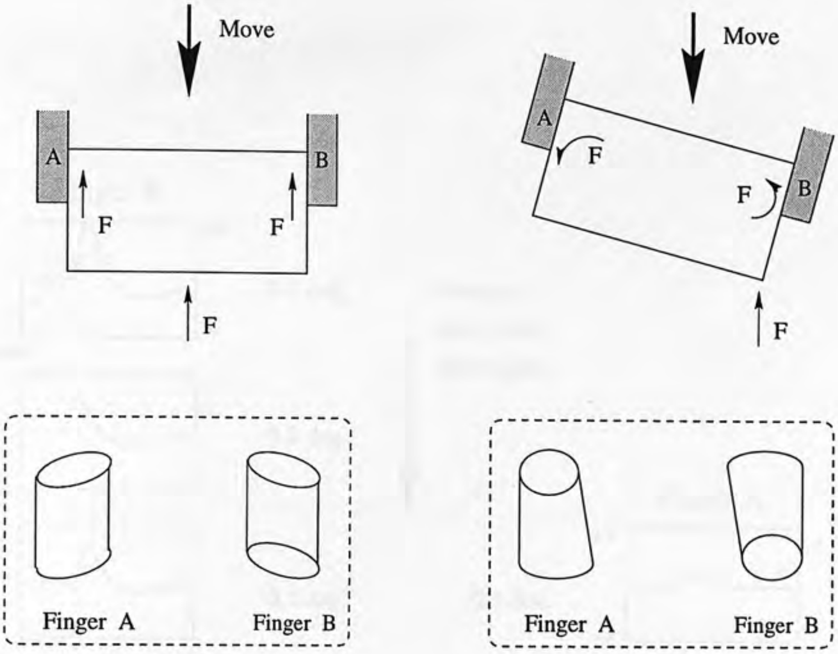


Figure 6.11: The reaction of the PVDF films to the slanted angle of the robot wrist

of 0.1 degree interval. There are two types of transition. Let us introduce the terms *Flat component* and *Slanted component* for convenience, to represent signal patterns in each situation. The Flat component (as shown in both the top graphs in the figure, which goes to plus) dominates when the wrist angle is 0, whilst the Slanted component (as shown in both the bottom graphs in the figure, which goes to minus) dominates when the wrist is slanted. Both components compete and as the slanted angle increases the Slanted component becomes more dominant. The figure illustrates two different cases of the procedure in which the Slanted component become dominant.

In Figure 6.12, it is apparent that as the angle increases, the Slanted component comes to dominate at the beginning of the signal, while the Flat component vanishes behind. Eventually the Slanted component overides the Flat component as the angle becomes greater. On the other hand, for finger B, the two components are cancelled by the aligned phase during the transition. Eventually, the Slanted component dominates. Different behaviours of the sensors during the transition are probably due to slight differences in physical construction and frictional variations.

At contact, the torque pattern would depend on the part dimensions. A smaller wooden

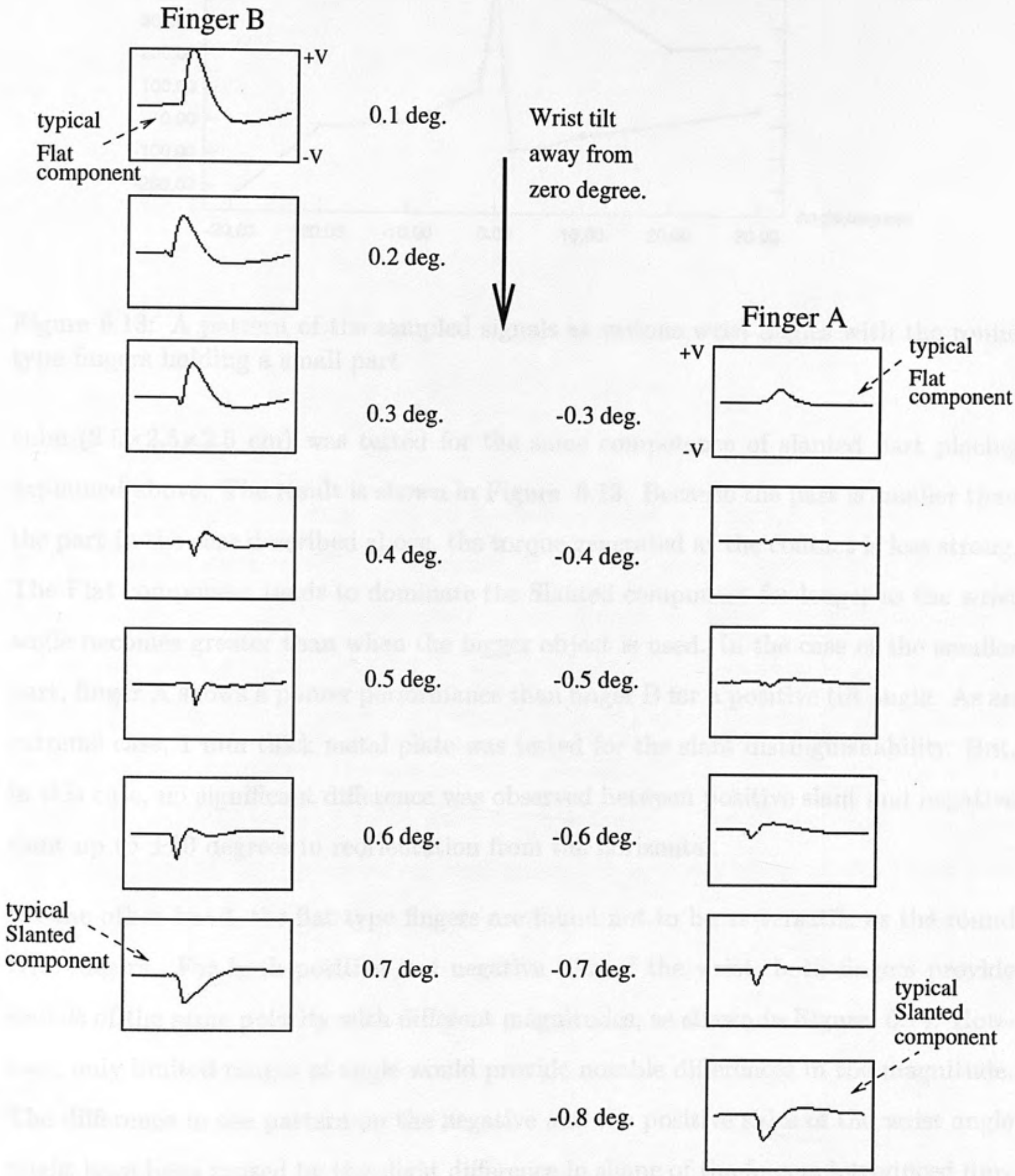


Figure 6.12: The transition of the sensor signals

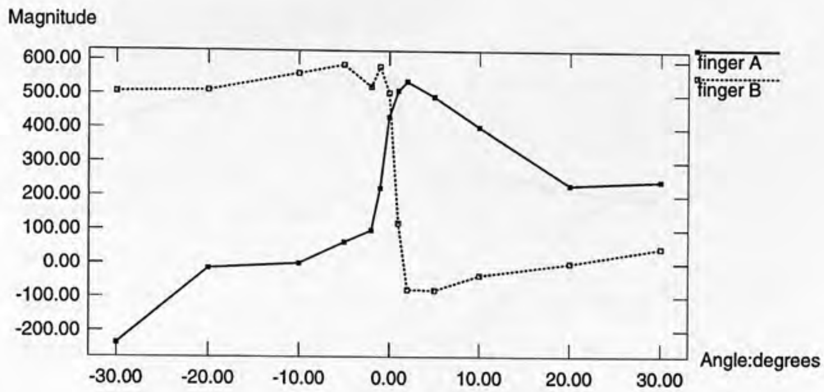


Figure 6.13: A pattern of the sampled signals at various wrist angles with the round type fingers holding a small part

cube (2.5×2.5×2.5 cm) was tested for the same competence of slanted part placing explained above. The result is shown in Figure 6.13. Because the part is smaller than the part in the case described above, the torque generated at the contact is less strong. The Flat component tends to dominate the Slanted component for longer as the wrist angle becomes greater than when the bigger object is used. In the case of the smaller part, finger A shows a poorer performance than finger B for a positive tilt angle. As an extreme case, 1 mm thick metal plate was tested for the slant distinguishability. But, in this case, no significant difference was observed between positive slant and negative slant up to ±30 degrees in reorientation from the horizontal.

On the other hand, the flat type fingers are found not to be as versatile as the round type fingers. For both positive and negative tilts of the wrist, both fingers provide signals of the same polarity with different magnitudes, as shown in Figure 6.14. However, only limited ranges of angle would provide notable differences in the magnitude. The difference in the pattern on the negative and the positive sides of the wrist angle might have been caused by the slight difference in shape of the fingers introduced during fabrication. Although built as identically as possible, these hand crafted fingers still have limitations. Nevertheless, the round type fingers, also hand crafted, provide a good distinction between the angles of different polarity. By their particular physical configuration, minor variations in fabrication are irrelevant since the PVDF films are forced to be bent radically differently when the direction of the applied force is

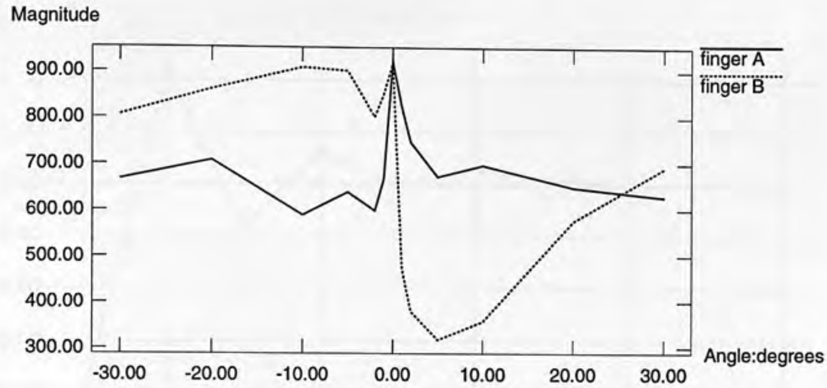


Figure 6.14: A pattern of the sampled signals at various wrist angles with the flat type fingers

changed.

The round type fingers were used on the robot whilst placing the wooden object of 4.5×4.5×9 cm, starting with an unknown robot wrist angle. Greater than 200[14] difference in the absolute magnitude was regarded as being tilted, so that the wrist is rotated in an appropriate direction by starting with 30 degrees then half of this angle for the second trial (see Figure 6.10, for the scale of the magnitude). The robot keeps performing parameterised guarded moves while reducing the rotation angle until the block sits flat. The track of robot wrist angle is shown in the Figure 6.15. The final angle of the wrist was -0.006 degree read by the robot. Some other results of the final wrist angle were: 0.040; 0.089; -0.438 degrees. Even a very small amount of permitted slip may change the way the object is settled between the fingers. When this error is introduced, the final angle between the block and the table was within a few degrees rather than perfectly flat. Figure 6.16 shows a series of photographs which were taken during a demonstration of the ADEPT robot placing a wooden block on a table of unknown slant. Before the block sat flat on the table, there were six more tries with smaller wrist angle changes, which are omitted as dots in the figure.

The Push Sensors' capability in disambiguating the force directions is limited in one particular direction of force or torque. It is also limited by the object size.

[14] The unit corresponds to force, as explained in Section 6.5, page 139.

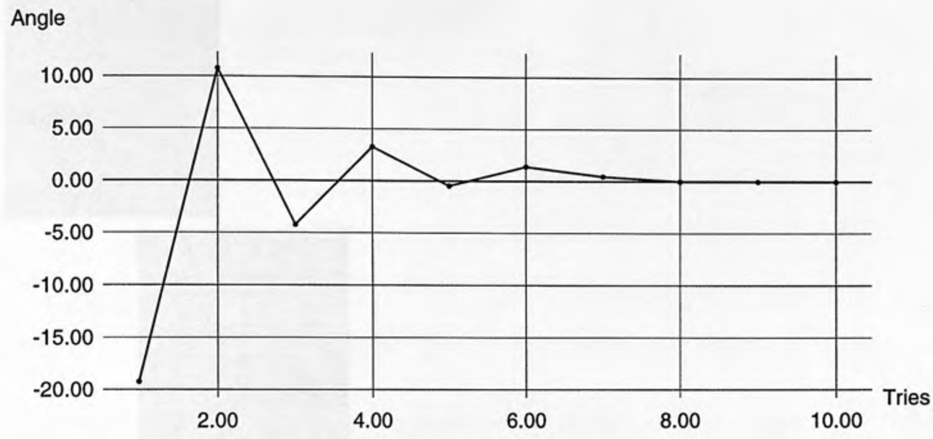


Figure 6.15: The track of the wrist angle at each guarded move for a part placing with unknown slant

6.6 Application areas

This section describes various application areas tested by the sensored fingers developed. These are useful competences which an assembly might require. First, sensor applications to guarded moves are described. Next, the use of sensors to detect force profiles is explained with an example of a snap fitting competence. The successful use of the sensors with the strategies employed by these varied applications suggests that the sensors developed are of use in robot assembly.

6.6.1 Guarded Moves

The sensors developed have been used for guarded move applications to show these competences:—

- **Vertical placing** The robot places the gripped part vertically, guarded by the sensors.
- **Horizontal placing** The robot places the gripped part horizontally, guarded by the sensors.

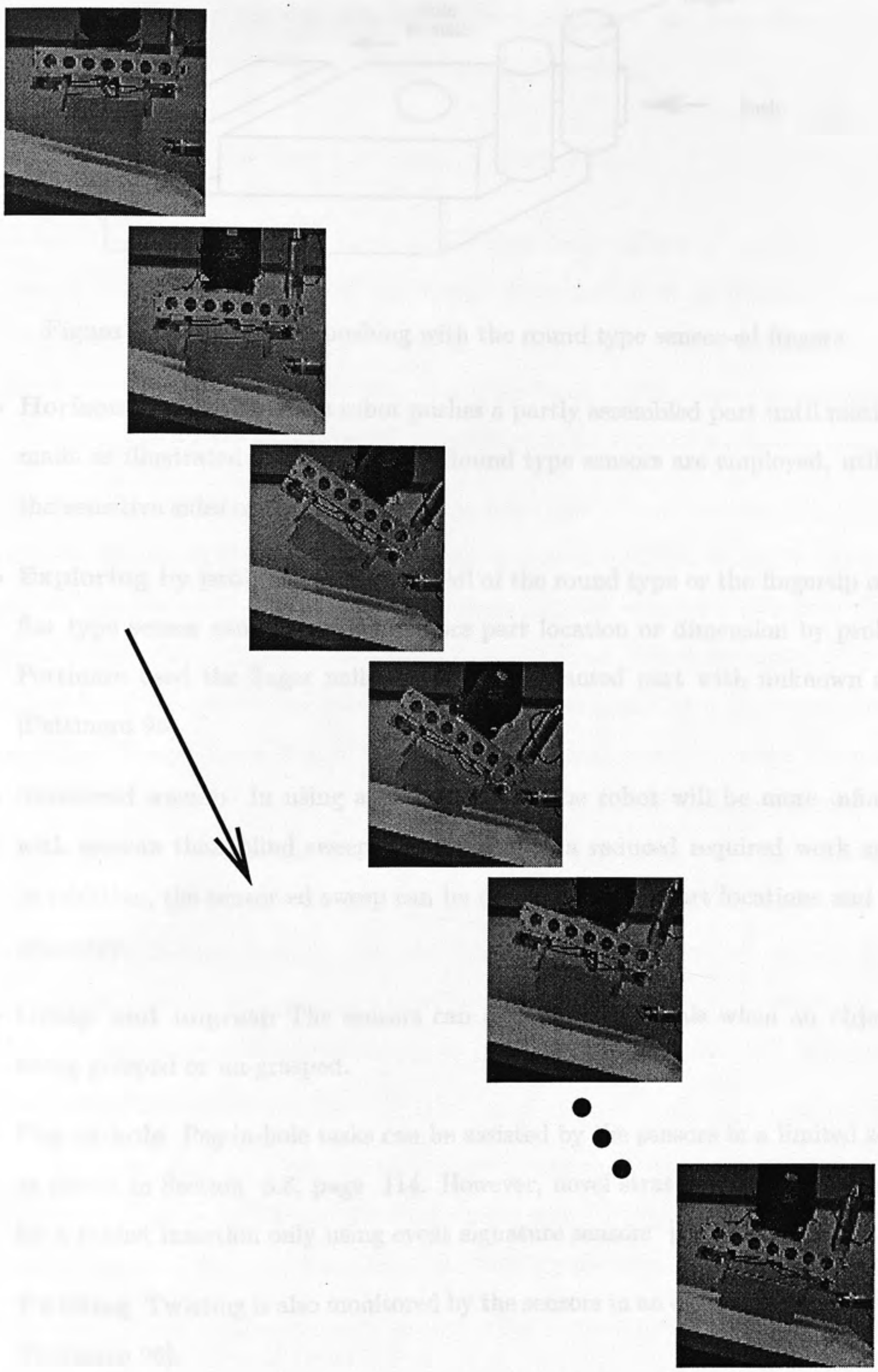


Figure 6.16: Photos taken during the robot placing a wooden block on a table of unknown slant using round type fingers

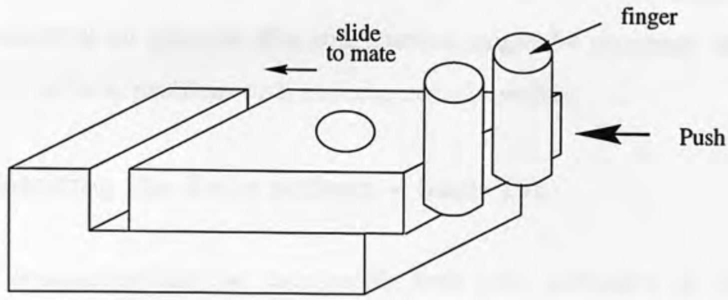


Figure 6.17: Horizontal pushing with the round type sensor-ed fingers

- **Horizontal pushing** The robot pushes a partly assembled part until mating is made as illustrated in Figure 6.17. Round type sensors are employed, utilising the sensitive sides of the fingers.

- **Exploring by probing** The fingernail of the round type or the fingertip of the flat type sensor can be used to explore part location or dimension by probing. Pettinaro used the finger nail to adapt to a slanted part with unknown slope [Pettinaro 96].

- **Sensored sweep** In using a sweeping tool, the robot will be more informed with sensors than blind sweeping, resulting in a reduced required work space. In addition, the sensor-ed sweep can be used for finding part locations and part geometry.

- **Grasp and ungrasp** The sensors can also provide signals when an object is being grasped or un-grasped.

- **Peg-in-hole** Peg-in-hole tasks can be assisted by the sensors in a limited scope as shown in Section 5.8, page 114. However, novel strategies could be devised for a robust insertion only using event signature sensors [Pettinaro 96].

- **Twisting** Twisting is also monitored by the sensors in an electric torch assembly [Pettinaro 96].

- **Parameterised guarded moves** By exploiting the capabilities of the sensors in terms of the polarity and magnitude information as described in Section 6.5, placing a part on other objects with an unknown angle of slope can be achieved.

These capabilities can be used to solve a problem with one degree of freedom. More research on physical film construction might be necessary in order to attempt to solve a problem with two degrees of freedom.

6.6.2 Exploiting the force pattern – Snap Fit

Problems of comprehending the force profile have been addressed by Selke *et al.* in the case of peg-in-hole [Selke *et al.* 91], and by McManus *et al.* in the case of a snap-fit [McManus *et al.* 92], using force sensing. In their works, input signal profiles are divided into sections in terms of time, then rule based interpreters comprehend the state of the assembly, in order to notify the failure or the termination of the assembly.

The Push Sensor developed was applied to the same kind of task, reading a force profile for snap-fit. Figure 6.18 shows a snap-fit task and its possible force profile along with a possible relative force profile after smoothing. In the actual snap-fit experiment, an electric fuse and its holder was used. Since the Push Sensor is responsive to the change of force applied to the gripped part, the force profile is a derivative form of the force pattern shown in Figure 6.18. A force profile sampled in the experiment is shown in the Figure 6.19[15]. Distinctive patterns of initial mating, insertion, and final contact with the bottom can be seen. The distinctiveness of the initial mate and release patterns depends on the tightness of the entrance of the holder. As long as the entrance is tight enough, and the robot speed is appropriate to the size of the parts, the physics and geometry of the task will produce similar patterns. The problem of truncating out-of-range values, as shown in Figure 6.19, could be overcome by employing log-amplifiers.

This result shows that the Push Sensor could be used to monitor the success of such tasks as snap-fit and peg-in-hole. The Push Sensor would provide an economic solution, over more expensive force/torque sensors in these applications.

[15] In the figure, polarity of the signal can be changed so that the force pattern can be compared with the imaginary pattern shown in Figure 6.18

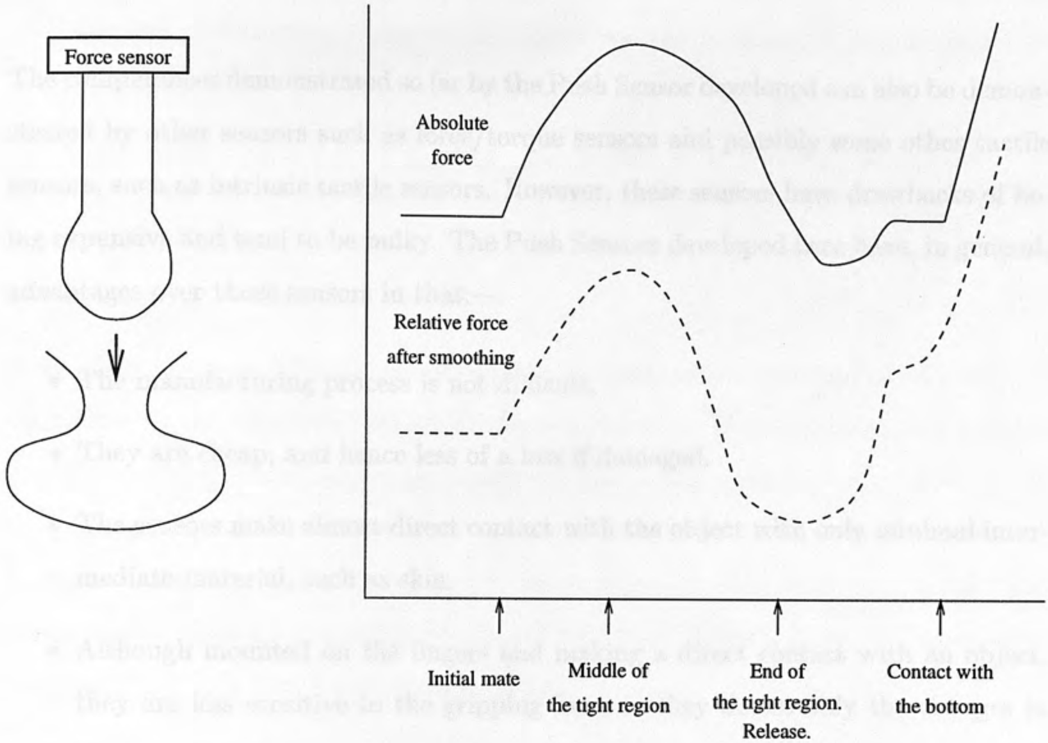


Figure 6.18: A snap-fit process and its possible force profile

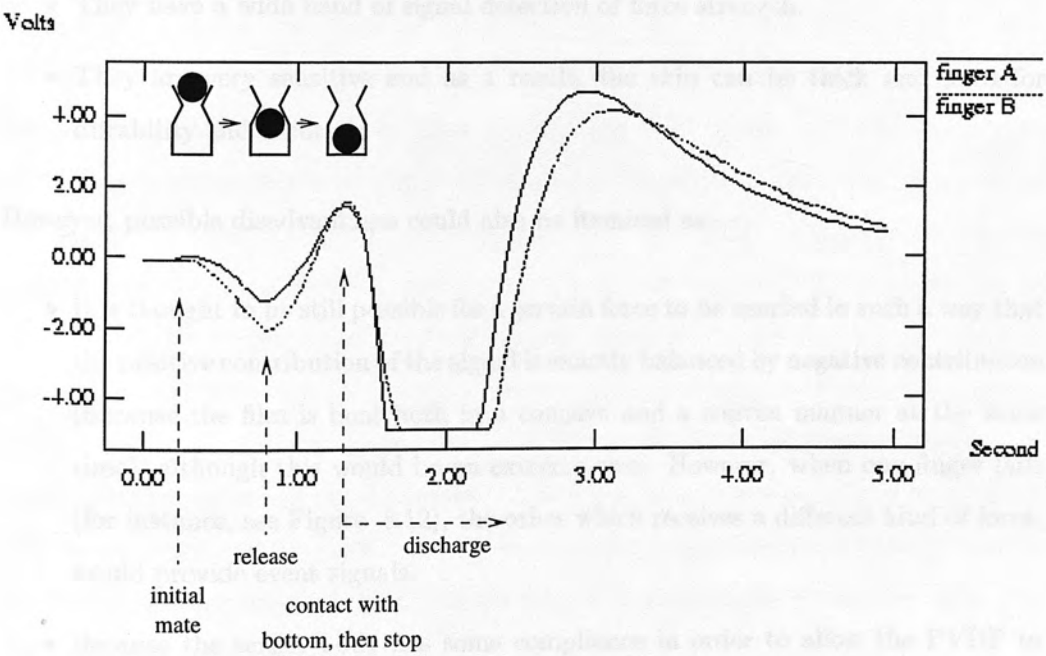


Figure 6.19: A resulted force pattern of snap-fit using a set of fuse and holder

6.7 Summary and Discussion

The competences demonstrated so far by the Push Sensor developed can also be demonstrated by other sensors such as force/torque sensors and possibly some other tactile sensors, such as intrinsic tactile sensors. However, these sensors have drawbacks of being expensive and tend to be bulky. The Push Sensors developed here have, in general, advantages over those sensors in that:—

- The manufacturing process is not difficult.
- They are cheap, and hence less of a loss if damaged.
- The sensors make almost direct contact with the object with only minimal intermediate material, such as skin.
- Although mounted on the fingers and making a direct contact with an object, they are less sensitive to the gripping force as they detect only the changes in force.
- The fingers can be shaped in various forms since PVDF films are flexible.
- They have a wide band of signal detection of force strength.
- They are very sensitive and as a result, the skin can be thick and hard for durability and accuracy.

However, possible disadvantages could also be itemised as:—

- It is thought to be still possible for a certain force to be exerted in such a way that the positive contribution of the signal is exactly balanced by negative contribution (because the film is bent both in a concave and a convex manner at the same time), although this would be an extreme case. However, when one finger fails (for instance, see Figure 6.12), the other which receives a different kind of force, would provide event signals.
- Because the sensor body has some compliance in order to allow the PVDF to deform, although it can be small, there is always some position error introduced in part contact.

- Although detection of changes of force in part contact is considered to be important, information about the absolute force would be still required for certain tasks, to which the Push Sensor is not well suited.

The simple electronic interface used in the experiments is good enough for the applications listed above. However, sensitivity can be greatly increased, if desired, by employing more sophisticated electronics.

In practice, the material remained deformed after being used for a long period, although no noticeable degradation of the sensitivity was observed. However, unless the robot is programmed not to consider precise finger positions, it may cause problems where finger position is crucial. The rubber sheet is not perfect for the skin, but our experiment shows it to be fairly good. It can also be easily replaced when worn out. In addition to the rubber sheet used, durable fabrics and even flexible metal sheet can be explored in an attempt to improve skin characteristics.

Although the Push Sensor developed is a skin mounted sensor, *i.e.* similar to other extrinsic tactile sensors, it is in fact an intrinsic tactile sensor, since it provides information about the force acting on the finger[16]. The Push Sensors are skin acceleration sensors like the one developed by Howe and Cutkosky [Howe & Cutkosky 89], with different physical construction, material, and frequency characteristics. Their sensor has 20 mm of soft foam rubber filled between the hard plastic core and the rubber skin. The 5 mm in diameter and 6 mm high accelerometer is mounted on the inner surface of the skin. Their skin acceleration sensor is a quartz crystal, built for slip and texture detection, hence is responsive to a higher frequency components (up to approximately 1000 Hz) than the Push Sensors. Howe *et al.* used the skin acceleration sensors, in addition to force torque sensors, to analyse sensory phenomena during grasp and load/unload processes [Howe *et al.* 90]. The acceleration sensors were used to indicate phase changes during the process. The Push Sensor developed can be used for similar purposes and yet could provide more robust and noise-less information since the low frequency band of the signal used by it is focused on where less noise can interfere with the signal.

[16] See Section 3.2, page 52, for the description of extrinsic and intrinsic tactile sensors.

The tensor cells [Shinoda *et al.* 95], as described in Section 3.3, page 56, are an interesting use of PVDF material to construct sensors. The transducer is embedded in a compliant material (which may limit the sensitivity of the transducer by absorbing some force), which is different from the Push Sensor that is mounted on the skin. Interpretation of the signal from the six transducers (PVDF films) fixed to the surfaces of the cube is required in order to figure out the force direction.

Force sensitive resistors are another good choice for contact force sensing, as Borovac *et al.* demonstrated [Borovac *et al.* 94]. Their sensor can be used for such tasks as peg-in-hole. They used soft sensor covers which introduce significant passive compliance during manipulation, whereas the Push Sensor developed here is less compliant in order to prevent excessive positional error. Since the sensors using force sensitive resistors measure absolute force, when significant gripping force is applied, minor changes in contact force are not well distinguished, whereas the Push Sensor developed here is less sensitive to the gripping force, since it only focuses on the change in deformation.

There are possible future extensions, such as:—

- More investigation on physical construction and PVDF film placement of the sensors. Combination of the round type and the flat type implemented in one finger is desirable in order to overcome the drawbacks of the individual types.
- Profile understanding: Based on the characteristics of the force pattern provided by the sensors, investigation into profile understanding engines is desirable.
- Seeking compliant materials for the skin: This shares with other tactile sensor research on the investigation of compliant materials.
- Active application of vibration[17] would help the Push Sensor to overcome the limitation of not being able to sense constant force.

The development of the event signature sensors for assembly tasks has given birth to innovative touch sensors exploiting PVDF film, which respond to the deformation of the finger body. They have advantages compared to other sensors for the same kind of

[17] Lee and Asada introduced sinusoidal vibration to robot motions in order to obtain rich and robust information from a force sensor under significant uncertainty [Lee & Asada 94].

tasks. Further exploration may result in more durable and practical sensors. Moreover, further investigation on other application areas is desirable.

Study of the problems of soft materials for fingers (*e.g.*, [Shimoga & Goldenberg 92]) is essential in order to provide a fundamental basis of tactile sensor implementation. Study of the utility of information from impacts (*e.g.*, [Wang & Mason 87], [Söderquist & Wernersson 92]) is also essential, particularly when there is compliant material introduced in the contact (*e.g.*, [Bicchi *et al.* 90], [Fearing & Hollerbach 85]).

In the next chapter, the problems of assembly robot control architectures when multiple sensors are introduced are addressed. The utility of the Push Sensor is demonstrated in a proposed behaviour–based assembly system and the associated problems are discussed.

Chapter 7

Sensor Fusion for Behaviour-Based Assembly

7.1 Introduction

Sensor fusion was explained and other work was reviewed in Section 2.4. Two scopes of sensor fusion were explained as: 1) narrow scope: combining of sensors (or sensing) in order to obtain unified physical or logical explicit information in a certain format; 2) wide scope: including the narrow scope, combining sensing in order to obtain a unified effect (*e.g.*, action fusion).

This chapter deals with the architectural issues of behaviour–based assembly robotic systems in using multiple sensing. The behaviour–based approach proposes that robotic systems should be built in a distributed and incremental manner. One of the main problems of the behaviour–based approach is how different distributed sub-systems can be made to work together to produce a consistently unified behaviour, while combining multiple sensing. This chapter is interested in the problems of combining multiple sensing in order to obtain a unified behaviour, which falls into the wide scope of sensor fusion. This chapter suggests and tests a generalised system architecture for the executive agent hosting multiple sensors, based on parallelism and concurrency, for assembly robots.

In general, vision is non–contact sensing, suitable for obtaining spatial information, while touch is contact sensing suitable for local part manipulation. However, touch can be used to obtain spatial information when combined with motion (*e.g.*, [Allen 92]), and vision can be used for local part manipulation, which can be achieved by finding a geometrical relationship between two bodies (*e.g.*, [Yin 83] and [Chongstitvatana 92]). They are very different in the form of information source, and in the method of information processing, but they both can contribute to figure out the spatial configuration of objects. Since they are initially so different in nature but can finally be combined to provide geometric knowledge, they exaggerate the sensor fusion problem, and are therefore interesting candidates to combine. Along with the Push Sensors described earlier in this thesis, a specially built vision sensor system is employed for the articulated task to be tested, the ring and wire problem.

The ring and wire problem is an articulated task to test sensor fusion strategies in the proposed architecture for behaviour–based assembly robots. It consists of an arbitrarily bent wire, and a ring held by the robot. The robot threads the ring along the wire, the aim being to get from one end to the other, guided by the sensors. Any bump on the wire of the ring is detected by the touch sensor and the monocular vision guides the robot. This ring and wire problem is a 3D problem which cannot be coped with by any one of the sensors well, but by combining the sensing information, it is expected to be more efficient. The ring and wire problem only requires relatively simple vision. Since this implementation is not intended to explore extensive vision routines, but to investigate the structural problems of subsystems employing multiple sensors, this ring and wire problem provides an economical experimental base.

The Albus' and Powers' models (see Section 2.4, page 39), could provide a plausible generalised architecture for the behaviour–based approach which emphasises the purposefulness exhibited by the modularised and distributed structure of BM's. Based on the guidelines (see Section 2.1.3, page 21) of the behaviour–based approach, while adopting the Albus' and Powers' models for the physical structure and the sensory access, a more generalised architecture of a behaviour–based executive agent for assembly robots employing multiple sensors is proposed here. Based on the proposed architecture, three strategies of integrating BM's with two different types of sensors are implemented, then the use of multiple sensors for behaviour–based assembly systems is discussed. Guarded moves using the PVDF Push Sensors are made to form the basis of the proposed methods.

Three strategies of combining multiple sensing were proposed and tested:—

Interlacing: The behaviour associated with vision is interlaced with the execution of the behaviour using touch sensors. By switching between the behaviours using different sensors, sensor fusion is achieved in the task space.

Vector summation: The direction of motions proposed by different sensors are combined by vector summation into a single direction of motion. However, vector summation is performed at the start of the straight line moves, which terminate at any sensed event. A drawback of employing these straight line moves is that the robot is not informed of any changes during the moves. It is a common feature of position controlled robots in that the robot is driven in terms of target positions and it is not straightforward for any elaborated sensing to affect the motion once started. These robots and their control languages are mainly designed to incorporate sensor information *between* move commands rather than *during*. In the next strategy, the robot was programmed to simulate the feature which allows sensing to affect robot motion continuously.

Continuous vector summation: While performing the vector summation, the robot is driven by approximated velocity control, where the robot updates its target position quickly with regard to the available information originated from sensing.

The guideline of incremental growth (see Section 2.1.3, page 20) of the system proposed by the behaviour–based approach can have two practical aspects on the assembly systems. One is to add a new independent assembly competence to the rest of the competences, which is the aspect described in Section 5.10, page 118. Since an assembly is performed in a temporal order, adding an independent competence would not affect the rest of the system, while increasing the number of competences of the robot. The other aspect is to improve an existing competence in an incremental manner. This requires more consideration of the architectural design of the system, since the system requires the future improvements to be conducted in a neat manner. This chapter deals with the latter aspect of incremental growth. Note that incremental growth was proposed by Brooks [Brooks 86] partly in emulation of biological evolution, which has to work with that constraint, and partly to avoid the need to program complex robots always from the ground up. Brooks implements *behaviours* using layers of AFSM's (Augmented Finite State Machines) and achieves incremental growth by adding new AFSM's. His AFSM's are more primitive than our elementary units, *i.e.* BM's, which correspond to his layers. So we can expect to have difficulty with this guideline.

From the implementation, the following results and criticisms have been made:—

- The touch sensors developed were tested in guarded moves and provided sufficient information to cope with positional uncertainty in one direction when used with robot motion. Guarded moves form the basis of the control hierarchy on which other more sophisticated functions are added. Since for assembly robots, detection of part contact is fundamental, guarded moves can be extensively applied to assembly tasks. The sensors were robust enough to survive during the long term use involved in developing and testing these tasks.

- As an implementational guideline, the guideline of incremental growth of the system proposed by the behaviour–based approach is criticised by our experience of these cases of using multiple sensors. Suppose an existing two degrees of freedom arm robot is to be developed into three or more degrees of freedom robot, substantial modification would be required of the physical structure. The incremental growth can be better applied to a robot which has a full mobility to start with as the basic physical structure. Then, by adding more abstract BM's with sensors the robot can be developed in terms of versatility and flexibility.

- Although centralised information processing ought to be avoided in the behaviour–based approach, unless multiple sensing exploits the *tacit* or *unlocated representation* (as explained in Section 2.4, page 35), explicit fusion of sensing can turn out to be necessary, which implies centralised information processing. In the behaviour–based approach, it is proposed here to adopt explicit sensor fusion in a limited manner, *i.e.* localised fusion where necessary, although it is proposed that it be implemented at as low a level as possible of the system hierarchy, so that more abstract, higher levels could avoid explosive complexity. Although such sensor fusion can be explicit at the level of a module, it can be tacit representation seen from the top level of the system (see Section 2.4, page

35).

- With the current generation of position–controlled robots, programming of flexible assembly with multiple sensors is difficult, especially with any sensor providing guidance motion with vectors, such as vision and force sensors. A robot which allows motion commands to be specified in terms of vector or force is recommended, which would facilitate the low level amalgamation of sensing and action and the distribution of localised sensor fusion.

This chapter describes, firstly the problem domain; secondly the formalism in abstracting sensing; thirdly, the system architecture, and the methodologies to combine multiple sensing and action; lastly, the discussion. The future prospect is directed towards robot structures and programming manners suitable for the behaviour–based approach.

7.2 The Problem Domain

The problem to test sensor fusion is articulated: the ring and wire problem. A ring is held by an ADEPT1 robot threading an arbitrary bent wire, guided by vision and touch sensing. It is a 3D problem, which is unable to be solved by monocular vision, and is also very tedious to solve with simple binary touch sensing. Although the touch sensors can provide directional information, it is too limited to be applied to the 3D problem, unless extensive exploratory movements are used. Since the Push Sensors developed have limitations in disambiguating directional information, as noted in Section 6.5, page 138, only binary sensing is used (*i.e.*, it is not possible to distinguish between making and breaking contact). The ring and the wire held by the ADEPT1 robot is shown in Figure 7.1. The thickness of the wire is 6 mm with a length of 80 cm. The ring has a 2 cm inner diameter. It freely rotates on its handle, hence the robot does not need rotation on its wrist[1] as bumps of the ring on the wire are expected to rotate the ring appropriately to the contour of the wire. The white round board has a diameter of 7 cm; the bigger black rectangular board attached to the handle of the wire, and the black curtain behind the gripper, are the extras prepared in order

[1] The ADEPT robot used lacks this kind of rotation.

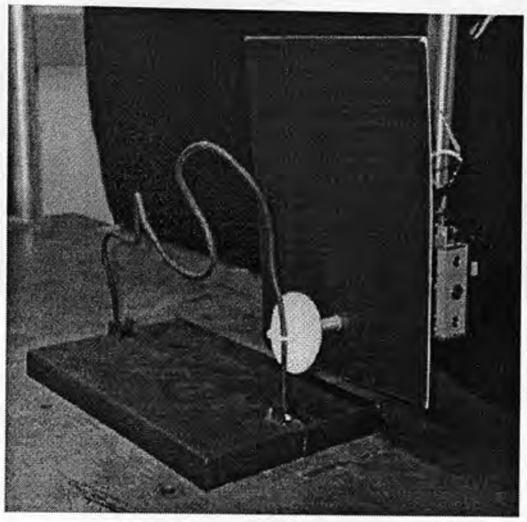


Figure 7.1: The task setup: the wire and the robot holding the ring

to obtain a silhouette image of a portion of wire projected on the white round board, which is near the ring. This simplifies the vision routine.

In a real assembly, vision can identify location or even the geometry of a part in order to direct the robot, which would require extensive visual processing. Since this thesis concerns the architectural problem of sensor fusion, extensive visual processing is out of scope. In this ring and wire problem, it is relatively easy for vision to acquire information about the contour of the wire. By only touch sensing, the robot might be sufficiently informed to complete the task. However, the robot would have to employ a large number of exploratory moves in order to figure out the contour of the wire. Vision would be helpful by figuring out the contour of the wire projected on a two dimension plane near the ring. One of the questions of this chapter is how these sensing modalities can be combined in order to have the robot performing this task effectively. By having more elaborated vision processing, this experience might be utilised for a realistic assembly. In this ring and wire task, vision resolves the Y–Z plane, and touch resolves the X axis in the robot frame of reference. Figure 7.2 shows the contour of the wire on the Y–Z plane and along the X axis. The pedestal has a considerable weight so that at the normal speed of the robot (10 mm per second) it would not move at any bump. But when excessive force is exerted due to any error, it would slip on the robot working table so as not to cause any damage of the facility.

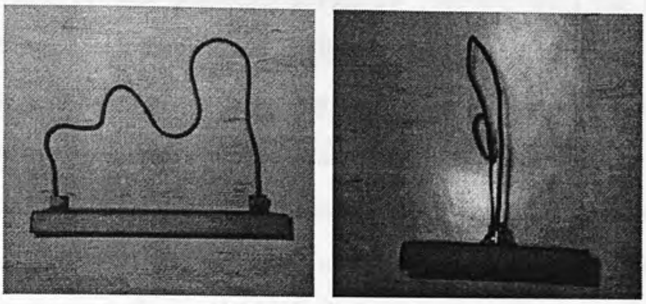


Figure 7.2: The contour of the wire seen on the Y–Z plane (left), and X axis (right), in the robot reference

7.3 The Concurrent System Architecture

This section describes the system architecture proposed for the sensor fusion experiment performed in this chapter. First the background is explained and the generalised communication architecture for BM's is proposed as the means of structural interconnection between the BM's. Then the actual communication methods between the BM's are explained.

7.3.1 Background

As mentioned earlier in Section 2.4, the problems of task level programming in using sensors would require fundamentally new approaches. A number of new approaches have been inspired by biology. As remarked in Section 2.4, many of these have in common the following general attitudes:—

- The system is distributed in terms of tasks, meaningful in environmental terms (*i.e.* behaviours).
- All control levels have their own access to sensing and action if necessary, thus are complete in their own scope of control.
- More abstract and sophisticated control levels use (subsume) the lower control levels.
- Sensing and action couplings tend to be implemented at as low a level as possible.

These notions might provide plausible guidelines together with the guidelines of the behaviour–based approach mentioned earlier in Section 2.1.3, page 21, for an appropriate system architecture for assembly robots. In the works of behaviour–based assembly in building executive agents so far, these notions have been partly adopted, and since sensors have been used in simple manners, extensive generalisation has not been yet made. Moreover, parallelism and concurrency have not also been yet addressed in the implementations, which the behaviour–based approach suggests should be useful in combining sensing and actions when multiple sensors are employed.

Parallelism and concurrency can be observed in the behaviour–based approach in Subsumption Architecture based mobile robots (see Section 2.2.3, page 24). Although mobile robots can be primitive and fairly reactive, assembly robots are different from mobile robots in that they perform tasks in a controlled geometrical environment and tasks are specified in a temporal order[2]. The executive agent needs to translate geometrical terms passed on by the planner or programmer in a behaviour–based assembly robot.

Parallelism and concurrency are often found in the control architectures of real animals as Albus and Powers modeled. In the modularised behaviour–based approach, BM's are kept as independent as possible of other BM's. Parallelism and concurrency promote independence between the BM's. This implies that adding or deleting any behavioural competence becomes convenient. The independence would also make it possible to add other sensors with associated behavioural competences to the existing system without modification. However, how to have independent distributed sub-systems to perform a specific task in a coherent manner remains a problem. Brooks' mobile robots have a single task, and he solves this problem with a specific task–oriented architecture. In assembly, the task must be programmable in a flexible manner, which gives a more general and complex problem.

The work proposed in this chapter uses parallelism and concurrency in the execution of BM's to form a behaviour–based assembly executive agent. This chapter suggests and tests a generalised system architecture for the executive agent to achieve task level

[2] However, more sophisticated mobile robots (*e.g.*, SHAKEY [Nilsson 84]), can be made to be commanded geometrically in a temporal order, but they are still different from common assembly robots in the degree to which the world is constrained.

assembly, based on parallelism and concurrency for assembly robots using multiple sensors.

7.3.2 Architecture for Behavioural Modules

The communication architecture between BM's determines how the system is built. A generalised BM is a complete control system where the incoming information is processed in a certain way until the appropriate actions are produced[3]. Hence, communication channels for the input and output are required. Moreover, a BM can be activated, inactivated, or parameterised, such as by commands. In addition, once a command is issued, a lower level BM may be asked to pass over the status or the result of its behaviour, *i.e.*, what it is doing or has done. A bidirectional communication for the command channel is built to meet these requirements. The output channel is connected to the command channel(s) of other BM('s), or directly to the robot. The output channel is bidirectional through which the output is provided and the status and/or the consequence of the actions is reported back if required. The input channel is also bidirectional in order for a BM to be able to parameterise sensing if necessary. Thus, as a generalised architecture of the information path, three way communication is proposed comprising *Input*, *Output*, and *Command* channel:—

Input Information about the world is passed from various source(s), such as sensory information and the status of other BM's. This channel can be multiple if there are more than one source. This may also be used in an output manner by BM's to parameterise sensing.

Output Actions are issued, such as a motor command to the robot and/or a command to another BM. The status and/or the consequence of the action of the recipient can also be obtained through this line, *i.e.*, use in an input manner. This could also be multiple if it concerns more than one output destination.

Command Commands from other BM's are passed, characterised as activation, inhibition, and parameterised commands. The status and/or the result of the

[3] The information is not necessarily received from sensors but may come from other BM's. Also a BM can work by simply following the steps instructed without any other information, such as from sensors.

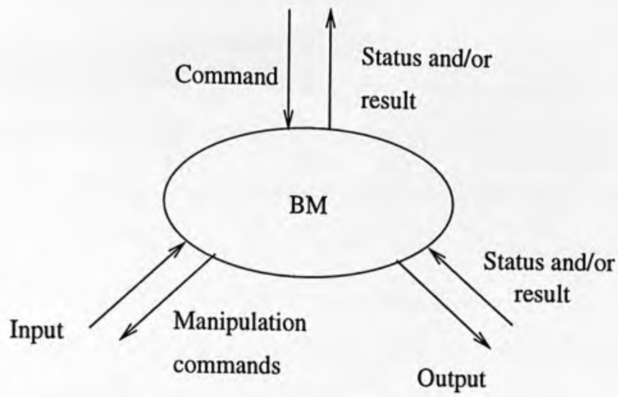


Figure 7.3: The generalised structure of a BM

behaviour may be passed back, *i.e.*, use in an input manner.

The generalised BM structure is shown in Figure 7.3. The communication channels are comprised of *Input*, *Output*, and *Command* channels which are multi-destination or multi-source. How they can be organised is exemplified in Figure 7.4.

This architecture of a BM is inspired by Albus and Powers. In Section 2.4, page 39, Albus' and Powers' hierarchical models of biological systems were introduced. In their models, modularised hierarchical subsystems are available with different sources of information at various levels of sensing abstraction hierarchy, and they issue commands to various levels of motor commands, ranging from single muscle fibre to an abstract movement of a limb.

In Section 2.2.4, page 25, comparison between **micro abstraction** and **macro abstraction** was made. The neuron level abstraction is regarded to be the micro-level of abstraction while behaviour level abstraction is at the macro-level of abstraction. This notion is also applied to the communication hierarchy. In physical communication between the actual biological subsystems, those clustered sets of neurons which exhibit meaningful behaviours, might be built with artificial neurons. It is a purist's problem whether to model every function of the single neurons involved in the communication (a micro-level of description), or to model them from the point of view of appropriate meaningful behaviours (a macro level of description). In this work, the communication means is modeled at the macro (behaviour) level.

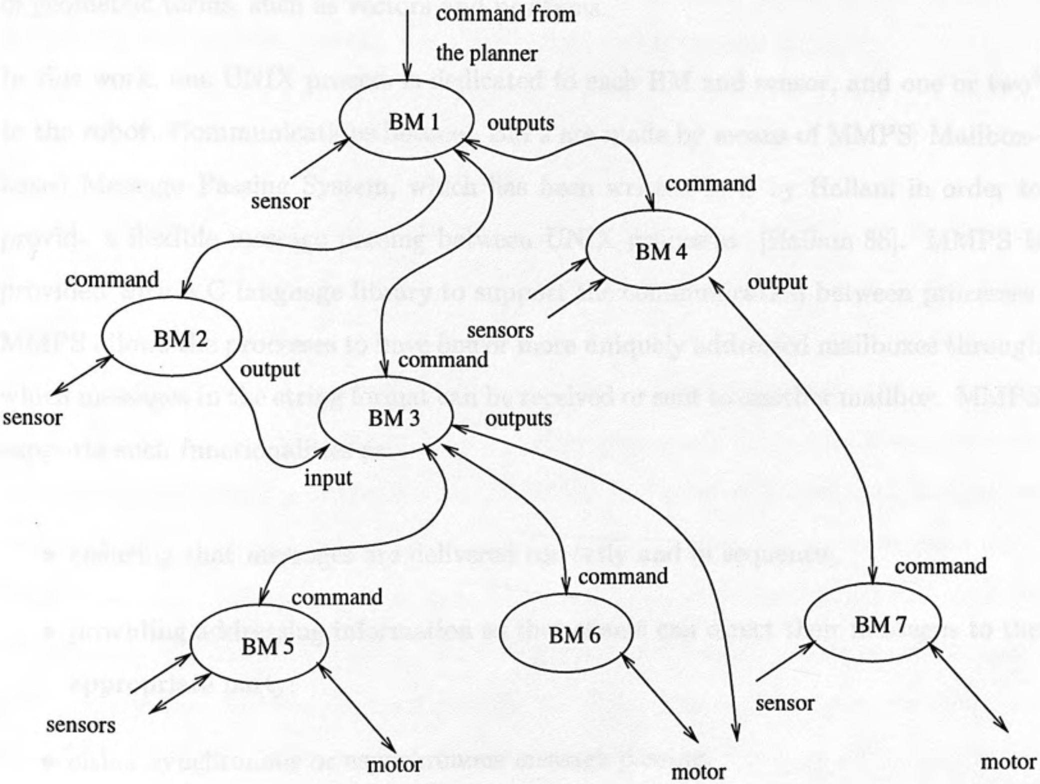


Figure 7.4: An example of organised multiple BM's

7.3.3 Inter-process Communication Between BM's: MMPS

The behaviour–based approach favours minimal communication between BM's. In the Subsumption Architecture, the simplest information passing is as simple as to suppress or inhibit [Brooks 89]. In the Behaviour Language [Brooks 90], Brooks specifies message passing facilities to provide communications between rules which describe AFSM's (Augmented Finite State Machines). On the other hand, an executive agent of a behaviour–based assembly system would require a form of communication of geometric terms, such as vectors and positions.

In this work, one UNIX process is dedicated to each BM and sensor, and one or two[4] to the robot. Communications between BM's are made by means of MMPS: Mailbox–based Message–Passing System, which has been written in C by Hallam in order to provide a flexible message passing between UNIX processes [Hallam 88]. MMPS is provided with a C language library to support the communication between processes. MMPS allows the processes to have one or more uniquely addressed mailboxes through which messages in the string format can be received or sent to another mailbox. MMPS supports such functionalities as:

- ensuring that messages are delivered correctly and in sequence;
- providing addressing information so that clients can direct their messages to the appropriate party;
- either synchronous or asynchronous message passing.

It is proposed that a BM should physically run independently of other BM's, that is, being created or deleted in running time of the whole system without interrupting other BM's. Hence, assigning one UNIX process to one particular BM is desirable.

The communication ports for *Command*, *Input*, and *Output*, are distributed by means of individual mailboxes addressed uniquely.

[4] Three strategies were proposed and tested in order to achieve the ring and wire problem. In the first two strategies, one process was dedicated to the robot, while in the last strategy, two processes were employed for the robot.

One process is dedicated to each of vision and touch sensing. Those processes make information available through the information channels by request from any BM.

The speed of message passing was measured under considerable load on the systems used for the sensor fusion application described later. In order to distribute the work load, 5 SUN work stations are used and the inter-process communications are established throughout the processes running on different machines. In the worst case, approximately 0.1 second was spent to send a message and receive a response immediately between two BM's. The time taken by the communication is not taken into account in the implementation, since it is considered to be fast enough[5].

7.3.4 MMPS Address Server

For a flexible and consistent management of addresses of the BM's, an MMPS address server is employed. It runs in a UNIX process while maintaining a symbol table [Hallam 94] which stores the names of BM's and their MMPS addresses. The addresses of BM's are not known in advance, since in order to eliminate any chance of conflict, MMPS addresses are intended to be assigned appropriately chosen from available addresses at the beginning of the processes (BM's). When a BM is activated, it registers its name (of the behaviour: identity) and the MMPS address of the *Command* port. This name and address couple is stored in the symbol table for further reference by other BM's. For instance, suppose there is a BM called Pushing. It registers its name, Pushing, and its MMPS address assigned for the *Command* port on the server. Suppose there is a BM called Alternating, which uses the behaviour Pushing. Behaviour Alternating knows that it is supposed to contact with the behaviour named Pushing. The Alternating behaviour asks the server for the address (*Command* port) of the behaviour called Pushing. Then, the server looks up the symbol table using the string Pushing as the key. The server then sends the MMPS address of Pushing back to Alternating. By obtaining the desired address, the Alternating behaviour directly contacts the Pushing behaviour with the address obtained. When the Pushing behaviour receives any

[5] The robot speed was set to 10 mm per second, and an maximum error of 1 mm can be introduced even when it is crucial. This is considered to be minor with respect to the dimensions of the test facility, ring and wire. Note too that this is not proposed as the best way to implement this architecture, but merely as a convenient experimental implementation. However, guarded move uses faster connection *i.e.*, from the PC, directly to an ADEPT input signal line.

message, it receives the address of the sender, hence it can send back any information requested, thus bidirectional communication is established. Sensors work in the same way. Any BM which needs to make connection with any sensor, can ask the address server for the MMPS address of the sensor by its name, such as Vision. Then, the BM directly requests for information from the sensor module with the address obtained. This allows BM's to be added or removed while the system is running.

7.4 Sensors

This section describes the sensors used: vision and touch.

7.4.1 Vision

A visual tracker is used for vision sensing. The tracking algorithm is imported from Chongstitvatana's work on an uncalibrated vision system [Chongstitvatana 92], while the feature retrieval algorithm and further processing is added.

As mentioned earlier in this chapter, in order to simplify the vision routine, a white object on a black background is used. On this white object, the contour of the wire is projected so that the vision can process the contour. The vision tracks the boundary between the white object contrasted on the black background.

Part of the black wire is projected on the 7 cm diameter white paper board behind, mounted on the ring handle, while the white coloured ring is seen by the vision to cut the black wire somewhere around in the middle of the 7 cm diameter white round background. Thus the projected contour of the wire is halved by the ring. The end of the robot manipulator including the gripper is hidden by a black paper board mounted on the handle behind the white round board. Figure 7.5 shows the top view of the layout (also see Figure 7.1, on page 160, for the overview). Figure 7.6 illustrates the object possibly seen by the vision[6], and how the outline is processed in order to provide an approximation of the wire contour ahead.

In Figure 7.6, (a) shows the front view of the objects. What is likely seen by the vision

[6] The universal aspect ratio of pixels is not considered in this figure, however in reality, the circle will look as an ellipse since the camera used has 1.5 universal aspect ratio of horizontal to vertical.

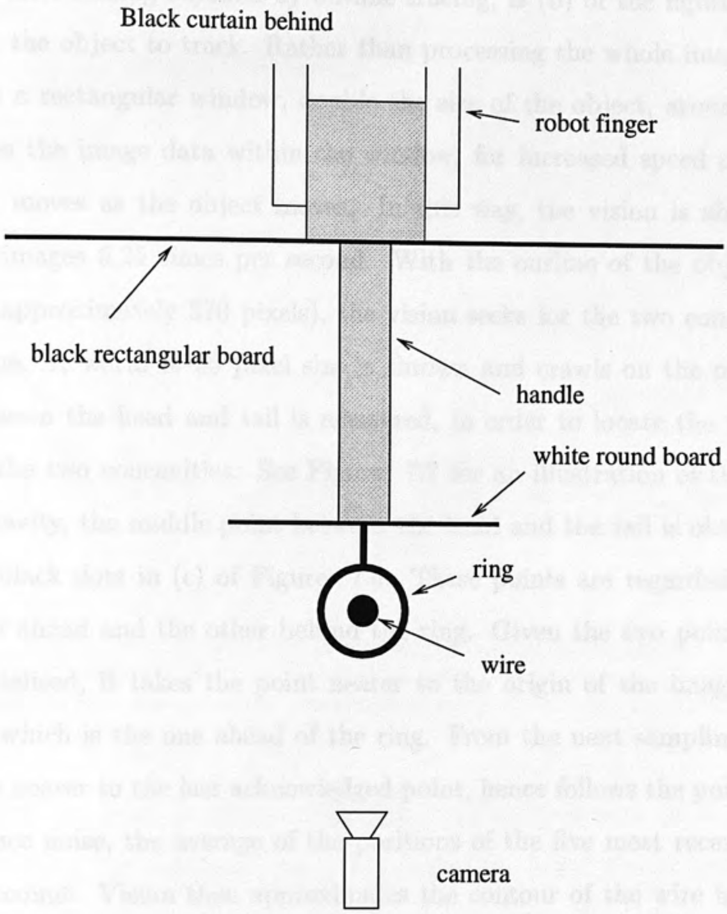
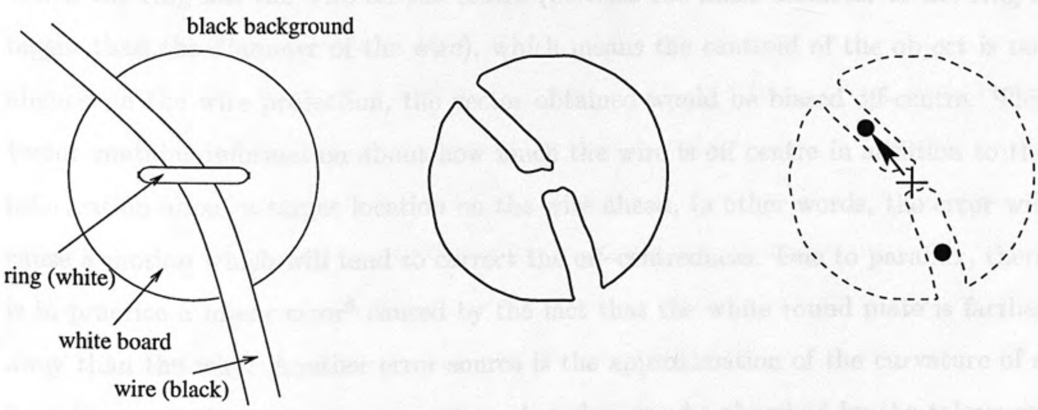


Figure 7.5: Top view of the layout of the objects in the experiment



(a) Actual Layout. (b) An example of outline seen by vision. (c) Vector approximation.

Figure 7.6: Illustration of setup, outline seen by vision, and processing

after binary thresholding followed by outline tracing, is (b) of the figure. The vision takes this as the object to track. Rather than processing the whole image plane, the vision makes a rectangular window, double the size of the object, around the object and processes the image data within the window, for increased speed of processing. The window moves as the object moves. In this way, the vision is able to sample and process images 6.25 times per second. With the outline of the object obtained (a length of approximately 370 pixels), the vision seeks for the two concave features on the outline. A worm of 36 pixel size is thrown and crawls on the outline[7]. The distance between the head and tail is measured, in order to locate the two shortest: presumably the two concavities. See Figure 7.7 for an illustration of the algorithm. At each concavity, the middle point between the head and the tail is obtained, which is shown as black dots in (c) of Figure 7.6. These points are regarded as spots *on* the wire, one ahead and the other behind the ring. Given the two points, when the vision is initialised, it takes the point nearer to the origin of the image plane (top left corner), which is the one ahead of the ring. From the next sampling, the vision takes the one nearer to the last acknowledged point, hence follows the point ahead. In order to reduce noise, the average of the positions of the five most recent points are taken into account. Vision then approximates the contour of the wire by taking the vector from the centroid (marked as a cross in (c), Figure 7.6) of the object to the point obtained, as the vector information which proposes the robot to move towards. When the ring has the wire off the centre (because the inner diameter of the ring is bigger than the diameter of the wire), which means the centroid of the object is not aligned on the wire projection, the vector obtained would be biased off-centre. This vector contains information about how much the wire is off centre in addition to the information about a target location on the wire ahead, in other words, the error will cause a motion which will tend to correct the off-centredness. Due to parallax, there is in practice a minor error[8] caused by the fact that the white round plate is farther away than the wire. Another error source is the approximation of the curvature of a line. However, these errors are so minor that they can be absorbed by the tolerance, the gap between the wire and ring, which is 16 mm.

---

[7] These vision routines employ a notional 'digital worm' as the central algorithm [Malcolm 83].

[8] The error can be approximately 3 mm maximum given the facility layout.

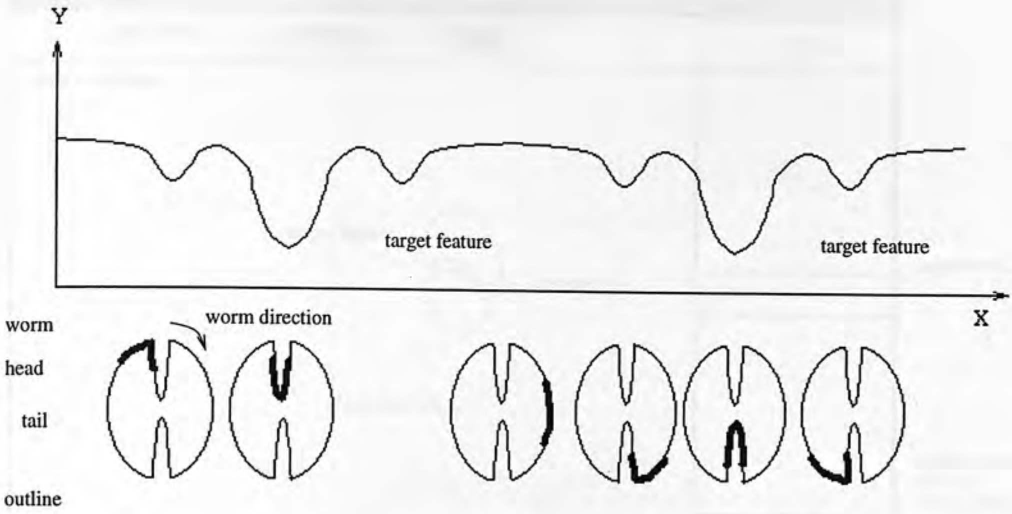


Figure 7.7: Illustration of concavity detection algorithm
(In this hand–drawn illustration, X denotes the positions along the outline and Y denotes the distance between the head and the tail of the worm. Associated positions of the worm on the outline are marked as thick lines below the graph.)

The vision is programmed to occupy a UNIX process and provides the vector information by request of other processes, *i.e.*, BM's, using the MMPS inter-process communication. Any BM, which requires vision information, uses the *Input* communication port to send a request signal and receive the information the vision sends back.

Figure 7.8 shows the display of the vision interface. The big window on the left shows what is seen by the camera: the outline of the object. The centroid is marked as a cross, and the small circle marks the centre of the wire as the imaginary target point ahead of the ring. The shape of the circle appears as an ellipse due to the 1.5 universal aspect ratio of the camera pixels. The actual target vector is produced from the centroid of the object and the smoothed position of the centre of the wire obtained. This vector is displayed on the top right window of the display. This vector is then compensated with the universal aspect ratio, and sent to the client BM's when requested. The middle right window in the interface shows the track of the last twenty moves of the robot sampled. Although this information is not currently made available to other behaviours, it is an extra to be used for a future extension. The bottom right window shows the reference as the z-axis of the robot obtained during the calibration session.

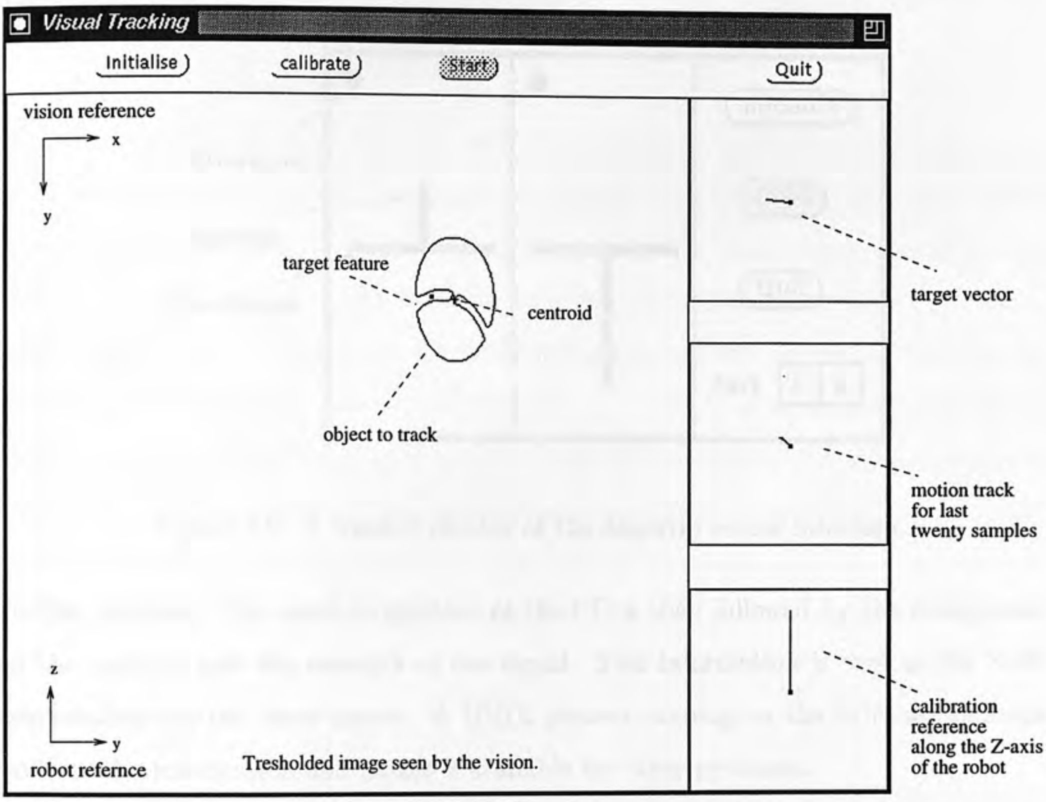


Figure 7.8: X window display of the vision interface

Calibration is required for the vision system in order to provide vector information with reference to the robot world coordinate system. During the calibration session, the robot is driven towards +z axis, while the vision system tracks the movement of the robot. The vision system takes this direction as the z axis. The target vector is later calculated using this reference. If the camera is later tilted, extra calibration will be required. In the figure, the calibration reference happened to be vertical to the ground, which is aligned with the z-axis of the robot.

7.4.2 Touch

The PC for the Push Sensors is instructed to directly notify the robot of any event from both fingers via two binary output channels. This short loop of signal transfer supports quick response of the robot in performing guarded moves. This event signal is then notified to a SUN workstation connected via an RS232 serial communication for

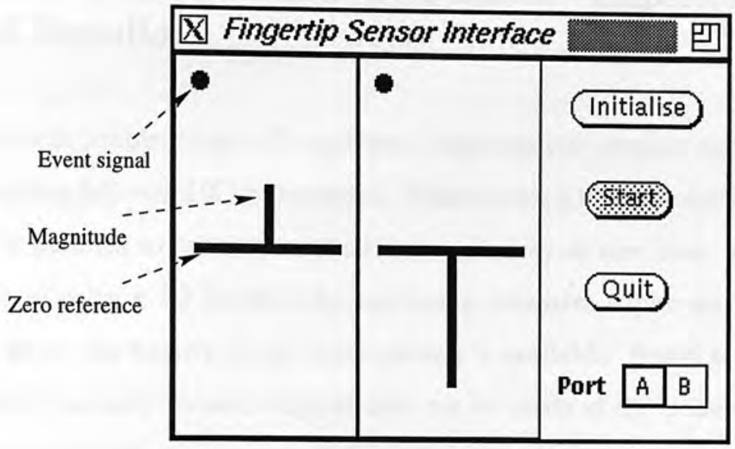


Figure 7.9: X window display of the fingertip sensor interface

further process. The event recognition of the PC is then followed by the recognition of the polarity and the strength of the signal. This information is sent to the SUN workstation via the same means. A UNIX process running on the SUN workstation collects the information and makes it available for other processes.

In this sensor fusion task, only binary event signals are used due to limitations of the touch sensing for the task attempted. Like the vision, the touch sensing interface runs as a UNIX process to distribute the information through MMPS communications. However, unlike the case with vision where clients (BM's) request for information when they need it (information reading), clients register their identity (MMPS address) to the touch sensor interface at the beginning of execution, then the touch sensor interface distributes any event signal to the registered clients as soon as it is detected. The X window display of the touch sensor interface is shown in Figure 7.9. It marks events as black dots on the top left of the sub-windows. The horizontal bars are zeros while the vertical bars represent strength of the signals. Although touch sensors are able to detect contact of ring and wire at the robot speed of 0.3 mm per second when the robot moves orthogonal to the wire, a speed of 10 mm per second is used to make the sensing more reliable, and speed up the performance, while not causing any significant distortion of any part due to the force of collision.

7.5 Three Sensor Fusion Strategies: Implementation, and Results

The ring and wire problem has a 3D problem. There are two sensors, each being incapable of providing full useful 3D information. Vision solves a two dimension uncertainty while touch is planned to solve one dimension uncertainty at one time. Touch sensors may be used to solve a 3D problem by employing extensive exploratory movements particularly when the history of the robot motion is available. Based on the contour of the curvature recently passed, a hypothesis can be made of the possible contour in front. Then, exploring behaviours could be activated to find out the actual curvature. Furthermore, the problems of sensor fusion can be addressed by adding vision to the existing configuration. This might be one method to achieve the fusion of multiple sensing with the current experimental configuration.

However, the problems of combining multiple sensing arise from more fundamental considerations. Vision and touch are very different from each other in terms of sensing modality. How they could be combined from the bottom is a fundamental problem particularly in a distributed system, such as built by the behaviour–based approach. In the behaviour–based approach, sensing and action couplings should be implemented at as low a level as possible; should exploit motion in assisting sensing; should use the environment if possible (from the guidelines, Section 2.1.3). Thus, these are the fundamental questions:—

- At how low a level can the sensors be fused?
- How can actions help sensor fusion?
- How can sensings be fused using the active involvement with the environment possibly using *unlocated* representations?
- How to organise distributed sub-modules, *i.e.*, BM's?

This section describes the implementations of the three sensor fusion strategies, followed by their results. In the first two strategies, *Interlacing* and *Vector summation*, the robot is used in a classical way of discrete position control. A robot interface was implemented to run on a UNIX process which passes commands from various concurrently running BM's to the robot. In the third strategy a robot motion command is issued with continuous positions, by means of which in effect the robot is driven in the manner of velocity control. For this purpose, the robot interface is updated and at the same time the capability of VAL II of hosting two concurrent processes is used.

7.5.1 Robot Interface

An ADEPT1 robot with VAL II is used for this experiment. It is accessible from a workstation via an RS232 serial port. Monitor commands can be issued, such as motion commands and execution of a VAL II program. Since SUN workstations running UNIX are used as the computing means with which sensors are physically connected and processed, a UNIX process is dedicated to the robot interface for a convenient inter-process communication. The robot interface receives motion commands from other processes and passes these to the ADEPT robot. A command can be a single VAL II monitor command, or a command to execute a pre-stored VAL II program on the ADEPT controller. The commands comprise un-guarded moves, guarded moves, taking current location, *etc.*

7.5.2 Strategy I: Interlacing

First, the ring and wire problem has been tackled in a primitive way. The robot pushes or pulls the ring guarded by the touch sensors, then shifts guided by vision, then pulls or pushes the ring. This is repeated in order to follow the contour of the wire.

The two dimensional vision is dedicated to the plane Y and Z, in the robot's global reference (see Figure 7.2). Touch is dedicated to the remaining axis, X. Guarded push and pull motions are repeated by a BM called Alternating. Alternating has two other BM's to deal with: Push and Pull. Push employs a guarded move towards +X until the ring makes contact with the wire, while Pull moves towards -X.

The Alternating behaviour alternates Push and Pull respectively. By this behaviour uncertainty along X can be solved because the robot has an idea about the position on the wire where contact is made. The Alternating behaviour has access to touch sensor

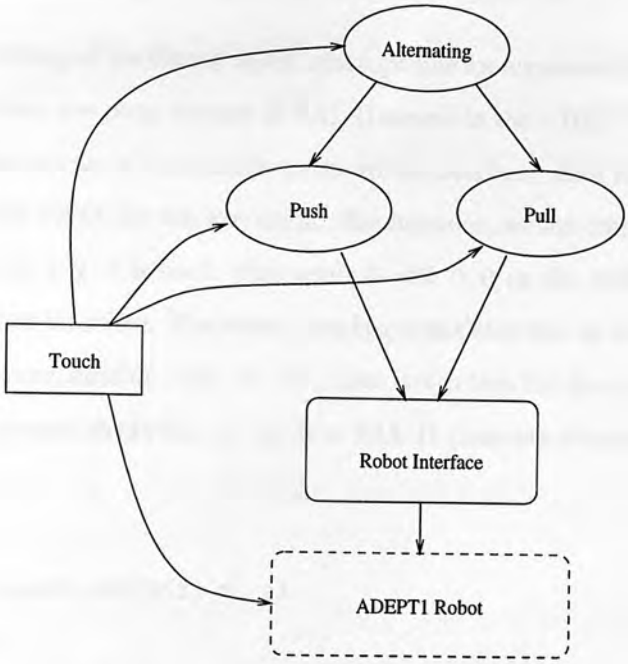


Figure 7.10: The system architecture with Alternating behaviour

interface for error detection. If any one task, Push or Pull, is terminated without any signal from the touch sensors, Alternating regards this as an error and gracefully exits (*i.e.*, the process is terminated). The system architecture is shown in Figure 7.10.

Push and Pull behaviours move the ring away from the wire after the contact by about a quarter of the inner diameter of the wire. If the robot is stopped by a guarded move with the touch sensors triggered, then the departing motion may also trigger the touch sensor in the opposite way. This is because having made a contact, the fingers stay compressed. Because the sensors discharged the previously applied charge at making the contact, they are ready for the next event to come. When the contact is broken, the compressed fingers will return to the normal state, and this will be detected by the sensors, because the sensors detect the change of the exerted force to the sensor body. This is an unwanted effect for the application described in this chapter. If the robot starts Pull with the ring still making contact, the guarded move of Pull will stop the robot immediately as the contact is cleared. A clearing motion from the contact is necessary in order to have the correct information. One quarter of the inner diameter of the ring for the distance of clearing motion was determined by experiment to have a successful contact clearing action.

Setting and un-setting of the binary input interrupt line for a guarded move, and a series of necessary motions are programmed in VAL II stored in the ADEPT controller. Push and Pull send the execution commands to the robot interface, then the robot interface passes these to the robot for the execution. For instance, as the command format for a guarded move, `G x y z` is used. Pull sends `G -20 0 0` as the string command via MMPS to the robot interface. The robot interface translates this to the actual monitor command `exe guard_shift( -20, 0, 0)`, then issues this for the ADEPT robot via an RS232 line. `guard_shift(x, y, z)` is a VAL II program stored in the ADEPT controller:

```
.PROGRAM guard_shift(x, y, z)
     BREAK
     SPEED 10 MMPS ALWAYS
     REACTI 1001, stophere       ; set motion interrupt
     REACTI 1002, stophere       ; for a guarded motion.
     MOVES SHIFT(HERE BY x, y, z)
     BREAK
     IGNORE 1001
     IGNORE 1002
     BREAK
     STOP
.END
```

The purpose of sensor fusion is to combine different sensing modalities into one towards the purpose of the task. The behaviour–based approach encourages the incremental growth of the system as the process of development of the whole performance. One of the ways to achieve this in the task is to add a behaviour which uses visual information to solve the Y–Z problem, on top of the existing structure. The *Interlacing* strategy is employed in order to achieve this: shift motions in the image plane, (Y,Z) are interlaced between Push and Pull, also Pull and Push. By Push or Pull, the location of the wire on the X axis is identified, then a step of the shift motion is performed towards the direction which vision tells. Hence, the procedure would be: ... Push – Shift – Pull – Shift – Push – Shift .... For the extension, Alternating is facilitated with the inhibit/uninhibit nodes and the reporting of its status (*i.e.*, such as whether it is performing pulling or pushing) through its *Command* communication channel.

A behaviour called Shift is employed. Shift takes the parameterised command, a command with a vector information, from any other BM through its *Command* communication channel. As soon as it receives the command, it converts the vector information to the YZ position information to which the robot is to travel, then sends the command to the robot interface. Shift drives the robot towards the vector for 6 mm, which is considered as a maximal appropriate lateral move given the geometry of the environment. This shift motion is also guarded.

A behaviour called Interlacing is employed which interrupts Alternating at the end of each pulling or pushing step, and interlaces Shift. While Alternating works normally, Interlacing requests Alternating for the information about the behavioural status. Once requested, the Alternating sends the information about the current behavioural status to Interlacing repeatedly. When Interlacing receives a signal from Alternating, either Push or Pull, Interlacing sends an inhibit signal to Alternating. At the termination of the current behaviour, Push or Pull, Alternating checks if there is an inhibit signal, and if there is, it sends back an acknowledgement and stops moving to the next step while waiting for an uninhibit signal. Confirming that Alternating acknowledged the inhibit, Interlacing activates Shift with the vector parameter, which is read from the vision sensor interface. At the termination of Shift, Interlacing sends an uninhibit signal to Alternating, then the next step of Alternating is resumed. By this procedure, Shift is interlaced between Push and Pull, and Pull and Push. Then the robot is able to follow the wire with the probing motions interlaced by Shift. The complete architecture of the BM's is illustrated in Figure 7.11, which embraces the architecture shown in Figure 7.10 for Alternating.

This method of fusion provides the feature of incremental growth of the system, proposed by the behaviour-based approach. On top of the existing Alternating behaviour, an Interlacing behaviour is created. The Interlacing behaviour adds the Shift behaviour on the existing structure by coordinating sub-modules in terms of timing sequence.

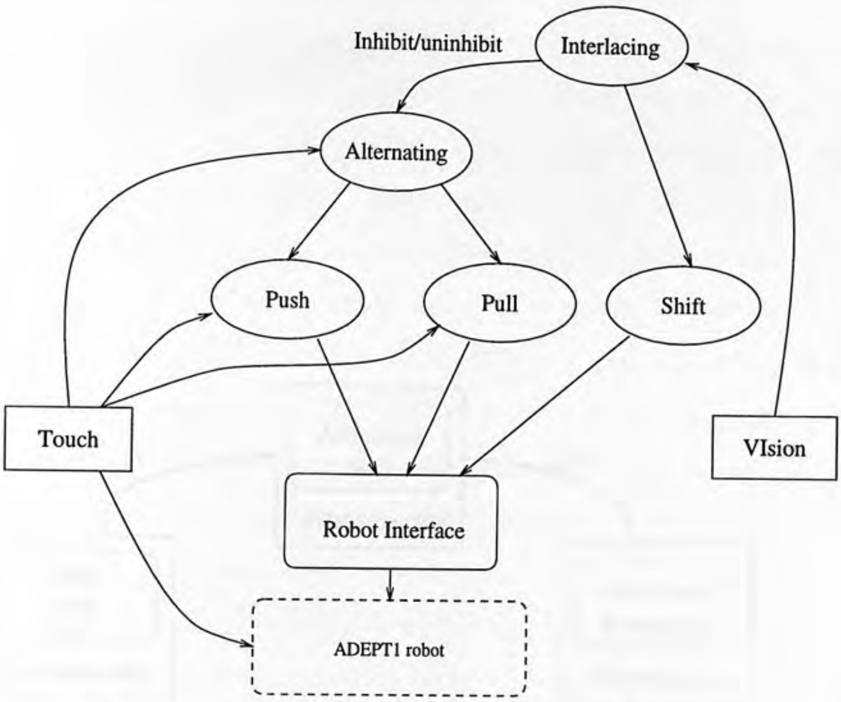


Figure 7.11: The architecture of BM's for the strategy I: interlacing

To increase the flexibility of the system, higher BM's can be added or deleted during running time. Hence, Interlacing can be started while Alternating is working and it can be also deleted while the system is running, leaving Alternating to continue its job. If any BM is disabled by accident, the rest of the system retains its functionality[9]. In the behaviour based approach, this feature is considered to provide robustness and convenience in adding to the system functionality. The distributed and concurrent architecture enabled this feature to be demonstrated in this experiment.

In order to distribute job load, 5 SUN workstations are used to run various BM's. Figure 7.12 shows how these machines are related.

One of the purposes of this experiment is to decide whether the incremental growth

[9] More precisely, if Interlacing is disabled, Push and Pull will work governed by Alternating. If Alternating is disabled, although Push and Pull would retain the functionalities, they will not be executed. The performance will fail by having just Interlacing and shift working. Push, Pull, or Shift can be disabled leaving the rest of the BM's working, but will cause failure of the performance. However, because the robot will continue pushing and pulling guarded by the touch sensors, with Interlacing or Shift disabled on line, this feature of concurrency and distribution can be claimed to support system robustness. Robustness is related to how BM's are designed and whether there is any redundancy. This point is discussed in more detail in Section 7.6.7, page 191.

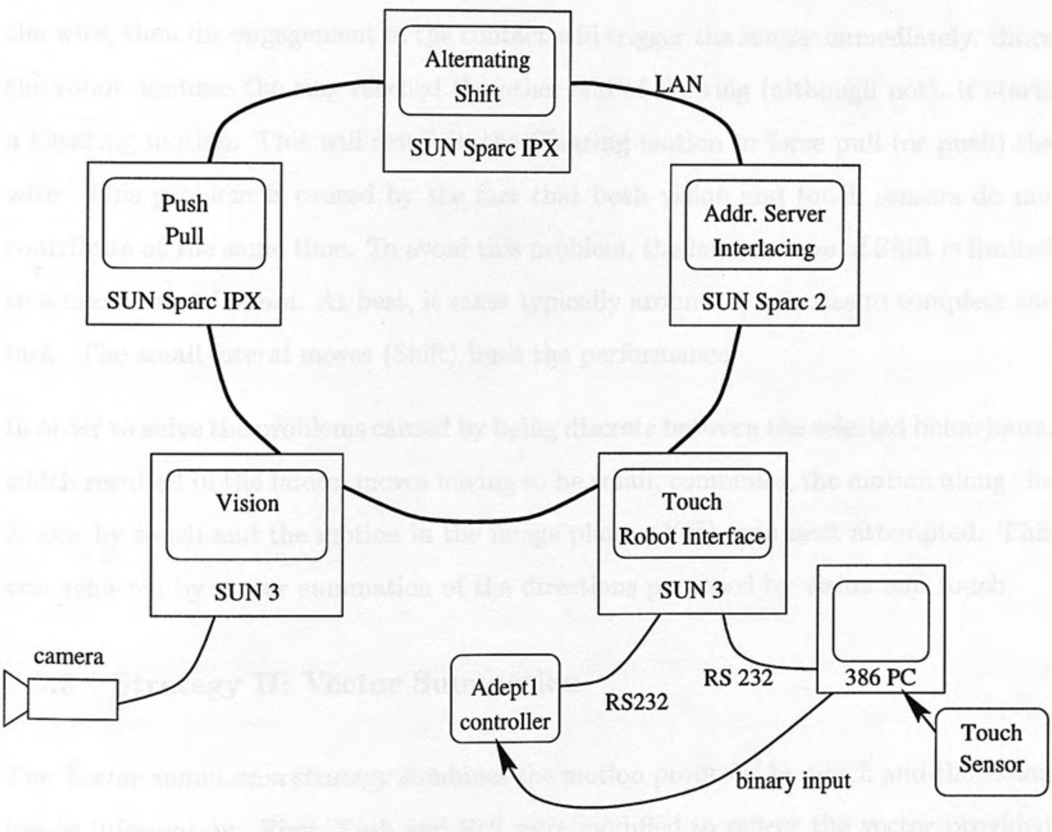


Figure 7.12: Interconnections between the various resources

of the system, proposed in the behaviour–based approach would be suitable for sensor fusion for assembly robots. By observation of the experimental implementation, it is argued that although the task is satisfactorily performed, the manner in which the robot performs is fairly primitive. The sequence of the actions are characterised as *bump, shift, bump shift,....* Although the sensors are fused in the behaviour space, the sensors are selected in a discrete manner. This discrete manner causes a functional problem: if the ring and wire is left touching after the behaviour Shift, the task's performance can be spoiled. If Push (or Pull) starts while the ring makes contact with the wire, then dis-engagement of the contact will trigger the sensor immediately. Since the robot assumes the ring reached the other end of the ring (although not), it starts a Clearing motion. This will result in the Clearing motion to force pull (or push) the wire. This problem is caused by the fact that both vision and touch sensors do not contribute at the same time. To avoid this problem, the lateral move of Shift is limited to a maximum of 6 mm. At best, it takes typically around 15 minutes to complete the task. The small lateral moves (Shift) limit the performance.

In order to solve the problems caused by being discrete between the selected behaviours, which resulted in the lateral moves having to be small, combining the motion along the X axis by touch and the motion in the image plane, (Y,Z), was next attempted. This was achieved by vector summation of the directions proposed by vision and touch.

7.5.3 Strategy II: Vector Summation

The *Vector summation* strategy combines the motion proposed by touch and the vision vector information. First, Push and Pull were modified to reflect the vector provided to the probing guarded moves, *i.e.,* to perform a vector summation between the two kinds of motions proposed by two different sensing: motion proposed by vision and motions along X axis for probing guarded moves associated with touch. Alternating was modified and named Vector Alternating, to get access to visual information directly. Push and Pull are improved to comprehend the vector information received from Vector Alternating, and named Vector Push and Vector Pull respectively. In the same manner as the previous *Interlacing* strategy, Vector Alternating activates Vector Push or Vector Pull in order, with the vector parameter. Vector summation physically takes place in

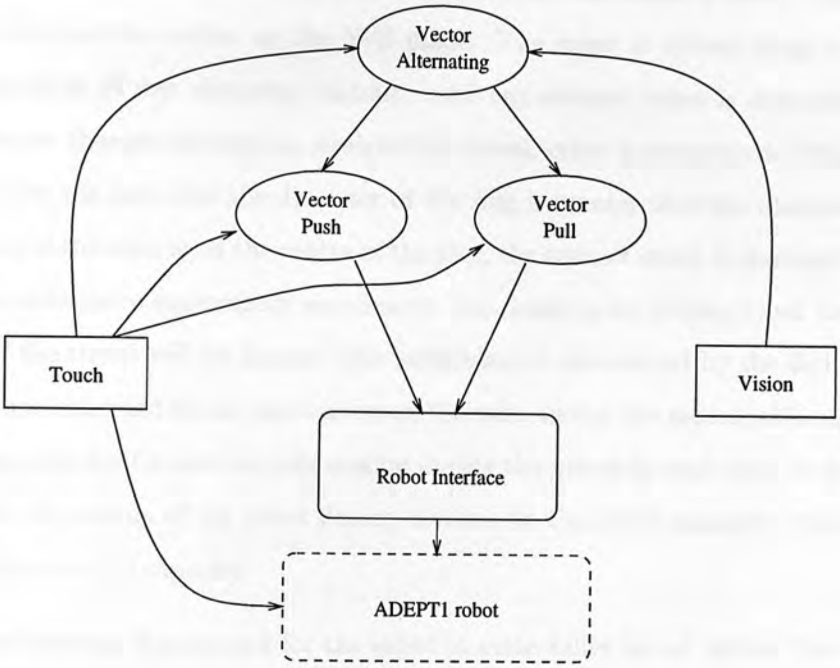


Figure 7.13: The architecture of BM's for the strategy 2: vector summation

Vector Push and Vector Pull by substituting decomposed vector values (Y and Z) with appropriate variables in the guarded move command `G x, y, z`, which is to be sent to the robot interface. Then the robot interface calls the VAL II program, guard_shift(x, y, z)[10], with the 3D information. In result, shift step is eliminated and the probing motion smoothly embraces the motion along the length of the wire: (Y, Z), guided by vision. The architecture of the BM's are illustrated in Figure 7.13. Vector Alternating has also access to the touch sensor interface for error detection, *i.e.* when there is no touch signal for a period of time after an execution command was issued, which means something must have gone wrong with one of the subordinating BM's.

In contrast to the *Interlacing* strategy, this *Vector Summation* strategy is more efficient. While the best performance of the interlacing architecture is typically around 15 minutes to complete the task, the vector summation architecture is able to complete it typically in 6 minutes, with a much reduced number of movements.

However, the *Vector summation* strategy still has a crucial deficiency. Before Push or

[10] See page 177 for the detail.

Pull is invoked, the contour vector information from the vision is read. This vector is used to direct the robot on the Y–Z plane. The robot is driven along a straight line, regardless of any changing contour, until any contact event is detected. If the wire contour changes during the straight line travel, error is introduced. This error is absorbed by the fact that the diameter of the ring is greater that the diameter of the wire. But, if the wire is off the centre of the ring, the contact event is met earlier. This will introduce more exploratory movements (*i.e.*, pushing or pulling), and the overall length of the travel will be longer. The inefficiency is introduced by the fact that the robot is not informed about the contour of the wire *during* the moves, although vision is able to provide the contour information during the moves in real time. It is difficult to update the status of the robot during motion. In the above example, vision is not exploited to its full capacity.

The third strategy is proposed for the robot to make fuller use of vision. The robot is driven in a manner such that the target position is kept changing as quickly as possible, which virtually simulates velocity control. The robot smoothly follows the contour of the wire on the Y–Z plane, which is more effective than the *Vector summation* strategy. This third strategy, *Continuous vector summation* is explained in the next section.

7.5.4 Strategy III: Continuous Vector Summation

In *Continuous vector summation*, the robot is instructed to adapt its motion smoothly to quickly changing incoming positional information, *i.e.*, from vision which (in this implementation) provides about six readings in a second. Although a position controlled robot should be able to approximate this kind of operation by changing its target location quickly, VAL II is not equipped with any appropriate straightforward facility. However, the capability of VAL II of allowing the user to run two processes simultaneously is utilised to find a way of doing this.

Two programs can be concurrently run in VAL II: Robot Control (RC) and Process Control (PC) programs [Ade85b]. An RC program directly controls the robot, while a PC program cannot directly influence the robot motion, but it can get access to binary signals and external communications. A PC program can communicate with an RC program through shared variables and software signals. A PC program was written such that it reads ASCII numbers from a serial port through which the positional information is passed. The PC program keeps changing three (XYZ) global variables (target positions) as quickly as possible accordingly to the incoming information. An RC program is written to repeat a motion instruction with the target position characterised by the three global variables, until any event signal is detected from the touch sensors. The PC program is left running all the time, while the RC program is invoked by any BM when necessary. By this way, the instantiation of the robot motions is governed by the BM's, while a stream of positional informations can be sent to the robot. In effect the robot is making lots of small movements run together, with the target destination being changeable between one move and the next.

The *Continuous vector summation* strategy is illustrated in Figure 7.14. The robot motion vector interface is employed to collect motion vector components from various sources asynchronously. It sends the vector information to the PC program of the robot via a serial link. The robot command interface relays any motion commands received from any BM. Vision is modified to provide vector information continuously at every sample to any registered client. The Vector decompose module converts incoming vision information to a desired target relative location of a distance of 2 mm, decomposes the vector into Y and Z components, then sends these Y and Z values to the robot motion vector interface. Since at the preset speed of 10 mm per second, the robot is not able to terminate one move, *i.e.*, a distance of 2 mm, before the next information from the robot motion vector interface (as the target position – relative to the current position) is received, the robot smoothly blends its motions according to the changing target positions. At an associated signal from Alternating behaviour, the Push or Pull behaviour issues a guarded motion command to the robot command interface, and at the same time provides appropriate vector component value of a size of 2 mm on the X axis for the robot motion vector interface.

While the robot pushes or pulls the ring, the robot follows the contour of the wire on the Y–Z plane which the vision covers. Touch is used by the robot control at the lowest level to interrupt actual motion of the robot for the guarded moves, and is also used by other BM's (*e.g.*, Alternating) to monitor for errors.

With only the Alternating behaviour activated, the robot moves about only the X axis

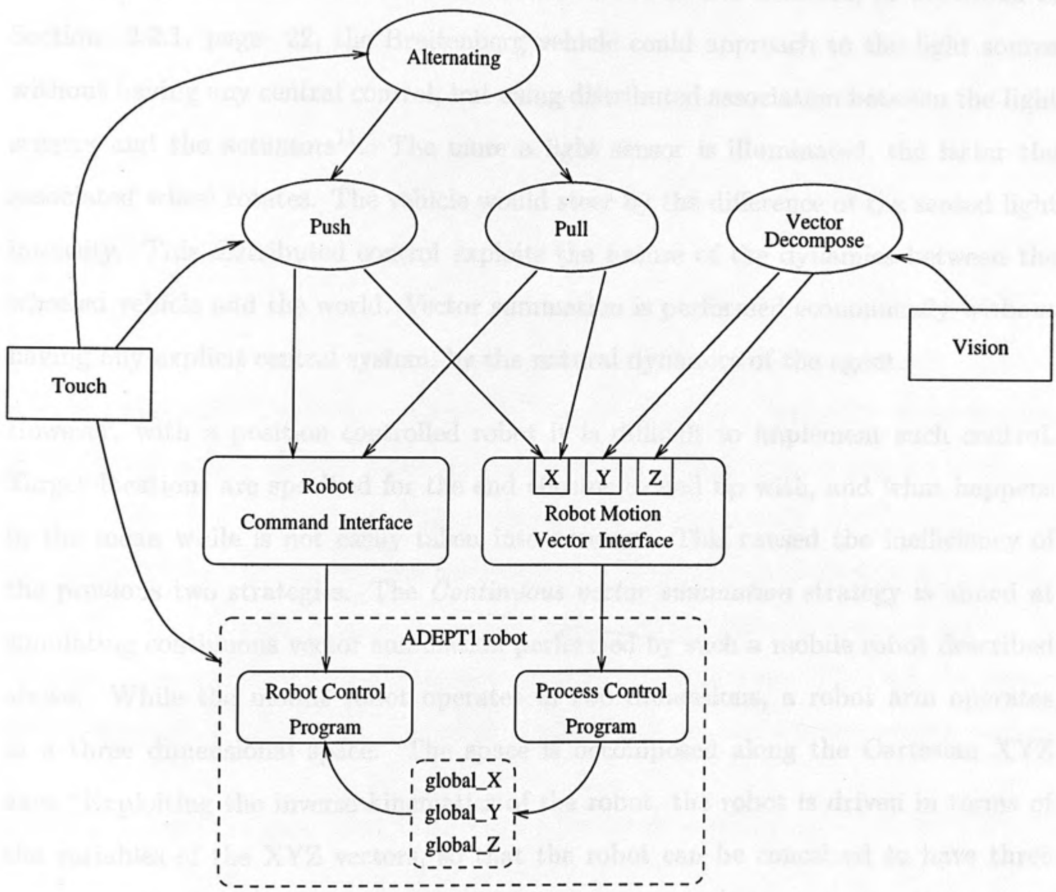


Figure 7.14: The illustration of the Improved Vector Summation strategy

with the guarded pushing and pulling motions. Vision can be activated while existing BM's are running. During running time, vision can be deleted without affecting the execution of other BM's associated with touch sensing, which will result in the robot to only push and pull the ring. This can be characterised as graceful degradation demonstrated by this distributed system.

Mobile robots can be built without having any central control, their actuators can be controlled by isolated sensors in a distributed manner. For instance, as described in Section 2.2.1, page 22, the Braitenberg vehicle could approach to the light source without having any central control, but using distributed association between the light sensors and the actuators[11]. The more a light sensor is illuminated, the faster the associated wheel rotates. The vehicle would steer by the difference of the sensed light intensity. This distributed control exploits the nature of the dynamics between the wheeled vehicle and the world. Vector summation is performed economically without having any explicit central system, by the natural dynamics of the agent.

However, with a position controlled robot it is difficult to implement such control. Target locations are specified for the end effector to end up with, and what happens in the mean while is not easily taken into account. This caused the inefficiency of the previous two strategies. The *Continuous vector summation* strategy is aimed at simulating continuous vector summation performed by such a mobile robot described above. While the mobile robot operates in two dimensions, a robot arm operates in a three dimensional space. The space is decomposed along the Cartesian XYZ axes. Exploiting the inverse kinematics of the robot, the robot is driven in terms of the variables of the XYZ vectors, so that the robot can be conceived to have three actuators along the axes. Visual information is fed to the Y and Z virtual actuators and Push and Pull generate the X vector, in association with touch sensing. Continuous vector summation is achieved in a distributed manner at effectively a low level. This is a way of implementing sensor fusion in a distributed manner where BM's, associated with different sensors could contribute to the robot motions independently.

The result is the robot motions smoothly following the contour of the wire on the Y–Z

[11] However, the Braitenberg's vehicle is not a real implementation, but rather imaginary for the purpose of discussion. A real implementation of the like is Webb's phonotaxis cricket robot, which is mentioned in Section 2.2.2, page 23.

plane which the vision covers. This reduces the number of bumps, and the task is typically completed in 4 minutes, which is two minutes quicker than in the case of the strategy 2: *Vector summation.*

However, in the implementation, the vector information channel of one axis is accessed by a single information source. If more than two information sources (*e.g.*, BM's) contribute to the robot motion along one axis, compromise and arbitration would be needed. Vector summation can be easily achieved, but more sophisticated treatment would require further consideration. Future work would be centred around the generalisation of the architecture particularly on the conflict resolution facility, and the robot command and motion vector interfaces.

7.6 Discussion

This chapter described the behaviour–based sensor fusion experiment using the touch sensors developed. Touch sensors are used for guarded moves, which lie at the bottom of the sensor fusion hierarchy. Vision is used as the complementary sensing to touch. Three sensor fusion strategies were proposed and implemented: *Interlacing*, *Vector summation*, and *Continuous vector summation.* The system architecture is based on the Albus' and Powers' models, and the guidelines of the behaviour–based approach. The BM's are executed in a distributed and concurrent manner. Through the sensor fusion experiment performed, a number of discussions are made under various topics.

7.6.1 Touch Sensors for Guarded Moves in Assembly

The touch sensors developed were tested in guarded moves and provide sufficient information to solve one degree of freedom when used with robot motion. The guarded moves form the basis of the control hierarchy on which other more sophisticated functions are added. Since for assembly robots, detection of part contact is fundamental, guarded moves can be extensively applied to assembly tasks.

Guarded moves are used in a dynamic manner by Nature. The fact that the robot moves is an extra information in addition to the event signature. More information is retrieved using a robot motion, *i.e.* motion helps sensing. In addition, the Push

Sensors proved to be robust enough to be used for a long term experiment (about a year) without any significant damage, although the rubber skin degrades to lose some compliance and hence becomes more fragile. Using a longer lasting plastics–based rubber, such as butyl, would remove this problem.

7.6.2 Incremental Growth

Based on a mobile robot with wheels for example in general, incremental growth of the system would start from the bottom: with the motors for the wheels. Although it looks primitive, it has at least a complete set of means of locomotion with regard to the robot's task space: navigation on the ground – a 2D problem[12]. However, the *Interlacing* strategy in the experiment in this chapter starts with 1D, then extends it to 3D by adding more BM's in a discrete manner. Compared to the mobile robot exemplified above, in *Interlacing*, the incremental growth of the system did not start from a full 3D mobility necessary for the task. *Vector summation* and *Continuous vector summation*, however, used their 3D mobility from the beginning. These showed more graceful growth of the system. In particular, the *Continuous vector summation* strategy employs a flexible robot interface which could interpret continuous vector information from different BM's with full XYZ mobility. Additional BM's could flexibly be added incrementally in order to sophisticate the robot motion.

7.6.3 Localised Fusion

The *Vector summation* strategy has a centralised control since the Vector alternating behaviour explicitly combines visual information with the guarded moves. However, this explicit central representation is confined to this module, *i.e.*, it is not a global system representation. Behaviour–based systems advise against the use of explicit knowledge. Here we have had to weaken this to mean that explicit global knowledge should be avoided, if explicit knowledge is necessary it is preferable that it be local.

How to organise distributed BM's? In addition to implicit fusion, localised fusion is proposed in line with the amalgamation of sensing and action proposed by the

[12] However, note that the first level competence of a hexapod robot, Genghis, for example, is just to stand up [Brooks 89].

behaviour–based approach. Rather than having a global centralised system, sensor fusion takes place at various levels as low as possible, distributed throughout BM's. This would provide a way to overcome the limitations in using the environment, while taking advantage of the feature of the behaviour–based approach. Given the concept of the robot interface built for the *Continuous vector summation*, any conflict resolution mechanisms could be built at the lowest possible levels as characterised by the localised fusion.

7.6.4 Low Level Amalgamation of Sensing and Action

In a broad sense, separation of the planner and the executive agent for behaviour–based assembly robots is a process of amalgamation of sensing and action. In behaviour–based assembly robots, the planner is ignorant of the details of sensing and action, while performing reasoning about the minimalised symbolic space. The details of sensing and action are amalgamated in the executive agent.

However, amalgamation of sensing and action is not only limited in the separation of the planner and the executive agent. A BM can have a sophisticated functionality managing other BM's. Amalgamation of sensing and action at the lowest possible level would be advantageous in building such a sophisticated BM. As mentioned above, sensor fusion can also take place at as low a level as possible in order to provide more simplicity for the BM's to be built upon.

Then, at how low a level can sensing and action or sensor fusion be amalgamated? The lowest possible level is thought to be the physical level, using *unlocated representations* (see Section 2.4). The *Continuous vector summation* strategy tried to simulate this feature, by separating the motion into XYZ components of the robot. Each component has its specific sensing incorporated, and the components are then recombined. Sensor fusion took place at the level of Cartesian motion level. It can also take place at the joint level in the same manner as Rucci and Bajcsy [Rucci & Bajcsy 95] showed by direct mapping in the case of using one sensor.

7.6.5 Position Controlled Robots Inappropriate

It is argued that current position controlled robots are not suitable for the behaviour–based approach by their nature. In the behaviour–based assembly system, sensing and actions are desired to be implemented at as low a level as possible — low level amalgamation of sensing and action. A robot must be able to comprehend a command in the form suitable for the nature of the sensor, in order to minimise any intermediate interpreter.

As seen in the experiment, a direction vector is more important than any absolute position for an assembly robot to interact with the uncertain environment assisted by the sensors, such as in a guarded move. In the experiment, vectors were turned into desirable end locations for the robot to move to. This step would be unnecessary if the robot could accept a vector command directly. Moreover, the direction of motion of the robot cannot be changed until it terminates the move or is interrupted. This is a drawback of the usual implementation of a position controlled robot, which happen to omit vectored motion[13]. Fortunately, it was possible, although not straightforward, to simulate updatable vectored motion in the ADEPT (VAL II).

Although for very different reason, it is interesting to note that Deacon too concluded that current position controlled robots are not suitable for assembly tasks [Deacon 97]. A force controlled robot was built using direct drive motors, to perform such tasks as peg-in-hole and crank turning very efficiently with simple sensing and motor loops. The arm is compliant and the commands are issued in terms of the force towards a target direction. It simulates human performance in assembly tasks in a more natural way than a position controlled robot.

7.6.6 Distribution and Concurrency

One of the difficulties in the sensor fusion demonstrated in this chapter is that a single robot interface had to be employed while the BM's are distributed and run in parallel, since the robot can only perform one motion command at one time. Although

[13] This is simply an observation of the way this particular implementation was developed.

the ADEPT1 robot used has five joint motors[14], it looks like one virtual actuator through inverse kinematics. Parallelism and concurrency is limited by this bottleneck. This problem is partly solved in the third strategy *Continuous vector summation* by separating information channels for the three Cartesian coordinates.

However, distribution enabled a comprehensive modularisation and maintenance of BM's, and the concurrency ensured the robustness against any failure. Moreover, the concurrency also helped the sensor modules to be used in a flexible manner. For instance, any new BM can get access to the sensor working for existing BM('s) on–line.

Although distributed, BM's require behavioural synchronisation. For instance, for Interlacing, Shift has to be called appropriately between Pull and Push. This is managed through the communication between Interlacing and Alternating. However, for more simplicity, in the behaviour–based approach, communication through the environment is encouraged, as the Herbert robot does [Connell 89]. Unfortunately, this is often an *ad hoc* matter of exploiting the unexpected, and depends on the ingenuity of the designer as much as anything else.

Sensings could be fused tacitly (*i.e.*, without using any explicit central knowledge) by making use of the servoing capability of the robot in the *Continuous vector summation* strategy. However, this is only one example of using active involvement with the environment. Many other different variations are possible, *e.g.*, as in Connell's Herbert robot, where communications between different behaviours are made through the environment [Connell 89], *i.e.*, execution of one behaviour affects the environment then some other behaviour is triggered by its own sensing of this change.

7.6.7 Redundancy and Robustness

Whether an assembly robot would need robustness is discussed. The robustness encouraged by the behaviour–based approach is characterised as graceful degradation: the failure of any one part of the system should not cause the whole system to fail, but merely loses some behavioural competence. However, in order to achieve this, sufficient redundancy should reside in the system to complete the tasks requested. During the

---

[14] Note, however, only three joints were used in the experiment.

Interlacing strategy, the Interlacing behaviour can be deleted leaving the Alternating behaviour successfully working, activating Pull and Push in turn. However, although the whole system still retains some mobility, it would not complete the whole task. This is because there is insufficient redundancy.

Whether the robustness is useful or not depends on the purpose of the task and what resources and functions the robot was provided with. For instance, suppose the purpose of a hexapod robot is walking, and this robot is designed to be able to walk with only four legs (*i.e.* with any two removed), although the motion would be jerky and speed be slow. The robot can be regarded to be sufficiently robust for walking if one does not care about the speed and behaving elegantly. However, if the robot were made to pick up an object using an arm, and by chance the robot happened to lose its arm function, then the robot will lose the competence of picking up an object, *i.e.* arm functions are not robust, unless some of the legs were designed to lift an object. The *robustness* discussed in the behaviour-based approach has an assumption that there be behavioural redundancies depending on the purpose of the task.

7.6.8 Active Information Reading vs. Passive Information Receiving

The manner of use of the touch sensor and the vision were different. Touch sensors were used to *notify* events, whereas vision makes information available for any BM to *read*. In other words, for vision, BM's actively read the value (except for the *Continuous vector summation* strategy), and for touch sensors, BM's passively received the signal when it occurred.

It is argued that rather than vision being used in the manner that BM's actively read information, vision can be better used by providing information continuously to BM's, in order to provide a low level with more functionality, such as servoing. Vision can be used in the same manner as in phototaxis and phonotaxis, where sensing is more directly connected to the actuator, as shown in the *Continuous vector summation* strategy.

7.6.9 How Action Can Help Sensing

In the *Vector summation* and the *Continuous vector summation* strategies, the touch sensor space was converted into a motion space by using the sensor with motion. Sensors (vision and touch) were fused in the motion space using vector summation. Since they have the same attribute, the sensor fusion could be straightforward. In this sense, action helped sensor fusion.

7.6.10 Generalisation of BM Structure for a Robot Controller

The generalised communication architecture for a BM, with *Input*, *Output*, and *Command*, was general enough for the requirements of the experiment. Multi-tasking with inter-process communication and flexible sensing accessibility features are desirable for an assembly robot controller. Generalised independently abstracted sensing is desirable which allows any BM to get access when interested. Programming using modularised BM's with assembly skills would ease the programming process of an assembly robot.

7.6.11 Comparison with Other Work

The sensor fusion problem dealt with in this chapter differs from problems of other conventional technical sensor fusion such as by data manipulation (*e.g.*, [Luo & Lin 88]) and symbol systems (*e.g.*, black board system [Harmon *et al.* 86]). The sensor fusion problem addressed in this chapter is an architectural issue of a behaviour–based system applicable to assembly tasks. Thus, this work addressed basic fundamental issues rather than technical issues. General comparisons between the behaviour–based approach and other approaches were mentioned in Chapter 2.

This work can be better compared with other approaches to behaviour–based systems, particularly on architectural issues and multi–agent arbitration. The BM's in this work are the same as AFSM's in the Subsumption Architecture [Brooks 86] in that they are purposeful sub-modules, but different in that BM's are more flexible in their structure, and are not limited to finite state machines. In their complexity, BM's are more like the layers of AFSM's in the Subsumption Architecture, but not identical. BM's have three general communication channels (*Input, Output, Command*), through which any type of messages can be passed (*i.e.* signals, numbers, *etc.*)[15]. In the Behaviour Language [Brooks 90], message passing facilities provide communications between rules which describe AFSM's, in addition to the simple signal communication.

Behaviour arbitration is an important issue in the study of the behaviour-based approach, where distributed sub-systems may require coherence. However, to date, there are not any fundamental insightful formalisms or detailed methodologies. Rather there are empirical explanations and task specific implementations. Brooks proposed ways to establish coherence between distributed sub-systems avoiding central representation, as described in Section 2.2.5, page 26: by exploiting natural coherence; and by exploiting certain mechanisms (using internal parameters; the environment for communication; and mutual exclusion). Brooks' suggestion is basically centred around the use of tacit and unlocated representations, although a practical implementation of *using internal parameters* could lead a system to have a sort of central representation, although it would be local. Matarić implemented two ways of explicitly combining different behaviours of a behaviour-based robot, in order to produce desired one output behaviour: switching and vector summation [Matarić 94]. Certain goals can be maintained by appropriately switching between basic behaviours. For instance, the basic behaviours: safe-wandering; dispersion, homing; and following, can be switched in order to produce the foraging behaviour. On the other hand, for vector summation, appropriately weighted vector summation of the output of the basic behaviours: homing; dispersion; aggregation; and safe-wandering, can generate the flocking behaviour. Matarić mentions: "In the spatial domain, the outputs of all of the basic behaviors are in the form of direction and velocity vectors, so appropriately weighted sums of such vectors directly produce coherent higher-level behaviors." Matarić's robots were mobile robots.

Vector summation is considered to be an effective yet simple method to combine sub-systems. It is relatively easier for a mobile robot to provide direction and velocity vectors to the actuators, than for a position controlled arm. One of the criticisms made in this chapter is that in order to utilise such an effective facility as vector summation in organising multiple behaviours, a behaviour-based assembly robot should have a

[15] Although the format being the type of string.

control mechanism which allows commands in the form of directional and velocity vectors.

The position of the work described in this chapter among other work can be summarised as:—

- This is behaviour–based sensor fusion, compared to other sensor fusion work using central symbolic knowledge representations and numeric manipulations.
- This work deals with problems of behaviour–based assembly in the study of the behaviour–based systems in general.
- This work concerns the problems of actual combining of multiple sensors for the executive agent of a behaviour–based assembly systems, while other approaches in behaviour–based assembly systems have so far been concerned only with single sensors (see Section 2.3, page 31).

7.6.12 Summary

In summary, this chapter described a behaviour–based sensor fusion using a proposed distributed and concurrent architecture. Guarded moves, which are useful for assembly robots, formed the basis of the control hierarchy. Difficulty and inefficiency in combining various motion commands from various sub-systems using a position controlled robot in a conventional way were highlighted in the early experiments. However, by using the robot to quickly and continuously update its target position, the robot was able to be driven using changing vectors in an XYZ space. By this way, the robot was able to adapt dynamically to the changing vector magnitudes and various BM's associated with different sensors could contribute to the robot motion independently. Low level amalgamation of sensing and action became easier and more economic.

Further research could be centred around generalisation of the architecture. Study on the compromise and arbitration facilities is also left for future investigation, where there is any conflict between any incoming vector informations from multiple BM's to a single channel output path, *i.e.* one of the axes (XYZ) in the case of *Continuous vector summation.*

In the next chapter, this thesis is concluded and future directions are suggested.

Chapter 8

Summary and Conclusions

This chapter concludes the work described in this thesis. This thesis is first summarised, then the contributions and future work are listed.

8.1 Summary

It being acknowledged that event sensing and its application to robot motion would provide useful information for assembly robots, the development of an event signature sensor for assembly robots was started, using PVDF piezoelectric film as a sensitive and flexible transducer. General experiments with PVDF materials on mechanical vibration characteristics were carried out. By the knowledge obtained from these experiments, two kinds of event signature sensors were developed: the Clunk Sensor and the Push Sensor. The Clunk Sensor responds to acoustic vibration, while the Push Sensor responds to the changes of force applied to deform the sensor body. Hence, Clunk Sensors are used to detect any 'clunk' noise during part mating in assembly, while Push Sensors were used to detect any change of force applied to the finger skin, hence to the gripped part. These sensors are very economic sensors with which to guard robot moves. The sensors developed were successfully applied to a benchmark assembly task.

With its high sensitivity and resistance to noise, the Push Sensor demonstrated much potential for further refinement during the experiment and the application to the benchmark assembly. Further development of the Push Sensor was carried out. The interesting points were whether the sensitivity could be increased, and how the physical structure of the sensor affected the sensitivity. In addition, the possibilities of more application areas were also investigated. In result, the sensitivity was improved to a level which easily suffices for the requirements of common robot assembly tasks. Two film configurations were tested: round type and flat type. The round type shows a good regular sensitivity for any direction of force applied, while the flat type is less sensitive to force in parallel to the flat table surface with the robot finger pointing vertically downwards (horizontal force), but extremely sensitive to force caused by the contact when the robot moves vertically downwards with the fingers pointing downwards (vertical force). Push Sensors can be used not only for binary event signatures, but also for reading force patterns over time, such as could be used for snap-fit monitoring. An example of this was demonstrated. They can also be used to obtain the direction of force applied to the gripped part in a limited manner, by using the difference in force applied between the two fingers or by combining the polarity of the signals.

These enhanced Push Sensors were applied to a sensor fusion problem, in order to test their utility and robustness, and to investigate how these sensors could be combined with other sensors in the context of the behaviour–based approach. Vision was used as the complementary sensor. The control methodology and hierarchy was developed from Albus' and Powers' models and the guidelines of the behaviour–based approach. BM's were distributed and run concurrently. Sensors were also incorporated into the modular structure independently. This control structure provided a flexible management of the system.

Three strategies of sensor fusion were tested: *Interlacing*, *Vector summation*, and *Continuous vector summation*:—

- The *Interlacing* permits incremental growth of the system[1]. But, the criticism can be made that in order to make the incremental growth work properly, substantial basic functional components must be provided in advance, such as free mobility in the 3D space. In other words, the kind of growth required must be foreseen

[1] System incremental growth — the addition of facilities (such as a new sensor) by adding modules, without changing existing modules – is one of the implementation guidelines of behaviour–based systems.

by the designer to some extent. As argued in Section 7.6.2, this guideline has special difficulties in assembly robots, compared to mobiles.

- The *Vector summation* strategy starts with a full 3D mobility and provides better performance. The required substantial basic functional components are determined by the physical structure of the robot, the characteristics of the environment, and the dynamics of the robot in the environment. To meet these requirements, a desirable behaviour–based assembly robot with sensors for guidance is proposed to have a feature to accept motion commands in terms of vector.

- The *Continuous vector summation* was implemented to cause the robot to move towards the vector proposed by the BM's or the sensors, changing the direction of motion as necessary while moving. The robot was able to follow a smooth curve using sensing without having a central system, hence the system was robust.

From experimental implementations based on these three strategies, it is argued that a distributed servoing mechanism is a good fundamental basis of a robot control hierarchy on which more abstract control schemes take can place. Parallelism and concurrency, provided by the MMPS system, are also features encouraged for convenient management of modularised subsystems and flexible execution. Servo control mechanism using external vector sensors such as vision and force sensors, could be built at the bottom of the control hierarchy of a behaviour–based assembly system in order to give the robot a well distributed multiple sensing management, hence high flexibility and distributed complexity.

In addition, touch sensors used with robot motions proved to provide a useful basis for a control hierarchy for assembly robots.

8.2 The Thesis Contributions

The work described in this thesis concerns the development of event signature sensors for assembly robots and their application to a behaviour–based sensor fusion problem. During the course of the work carried out, there were a number of contributions made as outlined below.

General Experiment on PVDF Films

Using PVDF films, signals from various sources were collected under various conditions and analysed. The information obtained by these experiments can be used to develop useful sensors. Based on these experiments, two types of event signature sensors were developed.

The Development of Clunk Sensors

Sound detecting Clunk Sensors were developed. Clunk Sensors can be used for guarded moves in assembly. They are particularly useful when rigid parts are used in the assembly, which usually generate detectable sound on contact. They can also detect slipping of the part in the fingers.

The Development of Push Sensors

This is one of the main contributions of the work described in this thesis. The Push Sensors are sensitive, cheap, relatively robust, relatively easy to build. Although the robot gripping force is significant (*e.g.*, to parallel human gripping force), these Push Sensors can still detect gentle touch (*e.g.*, 1 mm/second of a robot speed) on the gripped part. Force patterns in time (*e.g.*, snap-fit) can be also obtained using the Push Sensors. Discrimination of the direction of force applied is possible in a limited sense.

Exploring Guarded Moves

Although the introduction of guarded moves is not new, this thesis collectively deals with guarded moves using the touch sensors developed. Many examples of guarded moves were shown. Among which, sensored sweep (see Section 5.6, page 104) is considered as innovative: a robot being able to use a tool in order to exploit its sensors designed for gripped part manipulation. Guarded moves are highlighted in the context of the dynamical behaviour–based approach.

Investigating Structural Problems of Behaviour–based Arm Control Using Multiple Sensors

The sensor fusion problem of behaviour–based assembly systems was addressed. This is a particular problem in behaviour–based systems because one of the normal sensor fusion strategies — fusion *via* centralised symbolic knowledge representation — is regarded as undesirable. The suitability of Albus' and Powers' hierarchical models and the guidelines of the behaviour–based approach were examined in practice. A generalised behaviour–based architecture for assembly robots was proposed and tested in various ways. The result of the experiments suggested that a behaviour–based assembly system with multiple sensing could and should be implemented from the bottom up. This could be achieved more conveniently using a robot with vector servoing capability built at the bottom of the control hierarchy, rather than a traditional positional control robot. Sensor fusion problems can be solved despite using a distributed architecture which the behaviour–based approach favours. These sensor fusion experiments motivated criticisms and discussions for further development of behaviour–based assembly systems.

8.3 Future Work

In this section, possible future extensions are outlined.

Clunk Sensors

Clunk Sensors can be further developed to provide more information than just binary. More elaborate signal processing can be developed in order to discriminate signals between various types of bumping and sliding. Another interesting extension would be on monitoring assembly. The general sound pattern generated by an assembly is stored, then any failure of assembly could be detected by the comparison between the stored normal pattern and the currently obtained pattern.

Push Sensors

Possible future extensions are described in Section 6.7, page 153, as:

- More investigation on physical construction and PVDF film placement of the sensors;
- Generalisation and matching profiles of sensor response over time, *e.g.*, as in snap–fit;
- Seeking for better materials for the compliant foundation of the sensor film and for the skin;
- Investigation of the active application of vibration in order to help the Push Sensor to overcome the limitation of not being able to sense constant force.

Assembly Robot Architecture

This thesis proposed an architecture for a reactive assembly robot with multiple sensors. It showed that the bottom level control of the arm is better implemented with a servoing capability guided by external sensors using such control as force and velocity control, rather than position control. The experimental implementation described in this thesis is only enough to propose that such control hierarchy and control methods are appropriate for a reactive assembly robot with multiple sensors. Problems to be further investigated for scaling up in order to realise the complete control hierarchy proposed can be itemised as:—

- Investigation on vector arbitration: An arbitration engine is necessary between the vector commands from different BM's.
- The position control issue: An interpreter is necessary in order to interface the robot with a human programmer or other systems which use geometric terms to describe an assembly. It would be useful to investigate the possibilities of implementing an interpretation engine such as position control on top of the proposed velocity or force control incorporated with external sensors.
- Technical problems: More general means of execution of multiple agents and communication between them would be required.

8.4 Concluding Remarks

Biological sensors are effective in assisting their hosts to survive in the real world of uncertainty. They often report changes rather than measurements. It is not only because they have economical effective sensors but also because they have appropriate information processing ability. In managing multiple sensors, one of the features found in animals is to combine them at the lowest possible level, tightly incorporated with actions. Within the framework of the behaviour–based approach, these implications from real animals have been tested, in order to provide a method to organise multiple sensing and action for behaviour–based assembly systems. The two change–based event sensors proved to be sensitive and useful. The Push Sensor was robust enough to survive a variety of assembly–like tasks (*e.g.*, run daily for several months) without significant damage.

Unless a robotic system is implemented at a neuronal level, some abstraction is required and the behaviour–based approach provides reasonable guidelines. Since the behaviour–based approach is still in its infancy, those guidelines are not yet fully tested and generalised. In the sensor fusion work described in this chapter some of the guidelines were tested, particularly for those assembly systems using multiple sensing.

As a result, desired architectural features of a behaviour–based assembly system were outlined. Servo control mechanisms with external sensors are desired. Ideally, BM's convert information from multiple sensors into motion commands. With servoing, motion commands from various BM's can be combined economically. Implementation by incremental growth, and low level amalgamation of sensing and action, are argued to be convenient with servoing. Distribution and concurrency helps BM's to be independent, which results in robustness and flexibility. However, coordination remains a problem.

Action helped sensor fusion in the sense that a sensory modality was converted into a motion modality by a BM, then the motion modality was combined with other motion modalities generated by the BM's using different sensors. These different motion modalities could be simply combined by the dynamics of the action space. This is sometimes referred to by Flynn and Brooks as *behaviour fusion*, or *action fusion*

[Flynn & Brooks 89].

Generally speaking, the anti–symbolic and anti–central–control attitudes of behaviour–based systems are taken to imply that symbolic representation should be avoided. The knowledge representation terminology developed in Section 2.4, page 35 allows us to make more subtle distinctions, in particular to distinguish between knowledge which to a sub–system is central and symbolic, but to the whole system is tacit and local. In tackling the problems of arbitration and compromise between conflicting or additive BM's, it was useful to employ this kind of locally symbolic but globally tacit knowledge. It is therefore suggested that the behaviour–based guideline be refined by using these distinctions.

Bibliography

[Ade85a] Adept Technology, Inc., 1212 Bordeaux Drive, Sunnyvale, CA 94089, U.S.A. *Adept Manipulator System Basic Operation*, February 1985.

[Ade85b] Adept Technology, Inc., 1212 Bordeaux Drive, Sunnyvale, CA 94089, U.S.A. *Adept Manipulator System VAL II Reference Guide*, July 1985.

[Akbarally & Kleeman 95] H. Akbarally and L. Kleeman. A sonar sensor for accurate 3D target localisation and classification. In *Proceedings of IEEE International Conference on Robotics and Automation*, pages 3003–3008, 1995.

[Albus 81] J.S. Albus. *Brains, Behavior, and Robotics.* Byte Books, McGraw Hill, 1981.

[Allen 92] Peter K. Allen. *Robotic Object Recognition Using Vision and Touch.* MIT Press, 1992.

[Almand 85] B. Almand. Sensor information fusion and a robot decision making methodology. In *Society for Photo-optical Instrumentation Engineers, SPIE.*, volume 579, pages 436–441, 1985.

[Aloimonos & Bandyopadhyay 87] J. Aloimonos and A. Bandyopadhyay. Active vision. In *Proceedings of the 1^{st} International Conference on Computer Vision*, pages 35–54, 1987.

[Arbib 81] Michael A. Arbib. Perceptual structures and distributed motor control. In J.B. Brooks, editor, *Handbook of Physiology*, chapter 33, pages 1449–1480. American Physiological Society, 1981.

[Arbib 87] Michel A. Arbib. *Brains, Machines, and Mathematics.* Springer–Verlaf New York Inc., 1987.

[Balch 92] P.R. Balch. Force sensing in an industrial assembly. DAI Working Paper 237, Department of Artifical Intelligence, University of Edinburgh, 1992.

[Barnes *et al.* 83] D.P. Barnes, M.H. Lee, and N.W. Hardy. A control and monitoring system for multiple sensor industrial robots. In *Proc. 3^{rd} Int. Conf. Robot Vision and Sensory Controls*, pages 457–463, Cambridge, MA, November 1983.

[Bechtel & Abrahamsen 91] W. Bechtel and A. Abrahamsen. *Connectionism and the Mind: an Introduction to Parallel Processing in Networks.* Basil Blackwell, Inc., 1991.

[Beer 90] Randall D. Beer. *Intelligence as Adaptive Behaviour: an Experiment in Computational Neuroethology.* Perspectives in Artifical Intelligence. Academic Press, 1990.

[Beni *et al.* 83] G. Beni, S. Hackwood, L.A. Hornak, and J.L. Jackel. Dynamic sensing for robots: an analysis and implementation. *Robotics Research*, 2:51–61, 1983.

[Bicchi & Dario 87] A. Bicchi and P. Dario. Intrinsic tactile sensing for artificial hands. In *Proceedings of the 4^{th} International Symposium on Robotics Research*, pages 83–90, Santa Cruz, August 1987. MIT Press.

[Bicchi *et al.* 90] A. Bicchi, J.K. Salisbury, and D.L. Brock. Contact sensing from force measurement. A.I. Memo 1262, MIT, October 1990.

[Borovac *et al.* 94] B. Borovac, M. Nikolić, and L Nagy. A new type of force sensor for contact tasks. In *Proceedings of IEEE International Conference on Robotics and Automation*, pages 1791–1796, 1994.

[Braitenberg 84] V. Braitenberg. *Vehicles: Experiments in Synthetic Psychology.* MIT Press, 1984.

[Bridgeman 88] Bruce Bridgeman. *The Biology of Behavior and Mind.* John Wiley & Sons, Inc., 1988.

[Brooks 86] Rodney A. Brooks. Achieving artificial intelligence through building robots. A.I. Memo 899, MIT, May 1986.

[Brooks 89] Rodney A. Brooks. A robot that walks; Emergent behaviours from a carefully evolved network. A.I. Memo 1091, MIT, February 1989.

[Brooks 90] Rodney A. Brooks. The behavior language : User's guide. AI Memo 1227, MIT, Artificial Intelligence Laboratory, 1990.

[Brooks 91] Rodney A. Brooks. Intelligence without reason. A.I. Memo 1293, MIT, April 1991.

[Brooks 94] Rodney A. Brooks. Coherent behavior from many adaptive process. In *Proceedings of the Third International Conference on Simulation of Adaptive Behavior*, pages 22–29, 1994.

[Canny & Goldberg 93] J.F. Canny and K.Y. Goldberg. A risc approach to sensing and manipulation. In *6^{th} International Symposium on Robotics Research*, 1993.

[Chongstitvatana 92] Prabhas Chongstitvatana. *The Design and Implementation of Vision-Based Behavioural Modules for a Robotic Assembly System.* Unpublished PhD thesis, Department of Artificial Intelligence, University of Edinburgh, 1992.

[Churchland 86] P.S. Churchland. *Neurophilosophy: Toward a Unified Science of the Mind–Brain.* MIT Press, 1986.

[Clark & Yuille 90] J.J. Clark and A.L. Yuille. *Data Fusion for Sensory Information Processing Systems.* Kluwer Academic Publishers, 1990.

[Colquhoun 95] Alexander Colquhoun. Personal communication, June 1995. DAI Electronic Workshop, Edinburgh University.

[Connell 88] Jonathan H. Connell. A behavior–based arm controller. A.I. Memo 1025, MIT, June 1988.

[Connell 89] Jonathan H. Connell. *A Colony Architecture for an Artificial Creature.* Unpublished PhD thesis, Electrical Engineering and Computer Science, MIT, September 1989.

[Dario & Buttazzo 87] Paolo Dario and Giorgio Buttazzo. An anthropomorphic robot finger for investigating artificial tactile perception. *The Internation Journal of Robotics Research*, 6(3):25–48, 1987.

[Dario & De Rossi 85] Paolo Dario and Danilo De Rossi. Tactile sensors and the gripping challenge. *IEEE Spectrum*, 8:46–51, August 1985.

[Dario 89] Paolo Dario. Tactile sensing for robots: Present and future. In O. Khatib, J.J. Craig, and T. Lozano-Pérez, editors, *The Robotics Review 1*, pages 132–146. The MIT Press, 1989.

[Dario *et al.* 83] P. Dario, C. Domenici, R. Bardelli, D. De Rossi, and P.C. Pinotti. Piezoelectric polymers: New sensor materials for robotic applications. In *Proceedings of 13^{th} Symposium on Industrial Robotics*, pages 14.34–14.49, 1983.

[Deacon 97] G.E. Deacon. *Accomplishing Task-Invariant Assembly Strategies by Means of an Inherently Accommodating Robot Arm.* Unpublished PhD thesis, Department of Artificial Intelligence, University of Edinburgh, Forth coming 1997.

[Deacon *et al.* 93] G.E. Deacon, P.L. Low, and C.A. Malcolm. Orienting objects in a minimum number of robot sweeping motions. In *Proc. The Sixth International Conference on Industrial and Engineering Applications of Artificial Intelligence and Expert Systems*, pages 494–503, 1993.

[Dreyfus 92] Hubert L. Dreyfus. *What Computers Still Can't do: a Critique of Artificial Reason.* MIT Press, 1992.

[Durrant-White 87] H.F. Durrant-White. Consistent integration and propagation of disparate sensor observations. *The International Journal of Robotics Research*, 6(3):3–24, 1987.

[Durrant-Whyte 88] Hugh F. Durrant-Whyte. *Integration, Coordination and Control of Multi–sensor Robot Systems.* Kluwer Academic Publishers, 1988.

[Eberman & Salisbury, Jr. 94] B. Eberman and J.K. Salisbury, Jr. Application of change detection to dynamic contact sensing. *The International Journal of Robotics Research*, 13(5):369–394, 1994.

[Fearing & Hollerbach 85] R.S. Fearing and J.M. Hollerbach. Basic solid mechanics for tactile sensing. *The International Journal of Robotics Research*, 4(3):40–54, 1985.

[Fearing 92] Ronald S. Fearing. Planner elasticity for tactile sensing. In Howard R. Nicholls, editor, *Advanced Tactile Sensing for Robotics*, chapter 4, pages 75–93. World Scientific, 1992.

[Ferguson 92] Innes Ferguson. *TouringMachines: an Architecture for Dynamic, Rational, Mobile Agent.* Unpublished PhD thesis, Computer Labratory, University of Cambridge, 1992.

[Firby 89] R.J. Firby. Adaptive execution in complex dynamic worlds. YALEU/CSD/RR 672, Computer Science Department, Yale University, 1989.

[Firby 93] R.J. Firby. An architecture for a synthetic vacuum cleaner. In *Proceedings of the 1993 Fall Symposium Series, Workshop on Instantiating Real–World Agents*, 1993.

[Fleming 87] A.D. Fleming. *Analysis of Uncertainties and Geometric Tolerances in Assemblies of Parts.* Unpublished PhD thesis, Department of Artificial Intelligence, University of Edinburgh, 1987.

[Flynn & Brooks 89] A.M. Flynn and R.A. Brooks. Battling reality. A.I. Memo 1148, MIT, October 1989.

[Gat 92] Erann Gat. Integrating planning and reacting in a heterogeneous asynchronous architecture for controlling real–world mobile robots. In *Proceedings of 10^{th} National Conference on Artificial Intelligence*, San Jose, CA, July 1992.

[Georgeff *et al.* 87] M.P. Georgeff, A.L. Lansky, and M.J. Schoppers. Reasoing and planing in dynamic domains: an experiment with a mobile robot. Technical Note 380, SRI International, 1987.

[Grahn & Astle 84] A.R. Grahn and L. Astle. Robotic ultrasonic force sensor arrays. In *Proceedings of 14^{th} Symposium on Industrial Robotics*, pages 21.1–21.18, 1984.

[Hackett & Shah 90] J. Hackett and M. Shah. Multiple-sensor fusion: a perspective. In *Proceedings of IEEE International Conference*, pages 1324–1330, 1990.

[Hackwood *et al.* 83] S. Hackwood, B. Beni, L.A. Hornak, R. Wolfe, and T.J. Nelson. A torque sensitive tactile array for robotics. *International Journal of Robotics Research*, 2:46–50, 1983.

[Hallam 88] J. Hallam. MMPS: a mailbox-based message-passing system. DAI Software Paper 9, Department of Artifical Intelligence, University of Edinburgh, 1988. unpublished.

[Hallam 94] J. Hallam. Personal communication, 1994.

[Hallam 96] J. Hallam. Personal communication, 1996.

[Hardy *et al.* 92] N.W. Hardy, H.R. Nicholls, and J.J. Rowland. The design of sensing commands in the InFACT assembly machine. In *Proc. of* 23^{rd} *Int. Sym. on Industrial Robots*, pages 47–52, 1992.

[Harmon 92] L.D. Harmon. Tactile sensing for robots. In G. Beni and S. Hackwood, editors, *Recent Advances in Robotics*, chapter 10, pages 389–424. John Wiley & Sons, Inc., 1992.

[Harmon *et al.* 86] S.Y. Harmon, G.L. Bianchini, and B.E. Pinz. Sensor data fusion through a distributed blackboard. In *Proceedings of IEEE International Conference on Robotics and Automation*, pages 1449–1454, 1986.

[Haugeland 78] John Haugeland. The natural plausibility of cognitivism. *The Behavioral and Brain Sciences*, 1:215–226, 1978.

[Henderson & Shilcrat 84] T. Henderson and E. Shilcrat. Logical sensor systems. *Journal of Robotic Systems*, 1(2):169–193, 1984.

[Hillis 81] William D. Hillis. Active touch sensing. A.I. Memo 629, MIT, April 1981.

[Horowitz & Hill 89] P. Horowitz and W. Hill. *The Art of Electronics*. Cambridge University Press, 1989.

[Howe & Cutkosky 89] R.D. Howe and M.R. Cutkosky. Sensing skin acceleration for slip and texture perception. In *Proceedings of IEEE International Conference on Robotics and Automation*, pages 145–150, 1989.

[Howe & Cutkosky 92] R.D. Howe and M.R. Cutkosky. Touch sensing for robotic manipulation and recognition. In O. Khatib, J. Craig, and T. Lozano-Pérez, editors, *The Robotics Review 2*, pages 57–112. The MIT Press, 1992.

[Howe *et al.* 90] R.D. Howe, N. Popp, P. Akella, I. Kao, and M.R. Cutkosky. Grasping, manipulation, and control with tactile sensing. In *Proceedings of IEEE International Conference on Robotics and Automation*, pages 1258–1263, 1990.

[Hutchinson *et al.* 88] S.A. Hutchinson, R.L. Cromwell, and A.C. Kak. Planning sensing strategies in a robot work cell with multisensor capabilities. In *Proceedings of IEEE International Conference on Robotics and Automation*, pages 1068–1075, 1988.

[Int95] Interlink Electronics, 546 Flynn Road, Camarillo, CA 93012, U.S.A. *FSR Integration Guide and Evaluation Parts Catalog*, 1995.

[Ishikawa 92] Masatoshi Ishikawa. The sensor fusion system: Mechanisms for integration of sensory information. *Advanced Robotics*, 6(3):335–344, 1992.

[Kempf & Ambler 83] K.G. Kempf and A.P. Ambler. An experimental comparison of symbolic and graphic offline robot programming techniques. DAI Research Paper 207, Department of Artifical Intelligence, University of Edinburgh, 1983.

[Kim 92] Taehee Kim. The development of vibration sensors as event signature sensors in assembly. DAI Technical Paper 14, Department of Artifical Intelligence, University of Edinburgh, 1992.

[Klafter *et al.* 89] R.D. Klafter, T.A. Chmielewski, and M. Negin. *Robotic Engineering : an Integrated Approch.* Prentice-Hall, 1989.

[Košecká & Bogoni 94] Jana Košecká and Luca Bogoni. Application of discrete events systems for modeling and controlloing robotic agents. In *Proceedings of IEEE International Conference on Robotics and Automation*, pages 2557–2562, 1994.

[Kuperstein 91] Michael Kuperstein. INFANT Neural controller for adaptive sensory–motor coordination. *Neural Networks*, 4:131–145, 1991.

[Lee & Asada 94] Sooyong Lee and Haruhiko Asada. Assembly of parts with irregular surfaces using active force sensing. In *Proceedings of IEEE International Conference on Robotics and Automation*, pages 2639–2644, 1994.

[Lieberman & Wesley 77] L.I. Lieberman and M.A. Wesley. Autopass: an automatic programming system for computer controlled machanical assembly. *IBM Journal of Research and Development*, 21(4):321–333, July 1977.

[Lozano-Pérez & Brooks 85] T. Lozano-Pérez and R.A. Brooks. An approach to automatic robot programming. AI Memo 842, MIT, April 1985.

[Lozano-Pérez 82] Tomás Lozano-Pérez. Robot programming. A.I. Memo 698, MIT, December 1982.

[Lozano-Pérez *et al.* 92] Tomás Lozano-Pérez, Joseph L. Jones, Emmanuel Mazer, and Patrick A. O'Donnell. *HANDEY: a Robot Task Planner.* MIT Press, 1992.

[Luo & Kay 89] R.C. Luo and M.G. Kay. Multisensor integration and fusion in intelligent systems. *IEEE Transactions on Man, Machine, and Cybernetics*, 19(5):901–931, 1989.

[Luo & Lin 88] R. Luo and M. Lin. Robot multi-sensor fusion and integration: Optimum estimation of fused sensor data. In *Proceedings of IEEE International Conference*, pages 1076–1081, 1988.

[Lyons & Arbib 85] D. Lyons and M.A. Arbib. Task-level model of distributed computation for sensory-based control of complex robot systems. In *IFAC Symposium on Robot Control*, pages 349–354, Barcelona, Spain, 1985.

[Maes 90] P. Maes. Situated agents can have goals. In P. Maes, editor, *Designing Autonomous Agents: Theory and Practice from Biology to Engineering and Back*, pages 49–70. MIT Press, 1990.

[Maes *et al.* 96] P. Maes, M. Matarić, J. Meyer, J. Pollack, and S. Wilson, editors. *Proceedings of the Forth International Conference on Simulation of Adaptive Behavior.* MIT Press, 1996.

[Malcolm & Fothergill 86] C.A. Malcolm and A.P. Fothergill. Some architectural implications of the use of sensors. DAI Research Paper 294, University of Edinburgh, October 1986. (Also appears in *Proc. of Pisa NATO Conference on Robotics.*, September, 1986).

[Malcolm & Smithers 88] C.A. Malcolm and T. Smithers. Programming assembly robots in terms of tack achieving behavioural modules: First experimental results. DAI Research Paper 410, Department of Artificial Intelligence, University of Edinburgh, 1988.

[Malcolm & Smithers 90] Chris Malcolm and Tim Smithers. Symbol grounding via a hybrid architecture in an autonomous assembly system. *Robotics and Autonomous Systems*, 6:123–144, 1990.

[Malcolm 83] C. Malcolm. The outline corner filter. In B. Rooks, editor, *Proceedings of the 3rd International Conference on Robot Vision and Sensory Controls (ROVISEC)*, pages 61–68, Amsterdam, 1983. North Holland Publishing Company. Also available as University of Edinburgh DAI research paper No. 212.

[Malcolm 87] Chris Malcolm. Planning and performing the robotic assembly of soma cube constructions. Unpublished M.Sc. thesis, Department of Artificial Intelligence, University of Edinburgh, 1987. (Shared Rank Xerox Best Dissertation Prize).

[Malcolm 93] C.A. Malcolm. Behaviour, purpose, & meaning. in preparation, 1993.

[Malcolm 95] C.A. Malcolm. The SOMASS system: a hybrid symbolic and behaviour-based system to plan and execute assemblies by robot. In *Proceedings of AISB Conference*, Sheffield, U.K., April 1995.

[Malcolm 96] C.A. Malcolm. Personal communication, 1996.

[Malcolm *et al.* 89] C. Malcolm, T. Smithers, and J. Hallam. An emerging paradigm in robot architecture. In *2nd Intelligent Autonomous Systems Conference*, 1989. Also available as University of Edinburgh DAI research paper No. 447.

[Marr 82] David Marr. *Vision.* W. H. Freeman & company, 1982.

[Mason 85] M.T. Mason. On the scope of quasi-static pushing. In *Proceedings of the 3rd International Symposium on Robotics Research*, pages 229–233. MIT Press, 1985.

[Mat90] The Math Works, Inc., 21 Eliot Street, South Natick, MA 01760, USA. *Matlab : for Sun Workstations : User's Guide*, Jenuary 1990.

[Matarić 94] Maja J. Matarić. *Interaction and Intelligent Behavior.* Unpublished PhD thesis, Electrical Engineering and Computer Science, MIT, May 1994.

[Maturana & Varela 80] R. Maturana and F. Varela. *Autopoiesis and Cognition.* Reidel, Dordrecht, 1980.

[McClelland 89] Stephen McClelland. An overview of sensor development. *Sensor Review*, 9(2):105–109, 1989.

[McClelland *et al.* 86] J.L McClelland, D.E. Rumelhart, and G.E. Hinton. The appeal of parallel distributed processing. In J.A. Feldman P.J. Hays and D.E. Rumelhart, editors, *Parallel Distributed Processing*, chapter 1, pages 3–44. MIT Press, 1986.

[McManus *et al.* 92] D. McManus, K. Selke, B. Bannister, A. Joyce, S. McLachlan, and E. Nadarajah. Automatic generation of control structure for assembly monitoring. In *International Conference on Automation, Robotics and Computer Vision*, pages 15–18, Singapore, September 1992.

[Mel 88] B.W. Mel. MURPHY: A robot that learns by doing. In D. Anderson, editor, *Neural information processing systems*, pages 544–553. American Institute of Physics, 1988.

[Moravec 88] Hans Moravec. *Mind Children: the Future of Robot and Human Intelligence.* Harvard University Press, 1988.

[Morrow & Khosla 95] J.D. Morrow and P.K. Khosla. Sensorimotor primitives for robotic assembly skills. In *Proceedings of IEEE International Conference on Robotics and Automation*, pages 1894–1899, 1995.

[Moss-Salentijn 92] Letty Moss-Salentijn. The human tactile systems. In H.R. Nicholls, editor, *Advanced Tactile Sensign for Robotics*, chapter 7, pages 123–150. World Scientific, 1992.

[Murphy 96] Robin Murphy. Biological and cognitive foundations of intelligent sensor fusion. *IEEE Transactions on Man, Machine, and Cybernetics*, 26(1):42–51, 1996.

[Naidu & Fisher 91] D.K. Naidu and R.B. Fisher. A comparative analysis of algorithms for determining the peak position of a stripe to sub-pixel accuracy. In *Proceedings of the 1991 British Machine Vision Association Conference*, pages 217–225, 1991. Also available as University of Edinburgh DAI research paper No. 553.

[Nehmzow 92] Ulrich Nehmzow. *Experiments in Competence Acquistion for Autonomous Mobile Robots.* Unpublished PhD thesis, Department of Artificial Intelligence, University of Edinburgh, 1992.

[Newell & Simon 76] A. Newell and H. Simon. Computer science as empirical inquiry: Symbols and search. *Communications of the Association for Computing Machinery*, 19:113–126, 1976.

[Newell 82] Allan Newell. The knowledge level. *Artificial Intelligence*, 18:87–127, 1982.

[Nicholls & Hardy 92] Howard R. Nicholls and Nigel W. Hardy. Distributed touch sensing. In Howard R. Nicholls, editor, *Advanced Tactile Sensing for Robotics*, chapter 6, pages 107–122. World Scientific, 1992.

[Nicholls & Lee 89] H.R. Nicholls and M.H. Lee. A survey of robot tactile sensing tchnology. *The International Journal of Robotics Research*, 8(3):3–30, 1989.

[Nicholls 92a] Howard R. Nicholls. Introduction to tactile sensing. In Howard R. Nicholls, editor, *Advanced Tactile Sensing for Robotics*, chapter 1, pages 1–11. World Scientific, 1992.

[Nicholls 92b] Howard R. Nicholls. Tactile sensor designs. In Howard R. Nicholls, editor, *Advanced Tactile Sensing for Robotics*, chapter 2, pages 13–48. World Scientific, 1992.

[Nilsson 84] N.J. Nilsson. SHAKEY the Robot. Technical Note 323, SRI International, April 1984.

[Okada & Rembold 92] Tokuji Okada and Ulrich Rembold. A tactile sensor based on a suspension-shell mechanism for dexterous fingers. *IEEE Transactions on Robotics and Automation*, 8(1):126–131, 1992.

[Patterson & Nevill, Jr. 86] R.W. Patterson and G.E. Nevill, Jr. The induced vibration touch sensor — a new dynamic touch sensing concept. *Robotica*, 4:27–31, 1986.

[Payton 86] D.W. Payton. An architecture for reflexive autonomous vehicle control. In *Proceedings of IEEE International Conference on Robotics and Automation*, pages 1838–1845, 1986.

[Peshkin & Sanderson 88] M.A. Peshkin and A.C. Sanderson. The motion of a pushed, sliding workpiece. *IEEE Transactions on Robotics and Automation*, 4(6):569–598, 1988.

[Pettinaro & Malcolm 94] G.C. Pettinaro and C.A. Malcolm. Electric gripper development. DAI Technical Paper 28, Department of Artifical Intelligence, University of Edinburgh, 1994.

[Pettinaro & Malcolm 95] G.C. Pettinaro and C.A. Malcolm. Two–handed robotic system. DAI Technical Paper 34, Department of Artifical Intelligence, University of Edinburgh, 1995.

[Pettinaro 96] Giovanni Pettinaro. *Basic Set of Behaviours for Programming Assembly Robots.* Unpublished PhD thesis, Department of Artificial Intelligence, University of Edinburgh, 1996.

[Pfeifer 96] Rolf Pfeifer. Building "Fungus Eaters": Designing principles of autonomous agents. In *Proceedings of the Fourth International Conference on Simulation of Adaptive Behavior*, pages 3–12, 1996.

[Pie87] Piezo Film Sensors – Europe, Merrion Avenue, Stanmore, Middlesex, HA7 4RS, U.K. *Technical Notes*, 1987.

[Polanyi 67] Michael Polanyi. *The Tacit Dimension.* Routledge and K. Paul, 1967.

[Popplestone *et al.* 78] R.J Popplestone, A.P. Ambler, and I. Bellos. RAPT, a language for describing assemblies. *Industrial Robot*, 3:131–137, 1978.

[Powers 73] W.T. Powers. *Behaviour: the Control of Perception.* Wildwood House, 1973.

[Reber 85] Arthur S. Reber. *The Penguin Dictionary of Psychology.* Penguin Books, 1985.

[Ritter *et al.* 92] H. Ritter, T. Martinetz, and K. Schulten. *Neural Computation and Self–Organising Maps.* Addison–Wesley, 1992.

[Rosen & Nitzan 77] C.A. Rosen and D. Nitzan. Use of sensors in programmable automation. *Computer, IEEE.*, 10(12):12–23, 1977.

[Roukangas *et al.* 86] C.C. Roukangas, M.S. Black, J.F. Martin, and J.S. Schoenwald. Integration of multiple sensors to provide flexible control strategies. In *Proceedings of IEEE International Conference on Robotics and Automation*, pages 1947–1953, 1986.

[Rucci & Bajcsy 95] M. Rucci and R. Bajcsy. Learning visuo–tactile coordination in robotic systems. In *Proceedings of IEEE International Conference on Robotics and Automation*, pages 2678–2683, 1995.

[Russell 90] R. Andrew Russell. *Robot Tactile Sensing*. Prentice Hall, 1990.

[Salisbury 84] J.K. Salisbury. Interpretation of contact geometries from force measurement. In *Proceedings of the 1st International Symposium on Robotics Research*, pages 566–577. MIT Press, 1984.

[Schoppers 87] M.J. Schoppers. Universal plans for reactive robots in unpredictable environment. In *Proceedings of the 10th International Joint Conference on Artificial Intelligence*, pages 1039–1046, August 1987.

[Searle 80] J.R. Searle. Minds, brains and programs. *The Behavioural and Brain Sciences*, 3(3):417–424, 1980.

[Selke *et al.* 91] K.W. Selke, H.C. Shen, G.E. Deacon, and A. Pugh. Strategy for sensors and rules in flexible robotic assembly. *International Journal of Production Research*, 29(2):277–291, 1991.

[Shimoga & Goldenberg 92] K.B. Shimoga and A.A. Goldenberg. Soft materials for robotic fingers. In *Proceedings of IEEE International Conference on Robotics and Automation*, pages 1300–1305, 1992.

[Shinoda & Ando 94] H. Shinoda and S. Ando. Ultrasonic emission tactile sensor for contact localisation and characterization. In *Proceedings of IEEE International Conference on Robotics and Automation*, pages 2536–2543, 1994.

[Shinoda *et al.* 95] H. Shinoda, N. Morimoto, and S. Ando. Tactile sensing using tensor cell. In *Proceedings of IEEE International Conference on Robotics and Automation*, pages 825–830, 1995.

[Simon 81] H.A. Simon. *The sciences of the artificial.* MIT Press, 1981.

[Smith 82] B.C. Smith. *Reflection and Semantics in a Procedural Language.* Unpublished PhD thesis, MIT, 1982.

[Smithers 93] Tim Smithers. Taking eliminative materialism seriously: a methodology for autonomous systems research. In *Workshop on Artificial Life: a Bridge towards a New Artificial Intelligence*, San Sebastian, Spain, December 1993.

[Smithers 94] Tim Smithers. Why better robots make it harder. In *Proceedings of the Third International Conference on Simulation of Adaptive Behavior*, pages 64–72, 1994.

[Sobh & Valavanis 94] T.M. Sobh and K.P. Valavanis. Discrete event systems in robotics and automation. *Journal of Robotics and Autonomous Systems*, 13:151–152, 1994.

[Söderquist & Wernersson 92] Bertil Söderquist and Åke Wernersson. Information for assembly from impacts. In *Proceedings of IEEE International Conference on Robotics*, pages 2012–2017, 1992.

[Son *et al.* 94] J.S. Son, E.A. Monteverde, and R.D. Howe. A tactile sensor for localizing transient events in manipulation. In *Proceedings of IEEE International Conference on Robotics and Automation*, pages 471–476, 1994.

[Stanley *et al.* 84] W. Stanley, G. Dougherty, and R. Dougherty. *Digital Signal Processing.* Prentice-Hall Company, 1984.

[Steels 95] Luc Steels. When are robots intelligent autonomous agents. *Journal of Robotics and Autonomous Systems*, 15(1):3–10, 1995.

[Tortora & Anagnostakos 90] G.J. Tortora and N.P. Anagnostakos. *Principles of Anatomy and Physiology.* Harper Collins Publishers, sixth edition, 1990.

[van der Smagt 95] Patrick van der Smagt. *Visual Robot Arm Guidance Using Neural Networks.* Unpublished PhD thesis, Department of Computer Science, University of Amsterdam, 1995.

[Varela *et al.* 92] Francisco J. Varela, Evan Thompson, and Eleanor Rosch. *The Embodied Mind.* MIT Press, 1992.

[Wang & Mason 87] Yu Wang and Matthew Mason. Modelling impact dynamics for robotic operation. In *Proceedings of IEEE International Conference on Robotics*, pages 678–685, 1987.

[Webb 93] Barbara Webb. *Perception in Real and Artificial Insects: a Robotic Investigation of Cricket Phonotaxis.* Unpublished PhD thesis, Department of Artificial Intelligence, University of Edinburgh, 1993.

[Wen & Durrant-Whyte 92] W. Wen and H. Durrant-Whyte. Model–based multi-sensor data fusion. In *Proceedings of IEEE International Conference on Robotics and Automation*, pages 1720–1726, 1992.

[Will & Grossman 75] Peter M. Will and David D. Grossman. An experimental system for computer controlled mechanical assembly. *IEEE Transactions on Computers*, c-24(9):879–888, September 1975.

[Wilson 92] Myra Scott Wilson. *Achieving Reliability using Behavioural Modules in a Robotic Assembly System.* Unpublished PhD thesis, Department of Artificial Intelligence, University of Edinburgh, 1992.

[Yin 83] Baolin Yin. A framework for handling vision data in an object level robot language – RAPT. In *Proceedings of the 8^{th} International Joint Conference on Artificial Intelligence*, pages 814–820, Karlsruhe, Germany, August 1983.

[Yin 84] Baolin Yin. *Combining Vision Verification With a High Level Robot Programming Language.* Unpublished PhD thesis, Department of Artificial Intelligence, University of Edinburgh, 1984.

Appendix A

The Sampling Program

The sampling program written in Turbo Pascal is shown.

```
(*
    File:    SAMPLE.PAS
    Author:  Taehee Kim
    Updated: 5 July 1992
    Purpose: Samples data via A/D converter and stores as a DOS file
             under the name given.
    Description:
        The PC samples data via A/D converter. The sampling rate is
        8.33KHz(for 3 sec. in an XT PC). The Sampling rate can be
        adjusted by putting delays (NOP) in the loop between each
        sample. Once invoked, the PC asks for a file name in which
        the sampled data will be stored. Top level Turbo Pascal send
        the pointer of an array and number of data to be sampled
        to the inline Assembler. The inline Assembler samples data
        and stores the data into the array, then it passes back.
        Then, the top level Turbo Pascal saves the data in the array
        under the file name specified. The A/D converter used is A/D,
D/A converter (FPC-010), supplied by Chipboards plc.
*)

Program SAMPLE.PAS;

Uses CRT, DOS;

Const
  Scale_factor = 4096;
  Voltage_range = 9;
  Array_size = 24999;

Var
  Value : real;
  Data_array: array[1..Array_size] of word;
  Name_of_data_file : string[15];
  k : integer;
  f : text;
  hr, m, s, s100 :word;

Procedure Sample_it(Var Data_array; Array_size : word);

Begin
   inline(
      $FA/                  {      CLI   ; disable interrupt.}
      $C4/$BE/Data_array/ {      LES DI, Data_array[BP] ;load pointer.}
      $8B/$8E/Array_size/ {          MOV CX, Array_size[BP] }
```

```
$51/              { FIRST:  PUSH CX              }

    { pulse generator }

$B8/$08/$00/      { MOV AX, 0008  ; 0 VOLT. }
$BA/$77/$2/       { MOV DX, 0277 }
$EE/              { OUT DX, AL   }
$B8/$00/$00/      { MOV AX, 0000 }
$BA/$76/$02/      { MOV DX, 0276 }
$EE/              { OUT DX, AL   }
$90/              { NOP          }
$90/              { NOP          }
$B8/$04/$00/      { MOV AX, 0004  ; -4.5 VOLTS. }
$BA/$77/$2/       { MOV DX, 0277 }
$EE/              { OUT DX, AL   }

    { select AD port 1 }

$B8/$01/$00/      { MOV AX, 0001  ;select channel 1. }
$BA/$70/$02/      { MOV DX, 0270  }
$EE/              { OUT DX, AL    }

    { clear port }

$B8/$00/$00/      { MOV AX, 0000 }
$BA/$73/$02/      { MOV DX, 0273 }
$EE/              { OUT DX, AL   }

    { loop back }

$B9/$07/$00/      { MOV CX, 0007 ;seven times.}
$BA/$74/$02/      { MOV DX, 0274 }
$EC/        { LOOP1: IN AL, DX    :just read value.}
$90/              { NOP          }
$90/              { NOP          }
$E2/$FB/          { LOOP LOOP1   }
$B9/$07/$00/      { MOV CX, 0007 }
$BA/$75/$02/      { MOV DX, 0275 }
$EC/        { LOOP2: IN AL, DX   }
$90/              { NOP          }
$90/              { NOP          }
$E2/$FB/          { LOOP LOOP2   }
$B9/$06/$00/      { MOV CX, 0005 }
$90/        { LOOP3: NOP    ;delay }
$E2/$FD/          { LOOP LOOP3   }
$90/              { NOP    ;delay for an appropriate sampling freq. }
$90/              { NOP          }
$90/              { NOP          }
$90/              { NOP          }
$90/              { NOP          }
$90/              { NOP          }
$90/              { NOP          }
$90/              { NOP          }

    { read data }

$BA/$72/$02/      { MOV DX, 0272            }
$EC/              { IN AL, DX  ;higher byte.}
$88/$C4/          { MOV AH, AL              }
$BA/$71/$02/      { MOV DX, 0271            }
$EC/              { IN AL, DX  ;lower byte. }
$25/$FF/$0F/      { AND AX, OFFF  ;get valid data only. }

$89/$05/          { MOV [DI], AX  ;store data in an array. }
$47/              { INC DI        ;increment the array pointer. }
$47/              { INC DI        ;16 bits. }
$59/              { POP CX                  }
$E2/$A0/          { LOOP FIRST              }
```

```
      $FB);                   {STI    ;enable interrupt.}
   End;

Begin

    ClrScr;
    WriteLn('Type Data File Name...');
    ReadLn(Name_of_data_file);
    WriteLn('Start Sampling, please wait...');
    GetTime(hr,m,s,s100);
    WriteLn('Sec: ',s,'  ','mSec: ',s100);

    Sample_it(Data_array, Array_size);  {call procedure passing pointer}
                                        {of the data array and the array size }
    GetTime(hr,m,s,s100);
    WriteLn('Sec: ',s,'  ','mSec: ',s100);  {display the time elapsed}

    WriteLn('Done Sampling. Saving Data.');
    Assign(f,Name_of_data_file);
    Rewrite(f);
    For k := 1 to Array_size Do         {write data into disc under the}
      Begin                             {name specified               }
        Value := Data_array[k];         {this will write integer numbers}
       {if the value is required to be evaluated into real voltage}
       {value, thus real numbers, then, use the next line instead }
       { Value := (Voltage_range*Data_array[k])/Scale_factor -5;}

        WriteLn(f,Value);
      End;

    Close(f);
    WriteLn('*****  Done!  Saved to ==> ',Name_of_data_file);
    WriteLn('Press <Enter>.');
    ReadLn;
End.
```

Appendix B

Pro-Matlab Programs

The program codes which perform The Power Spectral Analysis of signal and general purpose high pass software filter are shown. They were written using Signal Processing Toolbox in Pro-Matlab.

B.1 Power Spectral Density

The discrete Power Spectral Density of a time domain signal $x(t)$ is denoted by:

$$S_{xx}(m) = \frac{|X(m)|^2}{N} = \frac{X(m)X(m)^*}{N}$$

where m and N denote discrete frequency and the number of samples respectively and $X(m)$ denotes the frequency function of $x(t)$.

This Pro-Matlab program displays the waveform of the signal and calculated the Power Spectral Density distribution of the frequency components on an X-window.

```
% File:    anal.m
% Author: Taehee Kim
% Updated:   5 July 1992
% Purpose:  To display time domain signal wave form and its frequency
%           spectral distribution.

% File name of the data file and number of data should be specified.
% Data file is regarded as a column vector. The number of points of Fast
% Fourier Transformation is 16184, but it can be modified. The power
% spectrum graph displays half of the frequency domain since the later
% part is symmetrical to the first half. The program is run in the Matlab
% interpreter.


clg                                 % clear the graphic window.
load filename.m;                    % load data file.
data = 9*filename/4096 - 5;   % evaluate data.

f = 3/24999*(0:24998);              % plot the waveform.
subplot(211),plot(f,data(1:24999)),title('Time domain signal'),
xlabel('Time(second)'),ylabel('Amplitude (V)'),
```

```
S = fft(data,16184);              % 16184 point FFT.
Pyy = S.*conj(S)/16184;           % calculate power spectral density.

f = 8333/16184*(0:8191);          % plot first half of PSD.
subplot(212),plot(f,Pyy(1:8192)),title('Power spectral density'),
xlabel('Frequency (Hz)'),
```

B.2 The Frequency Variable High Pass Filter

This is a generalised High Pass software filter (see the description below).

B.2.1 The Code

```
% File:     h_filter.m
% Date:     24 June 1992
% Purpose:  4th order Butterworth high pass filter with variable cut-off
%           frequency.
%
% When the program is invoked, it asks for desired cutoff frequency,
% then calculates coefficients of the filter and plots frequency response
% of the filter. Calculated coefficients remain in the current
% memory of Matlab interpreter under the name of 'a', 'b' which are used
% for the filtering of data.
% In order to filter the signal, a command 'filter' should follow.
% For example, now user data is loaded in Matlab under the name of
% 'data', then use:
%           > data = filter(b,a,data);
% After using this command the filtered data will be stored under the
% name of 'data'.
% NB. The sampling rate is assumed to be 8.333 KHz.


cut_freq = input('Desired Cut-off Frequency(Hz)? -->  ');
                                        % asks for the cut-off frequency.
N=4;                                    % the order of filter.
[b,a] = butter(N,cut_freq/4167,'high'); % 4167 is the half of the sampling rate.
n = 256;                                % for drawing....
hh = freqz(b,a,n);
hy = abs(hh);
ff = 8333/(2*n) *(0:99);
plot(ff,hy(1:100)),..
title('Frequency Response of 4th Order Butterworth  Filter (fc = 1200)'),..
xlabel('Frequency (Hz)'), ylabel('Gain'), pause  % when fc=1200
```

B.2.2 The Frequency Response of the Filter (fc=1200)

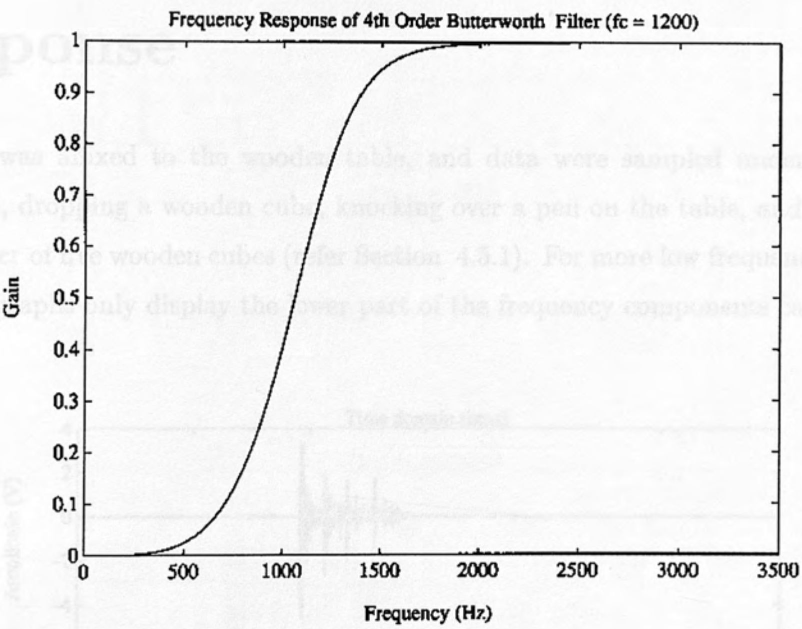


Figure B.1: The frequency response of software 4^{th} order Butterworth High Pass Filter

Appendix C

Some Examples of Indirect Response

The film was affixed to the wooden table, and data were sampled under different conditions, dropping a wooden cube, knocking over a pen on the table, and knocking over a tower of five wooden cubes (refer Section 4.5.1). For more low frequency details, following graphs only display the lower part of the frequency components calculated.

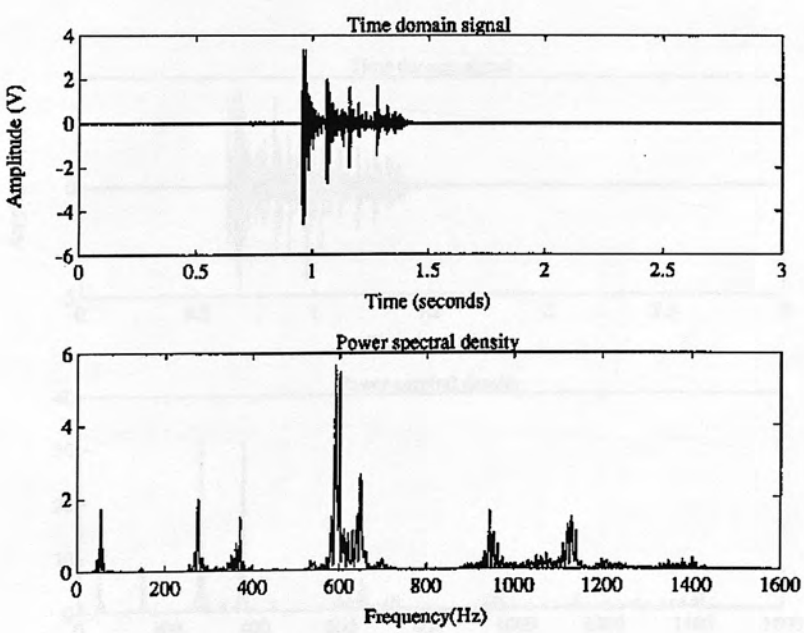


Figure C.1: Dropping and rolling a wooden cube on the table

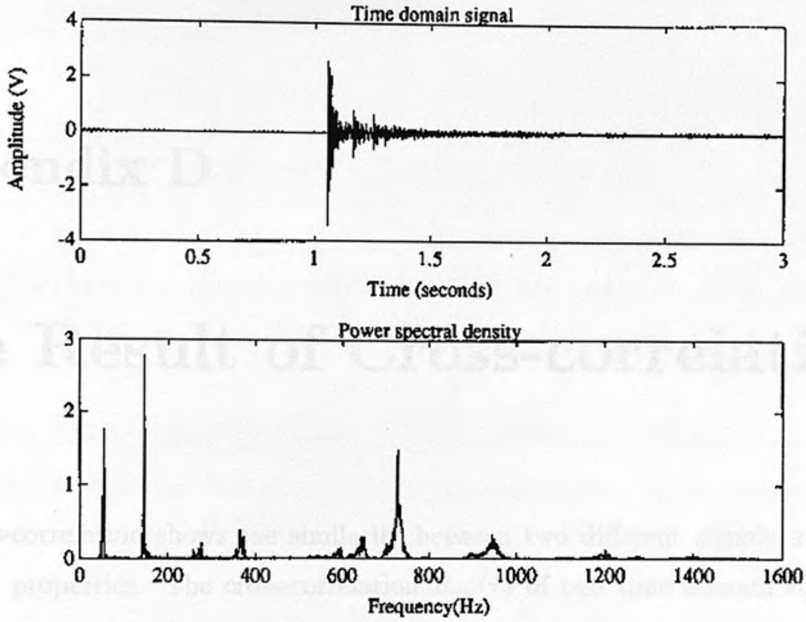


Figure C.2: Knocking over a pen on the table

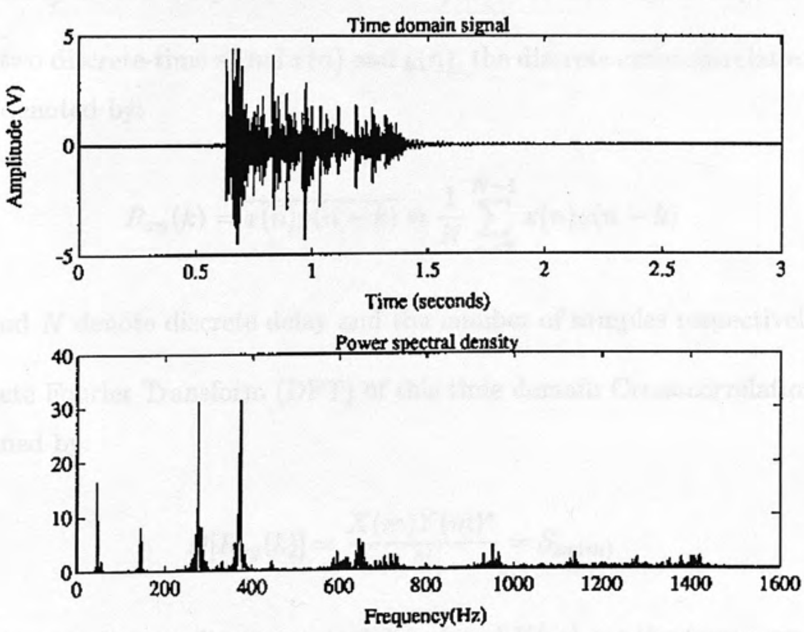


Figure C.3: Knocking over a tower of five wooden cubes on the table

Appendix D

The Result of Cross-correlation

The cross-correlation shows the similarity between two different signals in terms of frequency properties. The cross-correlation $R_{xy}(\tau)$ of two time domain signals $x(t)$ and $y(t)$ is denoted by [Stanley *et al.* 84]:

$$R_{xy}(\tau) = \overline{x(t)y(t-\tau)} = \lim_{t_p \to \infty} \frac{1}{t_p} \int_0^{t_p} x(t)y(t-\tau)dt$$

where τ and t_p denote delay and the effective period of time signal respectively.

For given two discrete-time signal $x(n)$ and $y(n)$, the discrete cross-correlation function $R_{xy}(k)$ is denoted by:

$$R_{xy}(k) = \overline{x(n)y(n-k)} = \frac{1}{N} \sum_{n=0}^{N-1} x(n)y(n-k)$$

where k and N denote discrete delay and the number of samples respectively.

The Discrete Fourier Transform (DFT) of this time domain Cross-correlation, $S_{xy}(m)$ is determined by:

$$D[R_{xy}(k)] = \frac{X(m)Y(m)^*}{N} = S_{xy(m)}$$

where m denotes discrete frequency, and $X(m)$ and $Y(m)$ are the frequency functions of $x(t)$ and $Y(t)$, respectively.

In the Pro-Matlab interpreter, using Fast Fourier Transform (FFT) (a simpler version of DFT), following lines will calculate the cross-correlation of two sampled time domain signals x, y:

```
Sx = fft{x,16184};                         %16184 point FFT.
Sy = fft{y,16184};
Corltion = Sx.*conj(Sy);                   %cross-correlation of x,y.
power = Corltion.conj(Corltion)/16184 %PSD of cross-correlation.
```

Feasibility to differentiating sliding and touching was investigated using cross-correlation method. It was expected that they could be discriminated in terms of frequency properties. But, following results shows it is difficult, since both sliding and touching have much similarity in terms of frequency properties (see Figure D.3).

Figure D.1 shows three samples of impact and their cross-correlation as the impact pattern. Figure D.2 shows three samples of sliding and their cross-correlation as the sliding pattern. Figure D.3 shows the cross-correlation of the sliding pattern and the impact pattern. The graph shows the similarity between the two. They have much similarity along the frequency domain, hence they can hardly be discriminated by filtering and cross-correlation method.

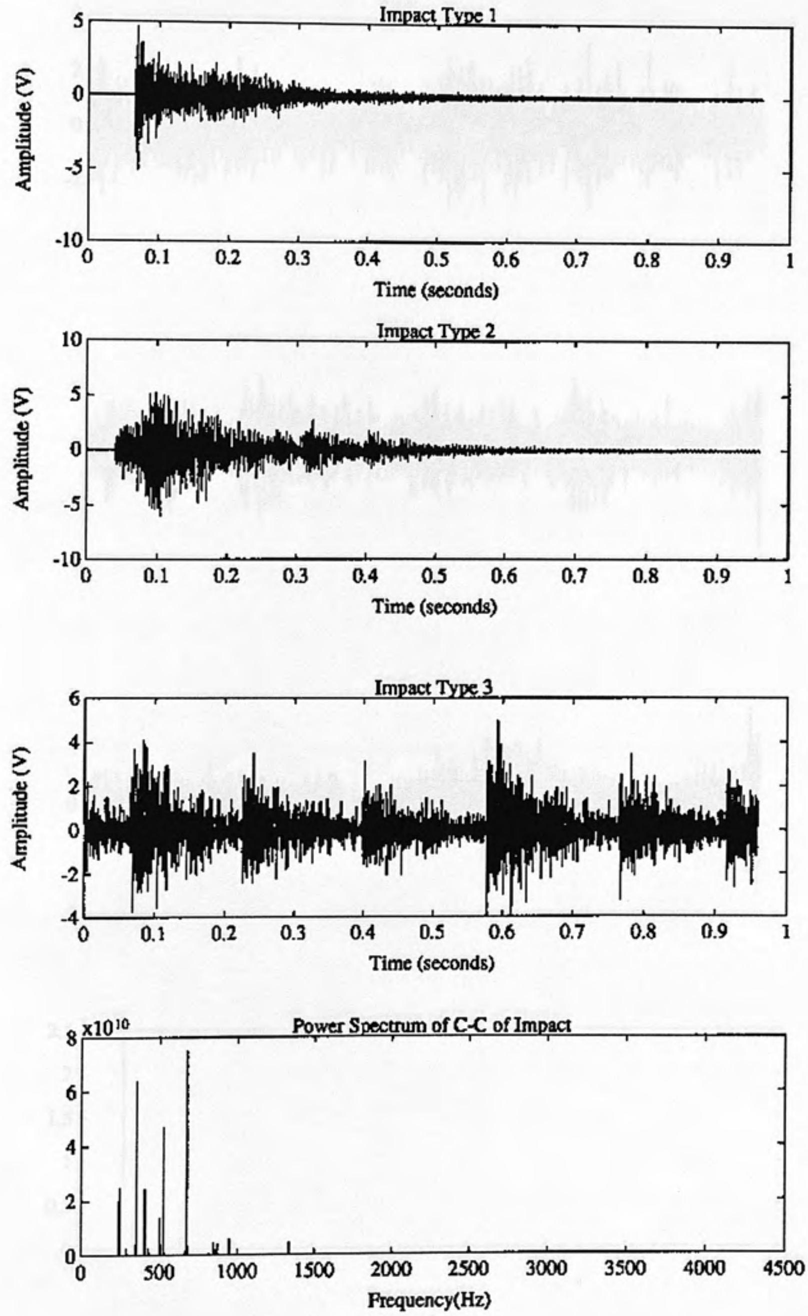


Figure D.1: Three impact samples and their pattern obtained by cross-correlation

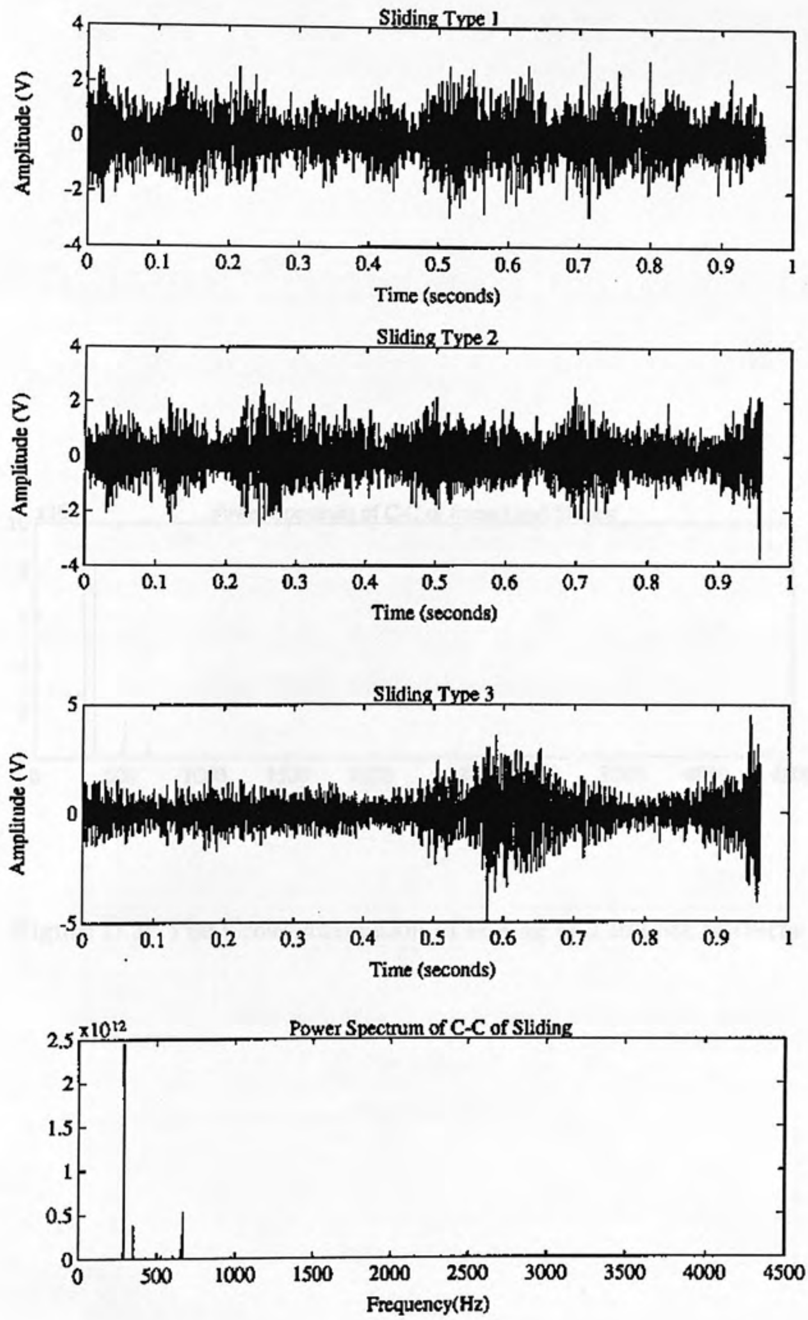


Figure D.2: Three sliding samples and their pattern obtained by cross-correlation

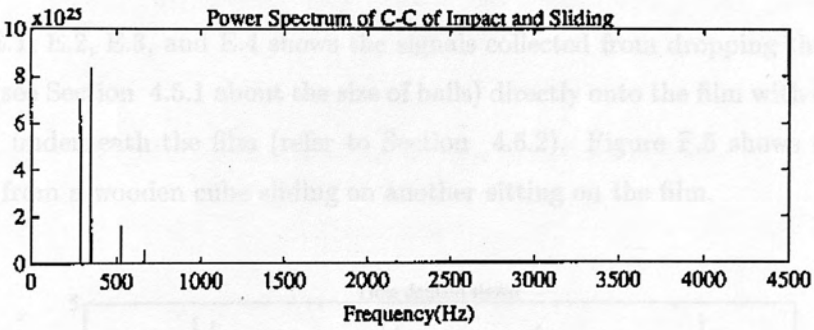


Figure D.3: The Cross-correlation of sliding and impact patterns

Appendix E

Vibrations from Touching onto the Film

Figures E.1, E.2, E.3, and E.4 shows the signals collected from dropping the smallest size ball (see Section 4.5.1 about the size of balls) directly onto the film with compliant materials underneath the film (refer to Section 4.6.2). Figure F.5 shows the signal collected from a wooden cube sliding on another sitting on the film.

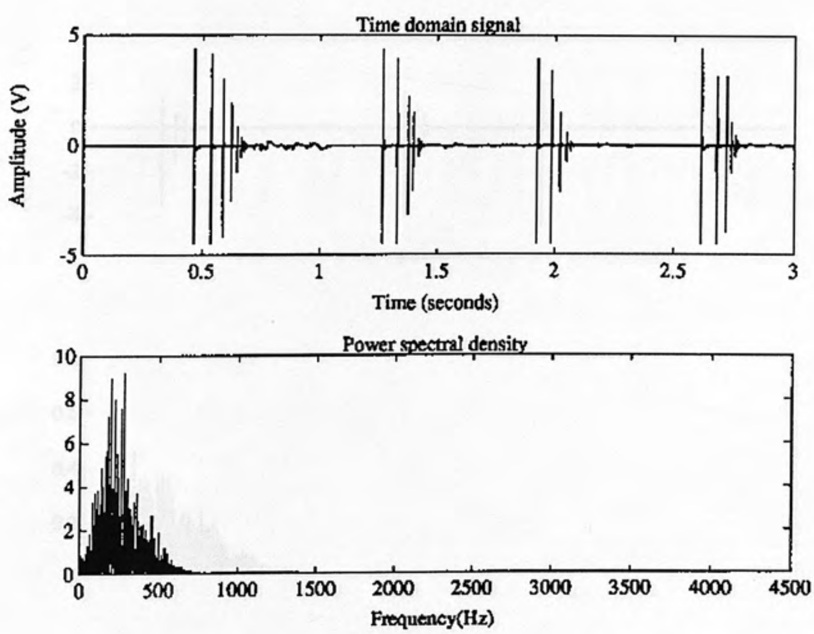


Figure E.1: Plane sponge as a compliant material

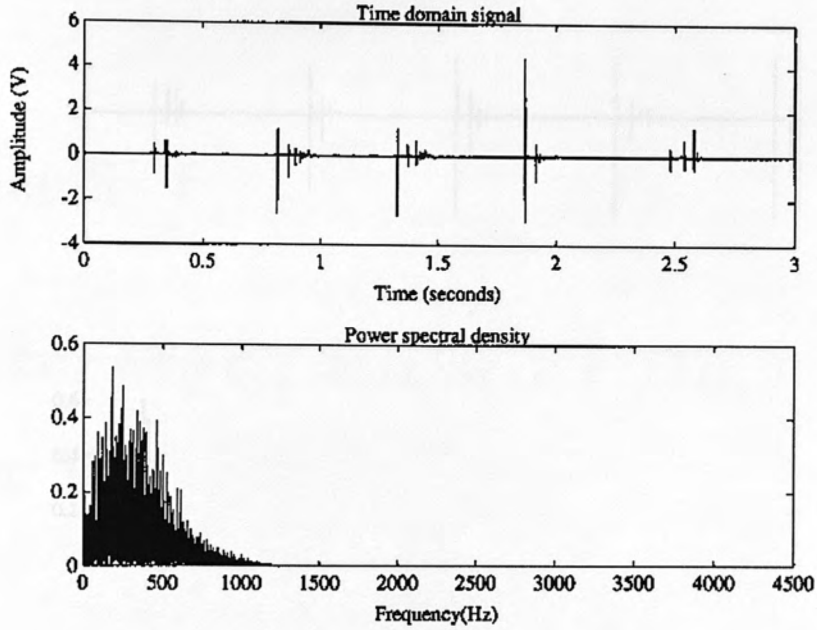


Figure E.2: Antistatic foam as a compliant material

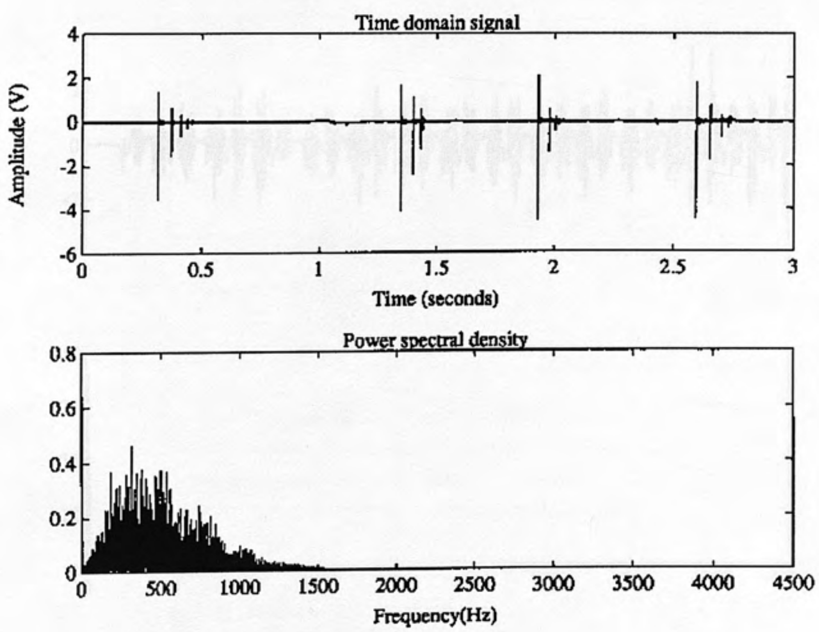


Figure E.3: Styro foam as a compliant material

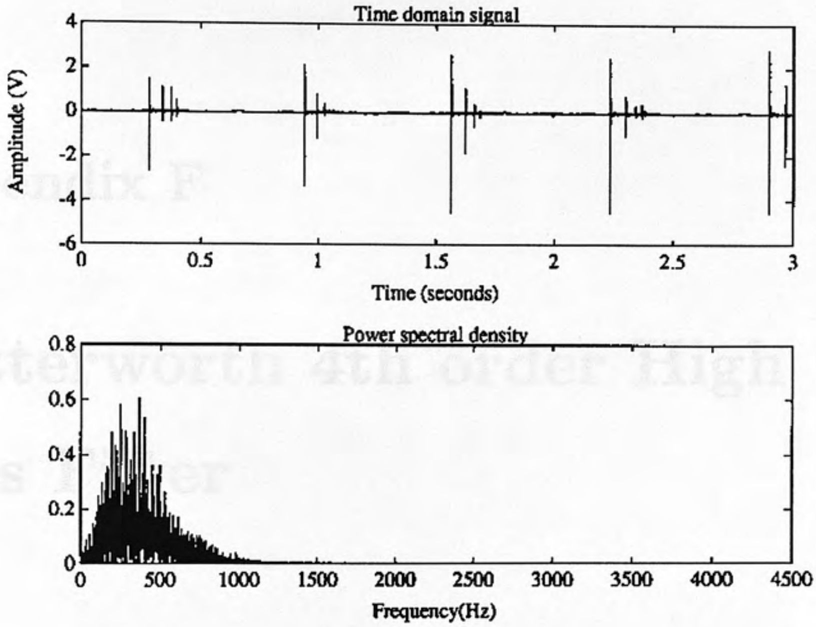


Figure E.4: Leather as a compliant material

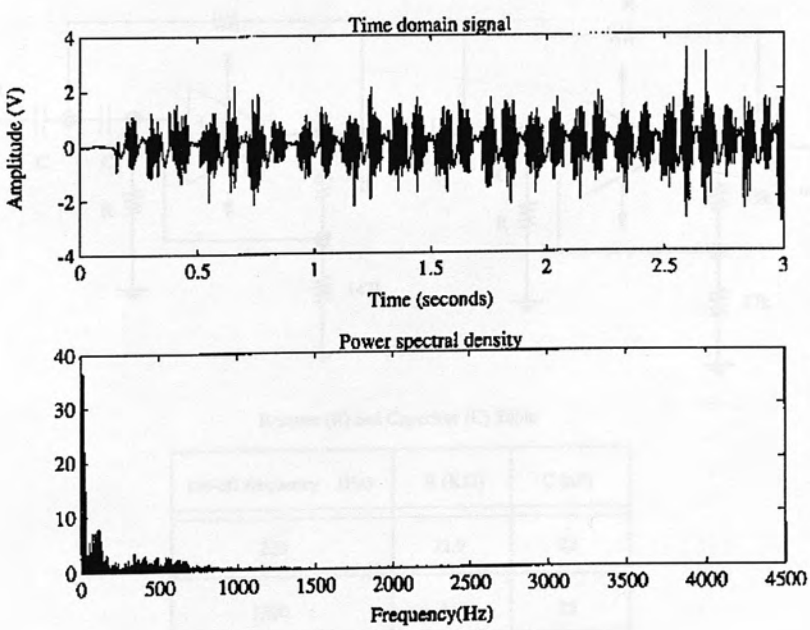
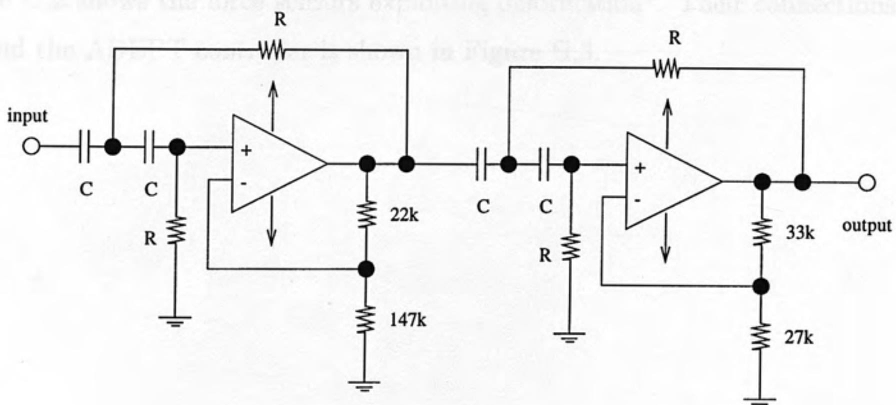


Figure E.5: Sliding of wooden cubes on the film

Appendix F

Butterworth 4th order High Pass Filter

Figure F.1 shows the circuit diagram of a Butterworth 4th order high pass filter from [Horowitz & Hill 89]. Calculated parameters are shown in the table, for the cut-off frequencies 220 Hz and 1200 Hz respectively.



Resister (R) and Capacitor (C) Table

cut-off frequency (Hz)	R (KΩ)	C (nF)
220	32.9	22
1200	6	22

Figure F.1: Butterworth high pass filter

Appendix G

Sensor Curcuit Diagrams and Connections

The electronic circuits of the sensors developed and their interconnections with the PC and the ADEPT controller is shown.

Figure G.1 shows the noise cancelling sensor (top), and the table sensor (bottom). Figure G.2 shows the force sensors exploiting deformation[1]. Their connections to the PC and the ADEPT controller is shown in Figure G.3.

[1] the $1\mu F$ capacitor and the $100k\Omega$ resistor at the end are redundant.

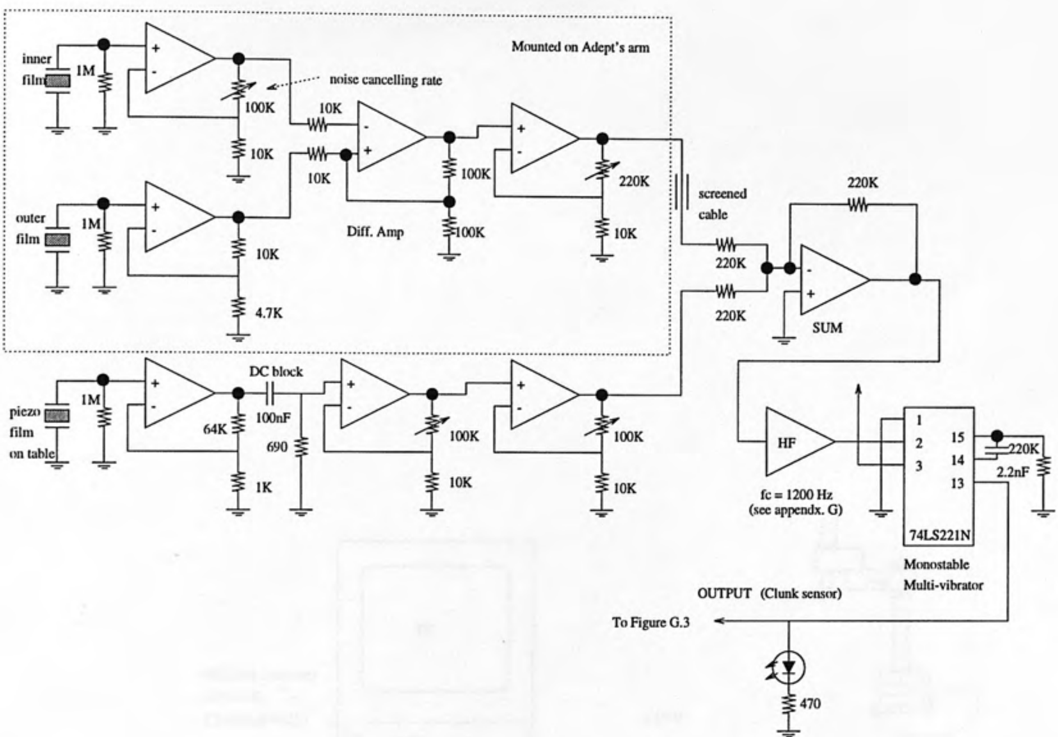


Figure G.1: The Clunk Sensor

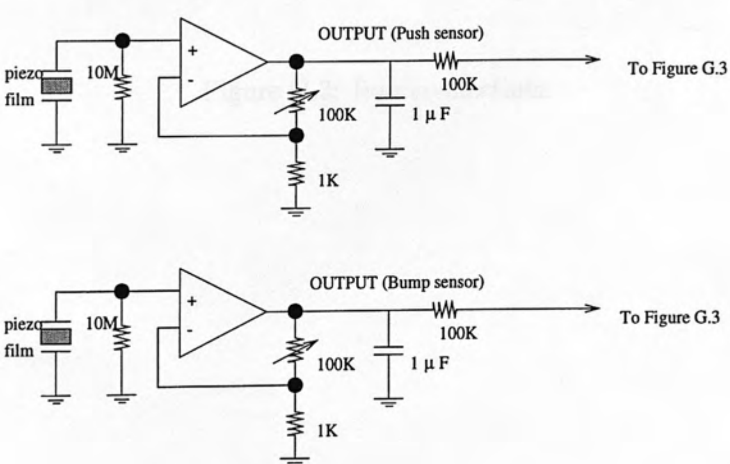


Figure G.2: The force sensors

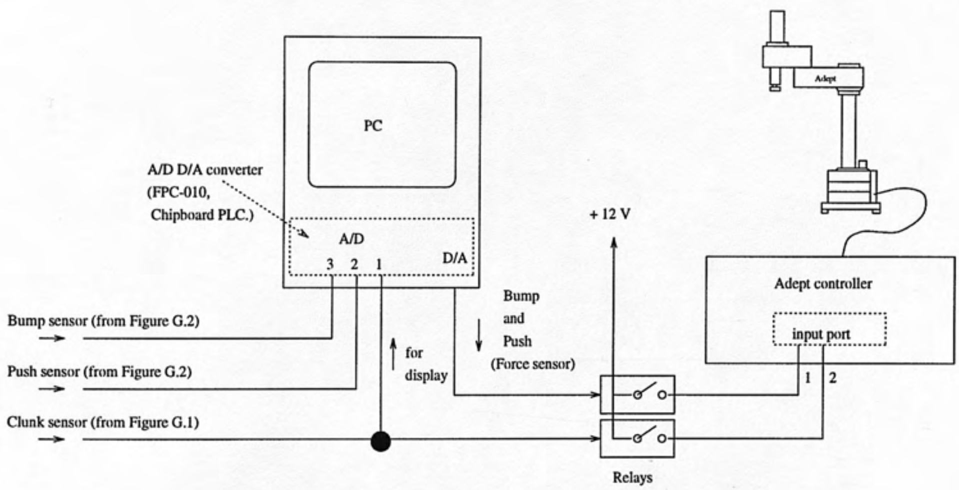


Figure G.3: Interconnections