Accomplishing Task-Invariant Assembly Strategies by Means of an Inherently Accommodating Robot Arm

Graham Edwin Deacon

ARTIFICIAL INTELLIGENCE LIBRARY
UNIVERSITY OF EDINBURGH
80 South Bridge
Edinburgh EH1 1HN

Ph.D.
University of Edinburgh
1997

Abstract

Despite the fact that the main advantage of robot manipulators was always meant to be their flexibility, they have not been applied widely to the assembly of industrial components in situations other than those where hard automation might be used. We identify the two main reasons for this as the 'fragility' of robot operation during tasks that involve contact, and the lack of an appropriate user interface. This thesis describes an attempt to address these problems.

We survey the techniques that have been proposed to bring the performance of current industrial robot manipulators in line with expectations, and conclude that the main obstacle in realising a flexible assembly robot that exhibits robust and reliable behaviour is the problem of spatial uncertainty.

Based on observations of the performance of position-controlled robot manipulators and what is involved during rigid-body part mating, we propose a model of assembly tasks that exploits the shape invariance of the part geometry across instances of a task. This allows us to escape from the problem of spatial uncertainty because we are no longer working in spatial terms. In addition, because the descriptions of assembly tasks that we derive are task-invariant, *i.e.* they are not dependent on part size or location, they lend themselves naturally to a task-level programming interface, thereby simplifying the process of programming an assembly robot.

However, to test this approach empirically requires a manipulator that is able to control the force that it applies, as well as being sensitive to environmental constraints. The inertial properties of standard industrial manipulators preclude them from exhibiting this kind of behaviour. In order to solve this problem we designed and constructed a three degree of freedom, planar, direct-drive arm that is open-loop force-controllable (with respect to its end-point), and inherently accommodating during contact.

In order to demonstrate the forgiving nature of operation of our robot arm we implemented a generic crank turning program that is independent of the geometry of the crank involved, *i.e.* no knowledge is required of the location or length of the crank. In order to demonstrate the viability of our proposed approach to assembly we programmed our robot system to perform some representative tasks; the insertion of a peg into a hole, and the rotation of a block into a corner. These programs were tested on parts of various size and material, and in various locations in order to illustrate their invariant nature.

We conclude that the problem of spatial uncertainty is in fact an artefact of the fact that current industrial manipulators are designed to be position controlled. The work described in this thesis shows that assembly robots, when appropriately designed, controlled and programmed, can be the reliable and flexible devices they were always meant to be.

For my parents: they deserve more.

Acknowledgements

I would like to take this opportunity to thank all those people that made the robot research lab. such a pleasant place to work in: Peter Balch, Prabhas ('Piak') Chongstitvatana, Alistair Conkie, Jim Donnett, Mitch Harris, Edward Jones, Taehee Kim, Lykourgous ('Akis') Petropoulakis, Giovanni Pettinaro, and Myra Wilson. Thanks especially to Jim for helping to get my practical work off the starting blocks. There's a distinct possibility that without his influence I would still be waiting for the starting gun! Thanks also to Mike Cameron-Jones for some interesting discussions during the early stages of this work, and to Mark Wright for some sanity-saving discussions during the final stages. Thanks again to Mark for generating the portion of configuration space obstacle in figure 3.3, and to Mark and Andrew Fitzgibbon for proof-reading various bits and pieces. Of course, any errors that remain are in the bits they didn't read.

I am grateful to the Science and Engineering Research Council (SERC) for providing two years of studentship money. An especially big thank you to the people at the Sowerby Research Centre of British Aerospace Plc, who agreed to sponsor me when my SERC money dried up, and who were able to be persuaded into financing the construction of a novel robot manipulator at a time when I should have been writing up! Thanks also to the people at the Edinburgh University Technology Transfer Centre who made sure this liaison had a firm footing.

More recently this work has been supported by the SERC ACME directorate, grant number GR/J94372, with Chris Malcolm as the principal investigator. Chris has been my supervisor throughout my long(ish) stay in Edinburgh. I'd like to thank him for not giving up on me, especially whilst I lurched from one financial crisis to the next, and for allowing me the chance to attempt something that appeared to fly in the face of common wisdom.

I also owe a massive debt of thanks to the Department of Artificial Intelligence's support staff. The administrative and computing staff have always been more than helpful, but my biggest debt is to the department's technicians for their work in the construction of EDDIE: Tom Alexander, Hugh Cameron, Alexander Colquhoun, Douglas Howie, Martin Hughes, Alasdair MacLean, Neil Wood, and David Wyse. I'd especially like to thank David, who did the bulk of the mechanical construction design work, and Hugh, who postponed his holidays to make sure that I had an experimental platform to work with. It was Tom that first mooted the idea of controlling the arm by a joystick. Thanks also to Dougie for taking the photos of EDDIE and Craig Strachen for scanning them.

I ought also to mention the Tuesday lunchtime badminton players. It just wouldn't have been the same without our weekly thrash-around at the Sports Centre. You really taught me the meaning of open-loop behaviour.

Finally I'd like to thank Louise and Lindsey for putting up with me, or rather putting up with a lack of me, over the last few years. (Actually, I have a suspicion that they prefer it that way!)

Declaration

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

G.E. Deacon
Edinburgh
11^{th} April 1997

Contents

List of Figures

Chapter 1

Introduction

1.1 The Promise of Assembly Robots

The high cost of dedicated assembly machinery, or hard automation, can be justified when the article being manufactured is going to be produced for a long period of time (typically several years). However, it has been estimated that such high volume production constitutes only 5% of all manufactured goods [Nevins & Whitney 78]. Robots were meant to make the automatic assembly of smaller batch sizes feasible by being sufficiently flexible to be applied to the assembly of more than one product. It would then be possible to recover the cost of buying a robot by spreading it over the returns from the assembly of a number of products distributed over a number of years. The main source of a robot's flexibility was considered to be due to the fact that a new product could be accommodated by changing the controlling software rather than the parts handling machinery itself.

However, despite initial optimism [Engelberger 80], robots have fallen short of fulfilling their promise. Shop floor installations have found it necessary to use special tools and parts holders in order to constrain the environment to a sufficient degree to make it practical to use a robot. Such jigs and fixtures are usually designed specifically for a particular object or product, thus detracting from the supposed flexibility of a robot workcell. Moreover, since typical assemblies contain up to 20 components [Redford & Lo 86], the total cost of auxiliary fixtures can end up comparable to the cost of the robot that they are intended to serve. As a result the batch size for which it is economically justifiable to invest in a robot can be strongly influenced by the

auxiliary hardware required, making claims about the flexibility of software much less significant.

In addition to this, the batch size for which it is feasible to apply hard automation techniques is reducing. In other words, if assembly robots are to find a niche in between dedicated, hard automation and the flexibility of human labour they need to be even more flexible than originally anticipated. It seems that the realisation of this fact has started to shift the interests of robotics researchers into much simpler systems that attempt to solve well circumscribed problems, rather than aiming for the originally envisaged "general purpose manipulator"—see for example [Canny & Goldberg 94].

So why have robots failed to deliver? The answer to this question is almost certainly the reason why specialised jigs and fixtures are necessary for any practical application: a major limitation of current industrial robots is their inability to cope with a feature of the world that we are able to take in our stride, that of the uncertainty inherent in dealing with the real world; more specifically, robots are limited by their inability to cope with *spatial* uncertainty.

This inability inevitably leads to assembly failures, thus detracting from the supposed advantages of robots over manual labour: "It is generally considered that robots, when compared to humans, yield more consistent quality, more predictable output, and are more reliable" [Owen 85, p.151]. In practice however, in order to keep a robotic workcell running it is invariably necessary to provide a human overseer, whose job it is to extract the robot from any failure situations, and either put the workcell into a state from which the assembly can be continued or, more likely, reinitialise the cell after removing the components of the failed assembly. The advantages of robots over manual labour amount to the fact that they don't get bored, tired, go on holiday or get involved in industrial disputes, *i.e.* they cope admirably with the repetitiveness of a production line.

There is however one major advantage that human workers have over robots: they are far better at dealing with unexpected situations. This is in part due to their ability to sense their environment, whereas a standard industrial manipulator is only able to sense its joint positions. Once this observation is made, augmenting robots' capabilities by providing them with exteroceptive sensors seems like the next natural step to take.

This however is no trivial proposition. Apart from problems arising due to the technology used (signal noise, drift, nonlinearities, incompatibility with other systems, *etc.*) there are problems concerning how sensor output should be processed in order to provide usable information (this is most apparent in machine vision systems). Moreover, there is currently no theory to guide users in how sensors would best be deployed[1]. As a result current applications use sensors in an *ad hoc* and unprincipled fashion: invariably sensors are only employed when a manufacturer finds that there is a susceptibility for a workcell to be unreliable at a particular point in an assembly sequence. A sensor is chosen in order to discern, whenever this point in the assembly cycle is reached, whether things are going as expected. In other words, sensors are most often used to try and detect the occurrence of anticipated errors.

Even in a well-tuned assembly cell on the factory floor, assembly errors can occur as much as 20% of the time (despite the use of a number of rudimentary sensing systems) [Davey & Selke 85]. In 'experimental' situations this figure can be much higher (typically 30%) [Hardy *et al.* 89]. According to [Malcolm & Fothergill 87] manufacturers often want failure rates that are less than 1 in 1000. As a consequence error recovery and motion planning with uncertainty have become increasingly active fields of research.

An alternative to using a sensing strategy to improve an assembly operation is to use a motion strategy. The idea here is that the parts are manipulated in such a way as to increase the likelihood that the outcome of the operation is the desired one: in the limiting case the desired outcome is guaranteed. An interesting discussion of various ways in which the mechanics of a task can be exploited in order to achieve some desired outcome is given in [Mason 85]. This has also become a very active field of research.

1.2 Raising the Level at which Assembly Robots are Programmed

Claims about the advantages accrued due to the use of software over hardware also lose their force when it is realised that even when the tasks involved are relatively

[1] Erdmann has in fact recently started a research programme into the design of sensors based on the information necessary to complete a task successfully [Erdmann 92].

simple, which is usually the case, the cost of programming a single robot application may be comparable to the cost of the robot itself [Lozano-Pérez 82]. Additionally, the expected portability of this software has not been realised: robot assembly programs tend to be *very* product-specific. In order to get round these problems attempts have been made at raising the level of abstraction at which robots are programmed. The reasoning being that, in the same way that computer programming languages in general have become more and more high level, and less dependent on machine details, a higher level robot programming interface would provide a means to get away from application-specific details. These attempts can be classified in various ways, see for example [Koutsou 81], [Bonner & Shin 82], [Lozano-Pérez 83a]. The following taxonomy of robot programming levels is after [Malcolm & Fothergill 87]:-

- Joint Level

  This is the lowest level. At this level the motions of the robot are specified in terms of the required positions of its joints. This is usually achieved by what is known as teaching by showing, where either the robot is moved manually through the required motion sequence, or a teach pendant is used to drive it, whilst the system either records the joint values at the end-points of each discrete motion, or samples the pattern in real time and records the joint values at fixed time intervals.

  The main advantage of this method is that it requires no specialised knowledge on the part of the programmer. The main disadvantage is that the programmer specifies a single motion sequence; there are no loops, conditionals or computations. It has been argued [Lozano-Pérez 83a] that if a robot's actions are to be more flexible, in particular, if it is to respond to sensory input and computations performed on sensor readings, then robot programming requires the capabilities of a general-purpose computer programming language.

- Manipulator Level

  At this level the required sequence of positions defining the robot's motion can be entered in a textual form, although this does not necessarily rule-out the ability to teach by showing. Most languages at this level extend more traditional languages

such as BASIC or PASCAL by the inclusion of the abilities to interact with sensors and specify robot motions. This level represents the current state of the art in commercially available industrial robots, *e.g.* VAL-II [Shimano *et al.* 84] (which has been used by both Unimation and Adept).

The main benefit of this level of programming is meant to be that information from sensors can be used to modify the actions of the robot, and as a consequence they can cope with a greater degree of uncertainty in the locations of objects in their environment. However, as mentioned above, it is still not well understood how best to employ sensors, and their addition increases both the cost and complexity of an installation. Another drawback is that the robot programmer needs to be a competent computer programmer, which is not the case for most workers on the factory floor.

- Object Level

This level of programming describes the assembly task as a sequence of situations, where a situation is a description of the spatial relations that are to hold between features of the objects to be assembled. RAPT [Ambler & Corner 82] and LM-GEO [Mazer 84] are two examples of languages at this level. The motions required of the robot are derived by a geometric inferencing engine which deduces the appropriate transformation between situations [Ambler & Popplestone 75], [Popplestone *et al.* 80], [Popplestone & Ambler 83], [Mazer 82].

A reason for raising robot programming to this level is that it allows the user to describe the task in what is considered to be the natural way for people to describe assembly operations [Corner *et al.* 83]. However, although once the spatial relations are specified these languages can compute the required positions, the user still has to specify how the assembly is to proceed, *i.e.* provide a parts mating strategy, as well as a geometric model of each of the objects involved (including the robot gripper). The main disadvantages of these languages is that they rely on the world to conform to their geometric model: this ignores the unpredictability of the real world and leads to the requirement that the world be constrained in such a way as to make the geometric model a reasonable approximation to

reality[2]. Programs written in LM-GEO generate code in the manipulator level language LM [Latombe & Mazer 81], which provides the ability to interact with sensors, but the manner of interaction has to be made explicit by the programmer, *i.e.* the criticisms of manipulator level programming still apply.

- Task Level

  The specification of an assembly operation in a task level language would resemble very closely the kind of instructions that a human might use, and any decisions on how the robot should move or how sensors should be used would be completely hidden from the user. Two such systems were defined some years ago, LAMA [Lozano-Pérez 76] and AUTOPASS [Lieberman & Wesley 77], but neither have been completed.

  The main advantage of such a system should be obvious; very little sophistication is required of the user. On the other hand, one criticism levelled against AUTOPASS is that the high-level of the statements in the language leads to ambiguities between the user's intended actions and the robot's interpretation of them [Bonner & Shin 82].

  It appears that the main reason why these task-level systems were never completely implemented is the lack of a clear choice of what motion and sensing strategies to employ for a particular task, *i.e.* the same old problem of spatial uncertainty and how best to cope with it. This is not surprising since the results of these systems were to be manipulator level programs, and if these problems are yet to be solved for that level of programming then they are not going to be solved by moving to a level that subsumes it.

As a consequence of the debilitating effects of spatial uncertainty on attempts to raise the level at which robots are programmed, research in robotic assembly in recent years has tended to be concerned with more delimited subproblems *e.g.* path finding, grasping and parts mating, hoping that the solution of these problems on an individual basis will ultimately lead to a high-level programming system via their union [Latombe 83].

[2] In fact it might be more accurate to say that the world needs to be constrained in order to make reality a reasonable approximation to the geometric model.

The work described in this thesis is an investigation into performing part mating operations. In order to avoid the additional complexities of grasping and path finding we will be working with objects that are physically attached to our manipulator, and will assume that the objects to be manipulated start off either touching, or very nearly so. We will further assume that there are sufficient kinematic constraints to provide motion guides towards a desired end configuration of parts. Since objects to be assembled are invariably designed such that various features are meant to be against one another, we don't consider this situation to be overly restrictive[3].

1.3 The Contribution of this Thesis

Whilst work towards a task-level programming interface has been going on in various research laboratories around the world, the spread of robots into manufacturing assembly has been very much slower than originally anticipated, and as a consequence a number of robot manufacturers have gone out of business. It is our contention that the root problem behind this phenomenon is the critical susceptibility of current industrial robots to spatial uncertainty, which seriously limits their reliability and even applicability to assembly tasks. Moreover, we claim that this arises from the fact that current industrial robots are position controlled devices; it is because current industrial robots are position controlled that the positions of objects in their environment need to be known in accurate geometric terms, this need makes the operation of current assembly robots susceptible to small positional variations, and it is this susceptibility that makes the operation of assembly robots so unreliable. This situation creates a requirement for precise, and therefore expensive, jigs and fixtures, thereby reducing the viability of robots in automated assembly. Additionally, it makes the task of programming a robot a major undertaking because each move needs to be exactly right. These two things together mitigate against the use of robots in assembly on the factory floor.

We intend to demonstrate that this root problem of spatial uncertainty is in fact an artefact of the way that robots are currently controlled. In order to do this we show that assembly tasks can be described in terms that are independent of metric spatial information, using instead the contact configurations possible between objects.

[3] The work described in [Lozano-Pérez *et al.* 84], for example, makes a similar assumption.

To facilitate performing assembly in these terms we have constructed an inherently accommodating, force-controllable robot arm. This makes it possible to adopt sliding behaviour as the basic motion primitive for assembly operations, and to navigate an assembly task in terms of contact configurations. As a consequence we arrive at a system where it is possible to issue such commands as "put the peg in the hole", and the system can be relied upon to achieve this goal without precise knowledge of the sizes of the parts involved, or the location at which the operation is to occur[4]. In other words, by moving away from position control and the trajectory model of robot programming, and exploiting instead sliding behaviour, we are able to specify task-invariant assembly strategies that apply to all instances of a task, thus creating the possibility of a task-level programming interface and all the benefits associated with it.

It is probably worth pressing home the point that it is the independence of these strategies from metric spatial information that allows us to escape the spatial uncertainty problem. After all, if you don't need to know something, then it doesn't matter if you are uncertain about it!

1.4 An Overview of this Thesis

In the next chapter we review some of the work that has attempted to make assembly by robot more robust and reliable. Most of this, in one way or another, attempts to tackle the problem of spatial uncertainty. In Chapter 3 we derive a representation of rigid-body assembly tasks in terms of the shape of the parts involved, rather than a detailed geometric model. In Chapter 4 we show that we are able to derive assembly strategies at the level of abstraction of this representation, and as a consequence we are able to escape the spatial uncertainty problem. Moreover, these assembly strategies are task-invariant, *i.e.* they apply to all instances of a task irrespective of the size of the objects involved or their location in space (provided, of course, we work within the physical limitations of the robot[5], *i.e.* within its reach, payload, *etc.*). Consequently

[4] In the assembly tasks described in Chapter 6 we use orientation information, but it used in a guiding sense, rather than being critical to the control of the assembly task.

[5] The notions of 'task' and 'task-invariance' are, of course, interrelated. In section 3.5 we consider in more detail what we mean by these terms.

they lend themselves naturally to a task level programming interface. We note here also that exteroceptive sensory information is intrinsically part of the approach, rather than being treated as a 'bolt-on' afterthought. However, this technique requires that we use a manipulator that is sensitive to environmental constraints. Since the inertial properties of standard industrial manipulators preclude them from this kind of operation we constructed a custom device with which to perform some experiments. In Chapter 5 we describe a three degree of freedom, planar, direct-drive arm, that we built in order to test the approach described in Chapters 3 and 4, and explain its advantages over robot manipulators constructed in a more traditional manner. Chapter 6 describes some representative experiments using our shape-invariant approach to assembly and our custom hardware. The final chapter concludes that the problem of spatial uncertainty is an artefact of designing robot manipulators to be position-controlled, and expounds how we think the work described here can be profitably extended.

Chapter 2

Spatial Uncertainty in Robotic Assembly

The work related to coping with spatial uncertainty to date can be divided into two basic approaches; the first arises from the view that errors are inevitable and hence require some method for recovery, the second attempts to optimise the possibility of success.

It is worth noting that it is possible for parts to be damaged or mis-manufactured such that they rule out the possibility of a successful assembly operation (without some form of error recovery). It is however possible to try and generate strategies that are guaranteed to succeed providing the parts involved are formed such that they can fit together in the intended manner. In practice parts sufficiently malformed to preclude the possibility of successful assembly are very rare. Consequently we will not be dealing explicitly with this problem in this thesis.

2.1 Error Recovery

Error recovery is a major research area in its own right and work tends to be broader in scope than just being concerned with rigid-body assembly. Since we can't hope to do justice to all of the work in this area we will content ourselves in this section with a brief description of the nature of this endeavour.

Some common features about the work in this area are the model of the assembly process that it uses, and the fact that it has a strong expert systems flavour to it. The

model of the assembly process invariably employed is that the robot and workcell begin in some initial state, and by a sequence of transformations through intermediate states arrive at some desired goal state. Such a sequence of transformations is the robot assembly plan. An error is considered to be a failure of one of these transformations which takes the robot (and workcell) instead into some error state. The error recovery problem then becomes that of taking the robot from the error state to the goal state. There are two basic approaches to this problem: forward error recovery and backward error recovery. Forward error recovery entails going from the error state to some point in the original plan closer to the goal than where the error occurred. Backward error recovery takes the robot from the error state to some point in the original plan earlier in the original plan than the point at which the error occurred. Forward error recovery is generally considered to be the more difficult of the two. An extreme form of backward error recovery is to remove all of the parts of the failed assembly and start the assembly cycle again. Representative work in this area includes [Srinivas 78], [Barnes *et al.* 83], [Lee *et al.* 84], [Smith & Gini 86], [Trevelyan & Nelson 87], [Chang *et al.* 89]. Other related work that places less emphasis on expert systems techniques includes that described in [Shen *et al.* 90], [Taylor 90], [Wilson 92].

2.2 Attempts at Improving the Likelihood of Successful Assembly

2.2.1 Worst Case Analysis

The approach most often taken to errors, as noted in Chapter 1, is to perform a particular sensory check at the appropriate time. If we wanted to avoid all errors this would mean providing sensory checks for all eventualities. There are certain problems associated with this:

> The laws of the universe state that there will be at least one failure mode for which a program check has been left out. This is God's way of teaching humility to engineers (who rightly regard the Babel affair as a management, and not a technical, failure) and computer programmers (who seem to like a profusion of languages). Even if it were, in fact, possible to anticipate

> and test for *all* failures, it would not necessarily be economical to do so. [Taylor 76, p.45].

This observation led Taylor to try and model explicitly the uncertainties inherent in the assembly of rigid body parts. Part of Taylor's solution to the parts mating problem (also known as the fine motion planning problem) was to use partially specified strategies, known as skeletons (see also [Lozano-Pérez 76]). These are parameterised robot programs for particular tasks that include motions, error tests, and computations, where many of the parameters are left unspecified. Taylor developed an algebraic system which derives constraints on the possible positions of parts from the relationships between their features. These error estimates were propagated through a structure of parts and used to make decisions for choosing a strategy and filling in the values of parameters.

This analysis was extended by Brooks who used a symbolic representation of the problem that allowed the calculations to go both ways [Brooks 82]. That is to say that Brooks' system is both able to infer the effects of actions and the propagation of errors, as well as being able to calculate the required tolerances on a set of parts given a desired end-configuration. It is also possible to check the progress of an assembly and should the spatial uncertainty be too large a sensor check to reduce the uncertainty could be included in the assembly plan. This system deals with robot motion planning with uncertainty in general and is not limited to fine motion planning.

Fleming has investigated the effects of the spatial uncertainties in a structure of parts [Fleming 87], by extending Requicha's work on geometric tolerancing [Requicha 83]. A tolerance zone is defined as a region in space in which a real feature must lie; a datum is defined as an infinite plane or straight line or a point embedded in a part. Fleming shows how a toleranced part can be represented as a network of tolerance zones and datums connected by arcs to which inequality constraints are attached, and how such a representation can be extended to deal with assemblies of parts. This work was intended to be integrated with the RAPT programming language, but proved to be very computationally demanding.

Configuration space was introduced into robotics as a representation for solving the problem of planning paths amongst obstacles. The basic ideas were presented in [Udupa 77], were later generalised in [Lozano-Pérez & Wesley 79], and formalised in [Lozano-Pérez 81]. [Lozano-Pérez 83b]. An object's configuration space is the space formed by the range of possible values that a set of parameters that uniquely specify an object's position can take. Configuration space obstacles correspond to real obstacles but in configuration space they occupy the regions that correspond to the values of the configuration parameters not available to the object whose configuration space it is (not available because another object is there). It is possible using this representation to shrink the moving object to a reference point whilst growing the configuration space obstacles by the appropriate amounts in the appropriate dimensions to compensate. The uncertainty of the situation can be accommodated by growing the obstacles a bit more than necessary so as to include a safety margin. This transforms the problem of planning a path for a polyhedral object in a cluttered environment to the problem of planning a path for a point in a (slightly more) cluttered environment. One problem with this is that in general an object's configuration space is six-dimensional which is difficult to visualise. It is still easier however, mathematically, to plan the motion of a point in configuration space than to plan the motion of an object with extent in Euclidean space. The culmination of a number of years research into this approach is described in [Lozano-Pérez *et al.* 92]. This documents a working system that is able to plan pick and place operations in relatively complex environments, given a specification of the desired final configuration of the part, and geometric and kinematic models of the robot and the environment. Configuration space techniques are also used for gross motion planning in the SHARP automatic robot programming system [Laugier 88].

The publication of [Lozano-Pérez *et al.* 84] introduced the use of configuration space to the development of a formal approach to the synthesis of fine motion plans based on a geometric description of the parts involved, and explicit estimates of errors in sensing and control. The problem is turned into planning the trajectory of a point by transforming the geometrical properties of the situation into the configuration space of the moving object. The basic technique is to define the goal region in configuration space that the reference point of the object is to reach, and then identify the range of positions from which this point can recognisably reach some point in the goal in

a single motion. This range of positions is known as the pre-image of the goal. If the pre-image doesn't contain the current position of the reference point of the object, then the pre-image computation can be applied recursively, using each pre-image as an intermediate goal.

The motion of the object is determined by the velocity of the robot. This is subject to a certain amount of uncertainty which is modelled by an uncertainty cone, the apex of which is the starting point. The starting point itself is not known with complete confidence and so is modelled as lying somewhere within a ball of uncertainty. The confidence with which the reference point of the object can be known diminishes with robot motion and as the range of possible values of the reference point grows. To accommodate this phenomenon a strong pre-image is defined as the "locations for which all motions within the range of velocity uncertainty will reach the goal." Position uncertainty is accommodated by shrinking the goal by an amount equal to the uncertainty in position, thereby guaranteeing that any sensed position value indicating that the reference point is in the goal region will in fact be correct in that respect. This provides the recognisability criterion mentioned above, provided that none of the dimensions of the goal region are less than the uncertainty in position. If they are, velocity and/or time can be used in the termination predicate (*i.e.* as the conditions that guarantee that the goal has been reached).

The range of strong pre-images is increased by introducing compliant motion using generalised damper dynamics [Nevins & Whitney 73]. This allows motion of the robot to be modified by physical constraints of the task such that the parts in contact can slide over each other. Whether the parts actually stick or slide depends on the friction between them. Friction is modelled by a "friction cone" with the apex at the point of contact and centre line along the surface normal. The angle at the apex of the cone is proportional to the friction coefficient[1]. In order to guarantee that the object won't stick, the elements of the velocity uncertainty cone must not point into the friction cone. A number of other authors have built on or elaborated on the procedure outlined in [Lozano-Pérez *et al.* 84].

[1] See Chapter 4 for a derivation of this representation of friction.

Donald's work on error detection and recovery [Donald 86], [Donald 87], [Donald 88] extends the [Lozano-Pérez *et al.* 84] framework to include geometric model uncertainty as well as the uncertainty arising from sensor errors and control errors. Each of the uncertain parameters in a model Donald treats as lying somewhere on an interval of possible values. The Cartesian cross-product of all the intervals of the uncertain parameters and configuration space Donald defines as "generalised configuration space". He defines a set of "physics axioms" for this space which allow the use of pre-images to generate guaranteed strategies in it. The physics axioms of generalised configuration space however, reduce the likelihood of finding (strong) pre-images in this domain, so Donald allows the possibility of using weak pre-images. (This is the set of start points from which a commanded motion could *possibly* reach a goal recognisably.) This modification is tantamount to relaxing some of the worst case analysis constraints. Now, however, the plans generated are not guaranteed to succeed so Donald requires that the failures be recognisable situations. If this is not the case his planner tries to generate a subsequent motion to disambiguate the situation [Donald 88].

A planner which is a variant on the original scheme is demonstrated in [Mason 84] to be correct and complete, that is, if a plan for a given problem exists, the planner converges on a correct plan. However, Mason's planner has been shown not to be generally computable [Erdmann 84]. Erdmann's system uses a termination predicate that only takes into account current sensor readings. This is in contrast to Mason's planner which used the entire sensor history as well as time. Using Erdmann's termination predicate the problem of computing pre-images reduces to the simpler problem of computing backprojections. A backprojection is defined as a region from which any motion commanded along the desired direction is guaranteed to enter the goal, but the goal need not be recognisable. If the information available to the termination predicate is restricted to the current sensor values, reachability and recognisability become separate issues [Erdmann 84]. Erdmann presents a method whereby a given goal can be replaced by one or more subsets which are guaranteed to be recognisable. By backprojecting from these a (strong) pre-image can be computed.

Erdmann implemented a 2-dimensional simulation of his scheme. Buckley adapted Erdmann's algorithm to three Euclidean dimensions and used a damped-spring com-

pliance model in a system for teaching compliant motion strategies, which converges on a solution (if there is one) by interacting with the user to define intermediate goals [Buckley 87], [Buckley 89].

2.2.2 Statistical/Probabilistic Approaches

Statistical and probabilistic methods are most often employed as a way of estimating the values of uncertain parameters. They can be used to form a best estimate of what noisy sensor signals are actually 'saying', or as a way of detecting more long term problems, such as drift in robot accuracy. They can also be used as an indication of the likelihood of the success of an action.

A probabilistic method for calculating the results of positional transformations is given in [Smith & Cheesman 86], [Smith *et al.* 88]. A transformation is considered to have a nominal value about which an uncertainty ellipse can be imposed. This ellipse is a contour of constant probability for a multivariate Gaussian distribution. Compounded positional transformations can be shown to increase this ellipse, whereas sensor readings can be used to reduce its size, making the position of a body more certain.

A related treatment of the problem is given in [Durrant-Whyte 88]. This approach considers geometric features as stochastic point processes described as a probability distribution on the parameter space of the associated object. The manipulation of uncertainty measures then becomes the transformation and combination of probability distributions. The invariant topology of relation between uncertain geometric features is used to develop a method for propagating observations through a world model.

Durrant-Whyte has also been concerned with the optimal combination of uncertain sensory data [Durrant-Whyte 87], [Hager & Durrant-Whyte 88]. These methods rely on models of the sensors, how their information relates to the readings of other sensors, and the best use of a sensor. The sensors are considered to be organised in teams, for which a theory of team decision making is developed.

Kalman filters can also be used for refining uncertain sensor information, see for example [Jones 95] where a Kalman filter is used to improve the estimate of the position of an object from the readings of a force/torque sensor mounted under a surface on

which the object rests. With an appropriate choice of state vector Kalman filters also allow the possibility of robots calibrating and monitoring themselves, correcting taught locations, and checking for long-term statistical effects such as positional drift [Whitney & Junkel 82], [Johnson & Hill 85].

A novel approach is developed in [Sanderson 84] where an entropy measure is used to indicate the degree of uncertainty in the positions of the assembly parts. These uncertainties can be reduced by either sensing or mechanical constraints, thereby reducing the overall entropy. Relative parts entropy is minimised upon completion of the assembly. This measure potentially provides a useful way of comparing the effectiveness of a sensing strategy compared to a motion strategy, and a basis for performing a cost to benefit trade-off.

In [Brost & Christiansen 93] consideration is given to extending the framework of [Lozano-Pérez *et al.* 84] to include probabilistic backprojections. Because strong backprojections can be conservative in their assessment of the success of an assembly operation, and weak backprojections tell you which actions might possibly succeed, but not how likely they are to succeed, Brost and Christiansen suggest adding a measure of the probability of success to regions in the pre-image of an action. This leads to contours of equal probability of success in the space of commanded initial positions. They note however, that the ability to generate these contours for arbitrary tasks doesn't look very promising due to problems of intractability.

2.2.3 Force Guided Motion

Not all attempts at improving the performance of robotic assembly has been in terms of being more precise about global positioning. The paper by Whitney [Whitney 82] summarises a number of years of research into improving the likelihood of successful assembly by studying the forces that can arise during assembly. In particular, the peg in hole problem with a chamfer was studied. A major product of this study was a device known as the remote centre compliance (RCC) [DeFazio *et al.* 84]. The design of this device was precipitated by the realisation that for a peg being fed into a hole by a robot under trajectory control, any misalignments between the peg and the hole acted in such a way as to make things worse. The RCC changes the effects of these misalignment

forces to being error corrective. that is they act so as to reduce the misalignments that caused them. It does this by putting the centre of compliance[2] 'below' the peg: this makes the peg behave as if it were being pulled into the hole rather than pushed. and forces and torques acting about this centre cause motion about the centre that reduces the misalignments that caused them [Nevins & Whitney 78].

Other researchers have used the success of this device as a rationale for constructing control systems that include matrices that encode this error-corrective property. The basic model utilised was established in [Nevins & Whitney 73]. In this paper Nevins and Whitney discuss various control schemes for a robot manipulator that relate force and velocity. In particular, they consider whether a manipulator should be velocity controlled and the forces arising during an assembly monitored and used to modulate the commanded velocity, or whether the manipulator should be force controlled and the resulting velocity used to modulate the commanded forces. They decided on the former option[3], which can be termed admittance control, and this has formed the basis for a significant amount of subsequent research. The work described in [Peshkin 90], [Asada 90], [Hirai & Iwata 92], [Schimmels & Peshkin 92], all describe different ways of synthesising admittance matrices. There has also been devices designed that can implement the strategies embodied by admittance matrices in hardware [Goswami & Peshkin 93].

2.3 Discussion

A general criticism to using a worst case approach to the problem of spatial uncertainty is that it might reject, at the planning stage, operations that could well work in practice because it is unlikely that all the uncertainty parameters relevant to a particular variable will all be at their maximum in a particular direction at the same time, *i.e.*

[2] The centre of compliance is defined as that point at which an applied force will produce a pure translation. and a pure moment applied about a line through the point will result in a pure rotation [Drake 77], [Lončarić 87]. It is conceptually similar to the notion of centre of mass: for a system of passive compliances (*e.g.* interconnected springs) the centre of compliance is the point where the collective compliance of the system appears to be, just as the centre of mass is the point where the mass of a body with extent appears to reside.

[3] This choice is contrary to Hogan's analysis of mechanical impedances [Hogan 85a], because it assumes that the manipulator has the behaviour of an admittance, and that the environment can be described as an impedance. We discuss the consequences of this in section 5.1.4.

these techniques may result in unrealistic overestimates. In addition, in the words of Dufay and Latombe:

> Our experience, ..., is that, while the strategy to be implemented as a fine motion program closely depends on the actual uncertainty of the geometry of the robot and its environment, in many cases it is practically impossible to have realistic prior knowledge about it. Overestimates of uncertainty frequently result in programs which are inefficient in terms of execution time and length of code, because they apply inappropriate strategies and/or they consider situations which will never occur. Underestimates of uncertainty result in unreliable programs." [Dufay & Latombe 84, p.5]

An objection that applies to the static analysis of the tolerancies in assemblies of parts is that the effects of exactly how the parts are to be moved (*i.e.* the mating strategy) ought to be taken into account. For example, [Inoue 79] shows that attempting to put a peg into a hole at an angle (rather than adopting the usual practice of trying to align the peg and hole centre axes) increases the likelihood of a successful operation (provided force information is available to guide the insertion at later stages), because it effectively increases the tolerance between the parts at the point where they are being brought together (see also [Mason 84]). The configuration space planner approach can model this because it specifies the motion of the object.

A general objection to methods that use probabilistic/statistical models is that it might make a robotic operation more likely to succeed, but it still doesn't provide a way of guaranteeing success (providing the parts involved will allow it, of course). Moreover, there is the problem of estimating the values of the various statistical parameters involved, such as the mean, variance and even the type of the probability distribution assumed.

Until recently it wasn't clear whether anyone had been successful in actually implementing an admittance control strategy. Peshkin had been unsuccessful due to "noise and bandwidth problems" [Peshkin 94]. However, there is film included in the 1995 IEEE Robotics and Automation Video proceedings that shows an Adept robot implementing the admittance matrices of Peshkin and Schimmels whilst running an impedance

controller based on Hogan's work [Mathewson & Newman 95]. Unfortunately, there appears to be no corresponding paper in the printed proceedings documenting the system. The video does, however, demonstrate that the implementation can be unstable for poor choice of gains in the feedback loops. This is an issue we consider further in Chapter 5.

All the above described work, and much more besides, has been aimed at improving the performance of robot manipulators by trying to cope in one way or another with spatial uncertainty, and its consequences. In the next chapter we will consider a model of assembly tasks that circumvents this problem.

Chapter 3

Modelling Rigid-Body Assembly Tasks

3.1 Programming in RAPT

In section 1.2 we described different levels of assembly robot programming. Probably the highest level language actually to be implemented is RAPT [Popplestone *et al.* 78], [Popplestone & Ambler 83]. The basic idea behind RAPT is to raise the level of programming by requiring that assemblies be specified in terms of the spatial relationships that hold between objects. The user needs to give the system a geometric description of both the objects involved in a task and the gripper to be used. The RAPT language primitives that can be used to describe how these objects relate to each other are AGAINST, FITS, ALIGNED and COPLANAR. A geometric inference engine translates a change in the specification of spatial relationships that holds between the objects into how the parts need to move in order to achieve this change.

In RAPT, changes in the dimensions of objects involved in some task are conveyed to the system by changing the part's geometric model. This means that we can have two instances of a task, involving parts of different sizes (but equivalent under scaling), where the programs to mate them are essentially the same in as much as the sequence of motions involved are the same even though the trajectories executed may be different. Any changes in the trajectory of the robot needed to accommodate a change in part size between instances of the task are calculated by the geometric inference engine. As far as the programmer is concerned the task is the same, *e.g.* a peg in hole task. In

addition, moving an assembly operation from one location to another requires minimal changes to a RAPT program. Say that the hole that some peg is to be inserted into is reoriented so that the central axis is horizontal rather than vertical, by conveying this fact to RAPT the motions of the peg that the robot executes are rotated to compensate. This of course does not address what is the best strategy to reliably achieve some task, but rather makes the (reasonable) assumption that if a strategy works for one set of parts of one size then it will work for another set of some other size that have an equivalent geometry under scaling.

RAPT turns assembly tasks described in terms of spatial relationships into commands for a position controlled robot. Consequently, the problems of spatial uncertainty, and questions about how robust a particular motion strategy is, still dominate. In addition, originally there was no provision for using sensory information. An attempt was made to change this with the vision system described in [Yin 84], but this was envisaged as a way of being more precise about RAPT's geometric world model.

However, the idea of abstracting away from position information to spatial relationships is an appealing one which we would like to preserve. Allowing assembly tasks to be specified in relational terms potentially makes the user interface much simpler, and permits the construction of more general programs, applicable to all instances of a task regardless of the dimensions of the parts involved, or where in the workspace the task is to be performed.

What we will attempt to do in this chapter is build upon the object-level description of a task employed by RAPT in order to generate a formalism suitable for a task-level assembly language. As we investigate the ramifications of this representation, we will find, in the chapters that follow, a need to discard the traditional manipulator and joint programming levels that RAPT subsumed (see section 1.2) and replace them with a more appropriate method of control. The first step in this process is to consider what kind of representation would be desirable for a system intended to operate at task level. We will develop this representation by analogy with a representation used in computer vision called the aspect graph.

3.2 The Aspect Graph

The "visual potential" or "aspect graph" of an object [Koenderink & Van Doorn 77], [Koenderink & Van Doorn 79] describes how singularities in the visual field of an observer come and go as the observer changes their location relative to the object. It is derived from optic flow [Koenderink 86] and differential geometry considerations, and concisely represents the experiences available to an observer as they traverse a path around an object. As an example the aspect graph of a cube[1] is given in figure 3.1.

The nodes in an aspect graph are the aspects: views of the object which are *qualitatively* the same. To see what this means imagine that you are looking at the cube in figure 3.1 with the vertex between faces A, B and C pointing towards you, and with the areas that you see of A, B and C roughly the same. If you were to move by 'small' amounts in any direction you would still see faces A, B and C, *i.e.* what you see is qualitatively the same despite small disturbances to your viewpoint. This would be true for most vantage points for some given combination of visible faces on an object.

The arcs of the graph represent the transitions between aspects. If we want to, for instance, go from viewing faces A and C to viewing faces B and C we can't do it without going through some other aspect or sequence of other aspects first (assuming no occlusion of the object *en route*). In other words it encodes the physically possible transitions available to an observer given their current relationship to the object.

This transition from one aspect to another is what Koenderink and Van Doorn call a catastrophic event. These events are associated with a discontinuity in the sensory field (actually the range of the apparent contour). For the cube example this corresponds to the observer's image of the object either losing or gaining a face. These catastrophic events occur due to relative motion between the observer and the object.

The features of an aspect graph that are interesting from our point of view are:-

- the representation is invariant to changes in the scale of the object; the aspect graph of the cube applies to all cubes regardless of their dimensions.

[1] An aspect graph can be derived for any object—it does not need to be polyhedral, but the aspect graph of a polyhedral object is most useful for our current pedagogical purposes.

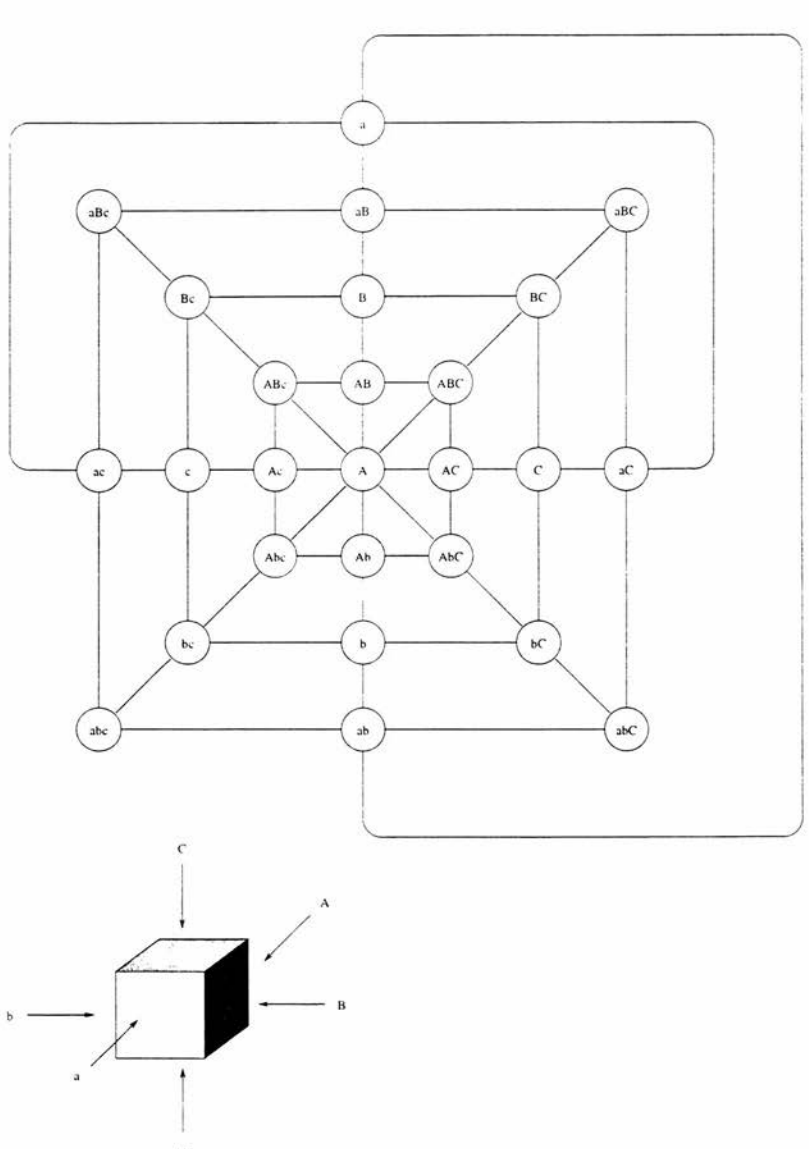


Figure 3.1: The aspect graph of a cube. (After [Koenderink & Van Doorn 77]).

- the representation is a relational one because the graph is derived from the relationship between the observer and the observed object, *i.e.* the definition of the aspect graph is independent of 'absolute' or 'world' coordinates.

These properties, we conjecture, are the properties both necessary and sufficient for the realisation of a task-level description of an assembly task. They enable the description of an assembly task to be made at an appropriate level of abstraction, since all the information necessary to specify a task in qualitative terms is retained, whilst instance specific, metric detail is ignored. Task-level assembly should, therefore be specified by a representation analogous to the aspect graph. That is to say, what we require is a representation that has the following properties:–

- the representation is composed of characteristic, or canonical, states that are inherent properties of a task
- the possible transitions between these characteristic states are the same for all instances of a task

The relational descriptions of RAPT almost achieve these things; they abstract away from metric details of the objects involved, and are relational in nature. However, these advantages are lost in the implementation. The relational descriptions are in fact just substitutes for numerical descriptions which are geometric transformations. It is our opinion that a more qualitative approach is needed. In the next section we will derive a qualitative description of assembly tasks that is analogous to the aspect graph.

3.3 Configuration Space Obstacles and Contact Configuration Graphs

A formalism that has been used in robotics for the purposes of path planning for some years is that of configuration space (see Chapter 2). Configuration space (CS) is spanned by a set of parameters that uniquely specify an object's position and orientation with respect to some (arbitrary) frame of reference. For spatial problems this requires six parameters: three to specify position, and three to specify orientation. In this thesis we restrict ourselves to working on planar problems for reasons explained in

Chapter 5 and Appendix D. Consequently we only require three variables to uniquely specify the location of an object: two for position, and a third for orientation.

The usefulness of CS for path planning arises from the fact that the problem can be transformed from trying to work out how an object with extent can be moved through an environment that includes obstacles, to the motion of a point in a higher dimensional space where the environmental obstacles have been 'grown' by an amount that takes into account the 'shrinkage' of the moving object to a point. An obstacle grown in this way is known as a configuration space obstacle (CSO).

A procedure for generating these obstacles is as follows: A fixed world coordinate system is chosen, as is a reference point which maintains a constant relationship to the moving object. The moving object is then, for a given orientation, swept about the stationary object. The locus of the reference point gives the surface of the configuration space obstacle for that orientation, often referred to as a slice of the CSO. The union of all of the loci generated for all relative orientations between the objects gives the CSO. Figure 3.2 shows this process for the part of the CSO that corresponds to placing a block in a corner. We choose the centre of the block as the reference point. Figure 3.3 shows the portion of the configuration space obstacle of interest to us. The x dimension corresponds to the horizontal part of the corner constraint, and the y dimension to the vertical. The full CSO would include the result of sweeping the block around the 'back' of the corner. For examples of the kind of results generated under these conditions see [Koutsou 86], [Brost 89], [Brost 91a], [Caine 93]. For our purposes it is sufficient to consider only the portion of the CSO that represents the contact configurations likely to occur.

The facets of the CSO correspond to a vertex-edge contact between the objects involved. This is because maintaining this kind of contact removes one of the degrees of freedom available to the moving object. The remaining two degrees of freedom allow a surface in CS to be swept out. 'Creases' in the CSO correspond to situations where two vertex-edge contacts hold. (The edge may be the same at both contacts.) In this case there is only one remaining freedom giving rise to a contour in CS. A vertex on the CSO corresponds to three simultaneous contact constraints leaving no possibility of motion between the two objects (in our planar domain).

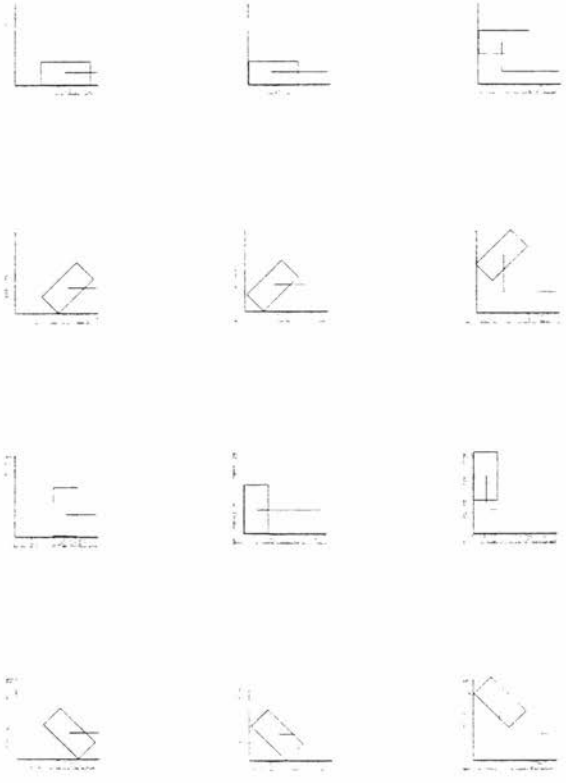


Figure 3.2: Constructing a configuration space obstacle.

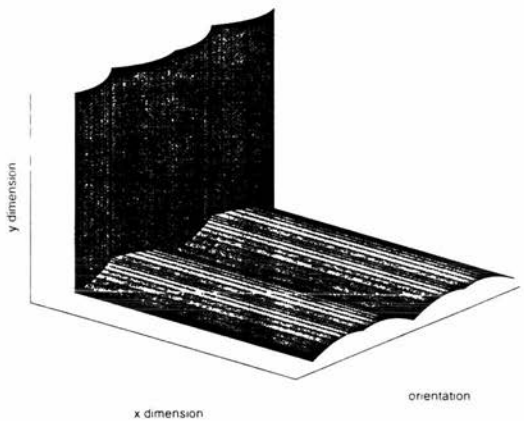


Figure 3.3: Part of the configuration space obstacle for a block and a corner.

A CSO is a part of the subspace of CS known as contact space: this is just the part of CS for which the moving object is in contact with another object without penetrating it. The volume inside the CSO represents situations where one object would be penetrating the other. The volume exterior to the CSO represents situations where the objects are not in contact. For the purposes of path planning this is the part of CS of most interest: the usually adopted technique is to grow the CSO by a little more than necessary in order to allow for geometric uncertainty, and then plan a trajectory for the reference point of the moving obstacle that avoids all the grown CSOs.

The property of a CSO that is of most interest to us is the fact that the surface of the CSO represents all the situations where the object boundaries are touching. Each facet, crease and vertex of a CSO represents a distinct contact configuration between the objects. Moreover, the adjacency of these CSO features encodes the physically feasible transitions between distinct contact configurations. This allows us to construct from the CSO a directed graph representation of all the possible contact configurations and the physically feasible transitions, where contact configurations are the nodes and the transitions the arcs. We will call this directed graph a contact configuration graph

(CCG). This graph is the representation analogous to the aspect graph that we require to be able to realise task level descriptions of assembly tasks.

Figure 3.4 shows some of the possible distinct contact configurations between a block and a corner constraint. If we allowed the block to rotate through 360°, then due to the symmetry of the block the icons representing the contact configurations would repeat, *i.e.* the next row at the bottom of the figure would look like the row below *C*0. In fact, the whole of the figure below *C*0 repeats and then wraps around, *i.e.* the configurations at the bottom of the page could transition to those at the top of the page. This is consistent with the fact that the CSO wraps around along the orientation axis.

The state represented by *C*0 is a special case, and doesn't actually form part of the CSO (because no contact is involved), but it is included here because it is a situation that might occur during any practical attempt to put a block in a corner so we will find it useful to refer to. Bidirectional transitions exist between it and all the other contact configurations.

Note that although the dimensions of the objects used in the construction of a CSO may vary, providing the shape of the objects remain constant, then the topology of the CSO will remain the same[2]. To see this consider that for polygonal objects, it is the point, line and (in the general case) planar contacts that create the various features that appear on the CSO. Note also that RAPT descriptions of assembly task states are implicitly couched in these terms (*cf* [Arai 85], [Koutsou 86]). Although it is easier to explain the CCG for polygonal objects, its applicability is not limited to polygonal objects.

Where the CSO encodes both metric and topological information about the contacts between objects, the CCG encodes only the topological information. In other words the graph structure is invariant to the dimensions of the parts involved, and where in the global frame of reference they occur, just like the aspect graph. Consequently, what we have here is a representation common to all instances of an assembly task.

[2] There are some caveats on this, for example, the width of a peg in a peg in hole task, being scaled such that it no longer fits the hole. We discuss this further in section 3.5.

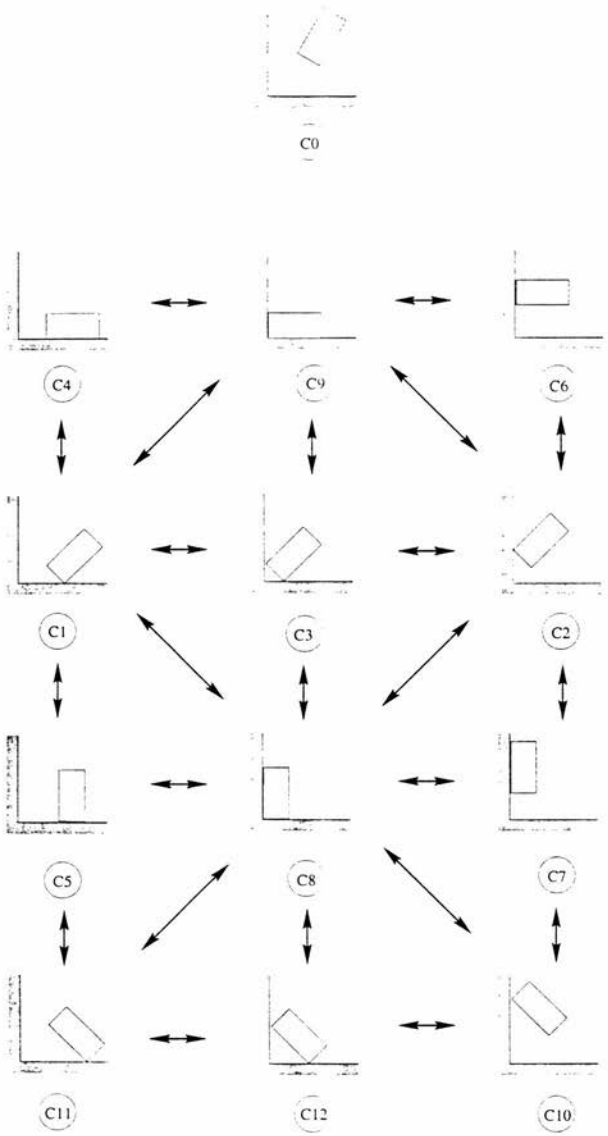


Figure 3.4: Some of the distinct contact configurations of the block in corner task and the physically feasible transitions between them.

3.4 Utilising Qualitative/State-Based Representations of a Task

What the CCG encodes is a kind of topological map of the assembly task. Because it shows all the physically feasible transitions between contact configurations it could be used to generate a nominal sequence of transitions through which the parts are to go in order to arrive at a goal configuration. In fact, because it is a map it is potentially far more useful than that, as the following analogy illustrates.

Consider that someone is coming to visit you who has never been to your home town before and you have given them explicit instructions to follow from the train station in order to reach your abode. Providing they adhere to those instructions they should arrive safely. If for some reason they stray from the prescribed path, perhaps because your directions were ambiguous, then they are completely lost unless they can get back to your original route and recognise it as being such. As an alternative to giving your visitor a prescribed route you could give them a map. Now if they stray from any prescribed route it doesn't really matter, providing they can identify where on the map they are, they know where they're going, and are able to implement a course from their current location to the desired one. In other words there is no need for them to ever be lost.

The situation where the visitor is only given one prescribed route is analogous to the normal state of affairs in robot programming where a sequence of prescribed actions intended to achieve a task are encoded in the assembly program. Error recovery can be interpreted as trying to get a system back onto this prescribed route once the fact that the system has strayed from it has been detected. If we were able to provide an assembly system with a map of a task we could effectively rule out the need for this kind of error recovery.

The exploitation of this representation as an assembly task map assumes that we are able to maintain some desired contact configuration whilst progressing towards the next desired one, and the ability to identify a transition from this configuration. As an example, if we were attempting to perform the task of placing a block into a corner, and we were in configuration $C1$ say, could we maintain this relationship whilst approaching

configuration $C3$? Additionally, could we detect the transition from this relationship and identify which of the possible transitions had occurred? In Chapter 4 we consider this issue in more detail. First, though, it is instructive to consider the relationship of both the aspect graph and the contact configuration graph to the synthesis of what might be called state appropriate actions.

Koenderink and Van Doorn derived the aspect graph as a minimal representation for the recognition of objects, from consideration of what an observer would perceive as they moved relative to an object. But can we invert this situation? That is, given an aspect graph, can we use it to navigate around an object? Similarly, if we were given the CCG of an assembly task would this be sufficient information on its own to navigate the task? The answer to this is probably 'No'; at least, not without some further information.

Consider the planar object shown in figure 3.5. By drawing the tangents to its faces we divide up the area surrounding the object into areas that can be labelled according to the faces that a point observer would see anywhere in that area. In other words the cells in this cell decomposition of the space surrounding the object correspond to nodes of the aspect graph[3]. Crossing one of the extended tangent lines corresponds to a node transition. If the observer has a geometric model of the object and their relationship to it they could plan motions that achieved particular crossings of the extended tangent lines, thereby effectively navigating around the object in terms of aspect graph transitions.

But if the observer only had the aspect graph model of the situation, that is, they had the description of the states that could be seen and the transition possible between them, and nothing else, how could they navigate the task? How would they know which way to go to achieve a particular transition? Since they would have no knowledge of the size of the object and the distance between them and it, they would have to navigate this object in a scale-invariant manner. We would suggest that they could do it by monitoring the way that things change as they move. The 'things' that we have in mind here are the magnitudes of the features which define an aspect, in our case the faces

[3] Note that the metric information encoded in the cell decomposition means that it bears a similar relationship to the aspect graph as the CSO does to the CCG.

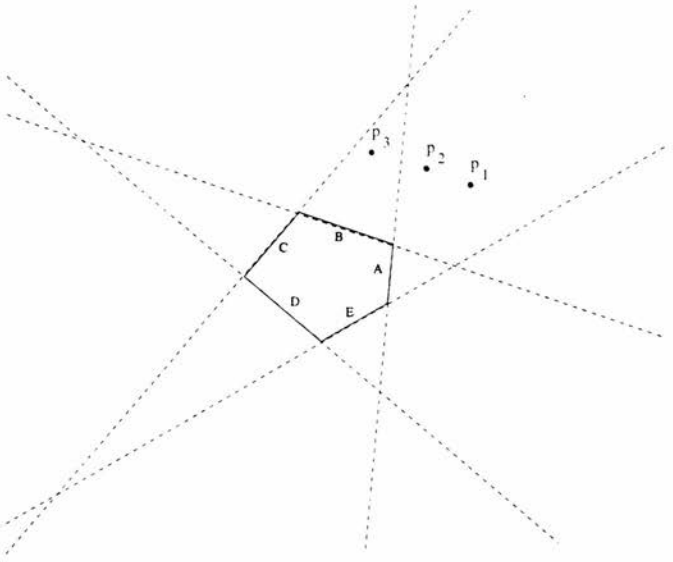


Figure 3.5: Cell decomposition of space surrounding an object on the basis of aspect graph nodes.

that they see at any point. With reference to figure 3.5 as the observer moves from p_1 to p_2 the size of face A seen diminishes[4], whilst the size of face B seen increases[5], until, as the observer crosses face A's extended tangent line, they lose sight of face A. From p_3 only face B can be seen.

For the visual case it is possible to gain knowledge about the relationship of the observer to the object by trying some motion and seeing what the consequence is. Then, after deriving the general direction of motion needed to effect a desired transition, they could just execute a "move until discontinuity sensed" strategy[6].

[4] The size of the angles subtended by the dotted lines at the point observer are proportional to the areas of the faces seen.

[5] In fact the relative rate at which these changes occur gives further information about the relative location of the observer to the object.

[6] Since the way that the occluding contour varies is part of the derivation of the aspect graph of an object, all we have done here is point out that the information necessary to infer the topological properties that define the aspect graph can be exploited in order to use an aspect graph as a navigational aid.

For the assembly task case it is not so clear that it is possible to try something exploratory in order to deduce the right way in which to act, particularly since we will find it necessary to specify actions in terms of changes in the the force and torque that should be applied, so that reversing of an action need not take you back to where you started should the exploratory action result in an undesired consequence[7]. In the assembly strategies that we deduce in Chapter 4 we assume that we know at the start that the relationship between the objects is one of a finite number of possible contact configurations. We then implement force application strategies that perform successive state transitions where each resulting state in the sequence is also composed of a finite set of possible contact configurations. This is conceptually similar to the work described in [Erdmann & Mason 88]. We can't navigate an assembly task in term of what current contact configuration we are actually in because we don't have a 'contact configuration sensor' that returns a value that is the current contact configuration[8]. We can however use the contact configuration graph as a framework to generate task-invariant assembly strategies. We will show that it is possible, given that we know the set of contact configurations that we are possibly in, to generate "act until discontinuity sensed" actions designed to take us progressively towards our goal. It should be clear, but is probably worth mentioning all the same, that the action associated with a particular state is not an intrinsic property of that state, but dependent on the context of the task in hand.

3.5 When is a Task not a Task?

In Chapter 1, when referring to our assembly strategies, we defined "task-invariant" as meaning "constant over all instances of a task". Up until this point we have been relying on the reader's intuitions concerning the definition of a task. But the question arises as to what do we mean by a task, or equivalently, under what conditions does task X stop being task X and become task Y? For instance, when does a peg-in-hole assembly task [Bland 86] become a block-in-detent assembly task [Peshkin 90] (otherwise known as a disc-in-hole task [Byrne 89])? In [Lozano-Pérez *et al.* 84] a

[7] This would work if static equilibrium applied throughout the 'exploratory action'.

[8] Although it might be possible, given our experimental set-up, to mount an overhead camera to do just this.

number of variations on the peg-in-hole task are shown that can be considered to be due to 'small' changes in geometry, and yet according to Lozano-Pérez *et al.* they "require substantially different programs to ensure reliable execution".

There has been some work surveying the kind of assembly tasks that are commonly performed, although maybe not as much as one might expect [Byrne 87]. Nevins and Whitney [Nevins & Whitney 78] list twelve common assembly operations, but some of these we would claim are not assembly tasks *per se*, *e.g.* "flip a part over", "provide temporary support", some include material deformation, *e.g.* "crimp sheet metal", and one is in fact a disassembly operation, *viz.* "remove locating pin". The classification given in [Swift 80] is performed with a view to applying design for assembly techniques [Boothroyd 80] and so, for instance, types of peg in hole insertions are distinguished by (amongst other things) the approach direction of the peg. The work that is probably most comprehensive and best suited to robotic assembly is that by Byrne *et al.* [Byrne 88], [Byrne 89]. In this work initially classification was performed on the basis of the geometry of the surfaces of a part that were directly involved in the mating process. A distinction was made between "primary features" and "secondary features". In essence the primary features determine the 'gross geometry' of objects, and secondary features are modifications to these, *e.g.* inclusion of a chamfer [Hopkins *et al.* 88]. The reason for this was the expectation that generic assembly strategies could be devised based on the primary features, that could be modified in some minor way to accommodate secondary features. Subsequent work included task kinematics as part of the classification criteria, *i.e.* which degrees of freedom are constrained as the task progresses. So, for example, a dual peg in hole might be considered as the same class of task as a square peg in hole [Byrne 89], [Byrne & Hopkins 91].

The inescapable conclusion of comparing these works is that there is no agreed formal definition of what constitutes a task, so we will attempt our own definition.

First we will specify that we are prepared to allow that not all features of an object will be relevant to the definition of a task: for a peg in hole task the shape of the underside of the part containing the hole we will consider to be irrelevant. Similarly, for a peg in hole task, we want to say that the fact that an object has more than one hole in it is also irrelevant; we will assume that the hole of interest can be discerned

by some means. The point is that there are invariably regions in close proximity to the parts of the objects that are used to define the goal configuration, that contain all of the features relevant to the successful completion of a task. On the CSO these features will correspond to a finite number of contiguous features in the neighbourhood of the feature that constitutes the goal configuration. We will only be concerned with the contact configurations that correspond to this neighbourhood. This convention is implicit in most work on robotic assembly, including the RAPT robot programming language (see section 3.1), is made explicit in the work of [Byrne 89], and is one we have adopted already by only showing part of the CSO for the block in corner task in figure 3.3. In what follows we will continue to adopt this convention.

For a given geometry between the robot held object and the object that it is to be mated to, the CSO represents all the possible contact configurations that can occur between them. This information is encoded in the CCG. If both objects are scaled uniformly this change in size will be reflected by a scaling of the x and y dimensions of the CSO, but the orientation axis will be unaffected. The CCG will remain the same because the possible contact configurations and the transitions between them will not have changed. The CCG is also independent of the location, *i.e.* the position and orientation at which the assembly process is to occur for the same reason.

The CCG will also be constant for some tasks when non-uniform scaling of the objects is allowed, but this will not be the case in general. However, for many tasks the CCG will be constant for non-uniform scaling over some finite interval. There is also the question of whether one of the objects involved in an assembly can be scaled independently of the other object. For a peg in hole situation the CCG will be the same despite changes in the width of the peg until such time as the peg's width is exactly the same as that of the hole. At this point the CSO features that correspond to the peg touching one side of the hole but not the other disappear. So our first requirement for determining whether two sets of objects are the same assembly task is that their CCGs match, or equivalently that both match a 'prototype' CCG for the task in question.

Our second requirement is that the definition of the goal configuration, in terms of contact configuration, be the same. It is quite feasible that two sets of objects that have the same CCG are required to be in different final configurations. This we want to

say constitutes a different task because the end objective is different. In the assembly strategies that follow in subsequent chapters we also assume that we can specify the initial or starting conditions. This is a matter of practicality since it avoids having to recognise the starting conditions and, though necessary to the success of a strategy, need not constitute part of a task's definition.

It may be possible to devise cases where small changes in geometry will have effects that do not alter the CCG, but that confound an otherwise robust assembly strategy. In which case we reserve the ability to modify our definition of a task so that isomorphism of CCG is a necessary but not sufficient condition for judging what task a set of objects belong to[9]. We will not know whether this modification to our definition is necessary, nor what form it would take, until further work has been performed that reveals its nature.

On the other hand, it is possible that some small changes in geometry may create or destroy one or more nodes in a CCG, whereas in practical terms there is little or no effect on the strategy needed to achieve a successful assembly. Similar effects happen with the (visual) aspect graph, where some small changes in geometry can essentially be ignored for the purposes of recognition. Such effects are the natural subject matter of empirical investigations [Fitzgibbon & Fisher 92].

This may sound as though we are avoiding the central issue since we are effectively saying that we want our definition of a task to be whatever it is that enables us to deliver a task-invariant assembly strategy. However, in lieu of any other agreed formal definition of a task, this is at least as good as any other, and has the additional benefit of potentially simplifying the programming interface to a robotic assembly system. Moreover we do not believe this will lead us into *ad hoc* and unprincipled definitions.

This definition, however, says nothing about how to actually select an assembly strategy that works over the range of possible instantiations of a task. This we consider in detail for some example tasks in the next chapter where we will see that the factors

[9] The kind of change that we have in mind here is a change in two of the interior angles of a peg in a peg in hole task along with a concomitant matching change in the geometry of the hole. This may or may not have serious consequences for the ability to derive an assembly strategy that is constant over all instances of a task. It may be that there is a range of variation in angle where this change has no effect, the bounding value being a function of the coefficient of friction.

involved in being able to do this are the magnitude of the coefficient of friction, the location of the centre of compliance, and how the components of the applied wrench are varied. Usually the value of the coefficient of friction is outwith our control and not accurately known anyway, so our objective is to implement a transition via choice of the location of the centre of compliance and relative magnitudes of the components of the applied wrench, despite possible variation in metric details of object geometry and the magnitude of the coefficient of friction.

Due to possible ambiguities that follow from the mechanics of contact we will have to work in terms of states that are defined by the set of contact configurations that a task could possibly be in at some point in its progress. If we can perform an analysis (possibly state by state) that shows us how to implement state transitions regardless of where in the range of possibilities consistent with the state definition that the objects lie, then we have abstracted away from the domain of metric details to that of contact configuration, or topology of contact. If each state transition works in these terms, then the strategy as a whole is in these terms, *i.e.* it works in terms of a description that is constant for all instances of a task: it is task-invariant.

As a corollary, if it were the case that the state transitions could be shown to 'converge' on the goal configuration from the initial configuration, then since these states are in terms of the defining, invariant characteristics of our task *i.e.* the contact configurations, the ensuing behaviour will be task-achieving. One way of realising convergent behaviour is to disallow cycles in the state transition graph. We have not concentrated on this issue in this thesis.

3.6 Discussion

In this chapter we have attempted to raise the level at which assembly tasks are described to that of a state-based representation. The basic hypothesis of this thesis, that we intend to explore in the following chapters, is that if we can manage to realise assembly programs at this level of abstraction then we ought to be able to escape from the spatial uncertainty trap, because we are no longer operating in spatial terms.

A number of other authors have made similar observations about the nature of an

assembly task being expressible as transitions between distinct contact configurations, for example [Arai & Kinoshita 81], [Desai & Volz 89], [Asada 90], [Peshkin 90], [Asada & Hirai 90], [McCarragher & Asada 93], [McCarragher 93], [Hirai 94]. It is also implicit in the work of [Whitney 82], [Caine *et al.* 89], [Laugier 89], [Brost 91a] and [Caine 93], amongst others.

The work of Hirai and Asada [Asada & Hirai 90], [Hirai 94], and McCarragher and Asada [McCarragher & Asada 93], [McCarragher 93], is particularly interesting, because they include attempts to recognise contact states. Hirai's work uses polyhedral convex cones [Goldman & Tucker 56], and a geometric model of the objects involved to discriminate contact configurations on the basis of the admissible forces. However the technique seems best suited to static situations since frictional effects, which cannot be avoided during motion, would increase the size of the polyhedral convex cones making some discriminations ambiguous. McCarragher's work uses qualitative models of the dynamics of contact transitions to determine which transition has occurred, though this need not provide unique discriminations of the 'exit conditions' of a state. It also uses a Petri net formalism that is able to synthesise automatically desired state transitions.

Chapter 4

Motion and Contact

If we are to be able to exploit the CCG of the previous chapter in order to arrive at task-invariant assembly strategies we require the ability to perform an assembly in terms of the making, maintaining and breaking of contact configurations. With this in mind it is instructive to consider the behaviour of a standard position controlled industrial manipulator during contact.

4.1 The Behaviour of Position Controlled Manipulators During Tasks Involving Contact

Consider trying to use a position-controlled manipulator to place an object on a table top. This sounds straightforward, however, anyone who has ever tried to do anything with a robot will know that this is not quite as easy as it sounds, and actually turns out to be a time-consuming and error-prone operation. The problem is that if the location is specified too low, the object the manipulator is holding will be pressed down in to the table top onto which it is meant to placed, with a force that could result in damage either to the objects involved, or the manipulator, or both; if the location is specified too high, when the part is let go it may fall, and the effects of the impact with the table top could either cause damage or cause the part to bounce into a location that is not consistent with the requirements of any further operations to be performed on it.

Now let us look at the problem of maintaining some desired contact in order to head 'towards' the next one in a desired sequence. Consider trying to get a position-controlled manipulator to follow the top of a table. Imagine that the trajectory that is specified is

a fraction of a millimetre too low: the end result is that the manipulator will attempt to make a furrow in the table top. If, on the other hand, the trajectory is specified a fraction of a millimetre too high, then the task is no longer being executed as specified. This assumes of course that you are able to specify a trajectory that runs parallel to the table top. It is just as likely, even if the manipulator starts off skimming the table surface, that as it executes the commanded motion, it either progressively buries itself into the table or progressively distances itself from it. Note that even if the manipulator exhibits perfect position control these situations can still occur because very few tables are perfectly flat.

It is considerations such as the above that lead us to say that it is the fact that standard industrial manipulators are designed to be position controlled that leads to the problem of spatial uncertainty.

The potential for high forces between the manipulator and the environment occurs because the robot is controlled by servo loops, and servos are designed to achieve their goal despite "disturbances"[1]. In the examples above if the robot's commanded motion is too low the table top will displace the position servos from their commanded set-points. The result will be that the servos treat this as a disturbance and exert a force to overcome it, which in this situation will be into the table. Because position servos tend to be designed with as high a gain as possible, specifically for the purpose of rejecting disturbances, the forces generated due to even a small displacement from the specified position can be sufficient to cause damage to the objects involved. These high gains are useful if one is attempting to get a manipulator to closely follow a specified path in free space, but they seriously limit the utility of industrial manipulators when in contact with the environment. At least one manufacturer realises this limitation and provides the option of reducing the position servo gains of their manipulators to ameliorate these effects [Cooper 87].

Although reducing the gains of a manipulator's position servos may improve the situation for simple situations such as placing an object on a surface, assemblies of objects create further complications. To a first approximation a position servo makes the end-point of a robot look as though it is held in place by a collection of springs in

[1] In Chapter 5 we highlight a related design decision that compounds this problem.

equilibrium. If the manipulator is displaced from the position that corresponds to the set points of its servos, the restoring force is proportional to the position servo gains *and* the amount of displacement. What this means is that if we are, say, trying to place a peg into a hole and the axis of the peg and hole are misaligned such that the robot is displaced from its commanded trajectory, the robot exerts a force on the part containing the hole that is proportional to the misalignment. The problem as far as successful assembly is concerned is that the reaction forces from the part containing the hole are likely to act in such a manner as to make the misalignment worse [Arai & Kinoshita 81], [Whitney 82]. Some researchers have taken this as implying that what is really required is a manipulation device that can use the forces that it experiences during contact to modify its trajectory in an error corrective fashion, as we discussed in section 2.2.3.

4.2 The Benefits of Being Able to Slide

First we would note that sliding is a useful strategy during the assembly of two objects, because it circumvents the spatial uncertainty problem. To see this consider the motion of a robot-held object sliding over the surface of another object fixed in the environment. By allowing the robot-held object to be guided by the surfaces of the object that is part of the environment, we do not need to know exactly where this environmental object is. By sliding one object over another we are in effect working in terms of the *relative* location of the two parts. This is a distinct advantage to us because assembly is just about achieving desired relative object locations.

In addition, if we are working in terms of contact configurations, then knowing that a desired contact has been maintained is sufficient knowledge to enable progress towards the desired final assembly configuration. We would note that the majority of assemblies are designed such that the parts involved need to be arranged in a manner where various faces come into contact. In fact, they may well be designed in a manner that is intended to guide the relative motion of the parts to the desired goal configuration [Boothroyd 80]. These facts work in our favour if we are attempting to exploit sliding behaviour. Indeed Koutsou [Koutsou 86] developed an automated planner that took this on board and attempted to deduce motions that would incrementally add kine-

matic constraints between the objects involved until such time as the goal configuration was reached.

The important point is that by organising assembly behaviour in terms of sliding motions we are able to operate in terms of the making, maintaining, or breaking of point, line and face contacts, *i.e.* in the terms that we decided are the desirable properties of the RAPT programming language and the CCG, because of their invariance to part size and task location.

4.3 Friction Cones

The CCG and CSO introduced in Chapter 3 are representations that encode purely kinematic facts: they express qualitative details about the relative locations between the objects. If we require the ability to slide one object over another in order to maintain, make or break a contact we will need to consider the nature of the mechanical interaction between the objects. There has been a significant amount of work in recent years by researchers who have been interested in predicting the behaviour of interacting objects in order to generate plans that achieve manipulation tasks reliably. Much of this work has been quasi-static in nature, including the geometric modelling of the effects of friction. The following sections provide a short overview of some of these models with a view to exploiting them in the derivation of task-invariant assembly strategies. We avoid performing a complete dynamic analysis because the qualitative predictions of these models are sufficient for our purposes.

4.3.1 Some Notation

In what follows we will find it useful to be able to refer to forces and torques in the plane simultaneously, so we will adopt the terminology of screw theory [Ball 00], [Hunt 78], and work in terms of wrenches. A planar wrench, $\mathbf{w}$, is given by:

$$\mathbf{w} = \begin{pmatrix} f_x \\ f_y \\ \tau_z \end{pmatrix} \tag{4.1}$$

where f_x and f_y are the components of the force aligned with the x and y axes (respectively) of some coordinate system, and τ_z is the torque (or moment) about the

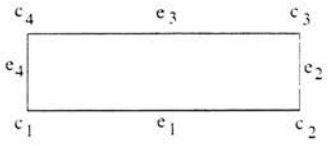


Figure 4.1: Labelling of a rectangular block.

corresponding z axis.

We will also find it useful to refer to features of a robot held object, in our case a rectangular block. Consequently we will use the labelling scheme shown in figure 4.1, where we pick a corner to label as c_1 and label the rest of the corners in order in an anticlockwise sense. Between each corner c_j and c_k lies edge e_j. Subsequent figures may exploit this ordering by labelling some reference corner or edge so that the reader may infer the rest of the labels.

When corner c_j of a block is in contact with an environmental constraint we will attach a frame of reference to the corner with the same label as the corner, and assign axes at the point of contact such that the positive y-axis is the environmental constraint surface's normal that points 'into' the block.

When edge e_j is in contact with a vertex of the environment we will attach a frame of reference to the contact point that is given the same label as the edge, and assign axes at the point of contact such that the positive y-axis is the block surface's normal that points 'into' the block.

We will also have recourse to refer to a frame of reference that is fixed with respect to the block. We will give this frame of reference the label w (not to be confused with the wrench $\mathbf{w}$). We will define ϕ_{c_j} (respectively ϕ_{e_j}) as the angle from the x-axis of frame w to the x-axis of frame c_j (resp. e_j) measured in an anticlockwise sense. Likewise we will define the angle θ_{c_j} (resp. θ_{e_j}) as the angle from the x-axis of frame c_j (resp. e_j) to the x-axis of frame w measured in an anticlockwise sense.

4.3.2 The Real Space Friction Cone

A useful place to start when considering the mechanics of contact is the affects of friction experienced by a particle in contact with a surface. Alternatively we could consider a body with extent in point contact with a plane, with the line of action of the applied force passing through the point of contact. We will assume that Coulomb's law of dry friction applies.

According to Coulomb's law, for a point sliding along a surface the tangential force generated due to friction, f_{f_t}, is related to the normal component of the applied force, f_{a_n}, by

$$f_{f_t} = \mu f_{a_n} \tag{4.2}$$

where μ is the coefficient of friction. Since the normal component of the force due to friction is equal and opposite to the normal of the applied force, *i.e.*

$$f_{f_n} = f_{a_n} \tag{4.3}$$

both components of the force due to friction are proportional to the normal component of the applied force. During sliding the ratio of the tangential component to the normal component is constant irrespective of the magnitude of the applied force, thus constraining the force due to friction to lie along a particular line. It can be seen from figure 4.2 that this line is at an angle ψ to the normal given by:

$$\tan(\psi) = \frac{f_{f_t}}{f_{f_n}} = \frac{\mu f_{a_n}}{f_{a_n}} = \mu \tag{4.4}$$

The set of vectors making this angle with the normal form the friction cone [Meriam 75].

If the applied force lies inside the friction cone then it is exactly balanced by an equal and opposite reaction force (unless, of course, so much force is applied that something breaks), and there is no relative motion between the objects. If the applied force lies outside of the friction cone, since the reactive force due to friction is constrained to lie along the edge of the friction cone, there is a net shear force, f_{net}, and relative motion occurs (see figure 4.2).

The angle ψ is the angle of friction. In fact there is an angle of static friction and an angle of kinetic friction [Meriam 75]. The angle of kinetic friction is defined during

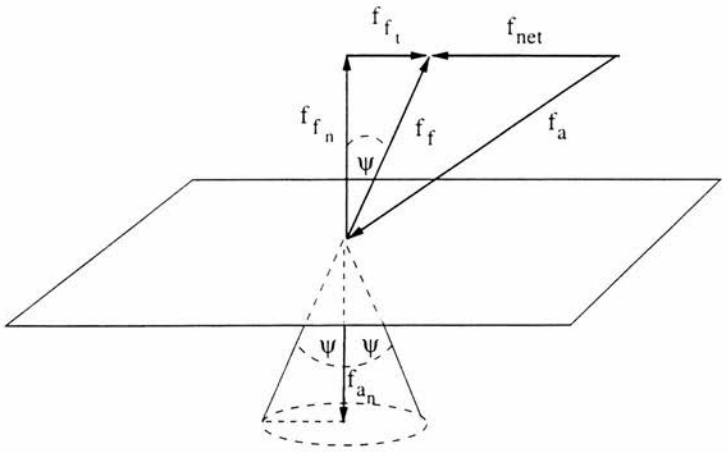


Figure 4.2: The real-space friction cone due to Coulomb friction.

sliding motion and is usually slightly less than the angle of static friction which is defined when there is no relative motion. There are also corresponding coefficients of kinetic and static friction. For our purposes we can ignore this distinction. It will become apparent that we employ techniques that are insensitive to these variations in the magnitude of the coefficient of friction: it is sufficient that we know that friction exists.

Although the friction cone in figure 4.2 is shown as a three dimensional cone, because the domain that we will be working in is planar, this three dimensional cone, for our purposes, can be projected into a plane orthogonal to the plane of contact of figure 4.2 resulting in an (open-ended) 'triangle'. Given this projection, the extent of the friction cone can be calculated as all non-negative linear combinations of the vectors representing the friction cone edges. Let us denote the left ray of the friction cone by the unit vector $\hat{\mathbf{r}}_l$ and the right ray of the friction cone by the unit vector $\hat{\mathbf{r}}_r$. For a point contact the friction cone will be centred about the y-axis of the contact reference frame (as defined in section 4.3.1) with a half-angle of ψ. Consequently we

can write the unit wrenches that represent the directions of the friction cone edges as

$$\hat{\mathbf{r}}_l = \begin{pmatrix} \sin(\psi) \\ \cos(\psi) \\ 0 \end{pmatrix} \tag{4.5}$$

$$\hat{\mathbf{r}}_r = \begin{pmatrix} -\sin(\psi) \\ \cos(\psi) \\ 0 \end{pmatrix} \tag{4.6}$$

and the planar real space friction cone can be represented by the open set [Nguyen 86]

$$W^{<} = \{\mathbf{w} \mid \mathbf{w} = k_l\hat{\mathbf{r}}_l + k_r\hat{\mathbf{r}}_r, k_l \geq 0, k_r \geq 0\} \tag{4.7}$$

where $W^{<}$ is known as a convex cone.

4.3.3 The Wrench Space Friction Cone

In [Erdmann 84], [Erdmann 94] Erdmann develops a generalisation of the real space friction cone described above. The need for this generalisation arises because any force applied to a body with extent, whose line of application does not pass through the point at which the body is in contact with the environment, will induce a torque on the body. Figure 4.3 illustrates this situation.

With reference to figure 4.3 if we have an applied force represented by the wrench ${}^{w}\mathbf{w}_a = ({}^{w}f_{a_x}\ {}^{w}f_{a_y}\ 0)^T$ in the frame of reference w, this appears at the point of contact as a wrench in frame of reference c_1. These wrenches are related by the geometry of the situation according to:

$$\begin{pmatrix} {}^{c_1}f_x \\ {}^{c_1}f_y \\ {}^{c_1}\tau_z \end{pmatrix} = \begin{pmatrix} \cos(\theta_{c_1}) & -\sin(\theta_{c_1}) & 0 \\ \sin(\theta_{c_1}) & \cos(\theta_{c_1}) & 0 \\ s_x\sin(\theta_{c_1}) - s_y\cos(\theta_{c_1}) & s_x\cos(\theta_{c_1}) + s_y\sin(\theta_{c_1}) & 1 \end{pmatrix} \begin{pmatrix} {}^{w}f_{a_x} \\ {}^{w}f_{a_y} \\ 0 \end{pmatrix} \tag{4.8}$$

which is an instantiation of equation C.7. However, since it is impossible to transmit a moment through a point contact this reduces to:

$$\begin{pmatrix} {}^{c_1}f_x \\ {}^{c_1}f_y \\ {}^{c_1}\tau_z \end{pmatrix} = \begin{pmatrix} \cos(\theta_{c_1}) & -\sin(\theta_{c_1}) & 0 \\ \sin(\theta_{c_1}) & \cos(\theta_{c_1}) & 0 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} {}^{w}f_{a_x} \\ {}^{w}f_{a_y} \\ 0 \end{pmatrix} \tag{4.9}$$

or

$${}^{c_1}\mathbf{w}_a = {}^{c_1}_{w}\mathbf{R}_z(\theta_{c_1})\,{}^{w}\mathbf{w}_a \tag{4.10}$$

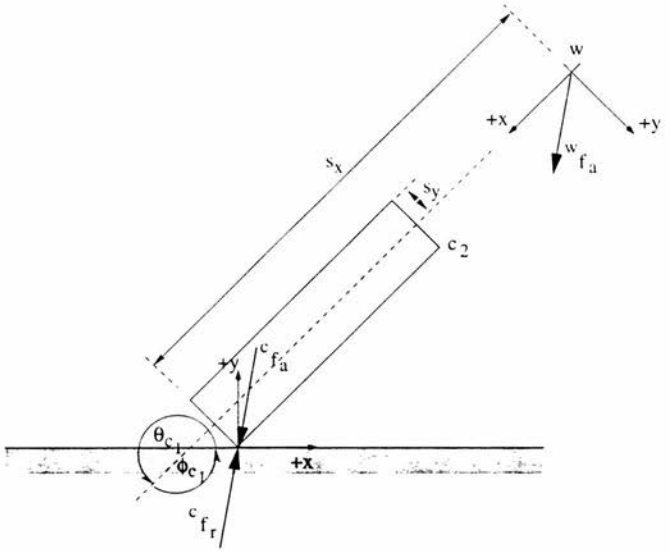


Figure 4.3: Diagram to aid in derivation of the wrench space friction cone.

where ${}^{c_1}_{w}\mathbf{R}_z(\theta_{c_1})$ is the rotation operator that maps frame w into frame c_1 (see equation C.6). Assuming this applied force points into the real space friction cone it results in an equal and opposite reaction force

$$ {}^{c_1}\mathbf{w}_r = -{}^{c_1}\mathbf{w}_a \tag{4.11} $$

which appears in the w frame of reference as ${}^{w}\mathbf{w}_r$ where

$$ {}^{w}\mathbf{w}_r = {}^{w}_{c_1}\mathbf{W}\,{}^{c_1}\mathbf{w}_r \tag{4.12} $$

and the operator that maps a wrench in frame c_1 to frame w is given by:

$$ {}^{w}_{c_1}\mathbf{W} = \begin{pmatrix} \cos(\phi_{c_1}) & -\sin(\phi_{c_1}) & 0 \\ \sin(\phi_{c_1}) & \cos(\phi_{c_1}) & 0 \\ s_x\sin(\phi_{c_1}) - s_y\cos(\phi_{c_1}) & s_x\cos(\phi_{c_1}) + s_y\sin(\phi_{c_1}) & 1 \end{pmatrix} \tag{4.13} $$

Working through these equations and noting that $\theta_{c_1} + \phi_{c_1} = 360^o$ illustrates how an applied force results in an induced torque given by

$$ {}^{w}\tau_{r_z} = s_y\,{}^{w}f_{a_x} - s_x\,{}^{w}f_{a_y} \tag{4.14} $$

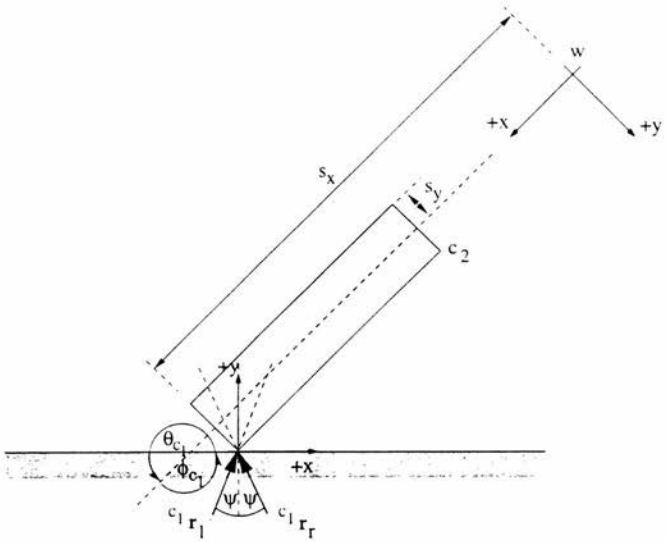


Figure 4.4: The friction cone rays of the contact shown in figure 4.3.

A result that is easily verified from inspection of figure 4.3.

Figure 4.4 shows the real space friction cone rays for the contact of figure 4.3 labelled as $^{c_1}\mathbf{r}_l$ and $^{c_1}\mathbf{r}_r$.

Since the friction cone rays represent the extremes in direction of applied force that the real space friction cone can balance, by transforming them into the frame of reference w we get the extremes of the wrenches applied in w that the contact can 'mirror'. Using the expressions for unit vectors aligned with the friction cone edges given by equations 4.5 and 4.6 and the transformation matrix of equation 4.13 we get the following convex cone edges in the wrench space associated with frame w

$$
{}^{w}_{c_1}\hat{\mathbf{r}}_r = \begin{pmatrix} \sin(-\psi - \phi_{c_1}) \\ \cos(-\psi - \phi_{c_1}) \\ \cos(\psi)[s_x \cos(\phi_{c_1}) + s_y \sin(\phi_{c_1})] - \sin(\psi)[s_x \sin(\phi_{c_1}) - s_y \cos(\phi_{c_1})] \end{pmatrix} \tag{4.15}
$$

$$
{}^{w}_{c_1}\hat{\mathbf{r}}_l = \begin{pmatrix} \sin(\psi - \phi_{c_1}) \\ \cos(\psi - \phi_{c_1}) \\ \cos(\psi)[s_x \cos(\phi_{c_1}) + s_y \sin(\phi_{c_1})] + \sin(\psi)[s_x \sin(\phi_{c_1}) - s_y \cos(\phi_{c_1})] \end{pmatrix} \tag{4.16}
$$

where we use ${}^{w}_{c_1}\hat{\mathbf{r}}_l$ to denote the (direction of the) 'left' real space friction cone ray of contact c_1 mapped into the wrench space associated with frame w. The expressions for the moment components of these wrenches can be simplified if we define the angle γ such that

$$
\sin(\gamma) = \frac{s_x}{s_r} \tag{4.17}
$$

$$
\cos(\gamma) = \frac{s_y}{s_r} \tag{4.18}
$$

The physical significance of these definitions for the contact configuration of figure 4.3 is shown in fig 4.5. From this definition of γ equations 4.15 and 4.16 become

$$
{}^{w}_{c_1}\hat{\mathbf{r}}_r = \begin{pmatrix} \sin(-\psi - \phi_{c_1}) \\ \cos(-\psi - \phi_{c_1}) \\ s_r \sin(\phi_{c_1} + \gamma + \psi) \end{pmatrix} \tag{4.19}
$$

$$
{}^{w}_{c_1}\hat{\mathbf{r}}_l = \begin{pmatrix} \sin(\psi - \phi_{c_1}) \\ \cos(\psi - \phi_{c_1}) \\ s_r \sin(\phi_{c_1} + \gamma - \psi) \end{pmatrix} \tag{4.20}
$$

As for the real space friction cone we can use equation 4.7 to evaluate the scope of the effects of friction; this time though, we use the wrenches of equations 4.19 and 4.20. We will call the resulting convex cone the wrench space friction cone. It is illustrated in figure 4.6.

In Erdmann's original formulation the moment dimension was scaled by the object's radius of gyration, ρ, giving an $(f_x, f_y, \tau_z/\rho)$ generalised force space. Because there is a direct correspondence between the contact normal expressed in this force space and the outward pointing normal in the corresponding $(x, y, \rho\theta)$ configuration space, Erdmann called the resulting friction cone the configuration space friction cone [Erdmann 84], [Erdmann 94]. A related representation appears in [Brost & Mason 90] and [Brost 91a].

Functionally the wrench space friction cone is equivalent to the real space friction cone: that is to say, any wrench in frame w pointing into the wrench space friction cone can be balanced by an environmental response. The application of a wrench that is pointing outside of the wrench space friction cone, but into the surface of contact,

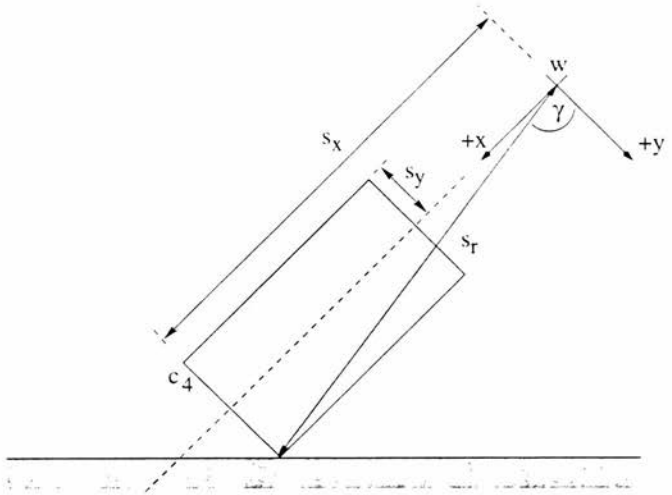


Figure 4.5: Definition of the angle γ for the contact configuration of figure 4.3.

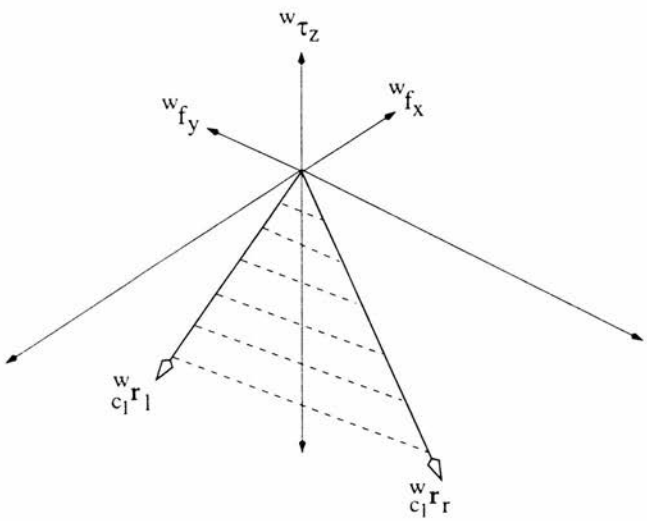


Figure 4.6: The friction cone rays of figure 4.4 mapped into wrench space.

can lead to sliding with a response force constrained to lie on the edge of the wrench space friction cone, that corresponds to the real space friction cone ray, that opposes the direction of motion.

We derived equations 4.19 and 4.20 from the direction of the edges of the real space friction cones. We could write the direction in wrench space of an arbitrary real space reaction force that makes an angle of κ with the contact normal at contact c_j as:

$$ {}^{w}_{c_j}\hat{\mathbf{w}}_r = \begin{pmatrix} \sin(\kappa - \phi_{c_j}) \\ \cos(\kappa - \phi_{c_j}) \\ s_r \sin(\phi_{c_j} + \gamma - \kappa) \end{pmatrix} \tag{4.21} $$

We would note that if the direction of an applied force is into the friction cone at point of contact c_j then

$$ \kappa \in [-\psi, \psi] \tag{4.22} $$

whereas if the direction is such that sliding occurs then

$$ \kappa \in \{-\psi, \psi\} \tag{4.23} $$

depending on the direction of motion.

We will find it useful in what follows to be able to refer to the moment component of equation 4.21, *viz.*

$$ {}^{w}\tau_z = s_r \sin(\phi_{c_j} + \gamma - \kappa) \tag{4.24} $$

since from it we can deduce the sign of the turning moment that a reaction force produces on a reference point.

4.3.4 Composite Friction Cones

When multiple contacts exist between objects the possible response forces that the contacts can exert are those that lie within the linear combination of the possible forces that each individual contact can balance [Erdmann 84], [Erdmann 94]. This is represented by the convex, or Minkowski, sum [Nguyen 86]

$$ W_1^{<} \oplus W_2^{<} = \{\mathbf{w} \mid \mathbf{w} = k_1\mathbf{w}_1 + k_2\mathbf{w}_2, k_1 \geq 0, k_2 \geq 0, \mathbf{w}_1 \in W_1^{<}, \mathbf{w}_2 \in W_2^{<}\} \tag{4.25} $$

resulting in a polyhedral convex cone [Goldman & Tucker 56], [Hirai & Asada 93] in wrench space.

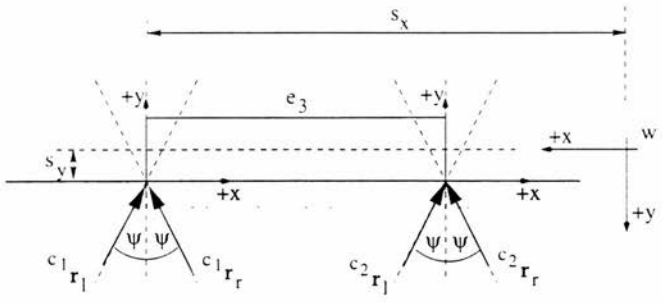


Figure 4.7: Planar contact between a block and a constraint surface.

As an example consider the situation shown in figure 4.7. In this face against face situation we take the friction cone rays at the two extreme points of contact and map these into wrench space, since we can then interpolate between them to take into account all the other possible point contacts along the contact surface. The resulting wrench space friction cone is shown in figure 4.8.

4.4 Predicting the Motion of Objects with Frictional Contacts

In recent years a number of authors have studied the behaviour of objects that are in frictional contact with some environmental object. There are two basic approaches: you can either define what your goal is in terms of desired motion, and work out the conditions necessary to achieve this behaviour, or you can take a given situation and try and calculate what will happen next for some applied wrench.

The first of these is conceptually similar to the backchaining approach to fine motion planning discussed in Chapter 2, and is an approach that Brost uses in his Ph.D. [Brost 91a]. Brost considers goals defined in terms of a location on the relevant CSO, and, with the aid of a model of the mechanics of interaction, progressively constructs contours which bound regions of guaranteed success[2].

[2] Note that our use of the CSO in Chapter 3 was agnostic about the mechanics of frictional contact, and only considered relative geometry. In the following sections we will consider the mechanics involved in realising particular desired transitions between contact configurations.

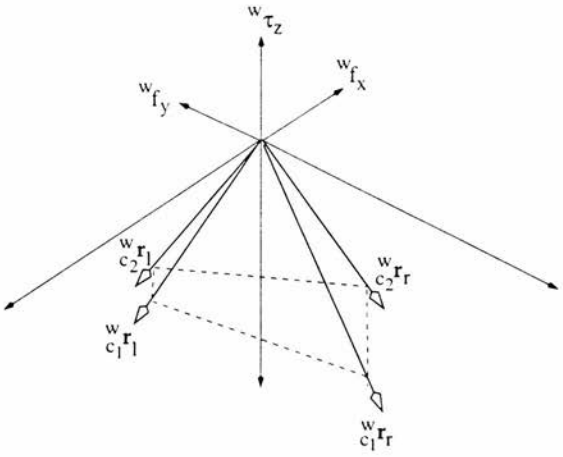


Figure 4.8: The wrench space composite friction cone of the contact configuration shown in figure 4.7.

As part of this model of mechanical interaction Brost and Mason [Brost & Mason 90] developed a technique for representing the forces acting on a planar object by mapping them to the space of acceleration centres, that results in a similar construction to that of Erdmann's configuration space friction cone. Brost uses this construction to test for the possibility of static equilibrium between objects [Brost 91a], [Brost 91b]. Mason has used a version of this representation [Mason 91] as a way of synthesising sets of forces consistent with a desired motion of an object in frictional contact with the environment [Mason 89]. Because some forces are consistent with more than one mode of motion, the procedure involves enumerating the possible contact modes (determined by whether a point of contact is left-sliding, right-sliding, sticking, or breaking free), and for each contact mode the sets of possible environmental response forces are subtracted from the set of acceleration forces to give the set of applied forces consistent with the motion mode. To get just one particular motion mode a force that is consistent with that mode, but inconsistent with all others needs to be chosen. A similar analysis is used by Lynch to determine motions that guarantee a pushed object maintain a stable contact with the pusher [Lynch 92].

Erdmann [Erdmann 84]. [Erdmann 94] considers the problem of predicting the outcome of applying a wrench to an object in frictional contact with another. He performs his analysis in configuration space and shows that the motion outcome of a particular situation may be ambiguous. A geometric test using the configuration space friction cone provides a simple way of establishing the possibility of this ambiguity.

Coulomb's law used in conjunction with Newtonian mechanics can produce motion predictions inconsistent with the constraints. A number of authors have noted these possibilities in recent years [Featherstone 87]. [Rajan *et al.* 87]. [Mason & Wang 88]. [Lynch & Mason 93]. They tend to arise when the conditions are extreme in some way, for example a long slender object with (virtually) all of its mass concentrated at one end. Consequently we will ignore these possibilities here. It is worth noting that the robot arm can be considered to constitute part of the object in contact with the environment, and would therefore need to be taken into account in the predictions of these models.

In the sections that follow we analyse the particular problems that we are interested in demonstrating in terms of a quasi-static analysis and impending motion. This proves to be sufficient for our purposes since we are only interested in qualitative predictions concerning motion: we do not require to be able to predict specific trajectories, since we are interested in exploiting motion in contact as a way of guiding the moving object to the goal.

4.5 Task-Invariant Assembly Strategies from the Mechanics of Contact

The previous sections looked at the effects of friction on the behaviour of objects in contact. They give useful qualitative predictions as to when an object will be stuck, when it will slide, and when it will turn. These models were expressed in terms of the forces acting on the objects, so in the following sections we will assume that we have access to a robotic manipulator that is able to control the force that it exerts on the object that it is holding.

The analysis of the following sections, and the results presented in Chapter 6, demon-

strate that it is possible to generate task-invariant assembly strategies. What it doesn't show is that generating these strategies is straightforward, despite the fact that we are offering the approach described in this thesis as a means of simplifying the programming of assembly robots. We have reason to believe, however, that there is a straightforward way of generating these strategies by exploiting the innate learning abilities that we all possess. We discuss this further in Chapter 7.

4.5.1 Sliding Along a Surface

Although it would probably be difficult to make a case for classifying sliding along a surface as an assembly strategy *per se*, it is an essential pre-requisite for our approach to performing assembly tasks, and so we include a brief discussion of it here.

A corollary of the discussion on the wrench space friction cone is that if a robot applies a force on an object that it is holding that is also in contact with an environmental constraint, and that force is 'into' the environmental constraint without passing through the point of contact, then there will be an induced torque on the object which will act to try and turn it. If we could get the robot to balance this induced torque then it would be able to resist this turning motion. One way of doing this would be to have a geometric model of the robot held object and its interaction with the environment. From this model it would be possible to calculate the induced torque from knowledge of the applied force. Our primary interest is in the development of task-invariant behaviour, *i.e.* in the development of strategies that require no geometric knowledge of the objects involved, only their shape, so we will reject this option. As an alternative we will assume the ability to measure the force and torque (or wrench) on the robot held object. Given this ability we can choose an arbitrary frame of reference about which to perform these measurements and adopt a strategy whereby the robot balances the induced moment that it senses by generating an equal and opposite torque in the same frame of reference.

Although in theory this strategy should be susceptible to uncontrolled (rotational) acceleration should there be even a small disparity between the sensed and commanded torques, empirical investigations have shown that in practice the robot wrist tends to maintain a constant orientation, probably due to friction in the wrist joint. We discuss

this further in sections 5.5 and 5.6. In Chapter 6 we show that this strategy produces a useful form of sliding behaviour.

It is important to note that this technique is not meant to be an attempt at accurately controlling the orientation of an object. Rather what we are interested in is preserving the contact configuration. This can be achieved via a whole range of possible orientations which, as far as the definition of the contact configuration is concerned, are each as valid as the other. This is why we are content with the fact that this strategy only 'tends' to maintain an orientation. We are also relying on the fact that should the contact configuration not be maintained, it is possible to detect this fact (see section 4.6), and that there would be an action associated with this transition appropriate to the task in hand.

We choose to work in a frame of reference that is aligned with the distal link of the robot. This way we can both specify and measure wrenches in a frame of reference that maintains a constant relationship with respect to the robot held object. The utility of this will become apparent in the next sections. However, since the size of objects for a given task may vary, this relationship is one of constant orientation with respect to the robot held object, but not necessarily constant displacement from the contact that the object makes with the environment for a given contact configuration.

Since, for a given contact configuration, both the size of the robot held object and its orientation to the environment can vary, the induced torque can vary over a wide range for conditions that our scheme considers to be equivalent. However, because the strategy that we are adopting is to "balance the induced torque", whatever it might be, this variation is irrelevant.

The introduction of a point on our force-controlled robot about which we both specify and measure wrenches is an important step. This point is in fact the centre of compliance, *i.e.* that point at which the compliance in the system appears to reside [Drake 77], [Lončarić 87]. (See section 2.2.3 for a more complete definition of the centre of compliance.)

Although our strategy for generating sliding behaviour is insensitive to the choice of location of the centre of compliance, our strategies for getting a peg into a hole and

rotating a block in a corner are not completely independent of the choice of its location. As an illustration of the issues involved, consider what happens to the induced moment as the centre of compliance is moved along the central axis of the distal link. For the situation shown in figure 4.3 with the centre of compliance near the wrist joint the sensed induced torque is negative. As the centre of compliance is moved away from the robot towards the point of contact there comes a point where the line of action of the response force goes through the centre of compliance giving a zero sensed induced torque. Beyond this point the sensed induced torque is positive. Because for our sliding behaviour we adopt the strategy of "balancing the induced torque" its sign is irrelevant. This example does, however, illustrate the fact that the location of the centre of compliance could have a marked effect on the behaviour that might ensue from the same wrench applied at different locations.

We would note here that this strategy of balancing an induced torque is effectively a form of closed-loop force control. This has consequences for robot controller implementation which we consider in section 5.6.

4.5.2 Rotating a Peg into a Hole

In [Caine *et al.* 89] the situation shown in figure 4.9 is analysed in order to determine a strategy to get a peg into a hole. The results of this analysis prompted the strategy that we have adopted for getting a peg into a hole so we will reproduce the important points here.

The parts are assumed to be in quasi-static equilibrium with the peg in a state of impending motion. The equations of static equilibrium are written down as:

$$0 = f\sin(\alpha) - \mu f_4 - f_1 \tag{4.26}$$

$$0 = -f\cos(\alpha) + f_4 - \mu f_1 \tag{4.27}$$

$$0 = \tau - \mu f_4 s_x - f_4(s_y - s_4) - \mu f_1 s_y - f_1(s_x - s_1) \tag{4.28}$$

These equations are then reformulated as inequalities by adding an extra component δ to the tangential components of the frictional reaction forces, *i.e.* each μf_j is replaced with $\mu f_j + \delta$. Solving for $\delta \geq 0$ gives a set of inequalities that specify the applied force and moment necessary to balance these extra tangential components thereby giving the

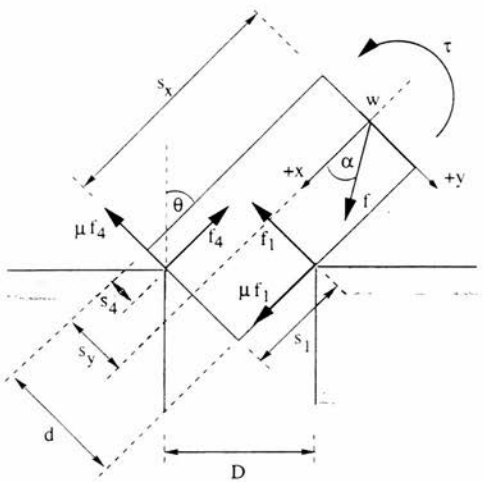


Figure 4.9: Parameters used in the analysis of a peg entering a hole by [Caine *et al.* 89].

desired motion in contact. The constraints derived from this analysis were as follows [Caine *et al.* 89]:

$$\alpha \geq \arctan\left(-\frac{1}{\mu}\right) \tag{4.29}$$

$$\alpha \geq \arctan(\mu) \tag{4.30}$$

$$\frac{\tau}{f} \geq \sin(\alpha)\left[s_x + D\frac{\mu\cos(\theta) - \sin(\theta)}{\mu^2 + 1}\right] + \cos(\alpha)\left[-s_y + D\frac{\cos(\theta) + \mu\sin(\theta)}{\mu^2 + 1}\right] \tag{4.31}$$

Inequality 4.31 suggests that the desired motion can be achieved by performing a 'search' in wrench space: if we have some applied force within the allowed direction constraint, and the applied moment initially set to zero, then we could just increment the applied moment until inequality 4.31 is satisfied. This would be detectable on-line by the fact that the parts start moving with respect to each other. The advantage of this strategy is that we don't need to know any of the values of the variables that define the constraint inequality: the limiting value of the inequality is effectively discovered on-line. This means that the strategy is both independent of the geometry of the parts involved and of the value of the coefficient of friction. Independence of part size has

been our goal all along: independence from knowledge of the coefficient of friction is useful because, as Prentis notes, "The amount of friction between two dry surfaces is difficult to predict with any degree of accuracy" [Prentis 79, p.45].

Since the contact configuration of figure 4.9 is a two point contact one, the wrench space friction cone will be a similar construction to that shown in figure 4.8. One possibility (depending on the actual value of the coefficient of friction and the dimensions of the parts) is shown in figure 4.10. Whilst the peg is in static equilibrium in the neck of the hole the applied wrench will be mirrored by a response wrench. As the moment component of the applied wrench is incremented the ray describing it in wrench space moves along a 'vertical' plane in wrench space. The response wrench moves along the same plane, but in the opposite sense, mirroring the applied wrench until it reaches the edge of the wrench space friction cone. At this point static equilibrium is lost, the response wrench is constrained to lie on the boundary of the wrench space friction cone, and motion ensues. Because we search for the edge of the friction cone we can assume that at the point that we leave the friction cone, the direction of the applied force is still into the constraint surface, so we will have motion *and* contact.

4.5.3 Sliding a Peg in a Hole

It is well known that once the peg has entered the hole the possibility of jamming and wedging occurs [Whitney 82]. Wedging requires the two point contact configuration shown in figure 4.11, and is due to an elastic deformation of the parts that results in the reaction forces at the two points of contact both lying inside their respective friction cones, and acting along the same line. Wedging usually requires that the peg be removed from the hole before the assembly can continue. Jamming is due to the proportions of the components of the applied wrench being such that static equilibrium occurs between the peg and the hole, and can be due to either the two point contact configuration shown in figure 4.11, or a one point contact configuration.

In [Caine *et al.* 89] Caine *et al.* perform an analysis of two point contact of a peg in a hole using the same technique described above for the rotation of a peg into a hole. They derive the following constraint equations that describe the conditions necessary

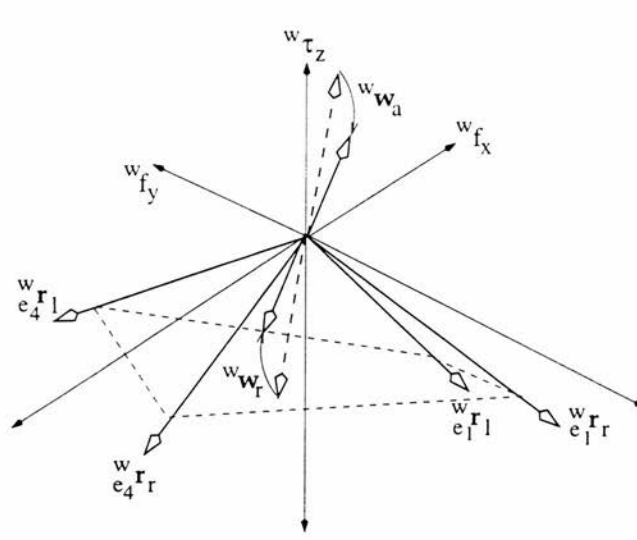


Figure 4.10: Whilst the applied wrench is in the negative of the wrench space friction cone the the response wrench 'mirrors' it.

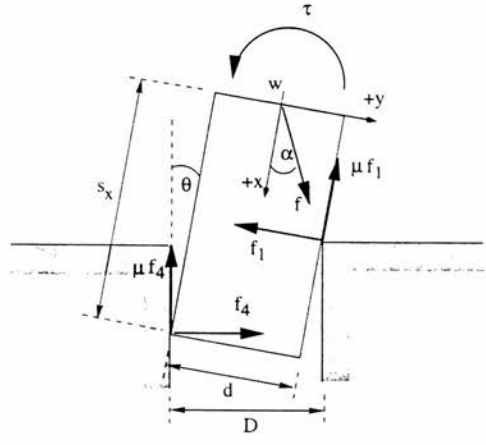


Figure 4.11: Two point contact of a peg in a hole.

for the peg to be sliding in two point contact into the hole:

$$\arctan\left(\frac{\mu\sin(\theta)-\cos(\theta)}{\mu\cos(\theta)+\sin(\theta)}\right) \leq \alpha \leq \arctan\left(\frac{1}{\mu}\right) \tag{4.32}$$

$$\begin{aligned}\frac{\tau}{f} \geq\ & \sin(\alpha)\left\{s_x - \frac{D[\mu\cos(\theta)+\sin(\theta)]+d\left[\frac{\mu^2-1}{2}\sin(2\theta)-\mu\cos(2\theta)\right]}{1-[\mu\sin(\theta)-\cos(\theta)]^2}\right\} \\ & +\cos(\alpha)\left\{\frac{d+D[\mu\sin(\theta)-\cos(\theta)]}{1-[\mu\sin(\theta)-\cos(\theta)]^2}-\frac{d}{2}\right\}\end{aligned} \tag{4.33}$$

As for the case of rotating a peg into a hole described in section 4.5.2, we see that the desired motion is achieved providing some lower bound on the ratio of the moment to the force magnitude, of the applied wrench, is met, whilst simultaneously satisfying a constraint on the direction of the applied force. This suggests that if the peg is jammed in the hole it is possible to find on-line the conditions that will break the jamming by incrementing or decrementing the applied moment in the sense determined by the orientation of the peg. As before, by doing this we are able to avoid having precise knowledge of the geometric details of the parts involved, or the coefficient of friction.

4.5.4 Rotating a Block into a Corner

The contact configuration central to this task is shown in figure 4.12. We assume that the interior angle of the environmental constraint forming the corner is 90^o. What we require for this task is for the robot manipulator to be able to rotate the block that it is holding, from some initial orientation, into a goal orientation that aligns the sides of the block with the planes of the constraint surface. The goal can either be that the sides of the block that are parallel to the x-axis of frame w are required to end up vertical, or it can be such that they are required to end up horizontal.

Our initial attempt at a strategy to get a block to rotate into a corner was to try and apply the same strategy that the analysis by [Caine *et al.* 89] had suggested to us as a viable means of getting a peg into a hole, *viz.* apply a force that would tend to keep the block in the corner, and increment the moment in the appropriate sense until the desired motion ensued [Deacon & Malcolm 94]. This however turned out to be rather naive: we found in practice that as the block approached its goal orientation it became

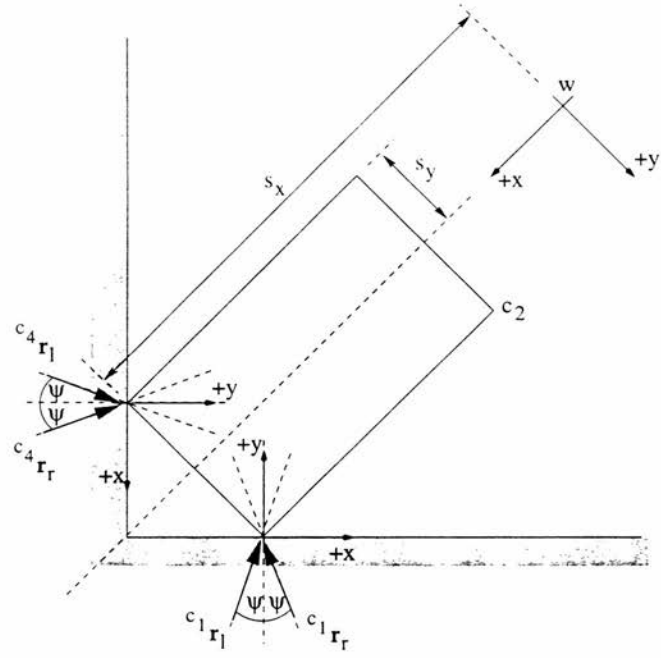


Figure 4.12: Diagram of a block in a corner showing the real space friction cones at the points of contact.

likely that the applied moment would cause the block to spin away from the constraint surface. In retrospect this is not really surprising.

Consider reorienting the block clockwise into the corner. The appropriate sense for the applied moment would also be clockwise. A moment applied in frame w would appear as a force at c_4 in a direction given by $\vec{s_r} \times {}^w\hat{\mathbf{k}}$, where

$$\vec{s_r} = s_x {}^w\hat{\mathbf{i}} - s_y {}^w\hat{\mathbf{j}} \tag{4.34}$$

and ${}^w\hat{\mathbf{i}}$, ${}^w\hat{\mathbf{j}}$ and ${}^w\hat{\mathbf{k}}$ are the unit vectors in the x, y and z directions, respectively, of frame w. As the block nears the goal the angle between this vector and the vertical constraint diminishes. Consequently the component of this force normal to the vertical constraint diminishes, resulting in a smaller and smaller frictional reaction force at c_4

as the block approaches its goal. The net force in the vertical direction away from the corner consistently increases, resulting in the block accelerating away from its goal location.

The strategy that we adopted as the result of some experimentation can be explained with the aid of figure 4.13. Assume that the block is in two point contact with the corner as shown.

Any applied force has its line of action through the centre of compliance, since this is the origin of the frame of reference that we choose to work in. We know that if there is to be no relative motion at a point of contact that the reaction force at that point of contact must lie in the friction cone. For static equilibrium the applied force must be balanced by a force which is the vector sum of the reaction forces at each of the points of contact (*cf.* equation 4.25): in fact for a planar system of three forces to be in static equilibrium they must be concurrent, *i.e.* they all intersect at a common point (see for example [Prentis 79]).

Consider the situation when the block is in a state of impending clockwise motion. The frictional reaction forces from both points of contact will lie on their respective 'left' friction cone edges. These reaction forces intersect at the point labelled *ll* in figure 4.13. For static equilibrium the applied force must also pass through this point. If the applied force were now rotated anticlockwise about the centre of compliance, it would pass outside of the composite friction cone, the frictional reaction forces would be unable to balance the applied force, and clockwise motion would ensue.

The situation just described shows that, assuming the parts are initially in static equilibrium, it is possible to achieve the desired motion in contact by performing a search in wrench space for the edge of the composite friction cone. In this respect the strategy is similar to those for getting a peg into a hole, and the breaking of jamming once in the hole. This time though, we rotate the applied force instead of incrementing the applied moment. Since the applied moment is zero this can be visualised as the applied force sweeping around the $\tau_z = 0$ plane in wrench space, until it reaches the edge of the composite friction cone.

The viability of this strategy depends, however, on (amongst other things) the location

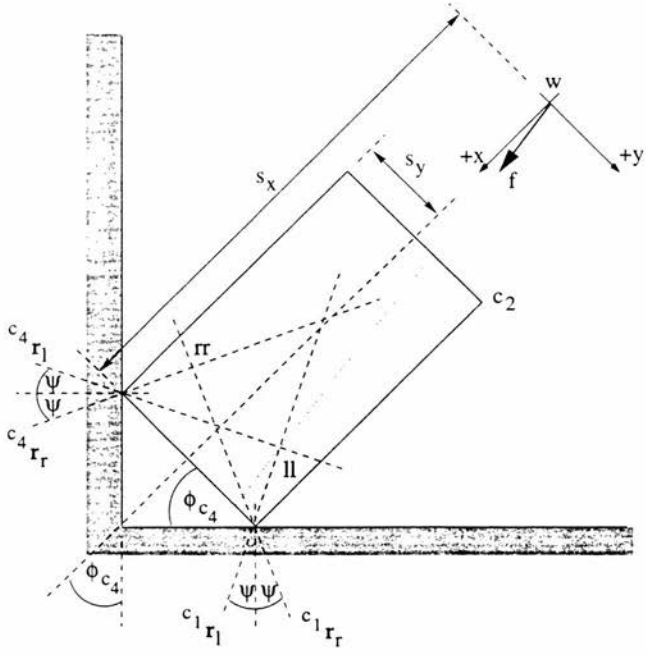


Figure 4.13: A block in a corner and the intersection of the real space friction cones.

of the centre of compliance. Figure 4.14 shows a situation where the centre of compliance lies on the other side of the gap between ll and rr to that shown in figure 4.13. The situation shown is that of impending clockwise motion again, this time however, the applied force needs to be rotated clockwise about the centre of compliance in order to precipitate the desired motion. In other words we need to know which side of the line joining ll and rr the centre of compliance lies in order to choose the appropriate sense of applied force reorientation.

In order to give us a concrete starting point from which to further investigate the limitations of this strategy we will assume that we are able to choose what side of the line joining ll and rr the frame w is on. For the sake of discussion we will assume that we have chosen the frame w to be 'above' the line joining ll and rr, as in figure 4.13, and that we are interested in getting a clockwise reorientation of the block. We will specify

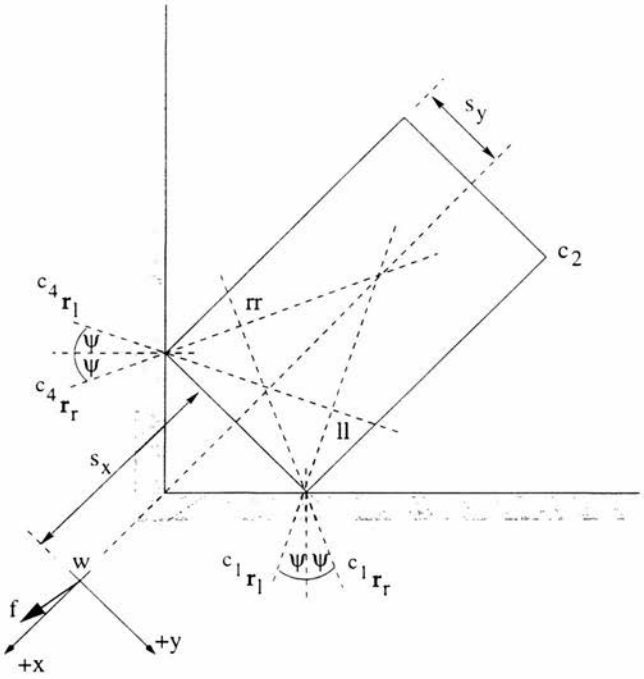


Figure 4.14: A block in a corner with the centre of compliance lying 'below' the points of intersection of the two left and two right real space friction cone edges.

that the applied force is to start off pointing along the positive x-axis of frame w. This gives us an initial condition that is independent of part geometry. We would also like the initial conditions to be static equilibrium (or such that they will take us toward our goal without intervention). Since in real space the points marked ll and rr form a sort of slalom gate that the applied force needs to pass through if static equilibrium is to be possible, this requires that the point rr lie on or above the block's centre line at the instigation of the strategy. This gives a bound on the allowable orientation of the block at the strategy's onset, but we will deduce a more restrictive bound presently.

Given these initial conditions we will now examine various cases to determine the effects of the location of the centre of compliance (*i.e.* frame w) under our strategy of

reorienting the applied force until motion ensues.

case 1 centre of compliance to the right of c_1's $\mathbf{r}_l$.

As the applied force is rotated anticlockwise it crosses ll before it points into the friction cone at c_1 so both contacts slide simultaneously in the desired sense.

case 2 centre of compliance lies along c_1's $\mathbf{r}_l$.

As the applied force is rotated anticlockwise it crosses ll when it lies on the $\mathbf{r}_l$ line associated with c_1. At this point there is zero induced moment. As the applied force passes $\mathbf{r}_l$ the direction of the applied force points into the friction cone at c_1, and there is a reaction force in the opposite direction that passes through c_1. This passes to the 'left' of the centre of compliance, inducing a negative torque that works to turn the block in the desired sense by pivoting at c_1. If the applied force maintains the same direction relative to the block the pivoting action will cause the applied force to leave the friction cone and the block will slide back into the corner. The centre of compliance will now lie in the realms of case 1 due to the effects of reorientation.

case 3 centre of compliance to the left of c_1's $\mathbf{r}_l$ line, but to the right of c_1's $\mathbf{r}_r$.

In this case it is possible for the direction of the applied force to lie inside the friction cone of the point of contact at c_1 before it crosses ll. This would cause a reaction force that passed through c_1 in the opposite direction which would pass to the 'right' of the centre of compliance, inducing a positive moment on the block. Anticlockwise motion would be resisted due to the kinematic constraint formed by the contact at c_4. As the applied force is rotated further anticlockwise it will pass through the point of contact at c_1 and an equal and opposite reaction force will mirror it through c_1. Once the applied force has passed through this direction there will be an equal and opposite reaction force that acts through c_1 passing to the 'left' of the centre of compliance causing the block to pivot about c_1 clockwise. (In principle, at least; in practice the induced moment might well be so small as to make no physical difference in the face of various inertial, stiction, *etc.* effects, but if the applied force were to continue rotating about w in an anticlockwise sense the induce moment would continue to increase.) Note

that, assuming the direction of the applied force remains constant with respect to the block once motion ensues, the pivoting action will act so as to rotate the line of action of the applied force outside of the friction cone at c_1, by crossing c_1's $\mathbf{r}_l$ line, causing the block to slide back into the corner (whilst still rotating in the appropriate sense).

If the centre of compliance were to lie any further down the centre line of the block it would lie beyond the line joining ll and rr before any further case need be considered. In fact a part of the centre line of the block that falls within the bounds of case 3 will lie below the line joining ll and rr.

The above considerations show that it is preferable to have the centre of compliance either on, or to the right of, c_1's $\mathbf{r}_l$. However, if we are prepared to tolerate a certain amount of pivoting, since the end-result is still consistent with our aims, then it is allowable for the centre of compliance to lie within the friction cone at the point of contact at c_1 providing it is 'above' the line joining ll and rr. But this constraint is of course determined by both the orientation of the block and its dimensions. What we require however, is a strategy that will work for all instances of the task without knowing the dimensions of the block.

Another feature that we would like our strategy to possess is the ability to rotate the block into the goal orientation from any initial orientation where the block is in two point contact with the corner.

In order to be able to achieve both of these requirements we will specify that $\gamma \geq 45^o$, or equivalently that $s_x \geq s_y$. We will call this the aspect ratio constraint. Note that this is a constraint on the location of the centre of compliance, rather than the dimensions of the block, although of course the former is not unrelated to the latter in this instance.

Given this restriction it is feasible to achieve reorientation of the block by applying a moment to the block for $\phi_{c_4} < 45^o$ (see figure 4.13), and then switching to the strategy described above once $\phi_{c_4} \geq 45^o$. The direction of $\vec{s_r}$ given by equation 4.34 will be into the vertical constraint until $\phi_{c_4} = \gamma$. Since $\gamma \geq 45^o$, the earliest that $\vec{s_r}$ will be parallel to the constraint surface is at $\phi_{c_4} = 45^o$. By switching from an applied moment

to an induced moment at 45^o we just catch the point at which the block is about to accelerate away from the constraint surface in the worst case. We can improve this situation by adding a 'bias' force along the x-axis of the frame of reference w, thereby making sure that the force experienced at c_4 is always into the constraint surface even in the worst case.

Now, since it is possible to get the orientation of the block to within 45^o of the goal using an applied moment, and because we have specified that $\gamma \geq 45^o$, we know that the centre of compliance, once we are ready to induce a torque on the block, must lie either on or to the right of the normal at c_1. This is consistent with our initial assumption.

In summary, we can guarantee the effectiveness of our strategy for inducing a torque providing the following constraints are met:-

$$\gamma \geq 45^o \tag{4.35}$$

$$\phi_{c_4} \geq 45^o \tag{4.36}$$

Taking into account the physical limits of the situation, and the fact that $\phi_{c_4} = \phi_{c_1} - 90^o$ we get the following allowed ranges on the variables involved:

$$\gamma = [45^o, 90^o] \tag{4.37}$$

$$\phi_{c_1} = [135^o, 180^o] \tag{4.38}$$

We would note that although we have concentrated on describing rotating a block clockwise into a corner the strategy is of course symmetric and applies equally well *mutatis mutandis* when rotating a block anticlockwise into a corner. Additionally, although we have to switch our wrench application strategy when $\phi_{c_4} = 45^o$, which requires a measurement of the relative orientation between the block and the corner, this value (providing the constraints listed above are met), is not dependent on part size or location, orientation being invariant under translation, rotation and uniform scaling (see for example [Mundy & Zisserman 92, Ch.1]).

4.6 Detecting Kinematic Transitions

So far we have considered force application strategies that can be used to achieve a desired motion of an object whilst in some contact configuration. Here we will consider the detection of the kinematic transitions that an object might experience during this motion. It might be possible to develop a taxonomy of possible transitions, but for the purposes of this thesis it will be sufficient to illustrate the kind of thing we have in mind with a couple of examples. We will assume the ability to sense the wrench acting on an object in some frame of reference w that maintains a fixed relationship to the object. It should become apparent that what we are doing here is developing a sort of 'robot's-eye' view of object interaction.

The first example we will consider is a block sliding along a smooth surface towards a corner constraint. We know from our discussion of Coulomb friction in section 4.3 that for a sliding point contact the environmental response force must lie on the edge of the friction cone at the point of contact. We may not know the orientation of the block with respect to the constraint surface, but this doesn't really matter. In fact the block could be constantly reorienting as it goes along, providing this reorientation is 'smooth', the change in the direction of the response force that is sensed in frame w will also be 'smooth'. However, when the block hits the corner constraint there is an abrupt change in the environmental response force that is experienced as a discontinuity in the wrench sensed in frame w. Figure 4.15 shows the applied and response forces before the collision, and some time after the collision. Note that all applied forces consistent with sliding the block along the surface into the corner are also consistent with static equilibrium once the block reaches the corner. This diagram is consistent with our previous quasi-static analysis, but if we were to take into account dynamic effects change in the direction of the response force would be even more pronounced. The sense of the change in the direction of the response force is predictable from the geometry. In frame w there is a discontinuity sensed in both the direction of the response force and in the magnitude of the sensed moment, and the sense of the 'jump' in the sensed moment is predictable from the expected sense of direction change in the response force. Of course, if we take into account dynamic effects there will also be a change in the sensed magnitude of the response force.

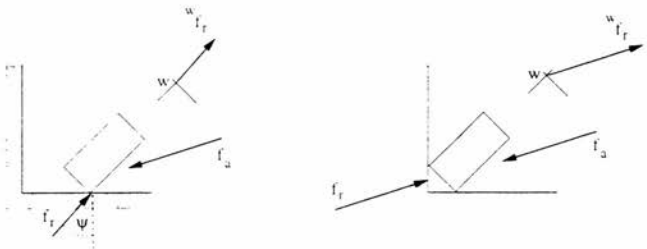


Figure 4.15: A block before (left) and after (right) colliding with a corner.

Note that in the above example, if we took the block from an initial state of equilibrium, and precipitated motion by rotating the applied force on it, as described previously, then the disparity between the coefficients of static and kinetic friction work in our favour. To see this consider that if the two values were the same, then the direction of the response force whilst the block were sliding and the direction once static equilibrium is regained would be very similar, because motion would have started when the direction of the applied force had just passed 'beyond' the direction at which equilibrium is possible. However, because the coefficient of kinetic friction is less than the coefficient of static friction, once the block starts sliding the response force rotates in the opposite sense to that of the rotation of the applied force that was needed to instigate the motion.

A situation that is of interest to us is a block that is rotating about a point contact that it is making with a constraint surface. An example is shown in figure 4.16. If we imagine the block starting with c_1 in contact with the constraint surface whilst reorienting clockwise, it will get to the situation shown in figure 4.7 with edge e_1 flat against the constraint surface. Its propensity to try to continue its rotational motion will cause the point of contact to shift from c_1 to c_2. This will result in a discontinuous change in the response force's moment arm about the wrist frame of reference (from a_1 to a_2 in figure 4.16). (There will correspondingly be a discontinuous change in s_x of equation 4.14 or, equivalently, a discontinuous change in s_r of equation 4.24). This will be 'seen' in frame w as a discontinuity in the moment sensed. In contrast, in this situation the sensed direction of the response force behaves 'smoothly'.

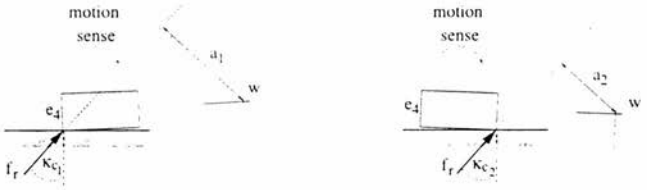


Figure 4.16: A block just before (left) and just after (right) lying flat against a constraint surface.

This scenario can be visualised in wrench space with the aid of figure 4.8. If the block is pivoting about c_1 then, just at the point that edge e_1 lies flat against the constraint surface, the response wrench will be lying somewhere between ${}^{w}_{c_1}\mathbf{r}_l$ and ${}^{w}_{c_1}\mathbf{r}_r$ in wrench space. When the point of contact passes from c_1 to c_2 the response wrench will 'jump' to a point between ${}^{w}_{c_2}\mathbf{r}_l$ and ${}^{w}_{c_2}\mathbf{r}_r$ that is directly 'above' its previous direction in wrench space, since there is no change in the sensed response force direction. If the block of figure 4.8 were sliding leftwards, say, as well as rotating clockwise, and the sliding motion persisted during the kinematic transition we are considering, then in wrench space the response wrench would initially lie along ${}^{w}_{c_1}\mathbf{r}_l$, and after the transition it would lie along ${}^{w}_{c_2}\mathbf{r}_l$. In this situation $\kappa_{c_1} = \kappa_{c_2} = \psi$ in figure 4.16.

4.7 Hybrid Force/Position Control

We started this chapter with a discussion of the behaviour of a position-controlled manipulator when in contact with the environment and used this as a way of introducing the benefits of being able to slide. From this we went on to a discussion of the behaviour ensuing between two objects due to the mechanics of contact. Many others have noted the benefits that arise from sliding, and the fact it tends to 'soak up' spatial uncertainty, and have used this as an argument for requiring that manipulators be able to control the force that they exert on the environment. However, it is argued that whilst force control is useful for maintaining a compliant contact with the environment, a manipulator should still be position controlled where it can be.

The formal codification of what has become the *de facto* standard methodology for this approach is due to Mason [Mason 81] where he argues that in degrees of freedom where a manipulator has positional freedom it should be position controlled, whereas in directions that the task constrains the manipulator it should be force-controlled. Mason formulates this approach in terms of the surface of the CSO for a task, *i.e.* in terms of the "C-surface": directions tangent to a C-surface should be position controlled, and those normal to a C-surface force controlled. This leads to a situation where there are two sub-spaces, one for the position controlled degrees of freedom, and the other for the force controlled degrees of freedom, and for any task these two sub-spaces are the "orthogonal complements" of each other. This is usually expressed mathematically in terms of "selection matrices", one for position and orientation, $\mathbf{S}_v$, and one for force and torque, $\mathbf{S}_f$. These are matrices whose only non-zero elements are 1's that lie on the leading diagonal and effectively act as filters determining which degrees of freedom are position controlled and which force controlled. So, for instance, for a spatial peg in hole insertion with the z-axis aligned with the peg's longitudinal axis we have

$$\mathbf{S}_v = \begin{pmatrix} 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{pmatrix} \tag{4.39}$$

$$\mathbf{S}_f = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{pmatrix} \tag{4.40}$$

If $\mathbf{t}$ is the end-effector twist, and $\mathbf{w}$ is the end-effector wrench, then these matrices express the fact that the peg is to be position controlled along and about the z-axis, whilst the other degrees of freedom are force-controlled, since $\mathbf{t}$ and $\mathbf{w}$ are required to satisfy the following:

$$\mathbf{S}_f\mathbf{t} = \mathbf{0} \tag{4.41}$$

$$\mathbf{S}_v\mathbf{w} = \mathbf{0} \tag{4.42}$$

The idea of orthogonal complementarity is expressed by

$$\mathbf{S}_f + \mathbf{S}_v = \mathbf{I}_6 \tag{4.43}$$

where $\mathbf{I}_6$ is the six dimensional identity matrix. A number of researchers have used this formulation as the basis for the design of robot controllers *e.g.* [Raibert & Craig 81], [West & Asada 85], [Zhang & Paul 85], [Khatib 87], [De Schutter & Van Brussel 88a].

Although this formulation is intuitively appealing its validity has been questioned by Lipkin and Duffy [Lipkin & Duffy 88], [Duffy 90] on the basis of its dimensional inconsistency (because it mixes linear and rotational sub-spaces), and the fact that the orthogonal complementarity of $\mathbf{S}_f$ and $\mathbf{S}_v$ is dependent on both the choice of units and the choice of origin. A practical consequence of this last dependency is that if the frame of reference of the model doesn't coincide with its expected location in the real world it is possible to generate undesired motions along force controlled directions, or *vice versa*, in a manner that is contrary to the requirements of the task [Lipkin & Duffy 88], [Duffy 90], [Abbati-Marescotti *et al.* 90].

By reformulating the problem in terms of screw theory it is possible to salvage the intuitively appealing notion of position and force controlled sub-spaces, and devise filters that are dimensionally consistent and invariant to change in units and choice of frame of reference [Lipkin & Duffy 88], [Abbati-Marescotti *et al.* 90], [Goldenberg & Song 96].

Let $\mathbf{T}_F$ belong to an n-system of twists $\mathbf{F}$, $0 \leq n \leq 6$, representing the twists of freedom available to a body during some state of a task, and $\mathbf{W}_c$ belong to an m-system ($m = 6 - n$), of wrenches $\mathbf{c}$ representing the wrenches of constraint, *i.e.*

$$\mathbf{T}_F = \mathbf{F}\varrho_F \tag{4.44}$$

$$= \mathbf{F}_1\varrho_1 + \ldots \mathbf{F}_n\varrho_n \tag{4.45}$$

$$\mathbf{W}_c = \mathbf{c}\eta_c \tag{4.46}$$

$$= \mathbf{c}_1\eta_1 + \ldots \mathbf{c}_m\eta_m \tag{4.47}$$

where the ϱ's, and η's are arbitrary scalars. Lipkin and Duffy [Lipkin & Duffy 88] point out that instead of orthogonal complements of configuration space being the basis for decomposition of a task into position and force controlled components, what is actually required is the recipricocity of the twists of freedom and the wrenches of constraint, *i.e.*

$$\mathbf{c}^T\mathbf{F} = 0 \tag{4.48}$$

Goldenberg and Song [Goldenberg & Song 96] have recently shown that the recipric-

ocity of screw spaces effectively maintains the orthogonality properties from the task space when the task is expressed as twists of freedom and wrenches of constraint. They also present a method for deriving invariant kinestatic[3] filters. In addition, since the degrees of freedom of a manipulator are unlikely to coincide with the axes of the frame of reference in which a task is described they also consider whether it is possible to maintain the orthogonality between position and force controlled subspaces when they are mapped into joint variable and joint torque space respectively. They note that this would only be possible when

$$\mathbf{J}(\mathbf{q})\,\mathbf{J}^T(\mathbf{q}) = \lambda \mathbf{I}_6 \tag{4.49}$$

where $\mathbf{J}(\mathbf{q})$ is the manipulator Jacobian (see Appendix B), and λ is a scalar. The design of controllers for hybrid position/force control is considered in [Song & Goldenberg 96] where it is shown that unless this condition holds it is not possible to design position and force controllers that do not influence each others' sub-space.

Despite the mathematical rigour of [Lipkin & Duffy 88], [Abbati-Marescotti *et al.* 90] and [Goldenberg & Song 96] this work rests on an assumption that is not in general true: the recipricocity of a twist and wrench relies on a workless constraint [Ohwovoriole 80]. [Ohwovoriole & Roth 81]. This means that any contacts should be frictionless; a situation that is rather unlikely in practice making disturbances between force and position controllers inevitable even if equation 4.49 does apply.

The fact that a contact is unlikely to be frictionless has a further consequence for the use of hybrid force/position control, one that is easily overlooked. Consider a planar object in contact with a flat constraint surface. We will assume, for the sake of argument, that the coefficient of friction is $\mu = 0.25$. The objective is to move the object from its current location along the constraint surface whilst maintaining contact with the constraint surface, *i.e.* we want the object to slide. Hybrid force/position control would require that there be force regulation normal to the constraint surface and position (or velocity) regulation tangent to the constraint surface. Assume that the force regulation value has been chosen as 10N. Does the object slide? The answer is indeterminate from the information given. Suppose that the position regulation

[3] This is a term introduced by Lipkin and Duffy to refer to the "dualistic properties and relations between the first order kinematics and statics of a rigid body" [Lipkin & Duffy 88].

direction is only able to exert 1N in order to realise the specified position command[4]. Now does the object slide?

From equation 4.2 we can see that a tangential force > 2.5N is required before the object will slide (otherwise the applied force, which is the vector sum of the normal and tangential forces, points into the friction cone). Note that this is purely conditional on the ratio of the magnitudes of the normal and tangential forces at the point of contact: the fact that position control is involved is incidental (which is why our first question was unanswerable). Only in the special case of a frictionless contact can a hybrid force/position control scheme guarantee sliding. The assumption normally made is that the position (or velocity) controlled directions will be able to apply sufficient motive force to achieve sliding without having to actually know exactly how much force is required. However, just because there is a motion regulated direction it does not mean that that motion will be achievable; its realisation is contingent on the prevailing force conditions. The point is that if we want to program a manipulator to work in terms of sliding it is best to deal in terms of forces directly, rather than indirectly via a motion servo.

Our approach avoids these problems associated with motion controlled degrees of freedom by either controlling the force (or torque) in any particular degree of freedom, or by utilising one or more uncontrolled degrees of freedom. In this respect it is more closely allied to Mason's Ph.D. work than hybrid force/position control (which was the subject of Mason's Masters dissertation [Mason 78]). We take this up in the next section.

4.8 The Relationship of Our Manipulation Strategies to Object Pushing Strategies

For his Ph.D. work Mason studied the reorienting behaviour of objects being pushed on a horizontal plane [Mason 82], [Mason 86]. This study can be considered the impetus

[4] To appreciate the plausibility of this scenario bear in mind that it is unlikely that a manipulator has the ability to apply forces at its end-effector isotropically, *i.e.* equally in all directions. The strongest and weakest directions usually vary throughout the workspace [Asada 83], [Asada & Youcef-Toumi 87]; near singularities the ability to apply a force in one direction 'swamps' that in another, whilst at a singularity the ability to apply a force in (at least) one direction is lost completely.

for a significant amount of subsequent work in the vein of exploiting the mechanics of manipulation [Brost 85], [Peshkin 86], [Wang 89], [Goldberg 90], [Christiansen 91], [Brost 91a], [Lynch 96]. In [Mason 85] Mason describes the basic notion that this work is intended to exploit, that of a manipulation funnel:

> "Just as a common funnel eliminates uncertainty in the location of a grain of sand, a manipulation funnel eliminates uncertainty in the location of objects."

In other words he is interested in exploiting mechanical behaviour to reduce spatial uncertainty. A good example of this is a rule that he deduced in order to predict the sense of reorientation of an object sliding on a horizontal surface due to a point contact with a pushing object [Mason 82], [Mason 86].

The rule in question states that if we have a point contact between two objects, and one is pushing the other, then we can determine which way the object will turn by considering how the friction cone rays, and a ray denoting the pushing direction, vote relative to the centre of mass[5]: if two or more rays pass to the left of the centre of mass, the object will rotate clockwise, if two or more pass to the right of the centre of mass, the object will turn anticlockwise, if the vote is a tie, *i.e.* one ray passes to the left, one to the right, and one passes through the centre of mass, the object will translate. Both Mason's rule and the work described in this thesis assume a quasi-static model of manipulation, and both are interested in predicting the sense of the turning moment on an object. In our case the sense of the turning moment follows from equations 4.22 to 4.24, which predicts the same sense of reorientation with respect to the centre of compliance that Mason's rule predicts with respect to the centre of mass.

[5] Mason cites MacMillan [MacMillan 36] as observing that during a pure translation the system of frictional forces arising due to the contact area between the sliding object and its support may be modelled as a single force acting through a fixed point called the centre of friction. Mason goes on to express a rule that predicts the sense of rotation of a pushed object, under quasi-static conditions, in terms of the centre of friction. This however is a misnomer: Mason's rule ought to be expressed with respect to the centroid of the contact pressure distribution [Brost 95]. For uniform isotropic friction and pushing in the plane of sliding that causes translation, the centroid of the pressure distribution coincides in the plane of sliding with the centre of friction [Goyal *et al.* 91]. But this is only true for translational motion. No such straightforward relationship exists when rotational motion is involved (see the effective centre of friction derived in [Lynch 92]). However, if all externally applied loads other than gravity lie in the plane of sliding, then the centroid of the pressure distribution coincides with the centre of mass (in the plane of sliding) [Goyal *et al.* 91]. Consequently, under these conditions Mason's rule can be stated relative to the centre of mass.

The crucial link here is that the centre of mass for the object pushing case is functionally equivalent to the centre of compliance for a robot held object[6]. This relation is also noted in [Akella & Mason 95], where they consider the use of a force-controlled robot to orient an object that the robot is holding, by means of an algorithm developed for parts feeding by pushing [Goldberg 90]. This work implicitly exploits induced torques, although it is not presented in these terms, and no consideration is given to the ability of a force-controlled robot to apply a torque. At the time of publication this work had not been implemented on an actual robot [Akella 95].

Another conceptual link between work on the mechanics of manipulation and the assembly strategies described in this thesis is the fact that the objects being manipulated are essentially uncontrolled in at least one degree of freedom. This is in contrast to the more usual approach adopted in assembly situations of hybrid force/position control (described in the previous section). We believe it is because we allow one or more uncontrolled degrees of freedom that the strategies described in this chapter work under such a wide range of conditions; we are effectively exploiting manipulation funnels that occur in an assembly task where the kinematic constraints form the (conceptual) walls of the funnel.

Consider trying to rotate a (planar) block in a corner whilst maintaining two point contact using hybrid force/position control. Since it is the rotational degree of freedom that is unconstrained the trajectory to be executed would most naturally be described in terms of the location of the centre of rotation that the block is to rotate about. As the points of contact are always sliding tangent to the constraining surface they must both be rotating about a point that lies on their respective normals, *i.e.* the instantaneous centre of rotation always lies where the two normals at the points of contact intersect. The locus of the instantaneous centre of rotation is, of course, dependent on the dimension of the block between the contacting points, and a reliable on-line estimate of its location would require accurate knowledge of this dimension as well as the orientation of the block with respect to the corner constraint. There would also be a significant additional on-line computational burden throughout the whole procedure

[6] In [Wright & Deacon 96] we show that this functional equivalence also extends to the centre of gravity of gravitational catastrophe machines resting on a constraint with friction. (See [Poston & Stewart 78] for a definition of a gravitational catastrophe machine.)

in order to calculate the instantaneous centre of rotation. Similar considerations apply to the operation of getting a peg into a hole when the peg is in two point contact at the mouth of the hole (see figure 4.9). The complexity of programming this operation is probably contributory to the fact that most people try and perform peg in hole insertion with a robot by aligning the axes of peg and hole and translating the peg along its axis into the hole, despite the fact that there is a far greater chance of getting (at least part of) the peg into the hole if it approaches the hole tilted [Inoue 79], [Mason 84]. Consider how many people try and place a pen-top onto a pen by aligning the axes of the pen and pen-top? It is far more usual to exploit the kinematic constraints of the situation to guide the relative motion of the parts.

By allowing an uncontrolled degree of freedom, we are able to deduce a generic strategy that requires no accurate knowledge of dimension or location, entails no significant computational burden, and is able to exploit environmental constraints. Instead of trying to coerce the world to conform to the expectations implicit in the design of the robot controller, it exploits the nature of the behaviours afforded by the mechanics of contact.

4.9 Summary

The point of this chapter has been to show that it is possible to manipulate objects in the terms that describe the states of the contact configuration graph. That is to say that it is possible to maintain a contact configuration during motion toward some other desired contact configuration. Moreover, it is possible to do this without knowing the size or location of the objects involved, the strategy of performing a search in wrench space being particularly useful in this respect. In addition, the way that the sensed wrench changes allows us to detect when a transition between contact configurations has occurred. However, in order to do this what we require is a manipulator that can slide along surfaces, possibly into a new kinematic constraint without causing damage. We also require the ability to specify the wrench to be exerted on the manipulator-held object. It is virtually impossible to achieve these things with a manipulator operating solely under position control. Hybrid force/position control improves the situation, but the position controlled degrees of freedom are still subject to spatial uncertainty,

can require significant amounts of computation to make their operation tenable, and may in fact fail because their specification is agnostic about the conditions needed to guarantee their success during sliding. The strategies described here that require force control only are computationally much simpler and are expressed in the same terms as the conditions that will 'decide' whether they are successful. These considerations suggest that we need a manipulator that is force-controllable. We take this up in the next chapter.

Chapter 5

The Design of a Sensitive Robot

The discussion of Chapter 3 left us with a way of modelling assembly tasks that abstracts away from metric properties. In Chapter 4 we showed by example that it is possible to generate assembly strategies that use this model, but they require a manipulator that is both able to exert controlled forces on the environment, as well as being accommodating to environmental constraints. The desirability of a robot manipulator that is able to react to forces and torques sensed during assembly has been realised for some time *e.g.* [Nevins & Whitney 73], [Inoue 79], and a significant amount of research has been invested in providing manipulators with a force control capability (see [Whitney 87] for a review). Despite this, force responsive manipulators are rarely found on the factory floor [Morris 92]. In this chapter we will introduce a force control scheme that will allow us to test our proposed approach to assembly, and compare it with other force control schemes.

5.1 Implementing Force-Controllability

5.1.1 Previous Work on Force Control

In order for a robot to be able to 'feel' how an assembly task is progressing it is natural to try and sense the forces and moments arising in a frame of reference attached to one of the components involved in the assembly task. This has given rise to the design of force/torque sensors that can either be attached to the end of a robot arm, or placed under an assembly jig (see for example [Watson & Drake 75], [Nakamura 91, Ch. 10]).

However, attempts at adding force control to manipulators designed to be position controlled by servoing on the information obtained from a wrist-mounted force-torque sensor have invariably turned out to be susceptible to instabilities during contact. Some of the documented reasons for these instabilities are:-

- Closed loop behaviour is dependent on the value of the product of the force feedback loop (forward) gain and the stiffness of the environment, thus requiring prediction of the stiffness of the environment for guaranteed stable operation [Whitney 77][1].

- A force control loop 'wrapped around' a position controlled arm results in a system that tries to achieve an endpoint force by changing the readings from its joint encoders. Because small changes in position can lead to large changes in forces, the forces attainable are limited by the resolution of the joint position sensors. This can lead to the system hunting for a desired end-point force [De Schutter & Van Brussel 88b].

- The non-colocation of the sensor and actuator results in unmodelled dynamics between them that could cause instability [Eppinger & Seering 86], [Eppinger 88].

- Certain robot kinematic configurations (*e.g.* revolute) in conjunction with hybrid position/force controllers (see section 4.7) have been found unstable. The original explanation given for this was that because only certain directions are position compensated the others can cause the arm to 'drift' towards singularities, and since calculation of the Jacobian inverse is included in the control loop, and at singularities loses full rank, the arm becomes uncontrollable [An *et al.* 88]. More recently Fisher and Mujtaba have argued that this problem can be avoided by using the pseudo-inverse of the product of the selection matrix and the Jacobian to map Cartesian space position errors into joint space [Fisher & Mujtaba 92].

- Colgate has shown that an architecture that consists of an external force controller and an internal position or velocity controller creates a competition between intrinsic and feedback-generated mechanical impedances that may lead to

[1] It is also worth noting that the stiffness of a manipulator varies with its configuration [Asada & Slotine 86].

contact instability [Colgate 89]. This problem results from the architecture itself: if the robot were inherently accommodating rather than non backdrivable (as with geared robots), it would not be necessary to add as much force feedback to achieve compliance and the competition would be avoided.

The scope of a number of these instability problems could be limited by including some passive compliance in the arm [An *et al.* 88], [Eppinger 88], [Trevelyan 94], but this is an unpopular option because it undermines the ability to accurately control the position of a manipulator's end-effector when required [Paul 87]. Note that this lack of accurate position control is not a problem for our approach since we are not interested in controlling positions directly, only indirectly through exploiting kinematic constraints.

In order to improve the performance of force control various researchers have tried to improve the quality of control of a manipulator's joint torque (for a review see [Stokić & Vukobratović 93]). The usual arrangement is that a strain gauge is added to the shaft attached to the load, and joint torque control is achieved by regulating the strain gauge output. This has the advantage of removing some of the unmodelled dynamics due to the non-colocation of sensor and actuator, but not all of them, *e.g.* a study that attempted to place a joint torque controller around a joint with a harmonic drive led to limit cycles due to the drive flexibility [Luh *et al.* 83]. In [Eloseegui *et al.* 90] it is shown that a controller designed around this kind of arrangement has low sensitivity to variations in the environment stiffness. It has also been argued that it can usefully compensate for the dynamic interactions between joints [Stokić & Vukobratović 93].

Disadvantages include the fact that it is mechanically difficult to mount the strain gauge and arrange for the output signals to be taken away from the arm (remember, the shaft will be rotating when the arm is moving). Moreover the mounting of the strain gauge requires that part of the mechanical structure of the arm be weakened so that the strain gauge gives a readable output. This reduces the structural stiffness of the arm which has undesirable consequences for position control in terms of both accuracy and stability. In addition, the study described in [Featherstone 85] found that (in a geared arm) the strain gauge signals were not only contaminated with electrical

noise, but were also corrupted by the mechanical noise and vibration present in a joint mechanism (*cf.* [Armstrong 88]).

Another problem encountered with adding force control to traditionally constructed arms is that in order to get a force servo to faithfully track a desired force 'trajectory' the arm structure requires a natural frequency higher than the highest frequency component in the signal [Fowler & Eppinger 91]. This however conflicts with the requirements for stability when an arm makes (a new) contact with the environment. In this situation it is desirable for the arm to have a low bandwidth so that it behaves like a low pass filter and cuts out any high frequency components that occur due to impact [An *et al.* 88], [Fowler & Eppinger 91]. However, with the scheme described in the preceeding chapters we are able to escape this paradox: we are not concerned with achieving and maintaining particular force values, our interest lies in being able to achieve the relative magnitudes of the wrist wrench that allows us to slide one part over another. For a given situation there is no one particular solution to this problem so the precise regulation of the elements of the wrist wrench are not an issue. In fact the scheme that we are proposing does not know *a priori* what the values of the desired wrench are, rather it 'discovers' them on-line by performing a search in wrench space: see section 4.5. Moreover, it is not the magnitudes of the components of the applied wrench *per se* that are important, but their *relative* magnitudes.

5.1.2 Exploiting the Natural Behaviour of a D.C. Motor

During the early stages of this work we considered adapting an existing robot arm in order to try and realise the force-controllable behaviour that the approach described in Chapter 3 requires. In order to be able to detect the making of new kinematic contacts we thought that it would be useful to try and exploit the natural behaviour of a direct current (d.c.) motor [Deacon 90].

Consider the operation of a d.c. motor. A schematic showing the electrical parameters is given in figure 5.1. The equation describing the relationship between these parameters is

$$V_{mot} - E = i_{mot} R_{mot} + L_{mot} \frac{di_{mot}}{dt} \tag{5.1}$$

where V_{mot} is the voltage applied across the motor, E is the back electro-motive force

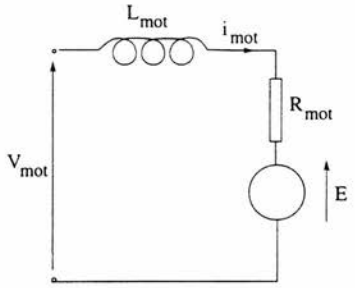


Figure 5.1: Electrical parameters of a d.c. motor

(e.m.f.), i_{mot} is the current flowing through the motor, R_{mot} is the motor's terminal resistance and L_{mot} is its inductance. The inductance of a motor is often ignored on the basis that it is orders of magnitude less than the motor's resistance [Luh 83]. This leaves us with

$$V_{mot} - E = i_{mot} R_{mot} \tag{5.2}$$

In steady-state, no load conditions we have

$$V_{mot} - E \approx 0 \tag{5.3}$$

i.e. the applied voltage and the back e.m.f. are about the same and virtually no current flows. Even when loaded, under steady state conditions the current through a d.c. motor is 'small'. If the motor became obstructed its angular velocity $\dot{q}_{mot}$ would drop to zero, causing the back e.m.f. to fall to zero since

$$E = \dot{q}_{mot} K_e \tag{5.4}$$

where K_e is the motor's back e.m.f. constant. This would cause the potential difference (p.d.) across the motor (*i.e.* the difference between V_{mot} and E) to rise creating a stall current proportional to the terminal resistance. In principle this could be used in determining the joint torque because the torque that the motor exerts is proportional to the current flowing through it according to:

$$\tau_{mot} = K_t i_{mot} \tag{5.5}$$

where K_t is the motor torque constant, and from this we could attempt to estimate the force that the arm was exerting on the environment[2].

This arrangement might have been useful for detecting collisions[3], *i.e.* the making of new kinematic constraints between the objects involved, but it doesn't make the arm force-controllable because the actual current going through the motor is affected by the motor's speed, which can be seen by substituting the expression for the back e.m.f. E from equation 5.4 into equation 5.2. In order to overcome this we decided to try and regulate the current through the joint motors using a feedback loop. The basic idea (expanded upon in section D.2) is that if a motor is obstructed so that its speed reduces, $\dot{q}_{mot}$ in equation 5.4 would reduce, causing E to reduce. From inspection of equation 5.2 we see that in order to keep the motor current constant V_{mot} would need to be reduced. In an extreme case, if the motor were forced to start turning in the opposite direction to that which it was originally turning, then E would change sign due to the change in sign of $\dot{q}_{mot}$, and, in order to maintain a constant motor current, V_{mot} would also need to change sign. This kind of behaviour can in principle be achieved using a circuit of the general form shown in figure 5.2, which realises a controllable current source [Electro-Craft Corporation 80]. In practice however, there are limitations imposed by the usual construction of a robot manipulator designed to be position controlled. We expand upon this in the next section.

5.1.3 An Underlying Problem

Realisation of the current regulation scheme just described relies on the motor being back-drivable. Unfortunately robots designed to be position controlled resist being back-driven for the following reasons:–

1. In order to try and ensure that the kinematic models of position controlled arms are accurate they are made from stiff material. This invariably results in a heavy device *i.e.* large inertia.

[2] See appendix B for the mapping between joint torques and end-point wrenches.

[3] Actually, operating standard d.c. motors under stall conditions is probably not a good idea because allowing large currents to flow through them can seriously degrade their performance due to demagnetization.

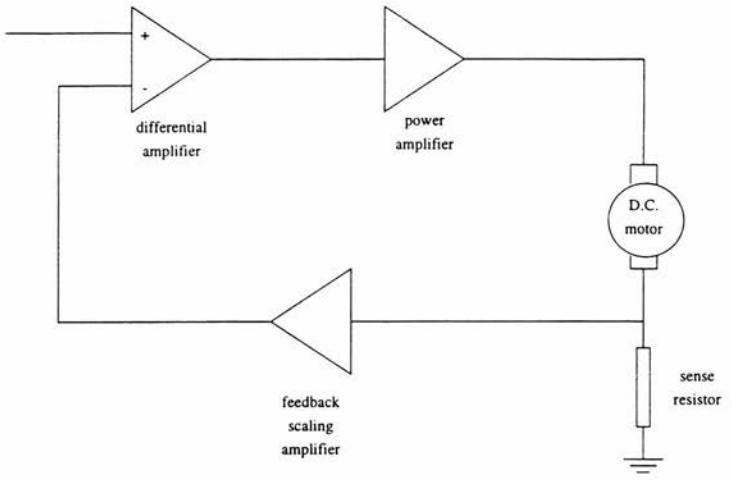


Figure 5.2: Schematic of a d.c. motor current regulating feedback loop.

2. The inertia seen by a force trying to push a geared robot joint is magnified by a factor which is a function of the gear ratio.

Point 2 is worth expanding on. Consider the situation depicted in figure 5.3. Referring to this figure the following relations are well known, (see for example [Klafter *et al.* 89]):

$$\frac{N_{mot}}{N_{load}} = \frac{\tau_{mot}}{\tau_{load}} = \frac{\dot{q}_{load}}{\dot{q}_{mot}} = \frac{\ddot{q}_{load}}{\ddot{q}_{mot}} \tag{5.6}$$

where *mot* subscripts denote a quantity associated with the motor's side of the gear train, and *load* subscripts a quantity on the load's side of the gear train; N represents the number of gear teeth, τ represents torque, $\dot{q}$ represents angular velocity, and $\ddot{q}$ represents angular acceleration. From this we can deduce

$$\tau_{load} = \tau_{mot} \frac{N_{load}}{N_{mot}} \tag{5.7}$$

and

$$\dot{q}_{load} = \dot{q}_{mot} \frac{N_{mot}}{N_{load}} \tag{5.8}$$

So if $N_{load} > N_{mot}$ then the speed is reduced and torque is increased. In other words, the gear ratio can be used to transform the usual high speed/low torque characteristics

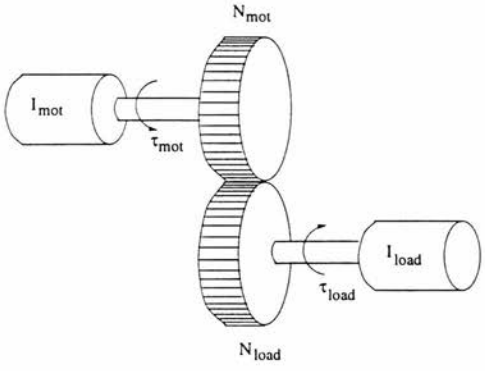


Figure 5.3: A geared motor with a load.

of a d.c. motor into the more desirable characteristics for robotics applications of low speed/high torque.

A design with a gear ratio such that $N_{load} > N_{mot}$ is also advantageous for position controlled robotic applications because of the way it affects the load inertia that the motor 'sees'. The motor torque necessary to accelerate a load is given by

$$\tau_{mot} = \frac{N_{mot}}{N_{load}} \ddot{q}_{load} I_{load} \tag{5.9}$$

where I_{load} is the load inertia. This can be viewed as if τ_{mot} is accelerating an equivalent inertial load I_{eq} at an angular acceleration of $\ddot{q}_{mot}$, *i.e.*

$$\ddot{q}_{mot} I_{eq} = \frac{N_{mot}}{N_{load}} \ddot{q}_{load} I_{load} \tag{5.10}$$

Substituting for $\frac{\ddot{q}_{load}}{\ddot{q}_{mot}}$ from equation 5.6 gives

$$I_{eq} = \left(\frac{N_{mot}}{N_{load}}\right)^2 I_{load} \tag{5.11}$$

If $N_{load} > N_{mot}$ then the motor sees a reduced load; reduced by a factor of $\left(\frac{N_{mot}}{N_{load}}\right)^2$.

For robot arms the situation is more complicated than for an ordinary gear train. The dynamics of a robot arm are represented by a set of nonlinear, coupled, differential equations. In appendix A we show that these equations are of the general form:

$$\tau_j = I_{arm} \ddot{q}_j + \tau_{coup} + \tau_{non} \tag{5.12}$$

where τ_j is the torque at joint j, τ_{coup} is the interactive inertia (coupling) torque, τ_{non} is the nonlinear torque due to Coriolis, centripetal and gravitational effects, $\ddot{q}_j$ is the acceleration of the joint j, I_{arm} is the effective inertia of the arm link(s). This relationship however, fails to model the effects of gearing and τ_j should be interpreted as the torque about the shaft on the output side of the gear train, similarly $\ddot{q}_j$ is the acceleration of the output shaft.

We can model the effects of gearing by considering how the motor torque is 'spent'. The nonlinear and coupling torque will be related to the motor torque according to equation 5.7:

$$\tau_{mot} = \frac{N_{mot}}{N_{load}} (\tau_{coup} + \tau_{non}) \tag{5.13}$$

The arm inertia will be related to the motor torque via equations 5.9, 5.10 and 5.11:

$$\tau_{mot} = \left(\frac{N_{mot}}{N_{load}}\right)^2 I_{arm}\, \ddot{q}_{mot} \tag{5.14}$$

If the motor were driving an unloaded gear train its acceleration would be determined according to

$$\tau_{mot} = I_{mot}\, \ddot{q}_{mot} \tag{5.15}$$

where I_{mot} is the inertia of the motor rotor and gears. By combining these effects we arrive at the dynamic equation for an individual joint that takes into account the effects of gearing [Asada & Slotine 86], [Asada & Youcef-Toumi 87]:

$$\tau_{mot} = \left(I_{mot} + \frac{I_{arm}}{r_j^2}\right) \ddot{q}_{mot} + \frac{\tau_{coup} + \tau_{non}}{r_j} \tag{5.16}$$

where $r_j = N_{load}/N_{mot}$ is the gear ratio for joint j ($N_{load} > N_{mot}$).

For a position controlled robot a high gear ratio has the beneficial effect of making the invariant motor and gear inertia, I_{mot}, dominate the dynamics. Note that not only is the arm inertia seen by the motor reduced by a factor of r_j^2, but the nonlinear and coupling torques are reduced by a factor of r_j. This reduces the effects of changing dynamics resulting from change of arm configuration and load variation, making it easier to design position servos with the aim of providing uniform performance over the arm's range of operation.

However, if we go back to our simple gear train and look at the opposite situation, where something in the environment is attempting to influence the motion of the

motor's output shaft, then the inertia experienced, I_{exp}, is given by

$$I_{exp} = \left(\frac{N_{load}}{N_{mot}}\right)^2 I_{mot} \tag{5.17}$$

As can be seen, the inertia experienced by the pusher is magnified by the square of the gear ratio. Again, for the case of a robot arm the situation is more complicated due to their articulated nature: a torque applied τ_{app} to joint j would experience the following

$$\tau_{app} = \left(I_{mot}\, r_j^2 + I_{arm}\right) \ddot{q}_j + \tau_{coup} + \tau_{non} \tag{5.18}$$

showing the increase in effective inertia[4] due to gearing (*cf.* equation 5.12). Gear ratios of 100:1 are common in robot arms resulting in the inertia of the motor appearing magnified by a factor of 10^4 from the point of view of an external applied force[5]. The practical consequences of this is that any arm with a gear train with a reduction ratio of the magnitude normally found in robot arms is very unresponsive to forces applied on it from the environment making it (as good as) impossible to implement force control using a motor current loop.

5.1.4 A Solution to the 'Underlying Problem'

We have shown that the natural mode of operation of an ordinary d.c. motor lends itself to being used for joint torque control if we regulate the current flowing through it, but that this mode of operation is severely hampered by the normal construction of robot manipulators, *viz.* they include gear trains that increases the inertia that would be experienced by something in the environment exerting a force on the arm. So, if gears are such a problem then why don't we just remove them from the picture? This has in fact already been done in the design of a number of manipulators where the designers have used what is known as direct-drive technology. The most obvious disadvantage of this is that you can't use gearing in a speed/torque trade-off. Rather, you need motors that are strong enough to bear directly the weight of the arm, including any distal direct-drive motors, as well as any load the arm might be carrying. Although

[4] The effective inertia for motor j being defined as the coefficient of $\ddot{q}_j$; see Appendix A.

[5] In fact the robot that we originally intended to use on this project, the Mitsubishi RM501 Movemaster robot, has a gear ratio of approximately 250:1 at its shoulder joint.

such technology exists in the form of rare-earth magnet motors, they are expensive, and can dissipate a lot of power as heat when operating.

Despite these drawbacks, there are a number of reasons why robot manipulators constructed from direct drive motors have been considered desirable (see for example [Asada & Kanade 83], [Asada & Youcef-Toumi 87], [An *et al.* 88]). Most obviously various problems associated with gearing are avoided: there is no backlash, friction is significantly reduced, and the compliance unavoidable in some gears, such as harmonic drives, is absent. These phenomena are all troublesome to the designer of position servos and their reduction makes the possibility of accurate position control more viable. Their suitability for force control has also been noted, that the actuators can be treated as torque sources, that they are backdrivable, and so do not suffer from dissipative losses due to friction, or the magnification of inertia due to the effects of gearing (as described above), are facts that have not gone unnoticed.

There is another consideration that is relevant to the construction of a sensitive robot that tends to substantiate what we have discussed so far: Hogan [Hogan 85a] has argued that the "causality" of a manipulator affects its performance during contact. His argument revolves around the fact that mechanical systems can be considered to be either admittances or impedances[6]. This distinction can be characterised in terms of the instantaneous power flow between interacting physical systems. The power flow along any particular degree of freedom is expressible as the product of two conjugate variables, an effort (*e.g.* a force, a voltage), and a flow (*e.g.* a velocity, a current), and no one of the two systems involved can determine both variables. An admittance accepts effort (*e.g.* force) inputs and yields flow (*e.g.* motion) outputs. An impedance behaves in a complementary fashion and accepts flow (motion) inputs and yields effort (force) outputs.

> Now, for almost all manipulatory tasks the environment at least contains inertias and/or kinematic constraints, physical systems which accept force inputs and which determine their own motion in response. However, ... while

[6] This notion is a generalisation of the electrical engineering concept. Consequently, mechanical impedance and admittance can usually be considered to be the reciprocal of each other, *i.e.* a high impedance is a low admittance, and *vice versa*. This is not the case however, when the system involved is non-linear. See [Hogan 85a] for more details.

> a constrained inertial object can always be pushed on, it cannot always be moved; These systems are properly described as admittances. Seen from the manipulator the world is an admittance. [Hogan 85a, p.4]

He argues that during physical interaction, along any degree of freedom, the causality of the two systems involved should complement each other; if one is an impedance the other should be an admittance, and *vice versa*. So, given that the world is best modelled as an admittance, a manipulator should be an impedance, *i.e.* it accepts motion and yields a force.

These considerations, in the light of our proposed scheme of controlling the progression of an assembly task, led us to decide to design and construct a manipulator whose actuators were current regulated direct-drive motors. Choosing the motors to be current regulated is consistent with Hogan's arguments on causality. From the discussion above it should be clear that a current regulated motor would accept a motion and yield a force, an impedance; voltage regulation of a motor results in a system whose natural propensity is to try and control the motor's velocity, which is the converse situation, *i.e.* an admittance[7].

In practice, of course, a robot manipulator possesses inertia, just as the environment does, endowing it with some degree of admittance-like properties. The use of gears makes these effects more pronounced (equation 5.18). This is in essence what is behind Colgate's observation concerning the competition between intrinsic and feedback generated impedances[8], and the problem of the natural mode of operation of non-backdrivable robots being inherently opposed to that of force feedback loops led him to suggest that either passive devices or direct-drive robots should be used in applications requiring contact.

Regulating motor currents allows the actuators to be treated as torque sources and thus

[7] Interestingly, despite the force of his own arguments, Hogan proposed implementing impedance control on manipulators of standard design (see for example [Hogan 84b]). This appears to be due to the fact that he considers position control an important part of a manipulator's repertoire and envisages specifying the impedance of a manipulator's end-point as a second order dynamic system that determines a manipulator's response to disturbances to its prescribed trajectory. The work of Colgate mentioned previously [Colgate 89] establishes some of the limitations of this approach.

[8] A description of the limitation on how much a manipulator's effective inertia can be reduced is given in [Colgate & Hogan 89].

the arm can be operated in an open-loop fashion with respect to the contact wrench. This is important because, since this scheme is open-loop with respect to the wrench at the end-point, it is not susceptible to the instabilities during contact described above.

It is probably worth pressing home the advantage of this control scheme: rather than having to perform high-rate real-time computations in order to artificially synthesise compliant motion in a manipulator whose behaviour naturally opposes it, the behaviour of our arm is such that it is automatically compliant. The 'decision', if you like, to be compliant, is made almost instantaneously by the current regulation loops which are implemented in analogue electronics: if a torque is applied to a joint that is greater than that which the motor is exerting due to the current flowing through it, the joint just swings out of the way. Another way of viewing this is to consider the actuators to be behaving as sensors as well[9]. In summary, our arm is designed from the bottom up to perform contact operations, stably.

Interestingly both [An *et al.* 88] and [Fowler & Eppinger 91] used direct-drive manipulators with joint torque control that is open loop with respect to the end-point force, but their main interest was in combining this with force/torque information from a wrist-mounted sensor in order to provide improved force servoing at the manipulator's end-point. This kind of scheme is particularly appropriate for grinding and polishing tasks, and would be useful in an assembly context if the arm were having to act against gravity and no accurate model of the configuration-dependent and load-dependent disturbance forces that gravity would cause (see Appendix A) were available. We consider this issue further in Chapter 7. For the experiments reported in this thesis we avoided this issue by confining our robot operations to being in the horizontal plane.

5.1.5 Some Comments on the Use of Direct-Drive Motors

We have shown above that not using gear trains in the joints of robot manipulators results in their motion being more sensitive to the effects of non-linear and coupling torques. For the designer of a position controlled manipulator these act as disturbances to the closed loop performance. Consequently research has been per-

[9] Although this possibility is noted in [Coiffet 83, p.56] we suspect that this author underestimated the effects of gearing.

formed into compensating for these terms explicitly [Asada *et al.* 83], [An *et al.* 88], [Khosla & Kanade 88]. There has also been work into designing arms that possess decoupled and invariant inertia matrices [Asada & Youcef-Toumi 87]. The rationale behind this being that if the inertial properties of an arm are such that their effects do not vary over the arm's work volume, then the system dynamics appear constant to the feedback loop trying to enforce the position control, thereby simplifying its design. We mentioned above that one of the effects of gearing is that the invariant inertia of the motor rotor dominates an arm's dynamics making it easier to design position servos that exhibit uniform performance throughout the arm's workspace. In effect this technique tries to emulate the geared situation in this respect.

5.2 The Basics of the Proposed Control Scheme

The work described in the previous section is directed at being able to achieve high speed accurate control of a direct-drive manipulator. We are primarily interested in the process of automatic assembly. During an assembly task motions of an arm are 'slow'. In addition our control scheme does not require the ability to accurately follow a trajectory in space: in fact, since we are assuming sufficient kinematic constraints to guide the robot-held part towards the goal configuration, it does not require the ability to follow a trajectory *at all.* The predominant mode of operation of our arm will be sliding behaviour, so that any disturbance that the inertial and velocity product terms might introduce to the arm motion will most likely be 'masked' by the more significant effects of the disturbance that is the environment's reaction to the arm pushing against it. Consequently we will assume that the inertial and velocity product terms in a manipulator's dynamics equations can be neglected during the mode of operation that we are interested in.

The complete expression for the dynamics of a robot arm is given by the following (see Appendices A, B and D):

$$\boldsymbol{\tau} = \mathbf{M}(\mathbf{q})\,\ddot{\mathbf{q}} + \mathbf{C}(\mathbf{q},\dot{\mathbf{q}})\,\dot{\mathbf{q}} + \mathbf{G}(\mathbf{q}) + \mathbf{J}^T(\mathbf{q})\,\mathbf{w} \tag{5.19}$$

where $\mathbf{M}(\mathbf{q})$ is the moment of inertia matrix, $\mathbf{C}(\mathbf{q},\dot{\mathbf{q}})$ is the matrix of velocity product terms, $\mathbf{G}(\mathbf{q})$ is a vector of gravitational loadings, $\mathbf{J}^T(\mathbf{q})$ is the transpose of the manip-

ulator Jacobian (as described in Appendix B), and $\mathbf{w}$ is the external wrench acting at the point that the Jacobian is defined for. Ignoring the inertial and velocity product terms effectively means setting the first two terms on the right hand side of this equation to zero. Since all the experiments reported in this thesis were conducted with our arm working in the horizontal plane we can also set the third term on the right hand side to zero. This leaves us with

$$\boldsymbol{\tau} = \mathbf{J}^T(\mathbf{q})\,\mathbf{w} \tag{5.20}$$

Because we want to be able to specify the wrench that is to be exerted on the manipulator held object, and this object is effectively just an extension of the manipulator itself, our assembly strategies encode what value of $\mathbf{w}$ should be applied at a particular stage of an assembly task. This is converted to the joint torques needed to realise this wrench according to equation 5.20. In other words, we continuously evaluate this expression in real-time in order to manipulate the robot held object, and this is tantamount to driving the arm itself 'through' an assembly task.

This control scheme can be represented by the diagram in figure 5.4, where $\mathbf{w}_{des}$ is the wrench we desire the arm to exert in the frame of reference that the Jacobian $\mathbf{J}(\mathbf{q})$ is defined for, $\boldsymbol{\tau}_{des}$ are the corresponding desired torques at the manipulator joints to realise this wrench, $\mathbf{V}_{D/A}$ is a vector of voltages from the digital to analogue converter that acts as the reference signal to the current regulation feedback loops, which produce the regulated currents denoted by $\mathbf{i}_{mot}$. See section D.2 for more on the relationship between the D/A voltages and the motor currents. This relationship effectively determines what the box labelled "scaling" does, although we also provide a means for making sure that the components of the drive torques maintain the desired relative magnitudes despite the possibility of motor saturation.

We limit the maximum torques that the shoulder and elbow motors can exert to the maximum that the wrist motor can exert so that they will never overpower the wrist motor. When the desired wrench is converted into desired joint torques, each torque is compared to the maximum allowed for that motor (including the wrist). If any torque is greater than the maximum allowed it is rescaled to its maximum value, and the other motor torques are rescaled by the same ratio. Thus the relative magnitudes of the desired wrench are maintained.

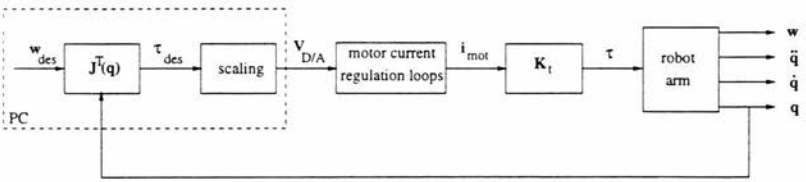


Figure 5.4: Schematic of the open-loop force control scheme.

The regulated currents, $\mathbf{i}_{mot}$, are 'converted' to the actuator torques $\boldsymbol{\tau}$ by the joint motors. This process is represented by a diagonal matrix of motor constants $\mathbf{K}_t$. The end-product is a set of joint positions, velocities and accelerations, represented by $\mathbf{q}$, $\dot{\mathbf{q}}$ and $\ddot{\mathbf{q}}$ respectively, and a contact wrench $\mathbf{w}$, that are determined by the physics of the situation according to equation 5.19. Note that although the values of the joint positions are fed-back and used in the evaluation of $\mathbf{J}^T(\mathbf{q})$, they are not used as part of a closed-loop regulation scheme.

In figure 5.4 we are agnostic about the frame of reference that the desired wrench and the Jacobian are defined for. In practice, whilst the arm is working autonomously, we always use a frame of reference attached to the distal link whose x-axis is aligned with the distal link (see section B.3). This way when we specify a desired wrench we are specifying it in a frame fixed with respect to the held object.

Part of our final design also includes a three degree of freedom joystick as part of the system in order that we are able to control the arm manually. The controlling software was written so that the deflections of the joystick could be interpreted as desired wrench components in either a frame of reference aligned with the wrist or a frame of reference aligned with the base (or world) coordinates. Often it seemed more natural to the user to be working in the latter frame. The diagram of figure 5.4 is applicable irrespective of the frame chosen, and indeed whether the desired wrench is generated manually via the joystick, or automatically by the controlling software. We would just note here that our joystick proved particularly useful in the development of the task-invariant assembly strategies described in Chapter 6.

For more details on the definition of the frames of reference used see Appendix B.

5.3 A Refinement to the Proposed Control Scheme

With the proposed control scheme described above the manipulator is essentially a force/torque source. The problem with this is that any force that it is commanded to attain it will try and achieve regardless of its situation. If it is constrained in the direction that it has been commanded to exert a force then this is fine, but if it is unconstrained then it will accelerate at a rate determined by its own inertia. In situations where it is intended to slide along an environmental constraint the commanded force will be partially balanced by the frictional response of the contact, but its motion will still be partially unconstrained. In order to 'moderate' the arms behaviour under these conditions we decided to include a velocity limiting feedback loop in the control scheme. However, because of the mode of operation of our arm it would be impractical to use a 'standard' servo design for this feedback loop.

The linear and rotational components of the desired wrench and sensed twist are treated separately because they are different types of subspaces. In order to understand the rationale behind the operation of our velocity limiting feedback loop consider the operation of the linear part. A standard servo feedback loop is designed such that the actuator is driven off the magnitude of the error, and the feedback signal is intended to be as close to the commanded signal as possible. This means that the error signal is small, so for high performance tracking of the commanded signal a high gain is required in the forward path of the loop. In our case, the feedback signal is the sensed wrist velocity. When our robot arm is in static equilibrium with the environment the velocity signal is zero. If we were to employ a normal feedback arrangement the error signal in this situation would be the commanded signal, and this would be multiplied by a large forward gain. In effect we would just have the system shown in figure 5.4 with a large gain multiplying the desired wrist force. This could easily result in the arm trying to apply dangerously high forces on the environment. Instead, our velocity limiting feedback loop uses unity gain in the forward path and modulates the applied force magnitude according to the magnitude of the gain in the feedback path. The size of this feedback gain was determined empirically, and can be varied over a wide range depending on the performance desired. A diagram of the loop is shown in figure 5.5.

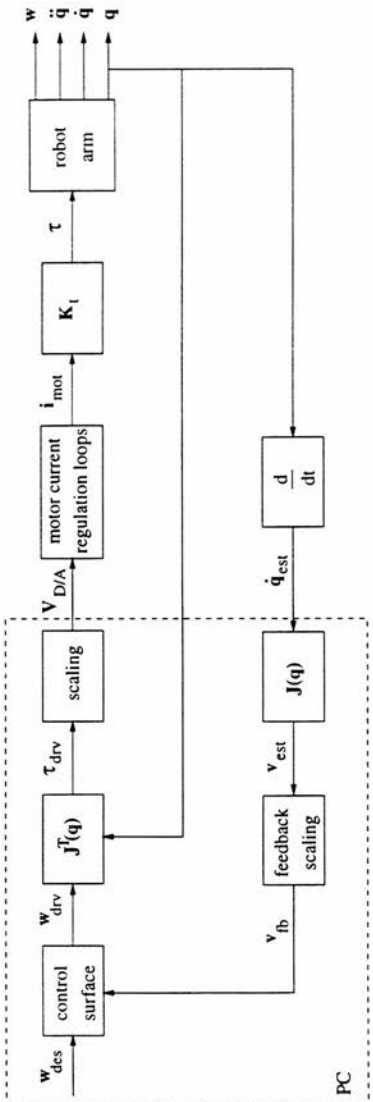


Figure 5.5: Schematic of the velocity limiting feedback loop.

The symbols in this diagram that do not appear in figure 5.4 are defined as follows: $\mathbf{w}_{drv}$ is the 'drive' wrench, that is, it is the desired wrench modulated by the wrist space Cartesian velocity estimate $\mathbf{v}_{est}$ multiplied by the feedback gain. τ_{drv} is the vector of 'drive' torques that correspond to the drive wrench, and $\dot{\mathbf{q}}_{est}$ is the estimate of the arm joint velocities derived from the joint position signals by analogue differentiator circuits (see sections 5.4 and D.4).

Although we require the velocity of the arm to be limited we require that this be achieved without affecting the direction of the applied force. This is because the direction of application of the desired force is likely to be important in the context of the task that the arm is performing. Consider, for instance, if the arm is sliding along a flat surface and the applied force just lies outside of the friction cone at the point of contact. The velocity of the arm will be in a direction tangential to the surface along which it slides. If both the magnitude and the direction of the velocity were used to modulate the desired force, the desired force would have a component added to it that was in the opposite direction to the sensed velocity. This would tend to rotate the drive force into the constraint surface, *i.e.* into the friction cone, causing the arm to stick. This illustrates why we only modulate the magnitude of the desired force and preserve the direction.

The box labelled "control surface"[10] modulates the magnitude of the drive force whilst maintaining its direction by implementing the following:

- For the linear components:–
  - (0) Convert the desired force from x and y components to magnitude and direction.
  - (1) if the desired force and the sensed velocity are not in the same direction (*i.e.* if the sensed velocity direction is not within $\pm 90^o$ of the desired force direction), then
    - the drive force magnitude is set equal to the desired force magnitude

    else
    - the drive force magnitude is set equal to the desired force magnitude minus the wrist velocity magnitude scaled by the feedback gain. If the

[10] For want of a better expression we adopt the terminology of [Barto 87] here.

resulting drive force magnitude is negative its value is halved.

(2) The x and y components of the drive wrench are set to the desired force direction multiplied by the drive force magnitude.

- For the rotational component:–

(1) If the desired moment and the sensed angular velocity are in opposite directions then

- the drive moment is set equal to the desired moment

else

- the drive wrench is set equal to the desired wrench minus the sensed angular velocity multiplied by the feedback gain. If the resulting drive moment is in the opposite sense to the desired moment its value is halved.

Note the algorithm is essentially the same for both the linear and rotational components.

If the drive signals' magnitudes are not halved when they are in the opposite sense to the desired signals, then the behaviour of the arm to an 'impulse' from the joystick is oscillatory. This value is not particularly critical: multiplying the drive signals' magnitudes by a number less than 1 progressively reduces the oscillatory behaviour as the number is reduced. Using a value of 0.5 results in the arm approximating well damped behaviour. Example data of the behaviour of the wrist joint is shown in figure 5.6.

The benefits of limiting the magnitude of the driving force by the sensed velocity in the way described above is most apparent when using the joystick to control the arm. It is important to note, however, that we have not compromised the basic desirable properties of the arm in any way.

5.4 The Overall System

Due to the expense of direct-drive motors, and the fact that it is still possible to demonstrate interesting behaviour on planar tasks, we decided to construct a planar, three degree of freedom manipulator. The finished device is shown in figure 5.7. The overall architecture of our robot system is shown in figure 5.8. We called this system

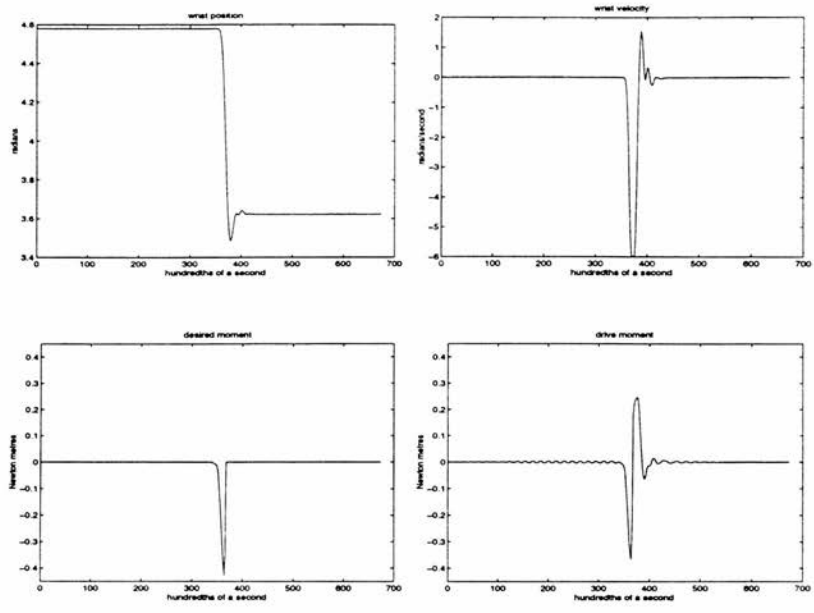


Figure 5.6: Example data showing the behaviour of the wrist joint when subjected to an 'impulse' from the joystick.

EDDIE, an acronym for Edinburgh Direct-Drive Invariance Experiment.

The controlling computer is an IBM compatible PC with a 486DX50 processor, 8MB of DRAM, and a VESA local bus. The controlling software is written in Borland's Turbo Pascal V7.0 running under Microsoft DOS V6.2. This software configuration actually proved to be problematic in the implementation of a real-time control capability, but it was possible to get round the various limitations sufficiently to demonstrate the operation of the arm. With the aid of a timer card we were able to get a guaranteed sample rate of 100Hz. In future work we intend to invest in an operating system designed for real-time control.

Joint positions are sensed using potentiometers in order to make the point that highly accurate position information is not necessary in order to reliably achieve an assembly. Velocity estimates are formed by taking copies of each of the position signals and

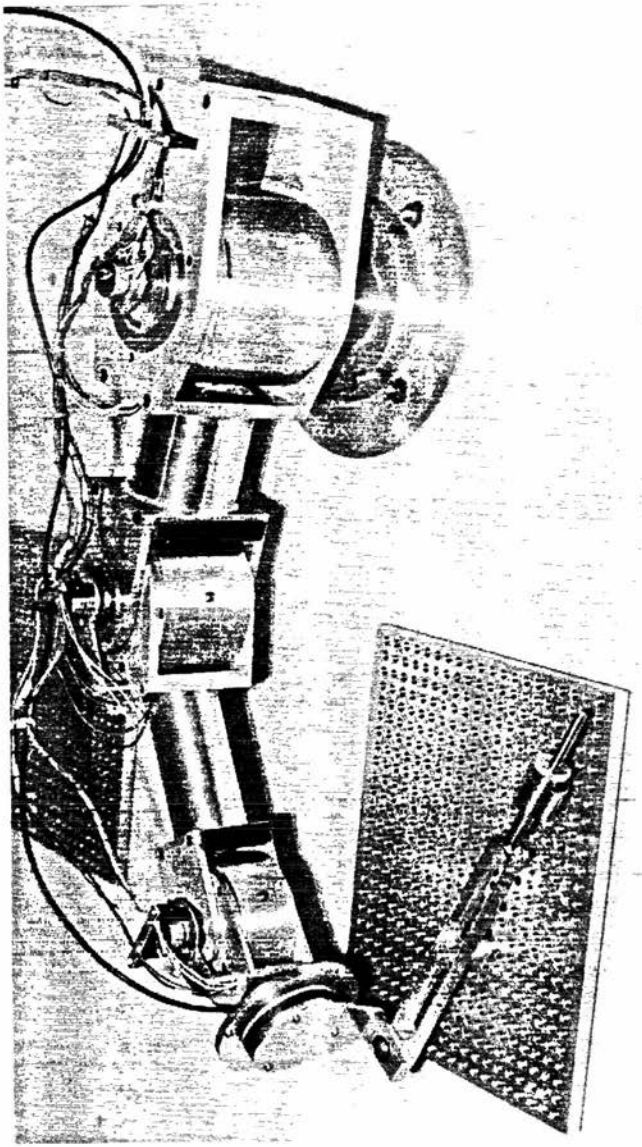


Figure 5.7: Our three degree of freedom, planar, direct-drive robot arm.

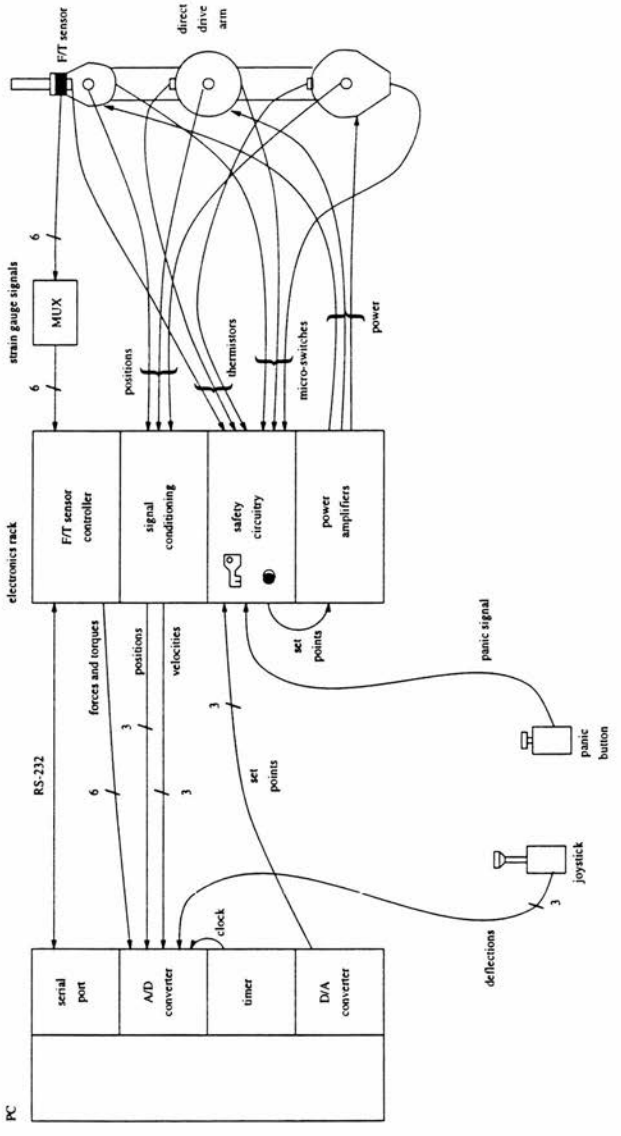


Figure 5.8: The overall design of our robot system.

feeding them through analogue bandpass filters with a resonant peak at approximately 16Hz (see section D.4). Over the speed of arm operation that we are interested in this method proved to be satisfactory.

An Assurance Technologies Incorporated six degree of freedom, mini force/torque (F/T) sensor [Little 92] is mounted at the wrist. Only three of the six degrees of freedom are relevant because we have chosen to restrict our operations to planar problems. The F/T sensor controller is programmed via an RS-232 line to perform the resolution of raw strain gauge information into axes aligned with the wrist frame of reference. It delivers these values to the controlling PC as analogue signals which are digitized using a 12 bit analogue to digital (A/D) converter.

The system also includes various safety features. The main concern is that the PC shouldn't start up with arbitrary voltages on the D/A converter causing the arm to bolt off wildly. Things are arranged such that, by default, the inputs to the power amplifiers are grounded by relays. These relays can only be switched to the position where the D/A card outputs control the power amplifiers if certain conditions are met. The first condition is that the D/A card outputs are at 0V (or thereabouts). The second is that the key lock on the front panel of the rack-mounted cage that houses the electronics is turned clockwise. The third is that the motor temperatures are within an acceptable range: thermistors inserted in the motor housings are used to monitor the motor temperatures. The switching of the relays is performed by a reset button on the front of the rack-mounted cage that houses the electronics.

The arm doesn't actually contain any mechanical brakes, mainly to keep the weight down (see Appendix D for an explanation of why this is an issue). Instead we limit the travel of the arm by using microswitches that are automatically depressed by protrusions on the arm if any joint travels beyond the prescribed working envelope. A panic button is also included as part of the overall package. If any of these are depressed, or the key in the electronics cage front panel is turned to the off position (or removed), the relays in line with the power amplifier set points are grounded. They are also grounded if the thermistors reach their switching point, approximately 80°C. Some hysteresis is included in the thermistor circuitry so that the relays in-line with the power amplifier can't be reset until the motors have cooled to below approximately

75°C.

The arm is mounted on a wooden board with a matrix of holes drilled in it. These are used to hold metal palettes with a finer grained matrix of threaded holes set at $\frac{3}{8}$" pitch between hole centres. These palettes are used to fix pieces of aluminium bar to in order to create fixtures for the robot held object to be manoeuvred into. The robot held objects are machined to have a constant diameter in the plane that the assemblies are performed, so that slight axial misalignments when they are screwed to the arm does not affect the plane of contact. The metal palettes are also used to hold the pivot of the crank during the crank-turning task described in the next section.

5.5 Putting EDDIE to the Test: Turning a Crank

In order to demonstrate that the construction of our manipulator is in fact inherently accommodating to environmental constraints we decided to implement a crank turning task. Attempts at turning a crank usually use information concerning the location of the task and the length of the crank's arm (*cf.* [Mason 81]). Due to the accommodating nature of our manipulator we were able to code the task such that no knowledge of the location of the axis of rotation, nor of the length of the crank is required. Moreover the same code is applicable whether a clockwise or an anticlockwise turning of the crank is required, all that needs to be changed is the direction of an initial push.

The strategy that achieves this behaviour is actually very simple and is summarised below:

(1) slowly increment the applied force in the direction of the x-axis until the maximum allowed force is reached.

(2) pause to allow static equilibrium to occur.

(3) slowly rotate the applied force in the direction of desired crank rotation until a velocity is sensed.

(4) repeat the following indefinitely:

   (a) push in the direction of the sensed wrist velocity.

   (b) read in the current wrist velocity.

In other words, once the crank starts moving the applied force continuously 'chases' the

sensed velocity, the velocity estimate providing a guide to where the motion freedom lies. Whilst this task is running the velocity limiting software continuously modulates the magnitude of the force commands output to the arm. Things aren't quite this simple due to noise, the setting of thresholds to overcome it, and getting DOS to 'pretend' to be a real-time system, but in essence this all that the program does. Figures 5.9 through 5.15 show an example of the behaviour of the arm using this algorithm.

The crank turning task has two degrees of freedom and so only requires a manipulator with two degrees of freedom. A manipulator with any further degrees of freedom would be kinematically redundant with respect to the task, requiring some constraint on the extra degrees of freedom [Whitney 69], [Nakamura 91]. Our arm possess three degrees of freedom, but we found that it was still possible to perform the crank turning task: as can be seen from the plot in figure 5.12 the wrist joint throughout this task was unpowered. We discovered empirically that provided the distal link was roughly aligned with the medial link, the friction in the wrist joint was sufficient to keep the wrist in roughly the same orientation. Measurement of the breakaway torques in the wrist joint showed that the stiction in the wrist joint is between 0.06 and 0.12Nm; values in the higher end of this range are obtained once the arm has been operating for a while[11]. We suspect this is due to different temperature coefficients of expansion between the motor spindle, bearings and housing, due to the different materials used[12]. Provided the maximum desired pushing force that the manipulator is allowed to exert is kept low enough (about 3.0Nm maximum) then the torques experienced by the wrist are insufficient to disturb its orientation, *i.e.* within $\pm$0.1Nm. By strapping the wrist joint so that our arm became a two degree of freedom device we were able to increase the maximum allowed desired force beyond 3.0Nm.

It can be seen that the characteristics of graphs 5.9 to 5.15 change around 31–32 seconds into the task. This is because we physically intervened and pushed the arm in the direction opposing its motion at that point. The upper plot of figure 5.9 shows that the arm changed its motion from turning the crank clockwise to turning it anticlockwise,

[11] In fact, all three joints exhibited similar characteristics.

[12] Armstrong [Armstrong 88] found that the friction in the joints of a PUMA manipulator varied with working temperature.

but soon settled into stable (*i.e.* consistent and repetitive) behaviour again.

The reason that the characteristics of the plots change when the arm changes direction is due to the fact that the arm's motion is affected by variation in its own effective inertia as the task is executed. This is the other side of the coin to making an arm accommodating to environmental constraints: it becomes sensitive to its own dynamic behaviour. Prentis [Prentis 80, §5.10] shows that for a four bar-linkage the effective inertia of the mechanism is dependent on its configuration. If we assume that the wrist joint of our arm remains fixed with respect to the medial link, then our arm becomes a four-bar linkage. There are two minima in the effective inertia of this mechanism, and they occur when the arm of the crank is aligned with the distal/medial link combination of the robot mechanism. Qualitatively these can be understood as the points in the arm's trajectory where minimal amounts of motion are required of the arm's mass—in particular the proximal joint and link. Between these minima occur local maxima in the mechanism's effective inertia.

The data plotted in figures 5.9 to 5.15 were generated with the arm in a right-handed configuration. It can be seen from the plots that the arm 'prefers' clockwise rotation of the crank under these circumstances, in as much as the behaviour is smoother than turning the crank anticlockwise. For a left-handed arm configuration the preferred sense of crank rotation is anticlockwise, as can be seen from 5.17, and by comparing figures 5.15 and 5.16. For the left-handed arm configuration, as before, the arm started rotating the crank in a clockwise sense and was manually disturbed into an anticlockwise motion about 30 seconds into the task. Based on Prentis' analysis of a four-bar linkage we conclude that the preferred sense of rotation is due to the rate of change of the mechanism's effective inertia. This is something we would like to investigate further in the future. In order to provide some extra stability to the crank turning motion we included the ability to add weight to the crank arm so that the robot experienced a smaller percentage change in effective inertia, because of a sort of flywheel effect. This weight can be seen in figure 5.7.

Just as the figures below show that we are able to change the sense of rotation of the crank whilst the program is running, and the arm will continue to perform the task successfully, we are also able to change the arm configuration, and/or crank length by

manual intervention, and again the system's performance is not detrimentally affected.

The data in figures 5.9 to 5.17 were generated with the pivot of the crank positioned at approximately (-0.198, 0.364) metres in base frame coordinates, and with a crank arm length of approximately 6.0cm.

5.6 Some Comments on Torque Balancing

In section 4.5.1 we considered the technique of balancing a torque induced on an object due to the reaction force at a point of contact. From the models of robot dynamics presented in this chapter so far this scheme would at first sight appear to be a marginally stable closed loop system: if we use an operational space dynamics formulation (see for example [Khatib 87], [Craig 89]) we can represent this situation with the diagram of figure 5.18, where $M_\theta(\mathbf{q})$ is the rotational inertia of the arm in Cartesian space, $\frac{1}{s^2}$ represents a double integration process, ${}^w\tau_z(\cdot)$ represents equation 4.24, H represents the feedback gain which we can initially assume to be given by $H = 1$, θ_0 is the wrist orientation (see equation B.8), and τ_d, τ_e, τ_a, τ_m are the desired, error, actual and measured torques about the z-axis, respectively. We ignore the velocity product terms (see Appendix A) on the assumption that their effects will be negligable.

This situation is marginally stable because the roots of the characteristic equation of the closed loop system has zero real parts. Stability could be ensured if we were to add some damping to the above closed loop system model. A full model of a d.c. motor would normally include a damping term [Tal 81], however, direct-drive motors are renowned for their lack of damping [Asada & Youcef-Toumi 87]. In fact, as we have shown in figures 5.6, our velocity limiting loop has a similar effect to damping, but due to its non-linear nature it is not a trivial matter to model its effects.

The above model is also an oversimplification in a number of other respects: it does not include the dynamics of the power amplifiers, the motors, the force/torque sensor, or the filtering that we employ. If we were to model all of these effects we would arrive at a system description that could predict the system's phase margin. However, due to the frequency components of the command signals (fractions of Hertz) instability due to phase reversal is unlikely to be an issue. Should the gain of the feedback path ever

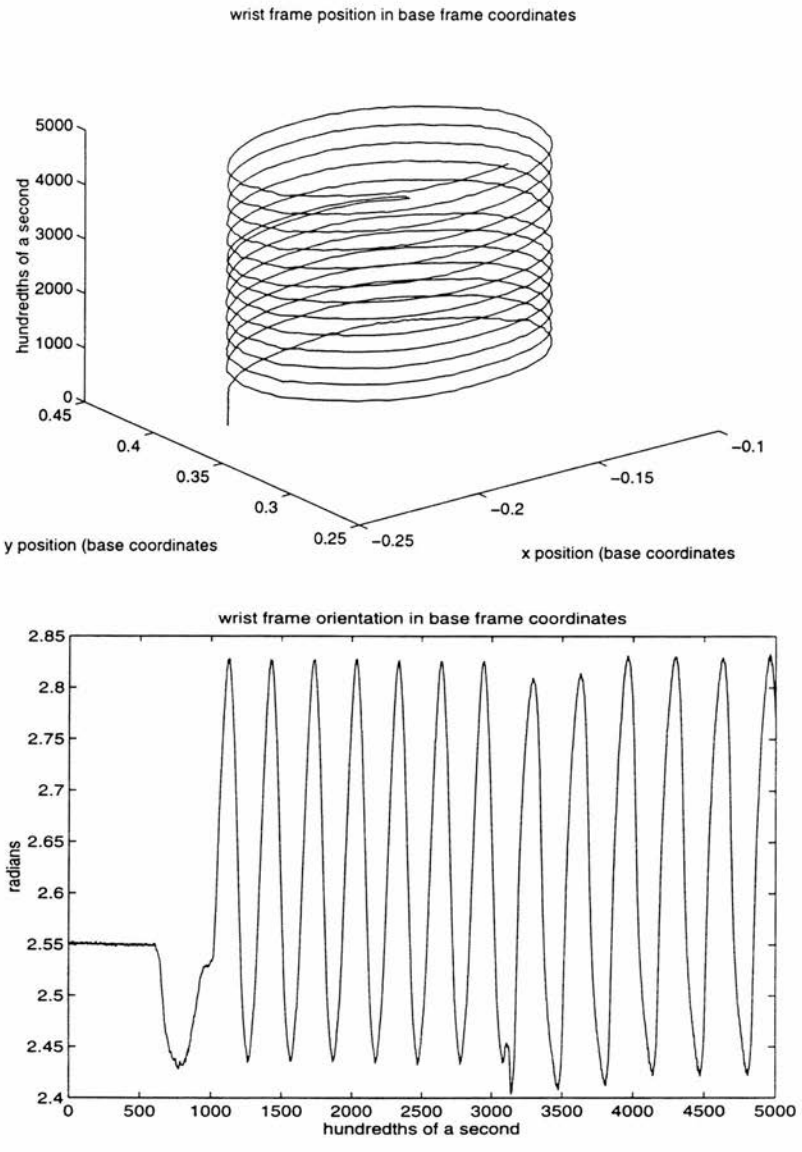
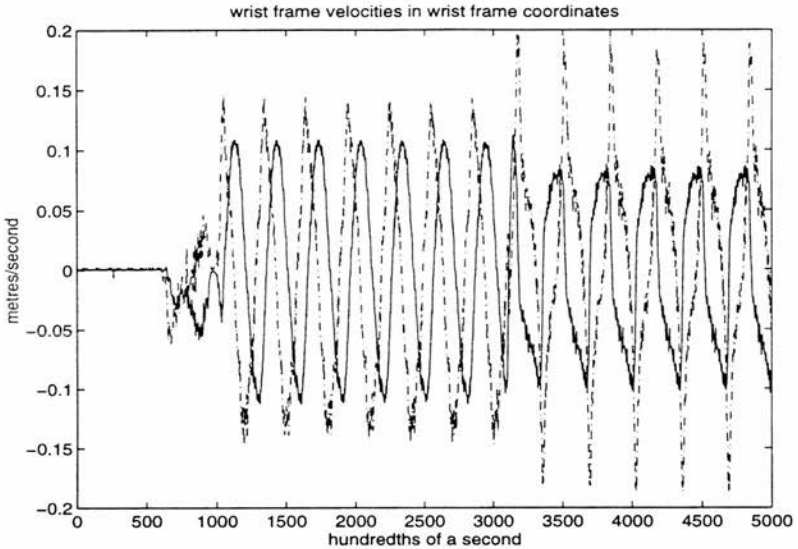


Figure 5.9: Example wrist frame position data generated by our direct-drive arm turning a crank in a right-handed arm configuration.



Solid line shows the x-component, broken line the y-component.

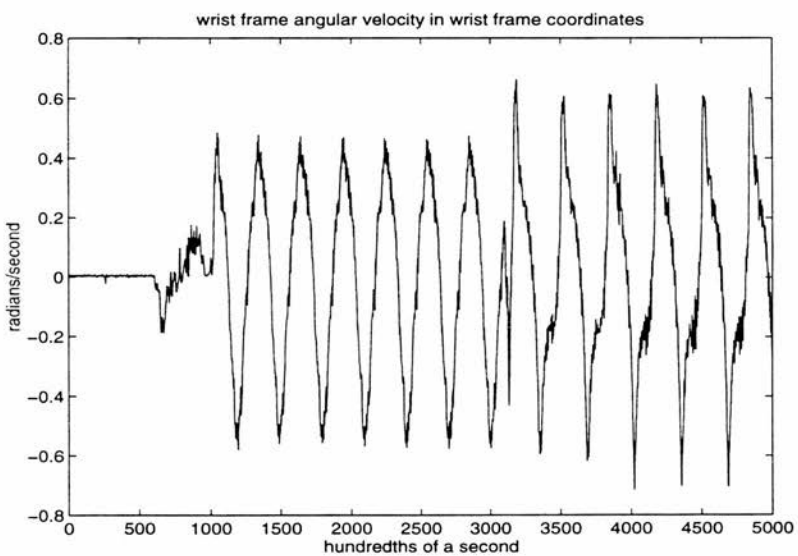
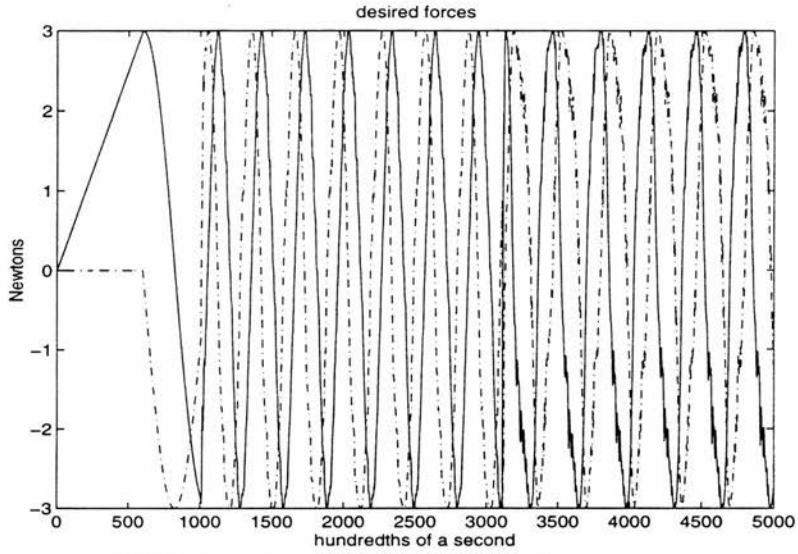


Figure 5.10: Example wrist frame velocity data generated by our direct-drive arm turning a crank in a right-handed arm configuration.



Solid line shows the x-component, broken line the y-component.

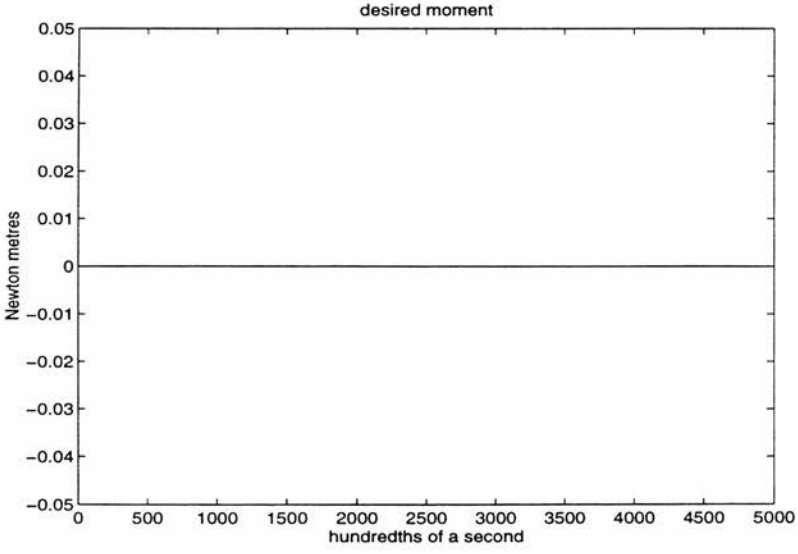


Figure 5.11: Example desired wrench data generated by our direct-drive arm turning a crank in a right-handed arm configuration.

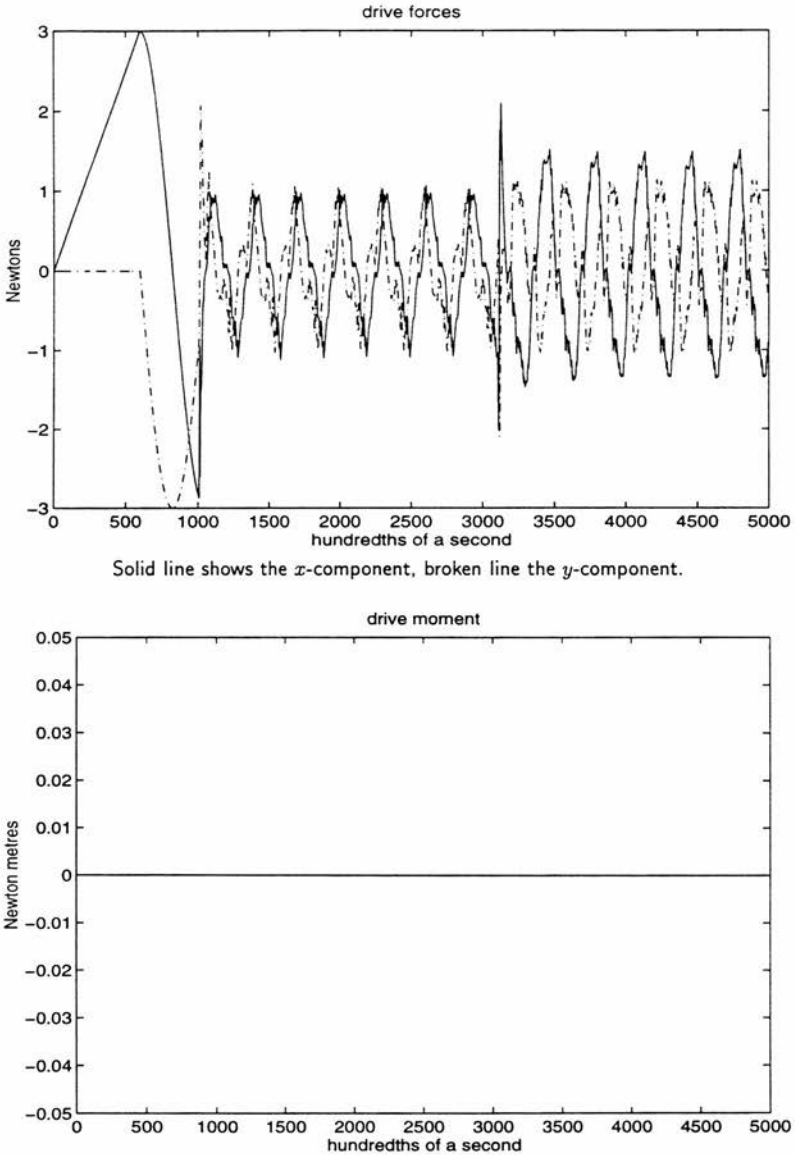


Figure 5.12: Example drive wrench data generated by our direct-drive arm turning a crank in a right-handed arm configuration.

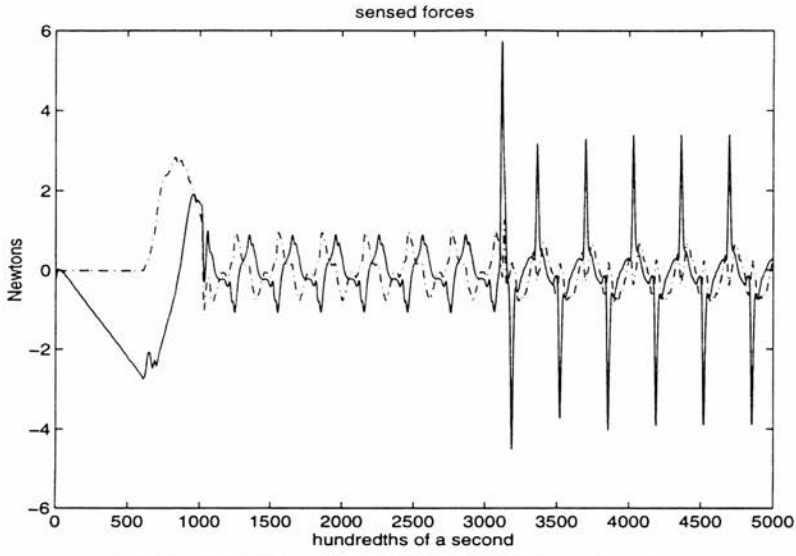


Figure 5.13: Example sensed wrench data generated by our direct-drive arm turning a crank in a right-handed arm configuration.

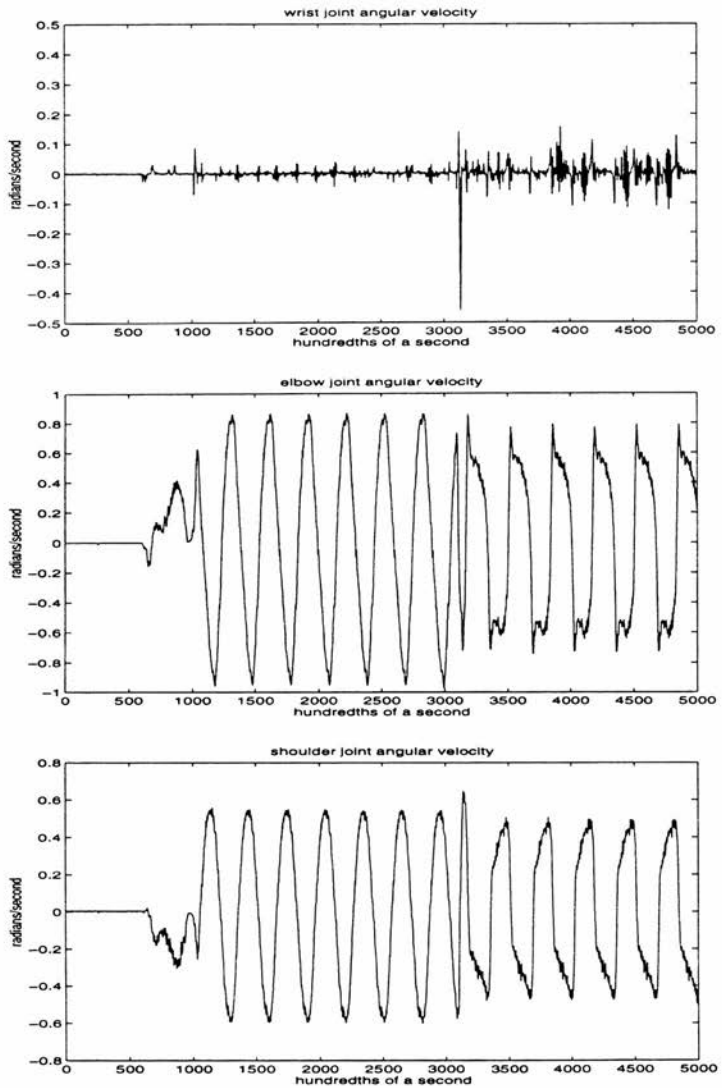


Figure 5.14: Example joint velocity data generated by our direct-drive arm turning a crank in a right-handed arm configuration.

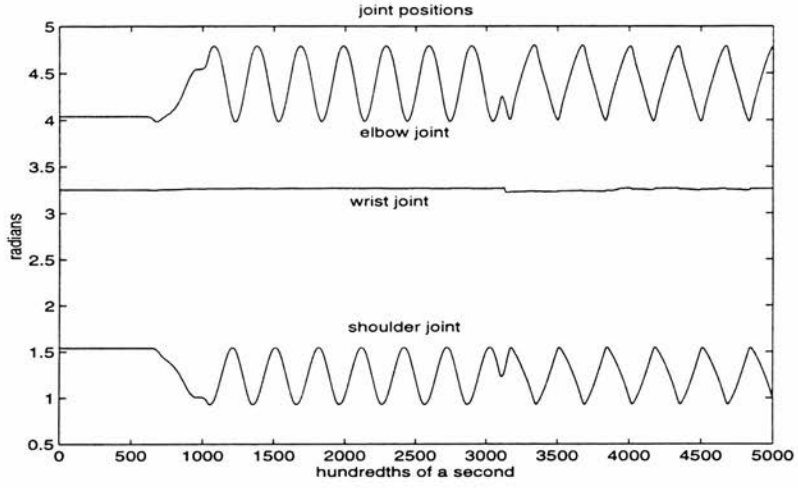


Figure 5.15: Example joint position data generated by our direct-drive arm turning a crank in a right-handed arm configuration.

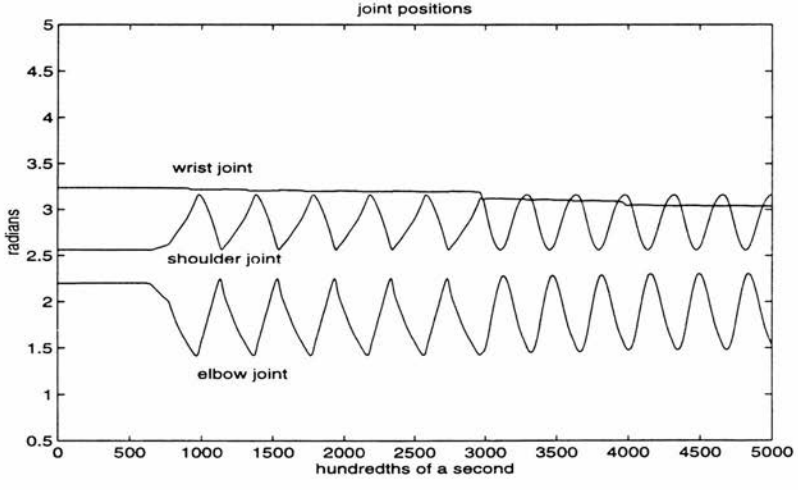


Figure 5.16: Example joint position data generated by our direct-drive arm turning a crank in a left-handed arm configuration.

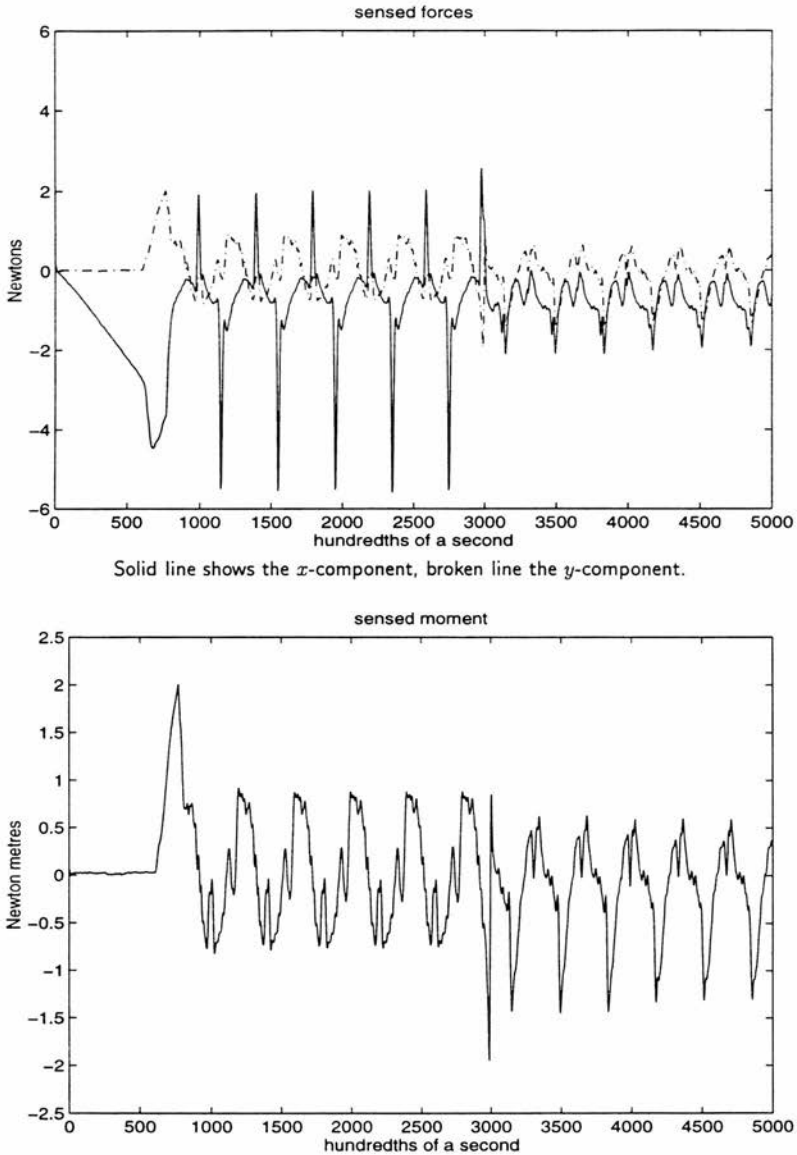


Figure 5.17: Example sensed wrench data generated by our direct-drive arm turning a crank in a left-handed arm configuration.

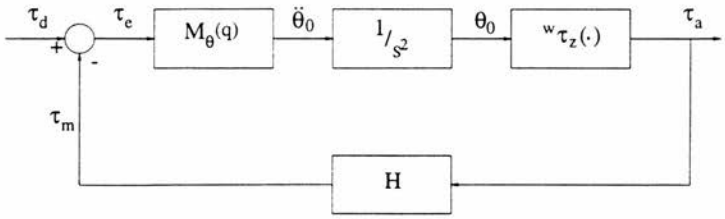


Figure 5.18: Simplified closed loop representation of torque balancing in operational space.

be greater than unity, however, then we could expect the control system to saturate to either the maximum or minumum allowed torque values. Measurements show that in fact the gain of the feedback path is approximately given by $|H| = 0.9$, most likely due to the fact that the assumptions required by equation D.38 are not entirely satisfied, but also in part due to the various filtering techniques employed in 'cleaning up' the torque signal. In effect the value $\frac{1}{0.9}$ is a measure of the stability margin for this closed loop system.

We would note here that the fact that the torque balancing tends to work in practice, despite $|H| < 1$,[13] may be due to the stiction in the wrist joint, that is so effective in stabilising the crank turning task (under the right conditions—see above). We would note that this effect warrants the inclusion of another component in the dynamics equations describing the behaviour of a robot arm: equation 5.19 should be augmented with a non-linear term that represents the effects of non-linear joint friction, *i.e.*

$$\tau = \mathbf{M}(\mathbf{q})\,\ddot{\mathbf{q}} + \mathbf{C}(\mathbf{q},\dot{\mathbf{q}})\,\dot{\mathbf{q}} + \mathbf{G}(\mathbf{q}) + \mathbf{J}^T(\mathbf{q})\,\mathbf{w} + \mathbf{N}(\mathbf{q},\dot{\mathbf{q}},\ddot{\mathbf{q}}) \tag{5.21}$$

There are a few research works that discuss the different forms that the nonlinear term $\mathbf{N}(\mathbf{q},\dot{\mathbf{q}},\ddot{\mathbf{q}})$ can take, for example [Gogoussis & Donath 87], [Gogoussis & Donath 88], [Armstrong 88], [Dupont 90].

We decided to adopt the use of torque balancing due to empirical observation when operating the arm under joystick control. Had we approached this problem from a text book perspective we would probably never have considered this option; as far as we are

[13] It would of course be straightforward to increase the value of $|H|$ in software should the need arise.

aware no robotics text book describes the effects of joint friction on robot performance that we encountered. In addition, it appears that those that have studied these effects do so in order to reduce them, see for example [Canudas *et al.* 87], [Armstrong 88]. Some might consider it just luck that these normally unmodelled frictional effects have worked in our favour. We believe that it illustrates the utility of experimenting with real hardware.

5.7 Summary

In this chapter we have argued that manipulators designed to be position controlled are fundamentally unsuited to performing motion in contact. This is true both at the command level, since it is virtually impossible to command a position controlled device to follow a surface (see Chapter 4), and at the mechanical level, since the inertial properties of standard industrial manipulators mean that their natural behaviour opposes accommodating environmental constraints. We have designed and constructed a three degree of freedom, planar manipulator that avoids these limitations, and demonstrated its accommodating nature on a crank turning task. In the next chapter we will exploit the capabilities of this manipulator in the realisation of task-invariant assembly strategies.

Chapter 6

Experimental Results

In order to establish the viability of our approach to robotic assembly we decided to test it empirically on some representative (planar) tasks. We have programmed our custom designed planar direct-drive arm to perform the insertion of a peg into a hole and the rotation of a block into a corner based on the analysis of Chapter 4. Interestingly the peg in hole program turned out to be so general that it is able to perform a dual-peg-in-hole assembly without any modification!

As mentioned in Chapter 5 the sample rate of the system is 100Hz. When the program looks for a discontinuity in the wrench readings the time interval used is 10ms, *i.e.* a difference is taken 10 sample instances apart. This is done to provide a certain amount of noise immunity. In order to detect discontinuities we basically have to set thresholds. If we are looking for a signal change of, say, greater than 1Nm per second, this would be equivalent to looking for a change in the signal of 0.01Nm per 10ms, or 0.1Nm per 100ms. If any one reading is likely to be off by 0.005Nm, say, due to the effects of noise, then obviously looking for a change of 0.1Nm is less susceptible to being corrupted by noise than looking for a change of 0.01Nm.

A number of parameters were varied in the experiments to demonstrate the invariant nature of our strategies. One of these parameters was the material that the robot held objects were fabricated from. This was in order to get some variation in the coefficient of friction. The environmental constraints that the robot held objects interacted with were always made of aluminium. We took some measurements of the coefficient of friction between the various robot held objects and the environmental constraints, the

robot held object material	friction cone half-angle
brass	21^o
aluminium	23^o
bright mild steel	24^o
ramon wood	29^o

Table 6.1: Estimated friction cone half-angles for the materials used in the assembly experiments.

results of which are shown in table 6.1. A handful of measurements were taken and the results averaged, but these numbers should not be interpreted too literally because, as noted before, it is very difficult to get an accurate measure of a friction coefficient. We include these numbers so that the reader can get a 'feel' for the kind of variation that our system operated under.

6.1 The Task-Level Software Structure

At the highest level the software is a state-interpreter. In order to perform an assembly task this interpreter needs to be provided with what is effectively a 'data file' for the particular task at hand. This data file is in fact just the state definitions that a task is composed of. A state definition consists of an identifier and an "action procedure" that 'decides' what to do, based on the current sensory information. This decision is implicitly taken within the context that defines the state. The general structure of an action procedure is as follows:

- If the wrist is moving then
  - If an expected sensory discontinuity is detected then
    - Perform the state transition signalled by this discontinuity (by changing the value of the current state)
- If the wrist is stationary then
  - Incrementally change the components of the desired wrench (in order to achieve some desired motion).

Notice that an action procedure is divided into two parts according to whether there is any detected motion of the wrist frame or not. We will borrow some terminology from

biological motor control to distinguish these two possibilities [Greer 84]. We will term situations where there is no motion as *isometric*, and situations where there is motion as *isotonic*. Isometric literally means constant distance: in the field of biological motor control this refers to the length of muscle fibres, and describes the situation where the fibres build up tension, but do not actually shorten. Isotonic means constant tension, though it is used in biological motor control to refer to the situation where the tension in a set of muscle fibres is greater than that opposing their motion. In our situation we can use these terms literally: during an isometric situation the arm is stationary whilst it changes the desired wrench in some way until there is motion, *i.e.* there is constant position, but change in force/torque; during an isotonic situation the desired wrench remains constant for as long as there is motion, *i.e.* constant force/torque, but changing position. We use sensed motion to signal the end of an isometric phase, and a discontinuity in the sensed wrench to signal the end of an isotonic phase.

As a safety precaution we also check to see that contact has been maintained. If it hasn't then we terminate the assembly task, and inhibit power to the robot arm, in case the arm is flying off wildly. Since we are dealing with a prototype device we consider it prudent to err on the side of caution in these matters.

There is actually only a limited number of things that an action procedure can choose to do: it can either change the value of the current state or the current value of the desired wrench. Ultimately any change to the desired wrench will amount to either incrementing, maintaining, or decrementing individual components of the desired wrench, but the 'intent' of these changes vary. For example, we might be interested in the ratio between individual components of the desired wrench, or we could treat the desired force as a magnitude and direction and increment or decrement either, or both, of these, thus 'indirectly' affecting the x and y components of the desired wrench.

As a consequence of this data structure all the state interpreter has to do is to continuously execute the following steps:

(1) read in the current sensory data;

(2) invoke the action procedure associated with the current state in order to change the current value of the desired wrench, or to change the value of the current state;

(3) convert the current value of the desired wrench to desired joint torques, and send these to the joint current regulation loops.

Since our assembly strategies are dependent on applying and sensing forces and torques, we require that the objects involved are touching. Consequently, we employ an initialisation stage to ensure that the objects are touching in a known relationship once the software starts to 'navigate' a task. At the moment, all this means in practice, is that an initial push is built up incrementally that results in static equilibrium in a known contact relationship. This is much the same as the crank turning task described in section 5.5. If any motion is detected once this initial push has reached its full magnitude the software assumes that something is wrong, *e.g.* the expected contact configuration has not been achieved, and the task is aborted.

At the other end of the scale, each task has a goal state defined that does nothing except register the fact that the goal has been achieved.

6.2 Rotating a Block into a Corner

Although conceptually a simple task we have been unable to find a description of any previous practical work concerned with rotating a block into a corner. In [Asada 90] mention is made of a block palletizing task, but there is no explicit consideration of rotating the block in the corner. The task of placing a cylinder into the corner of a box is used in [Asada & Izumi 89], in a system that attempts to automatically deduce a hybrid force/position assembly strategy by observing a human operator[1]. In our planar domain the corresponding task would be the placing of a disc in a corner, which is a simpler task because there is no rotational constraint on the goal configuration.

The strategy we have developed actually arose through experimentation with our robot arm under joystick control. The analysis of section 4.5.4 is a *post hoc* analysis where we attempt to derive the limitations on our strategy.

The block in corner data file consists of procedures that describe how to vary the applied wrench in three separate situations. These situations are described in terms

[1] Interestingly they used a direct-drive robot arm as the teaching pendant because of its backdrivability.

of the possible contact configurations that could be current. We will use the labels of the icons in figure 3.4 as a shorthand to describe these situations. It will become apparent as we describe the current implementation that it does not associate actions with all of the contact configurations of figure 3.4. This is partly due to the robustness of the behaviour that we have implemented, and partly due to time limitations. We intend to extend the range of operation of this program in the near future. One of the benefits of the approach described in this thesis is the ability to incrementally build an assembly strategy.

Before anything else can happen the system interrogates the user for the goal orientation. This is specified by taking the robot held block, placing it in the corner, and notifying the program to remember this orientation. This information is used in determining the sense of reorientation required, as a way of determining when to switch from an applied torque to an induced torque, and as a way of disambiguating the significance of a sensory discontinuity[2]. Although knowledge of the orientation is metric information it is not used in a way that requires precise knowledge of its value. This is just as well due to the low quality position information available to the system (see section D.4) and the fact that there is some compliant material included as part of the arm's distal link in order to protect the force/torque sensor. This is shown in the side view of the end-point of our arm in figure 6.1. The compliant material between the force/torque sensor and the robot-held object is visco-elastic poly-sulphide rubber, which is probably less compliant than an average eraser. At the end of section 6.3 we present some data from an experiment where there was no compliance in the wrist in order to demonstrate that it is the accommodating nature of our arm that allows us to perform assemblies via sliding motions rather than any passive compliance in the arm's construction.

For the sake of simplicity we will describe the task as though we are trying to get the block into orientation $C9$ of figure 3.4, *i.e.* a clockwise reorientation of the block. Because we tell the system the goal orientation the same program code is able to perform reorientation of the block in either a clockwise or anticlockwise sense, due to the symmetry of the task, by simply reversing the polarity of various parameters.

[2] We now believe that this is necessary due to a shortcoming in our program concerning the rate at which the applied moment is zeroed, see page 129 for details.

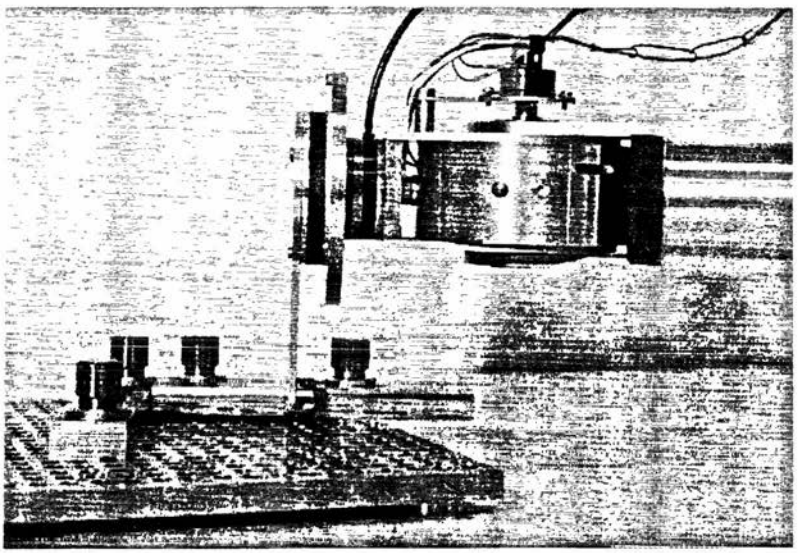


Figure 6.1: A side view of the end-point of our robot arm with a 2" long aluminium block attached.

After the initialisation step, where the arm builds up a push along the x-axis of frame w in increments of 0.005N per sample instant, the program assumes that corners c_1 and c_4 (see figure 4.1), are touching the environmental constraint surfaces. We would note here that if the block is oriented such that the centre of compliance is to the 'left' of the normal at contact c_1 (see figure 4.12), then a torque would be induced that would act to reorient the block in the opposite sense to that which we require. This motion, however, would require violating the kinematic constraint at c_4.

As mentioned in Chapter 3 the state definitions that we will use are composed of the possible contact configurations that the objects involved in the task could possibly be in. Each of these states has an action procedure associated with it. These action procedures are described below. All of the described changes to the applied wrench are made in 'small' increments. After each small change the flow of control goes to the state interpreter, which arranges for the motors to be driven appropriately. Large changes are produced by the integration of these small changes over time. Only one

rule is executed each time an action procedure is called: the result can either be a change in the desired wrench (including a null change), or a change in the definition of the current state.

All references to 'orientation' in what follows refer to the block orientation, which is the same as the orientation of the robot's distal link, and is given by equation B.8. The uncertainty in orientation information is catered for by specifying a small number of degrees[3], ϵ, such that any orientation measurement θ can actually be anywhere in the range $\theta \pm \epsilon$. During the experiments presented here ϵ was set to 1.5^o. This means a tolerance interval of 3^o. For most systems this would probably be an unacceptable level of ambiguity. This is not a problem for our system since we are not using this information to servo on, merely to decide what force application strategy to apply next. Achievement of the goal orientation is actually determined by the environmental constraint, which is detected by a discontinuity in the sensed moment.

The state definitions for the block in corner task are as follows:–

C0 There is no contact. For safety purposes if no contact is experienced the arm is powered down.

- The task terminates.

C3C8C9 The block is somewhere in the corner in either contact configuration $C3$, $C8$, or $C9$. The general aim of this state is to rotate the block towards the goal orientation whilst maintaining two point contact.

- If there is no contact then
  - the current state is set to $C0$
- If there is no motion then
  - if the difference between the current orientation and the goal is between -45^o and -90^o degrees then
    - if the applied force is pointing in the direction of the positive x-axis of the frame of reference w then
      - decrement the moment component of the desired wrench

[3] Although this value is specified in degrees the reader should be aware that all the angular measurements plotted as results are in radians.

else

* rotate the direction of the desired force towards the x-axis of frame w

- if the difference between the goal orientation and the current orientation is between $-\epsilon$ and -45^o then
  * if the direction of the applied force is greater than or equal to 90^o from the goal orientation then
    * the current state is set to $C1C3$

    else
    * rotate the applied force in a positive sense (in order to induce a turning moment on the block)
- if the difference between the goal orientation and the current orientation is between $-\epsilon$ and ϵ then
  * the current state is set to $C4C9$
- if the difference between the goal orientation and the current orientation is greater than ϵ then
  * the current state is set to $C10C12$

- If there is motion then
  - if the orientation is between $-\epsilon$ and -45^o then
    * if there is an applied moment then
      * null the applied moment (because we want static equilibrium to hold before inducing a torque)
    * if a discontinuity is detected in the sensed moment then
      * the current state is set to $C4C9$

C1C3 Could either be in contact configuration $C1$ or $C3$. The general aim of this state is to align edge e_1 of the block with the 'horizontal' environmental constraint.

- If there is no contact then
  - the current state is set to $C0$
- If there is no motion then
  - if the direction of the applied force is less than 90^o from the goal orientation then
    * reorient the applied force in a positive sense (in order to induce a

turning moment on the block)

- if the direction of the applied force is $90^o \pm \epsilon$ to the goal orientation then
  - increment the magnitude of the applied force, providing it remains below a prespecified maximum

  else
  - terminate the task

- If there is motion then
  - if a discontinuity is detected in the sensed moment then
    - the current state is set to $C4C9$

C4C9 Could either be in contact configurations $C4$ or $C9$. The general aim of this state is to rule out the possibility of being in contact configuration $C4$.

- If there is no contact then
  - the current state is set to $C0$
- If there is no motion then
  - if the current direction of the applied force is not along the x-axis of frame w then
    - rotate the direction of the desired force towards the x-axis of frame w
  - if the current orientation is within $\pm\epsilon$ of the goal orientation then
    - the current state is set to $C9$
  - if the current orientation is less than $-\epsilon$ then
    - the current state is set to $C3C8C9$
  - if the current orientation is greater than ϵ then
    - the current state is set to $C10C12$

C10C12 The block has overshot the goal orientation.

At the moment there is no action associated with this situation other than to signal its occurrence and terminate. This situation occurs very rarely (for reasons discussed below), and when it does it is usually a misinterpretation of the situation due to a poor orientation measurement rather than the block having actually overshot the goal.

- The task terminates.

C9 The goal configuration.

- The task terminates.

Note that there are a number of situations where the action procedure associated with a state has no rule to cover some set of likely conditions. For example, there is no explicit provision for what to do in state $C4C9$ if there is any motion. During these situations it is implicitly the case that the best thing to do is nothing. Doing nothing is tantamount to deciding to maintain the current desired wrench.

The software outlined above was used in conjunction with our custom designed robot arm to perform the rotation of a block into a corner under a number of different circumstances. Figures 6.2 to 6.14 illustrate some representative data. The parameters varied were the location of the corner constraint, the material that the block was constructed from, the length of the block, the sense of reorientation required, and the left or right handedness of the arm. The dimensions of the blocks given in the figure captions are described by the length of the block sides parallel to the x-axis of frame w times the length of the sides parallel with the y-axis of frame w.

These experiments were performed on two separate corner constructions: location 1 refers to a construction where the location of the vertex of the corner was at a location of approximately (0.288, 0.331) metres in base frame coordinates, and location 2 corresponds to a corner with a vertex lying at approximately (-0.122, 0.402) metres in base frame coordinates. The data was collected as the task proceeded in a data structure that is a dynamic variable, and the information written to disk once the task was finished, because we were unable to write the information to disk as the task proceeded whilst sustaining a sample rate that maintained the controllability of the arm. Which state the program is in is stored as an integer which is a code for that state. This coding is as follows:–

0 corresponds to the initial push that the arm exerts

1 corresponds to the initialisation being successful

2 corresponds to state $C0$, *i.e.* no contact

3 corresponds to state $C3C8C9$

4 corresponds to state $C9$, *i.e.* the goal state

5 corresponds to state $C4C9$

6 corresponds to state $C10C12$

7 corresponds to state $C1C3$

In order to get the system to perform it was necessary to set a number of thresholds. There is a threshold that is used to determine if the wrist is moving; if the magnitude of the velocity of the wrist frame of reference is above 0.004m/sec (regardless of direction) then the wrist is deemed to be moving. We would like this value to be as low as possible whilst still reliably detecting the onset of motion. We would like this value to be low so that the motion of the objects across each other is slow, thus maintaining a quasi-static character to the proceedings, including the benefit of avoiding potentially damaging high energy impacts. On the other hand we cannot allow this value to be so low that the software is continuously being 'confused' by the effects of the noise on the velocity readings.

The alignment of the block with the corner is detected by a 'discontinuity' in the sensed moment. This discontinuity arises because of the 'jump' in the length of the moment arm of the response force, as described in section 4.6. The detection of this kinematic transition was implemented by treating changes in the sensed moment with a magnitude greater than 0.25Nm/sec as being a discontinuity. Account was also taken of the sign of the change compared with the sign of any expected change. The magnitude of the sensed discontinuity for any particular instance of a task will be dependent on the rate at which the block is reorienting when it makes contact with the constraint (since we don't really live in a world governed by statics equations alone), as well as the length of the block since this determines the change in the moment arm of the environmental response force. In order to be able to detect the alignment of slow moving and short blocks we would like to set this threshold as low as possible whilst still reliably detecting the event that it signals. The lower bound on this value again being affected by noisy sensor readings. But there is also a case for making this threshold larger than the minimum feasible. Inspection of the plots of example data generated whilst rotating a block in a corner will reveal that a number of the tasks experienced a premature transition from $C3C8C9$ to $C4C9$. Since these all occur at about the point when the applied moment is removed (because the block is 45^o from

the goal orientation), it is likely that reducing the rate at which the desired moment is zeroed would also improve the situation. Time limitations have prevented us making this simple amendment.

At the end of the initial push the desired force is set at some nominal value. For the experiments documented in figures 6.2 to 6.14 this value is 5N and remains constant throughout the duration of the task. However, we have already noted that the length of the block can effect the magnitude of the discontinuity detected in the sensed moment, it also affects the magnitude of the induced torque. Figure 6.15 illustrates what happens when our system tries to rotate in a corner a block made of aluminium of dimensions $1" \times \frac{3}{4}"$ with a desired force of 5N. Figures 6.16, 6.17 and 6.18 show how the performance changes as the nominal applied force is set at 6N, 7N, and 8N respectively. It was actually the performance of our system with the 1" aluminium block that prompted us to include state $C1C3$.

Inspection of the plots of the x-y position of the wrist frame of reference in base frame coordinates of figures 6.15 through 6.17 will reveal that as the desired force is rotated away from alignment with the x-axis of frame w in a negative sense and approaches alignment with the normal at the contact at c_4, the behaviour of the block is consistent with the applied force having passed 'through' the friction cone at c_4 and the block slides away from the corner constraint. This is most pronounced in figure 6.15 and gets progressively less noticeable as the nominal desired force is increased, and if it is set to 8N does not occur at all.

A partial explanation of what happens here is that for the lower values of the applied force there is not sufficient turning moment on the block to bring it into alignment with the environmental constraint. As equation 4.14 makes clear the induced torque is a function of both the applied force and the dimensions of the block. The magnitude of the force required to turn the block will be dependent on factors such as stiction in the robot joints and the inertia of the arm. Our quasi-static model of the interaction between the objects says nothing about these phenomena. State $C4C9$ increments the magnitude of the applied force once its direction is normal to the constraint surface. In this way we again perform a 'search' for the conditions that will lead to the behaviour that we desire.

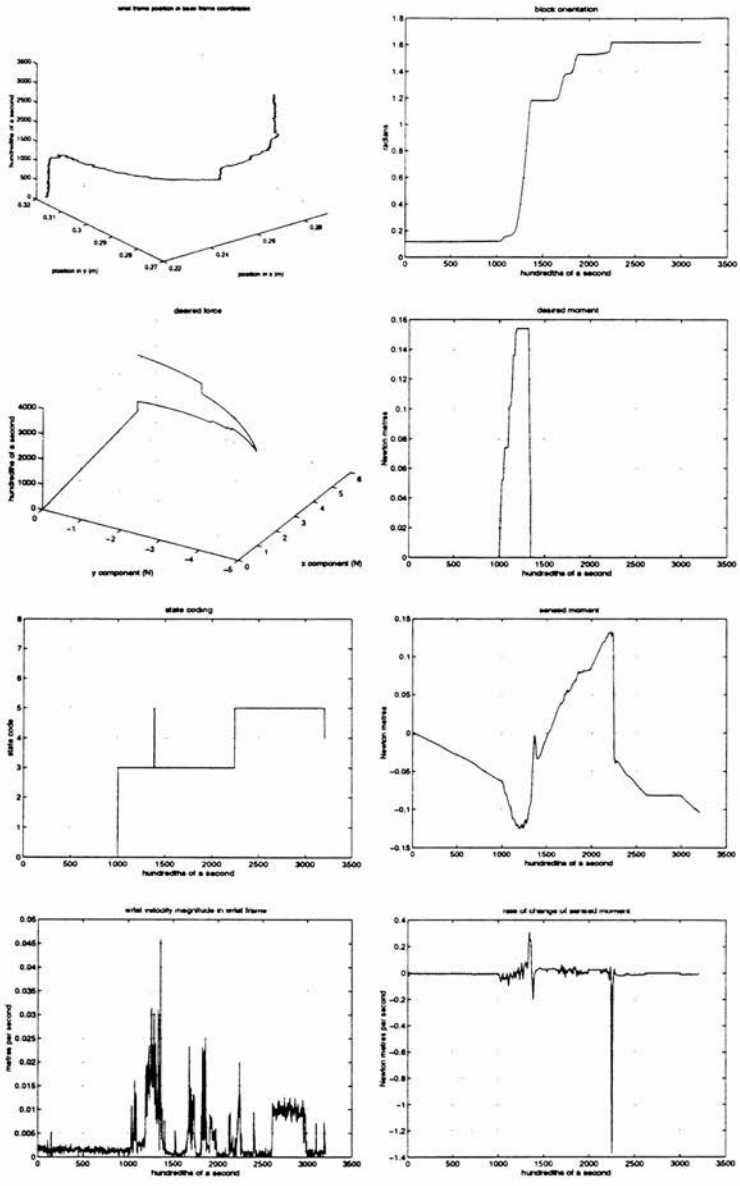


Figure 6.2: Example data generated rotating a block in a corner. Parameters: block, $2" \times \frac{3}{4}"$ aluminium; arm configuration, right handed; reorientation sense, anticlockwise; location 1; initial orientation, 0.12 rads; goal orientation, 1.62 rads.

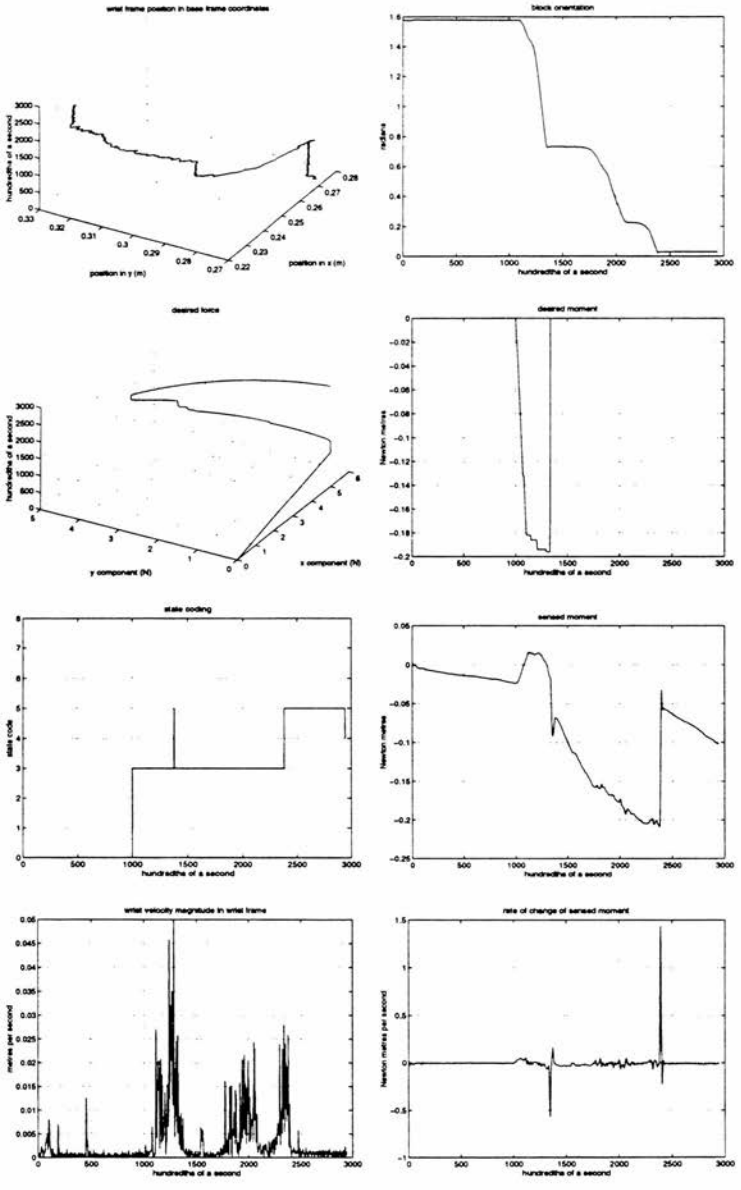


Figure 6.3: Example data generated rotating a block in a corner. Parameters: block, 2" $\times \frac{3}{4}$" aluminium; arm configuration, right-handed; reorientation sense, clockwise; location 1; initial orientation, 1.58 rads; goal orientation, 0.03 rads.

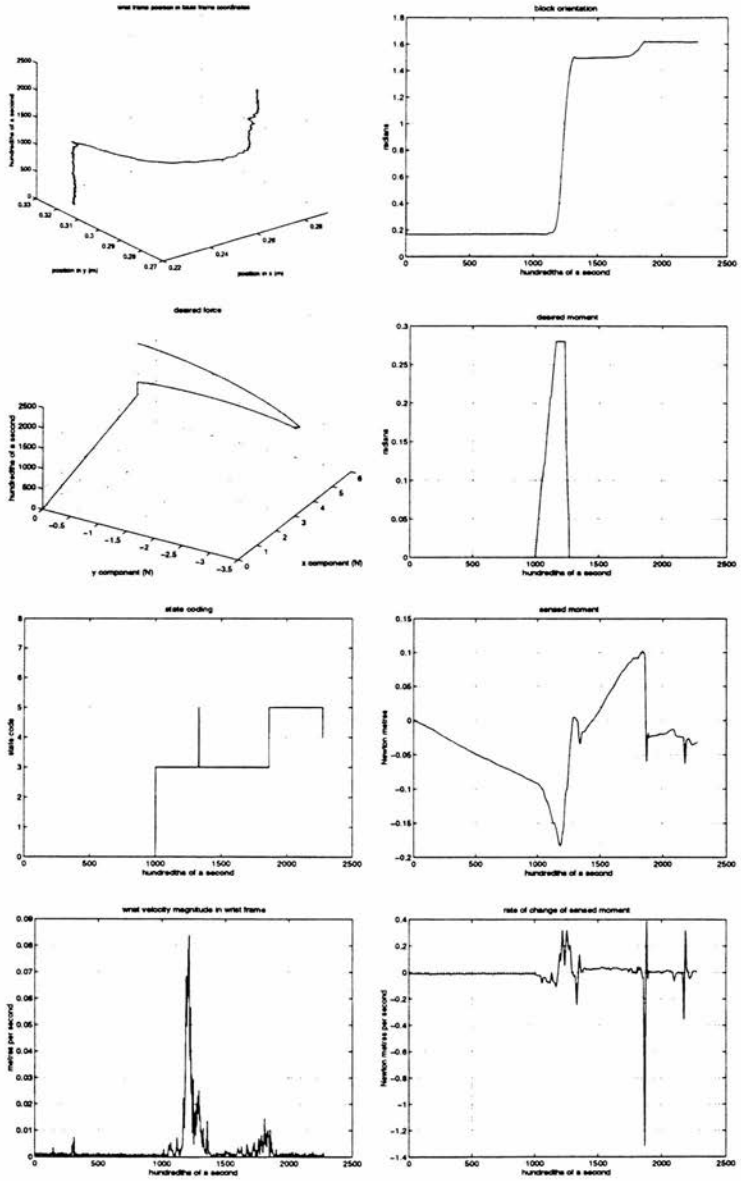


Figure 6.4: Example data generated rotating a block in a corner. Parameters: block, 2" × $\frac{3}{4}$" aluminium; arm configuration, left-handed; reorientation sense, anticlockwise; location 1; initial orientation, 0.11 rads; goal orientation, 1.55 rads.

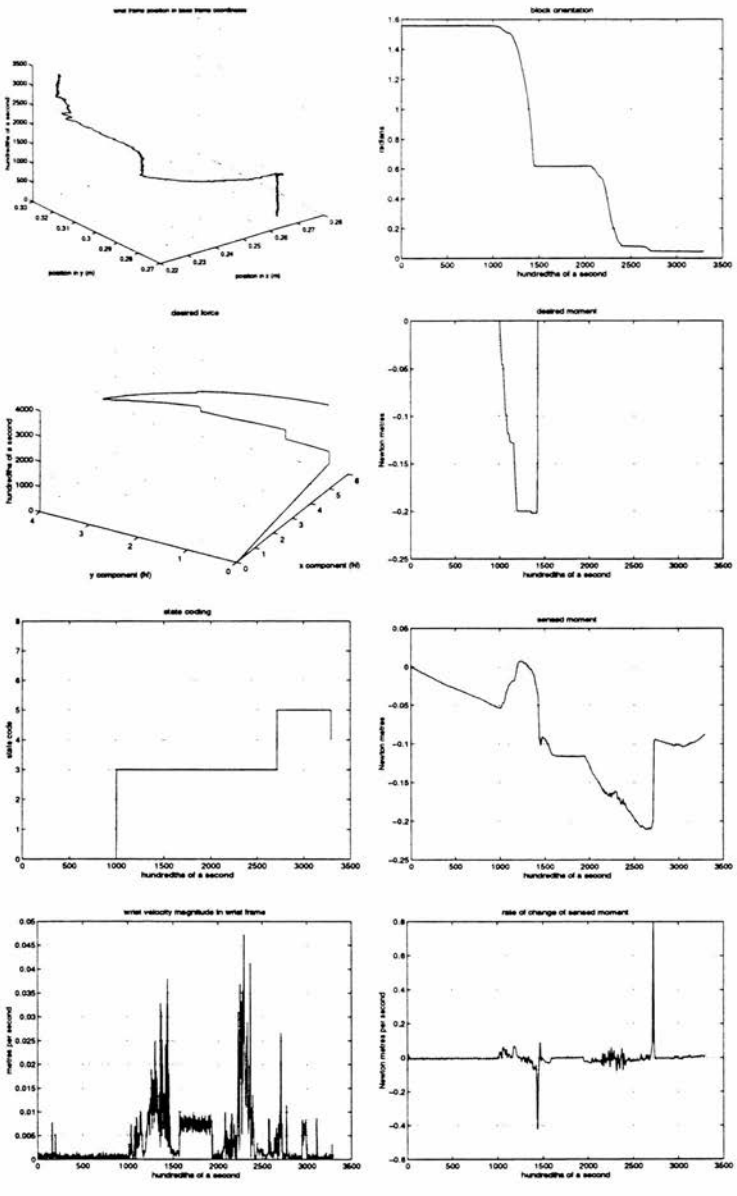


Figure 6.5: Example data generated rotating a block in a corner. Parameters: block, 2" $\times \frac{3}{4}$" aluminium; arm configuration, left-handed; reorientation sense, clockwise; location 1; initial orientation, 1.56 rads; goal orientation, 0.052 rads.

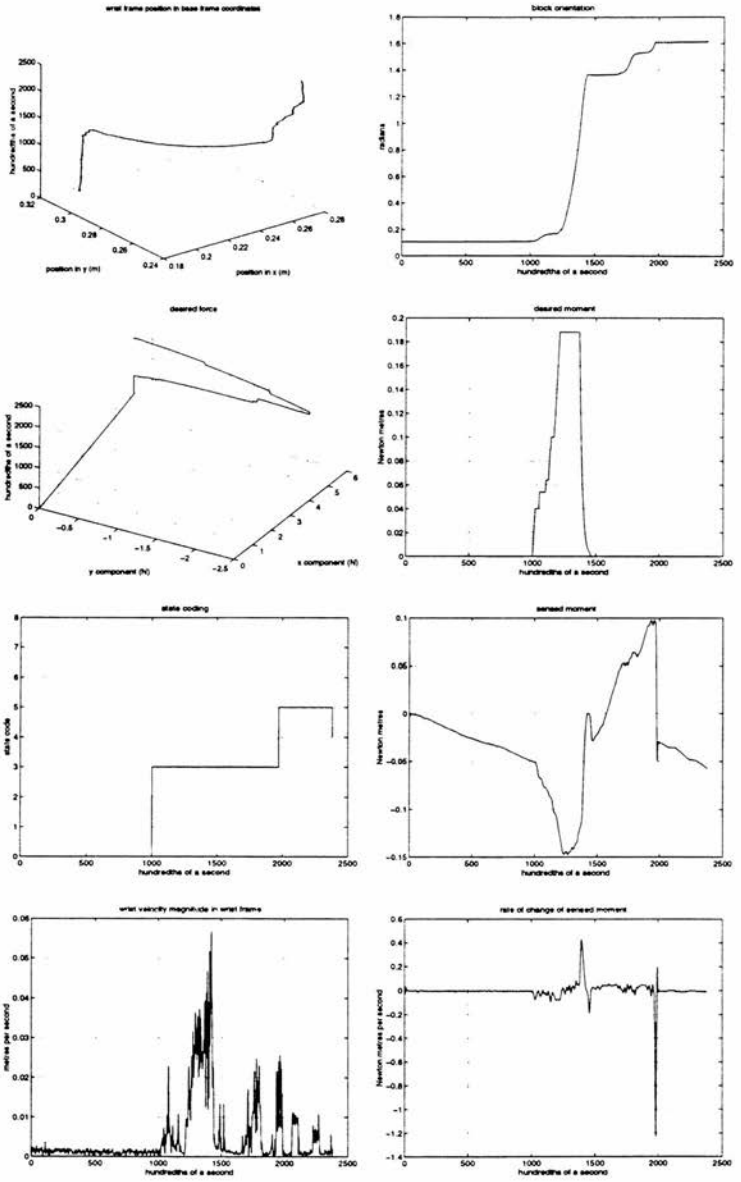


Figure 6.6: Example data generated rotating a block in a corner. Parameters: block, $3" \times \frac{3}{4}"$ aluminium; arm configuration, right handed; reorientation sense, anticlockwise; location 1; initial orientation, 0.11 rads; goal orientation, 1.62 rads.

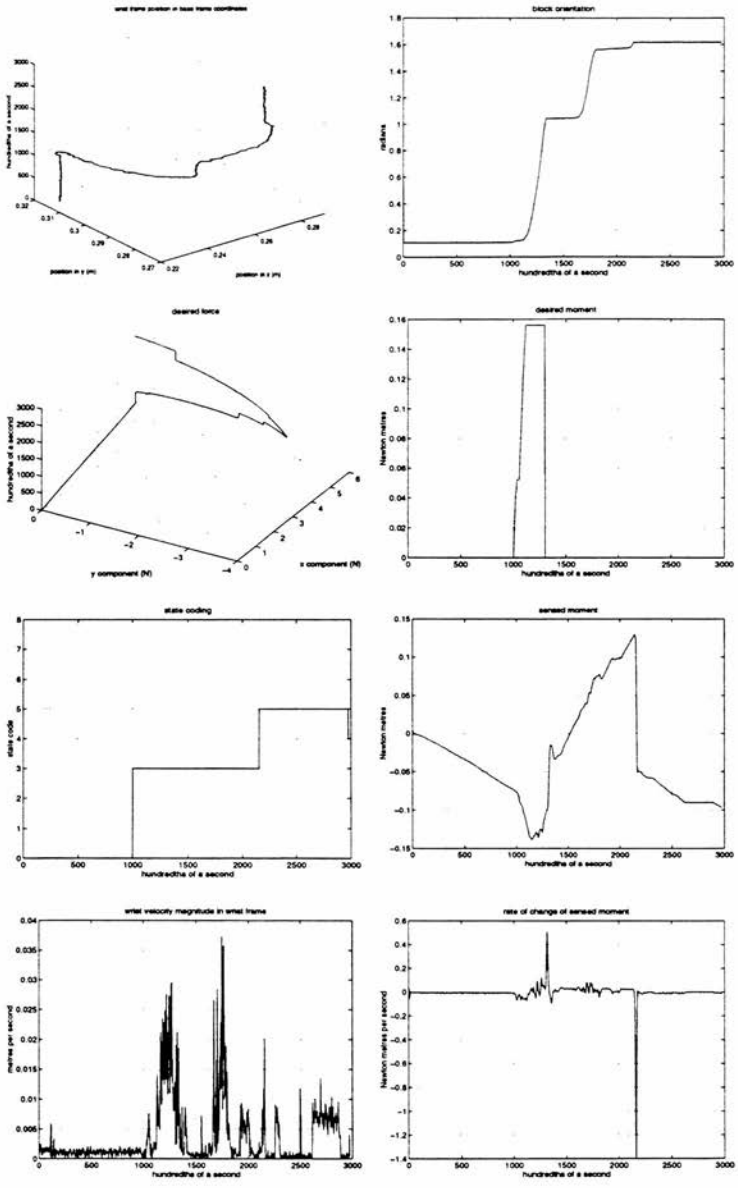


Figure 6.7: Example data generated rotating a block in a corner. Parameters: block, 2" $\times$1" aluminium; arm configuration, right-handed; reorientation sense, anticlockwise; location 1; initial orientation, 0.11 rads; goal orientation, 1.62 rads.

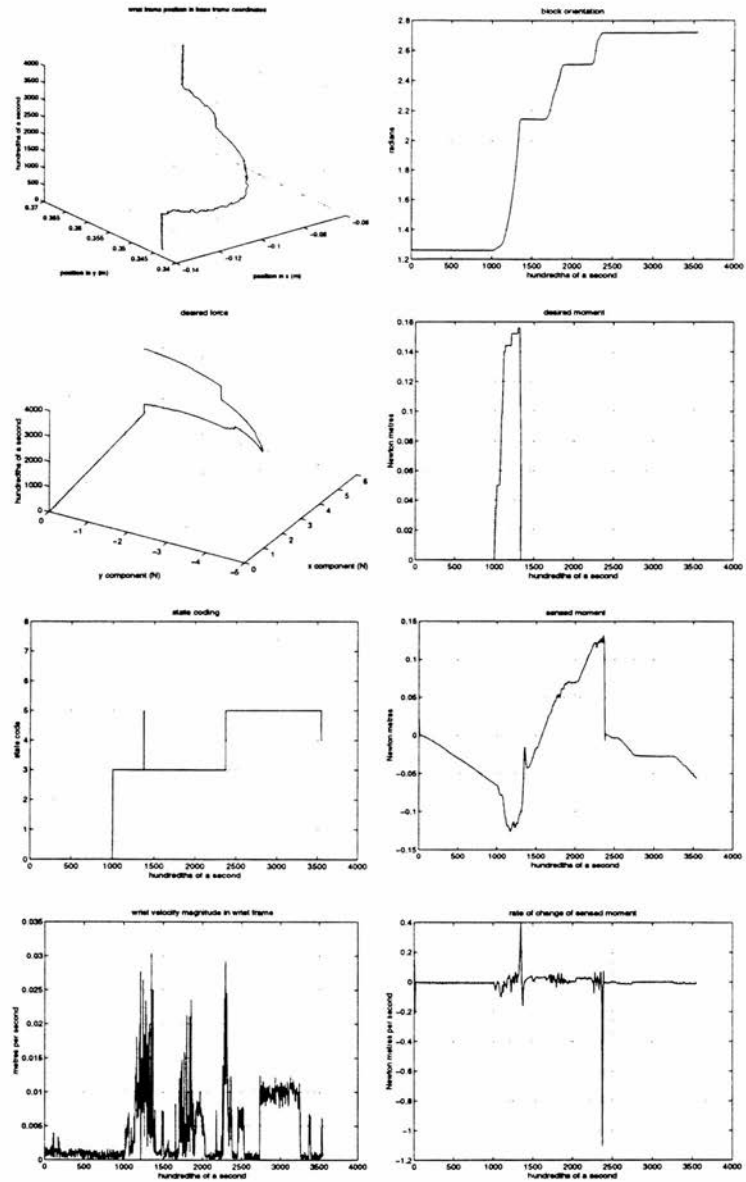


Figure 6.8: Example data generated rotating a block in a corner. Parameters: block, $2" \times 4\frac{1}{2}" \times \frac{3}{4}"$ aluminium; arm configuration, right-handed; reorientation sense, anticlockwise; location 2; initial orientation, 1.26 rads; goal orientation, 2.72 rads.

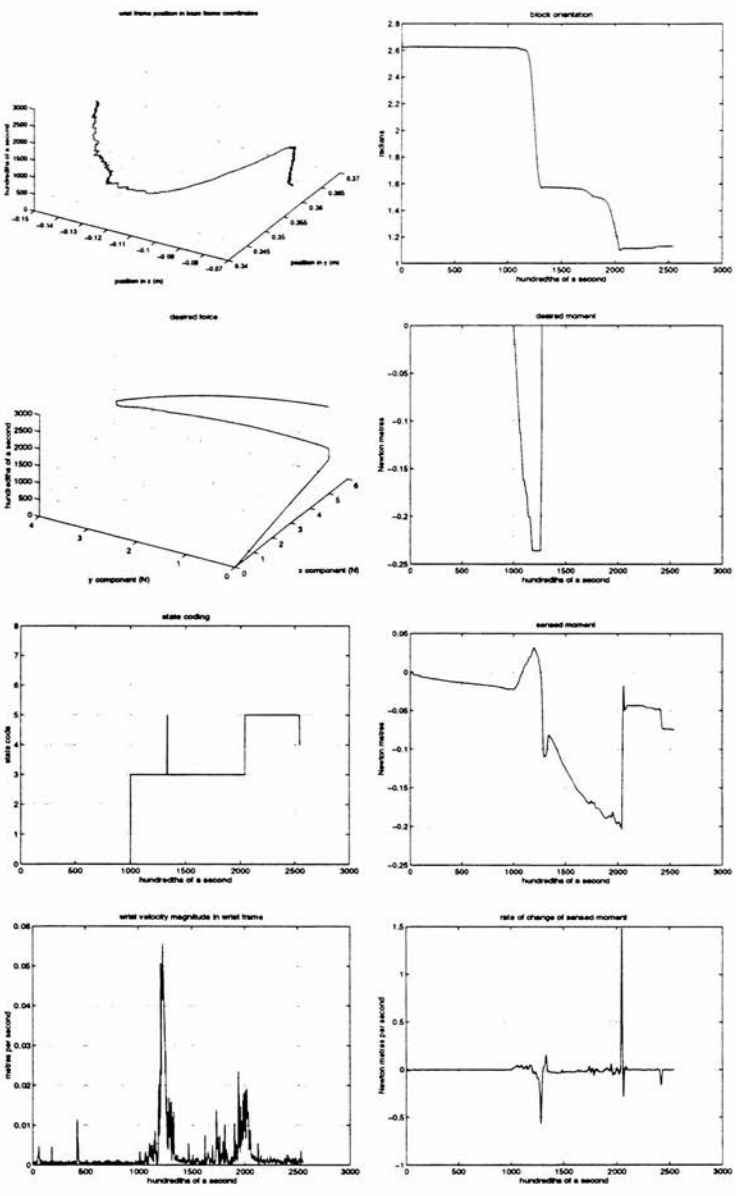


Figure 6.9: Example data generated rotating a block in a corner. Parameters: block, 2" $\times \frac{3}{4}$" aluminium; arm configuration, right-handed; reorientation sense, clockwise; location 2; initial orientation, 2.62 rads; goal orientation, 1.14 rads.

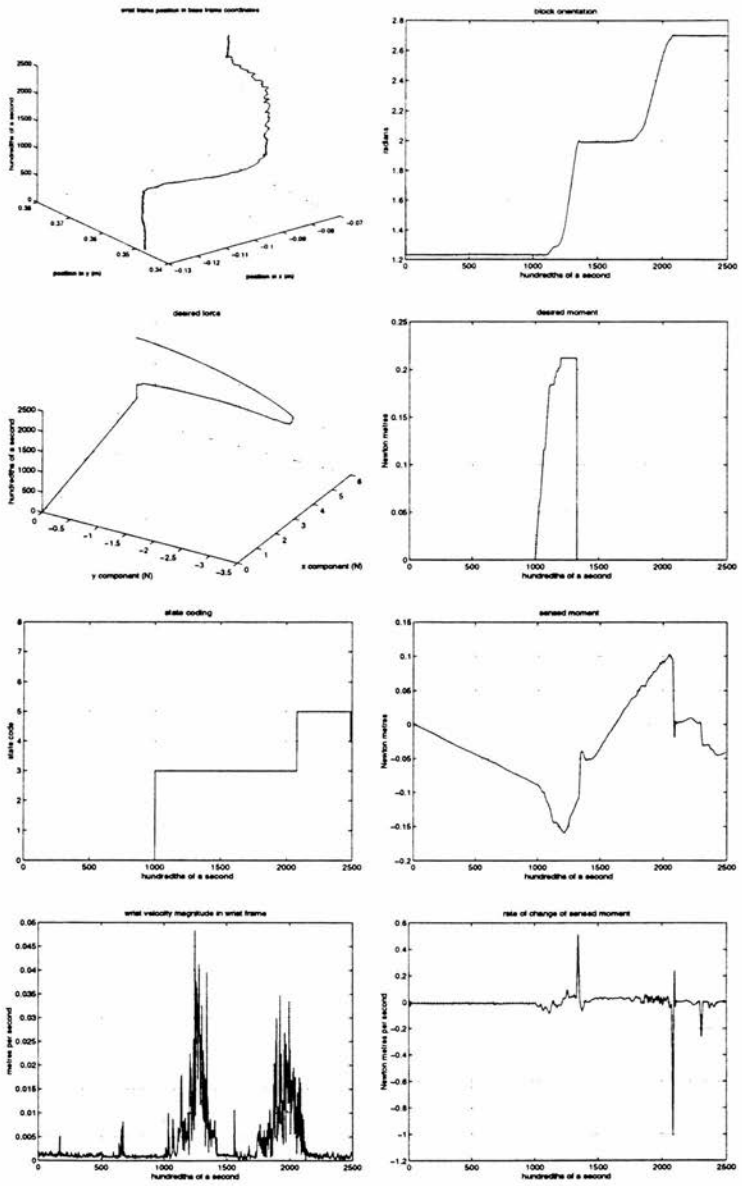


Figure 6.10: Example data generated rotating a block in a corner. Parameters: block, $2" \times \frac{3}{4}"$ aluminium; arm configuration, left-handed; reorientation sense, anticlockwise; location 2; initial orientation, 1.23 rads; goal orientation, 2.67 rads.

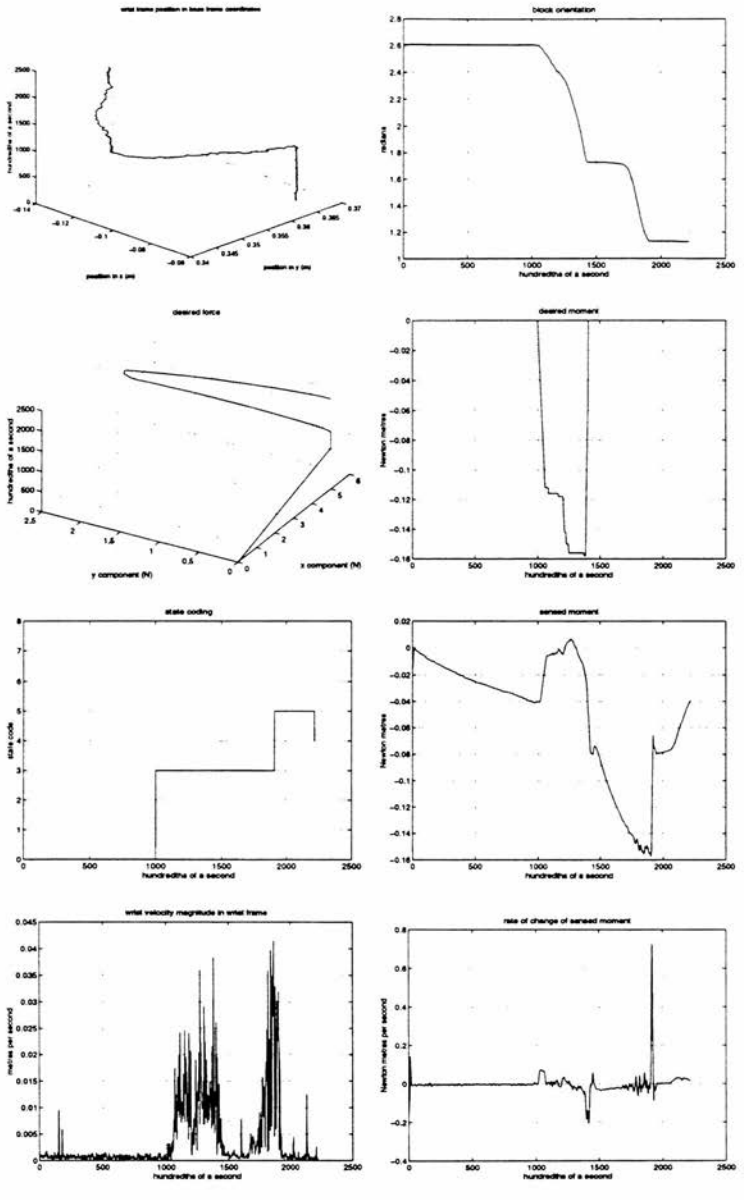


Figure 6.11: Example data generated rotating a block in a corner. Parameters: block, 2" $\times$ $\frac{3}{4}$" aluminium; arm configuration, left-handed; reorientation sense, clockwise; location 2; initial orientation, 2.61 rads; goal orientation, 1.13 rads.

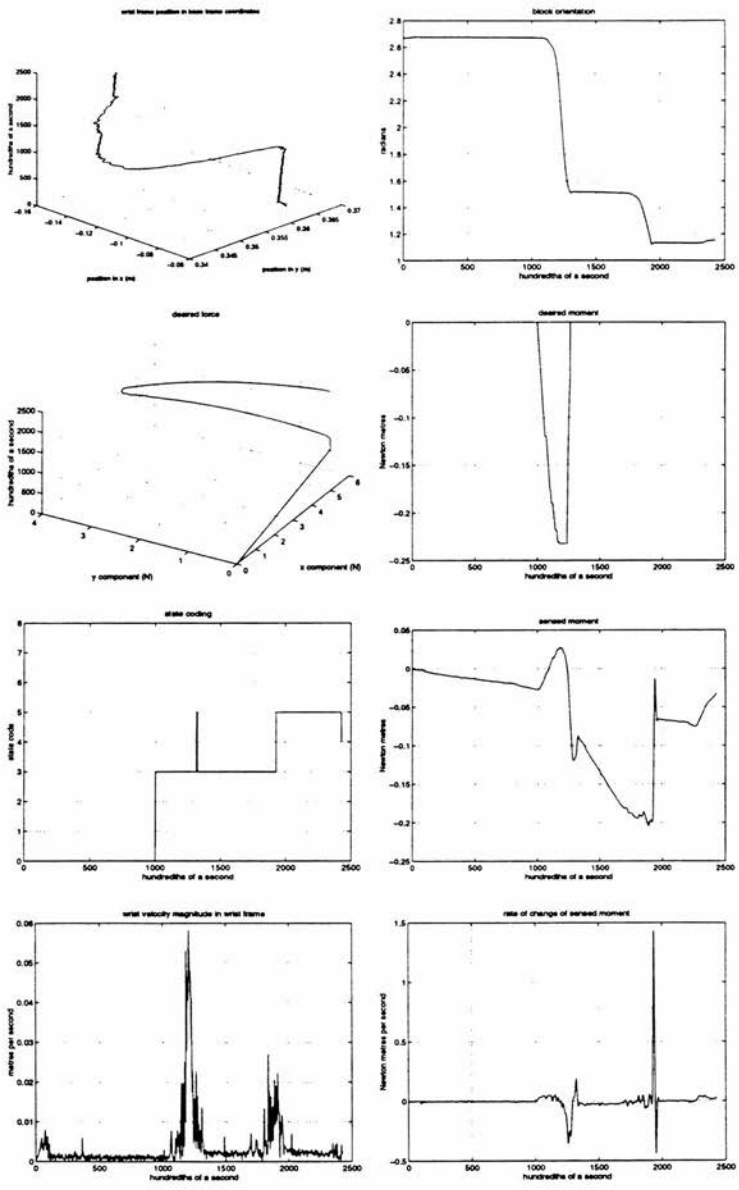


Figure 6.12: Example data generated rotating a block in a corner. Parameters: block, $2" \times \frac{3}{4}"$ ramon wood; arm configuration, right-handed; reorientation sense, clockwise; location 2; initial orientation, 2.67 rads; goal orientation, 1.17 rads.

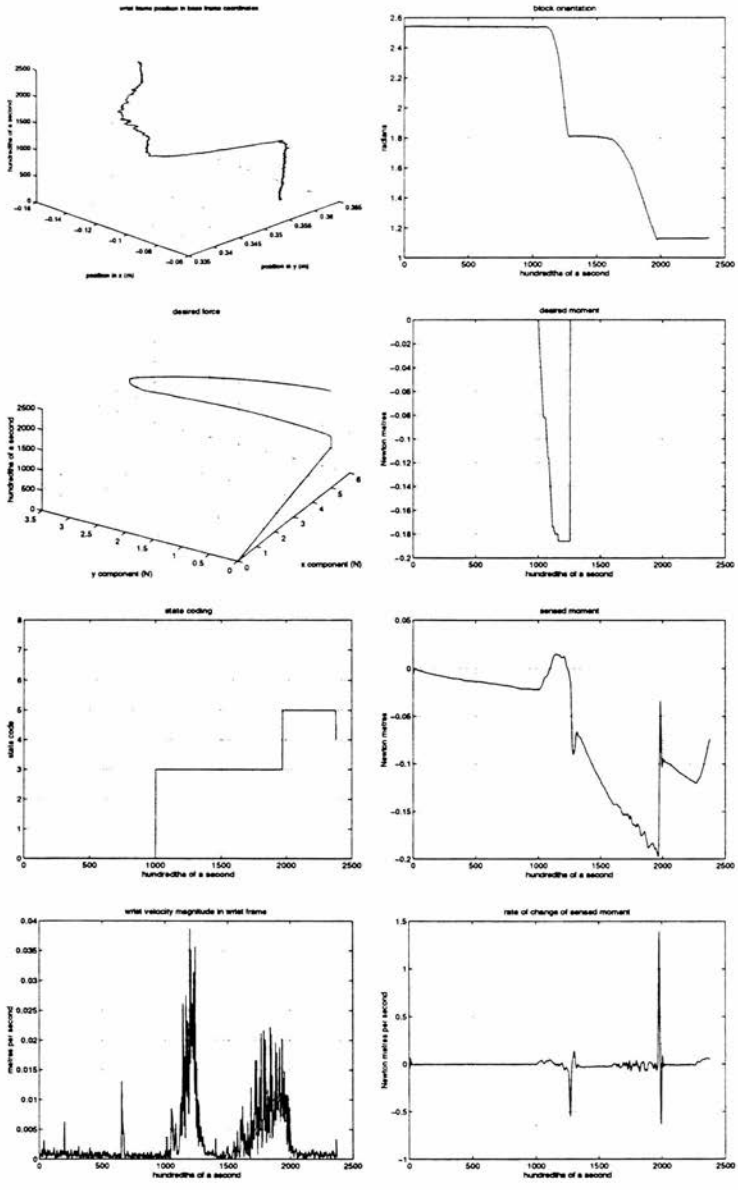


Figure 6.13: Example data generated rotating a block in a corner. Parameters: block, 2" $\times$ $\frac{3}{4}$" brass; arm configuration, right-handed; reorientation sense, clockwise; location 2; initial orientation, 2.54 rads; goal orientation, 1.14 rads.

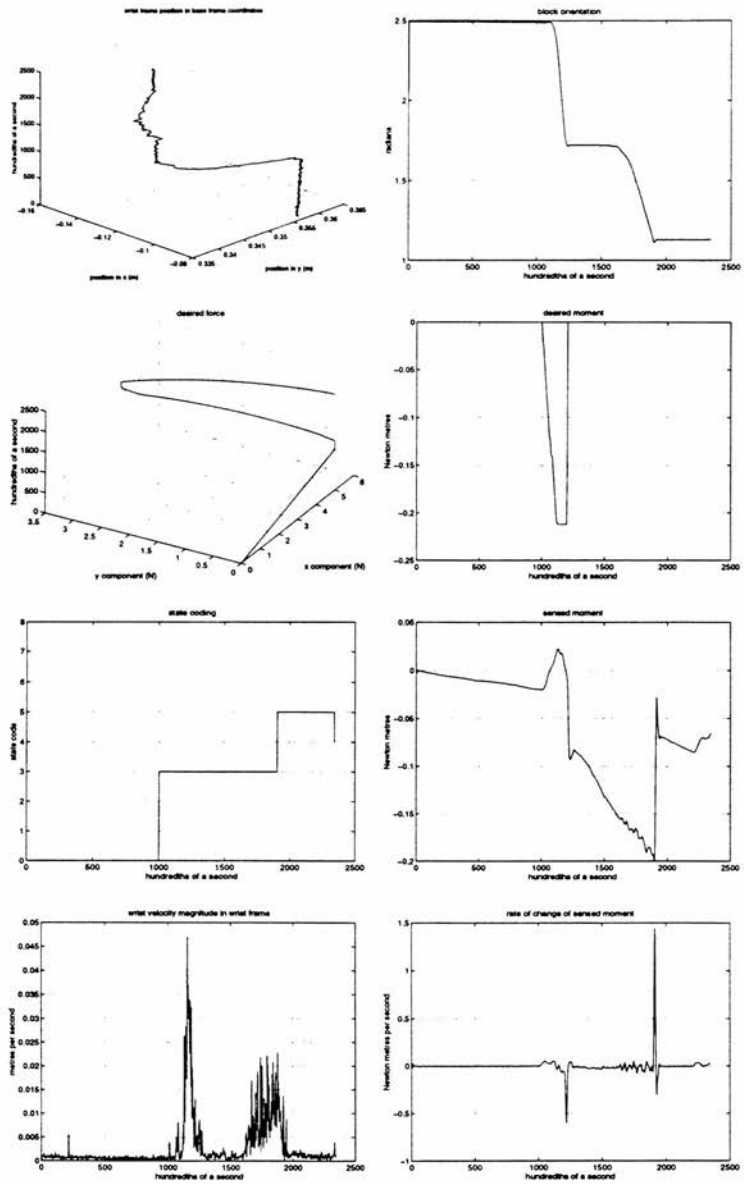


Figure 6.14: Example data generated rotating a block in a corner. Parameters: block, 2" $\times$ $\frac{3}{4}$" bright mild steel; arm configuration, right-handed; reorientation sense, clockwise; location 2; initial orientation, 2.49 rads; goal orientation, 1.14 rads.

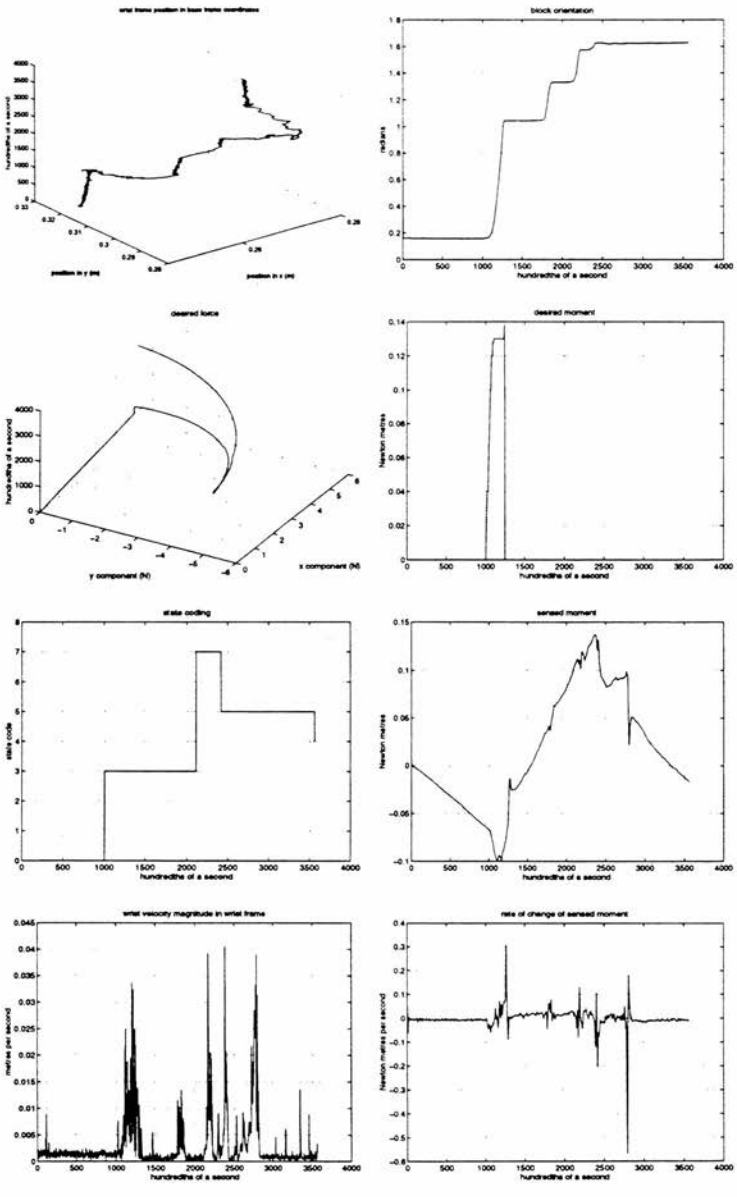


Figure 6.15: Example data generated rotating a block in a corner. Parameters: block, $1" \times \frac{3}{4}"$ aluminium; arm configuration, right-handed; reorientation sense, anticlockwise; location 1; initial orientation, 0.16 rads; goal orientation, 1.63 rads; nominal desired force, 5N

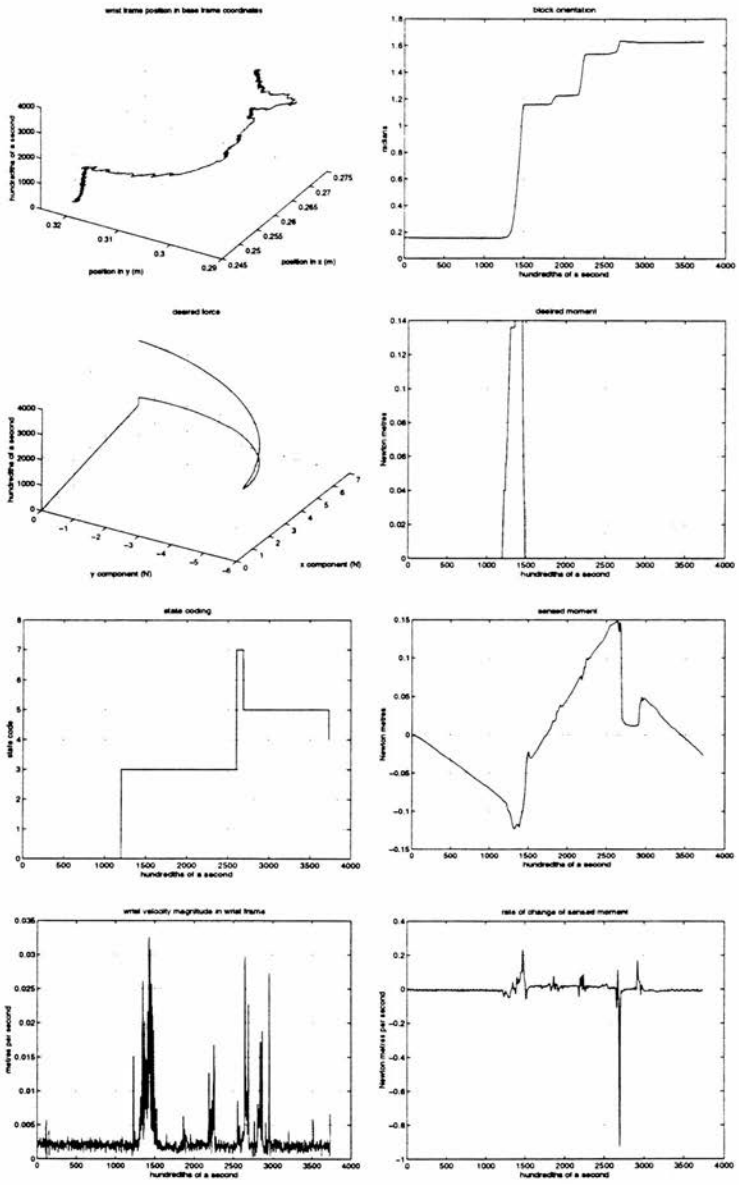


Figure 6.16: Example data generated rotating a block in a corner. Parameters: block, 1" $\times \frac{3}{4}$" aluminium; arm configuration, right-handed; reorientation sense, anticlockwise; location 1; initial orientation, 0.15 rads; goal orientation, 1.63 rads; nominal desired force, 6N.

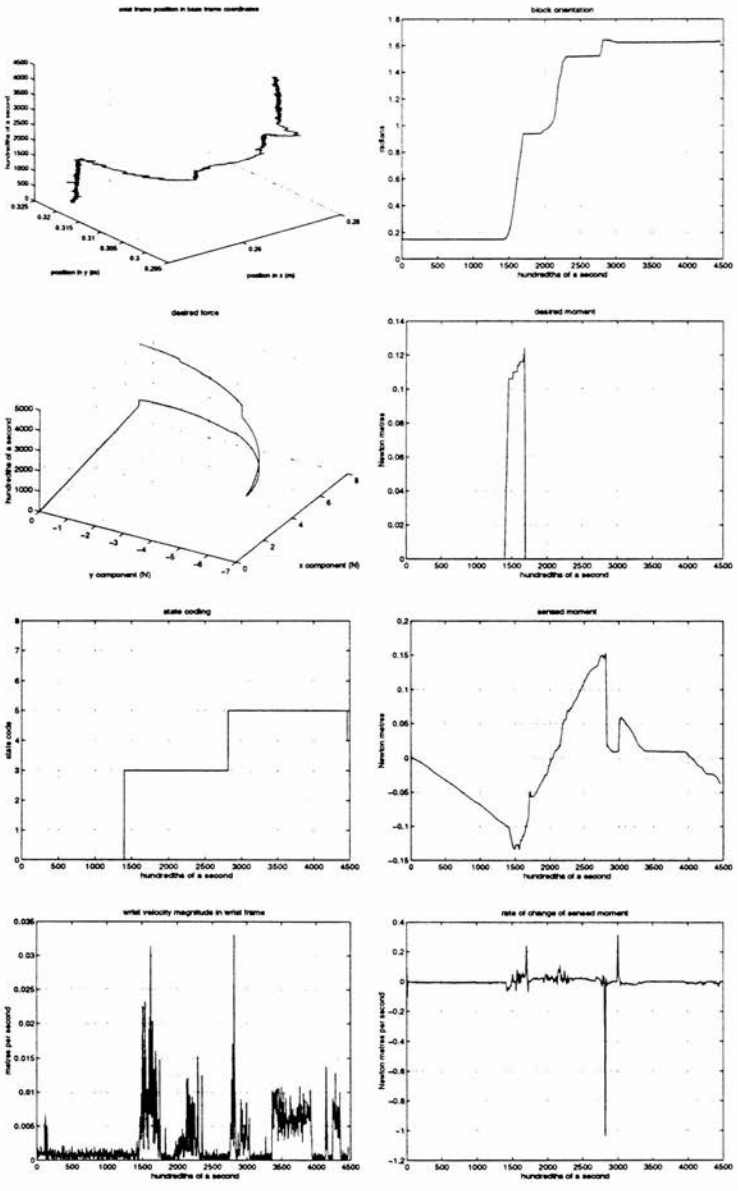


Figure 6.17: Example data generated rotating a block in a corner. Parameters: block, $1" \times \frac{3}{4}"$ aluminium; arm configuration, right-handed; reorientation sense, anticlockwise; location 1; initial orientation, 0.15 rads; goal orientation, 1.63 rads; nominal desired force, 7N.

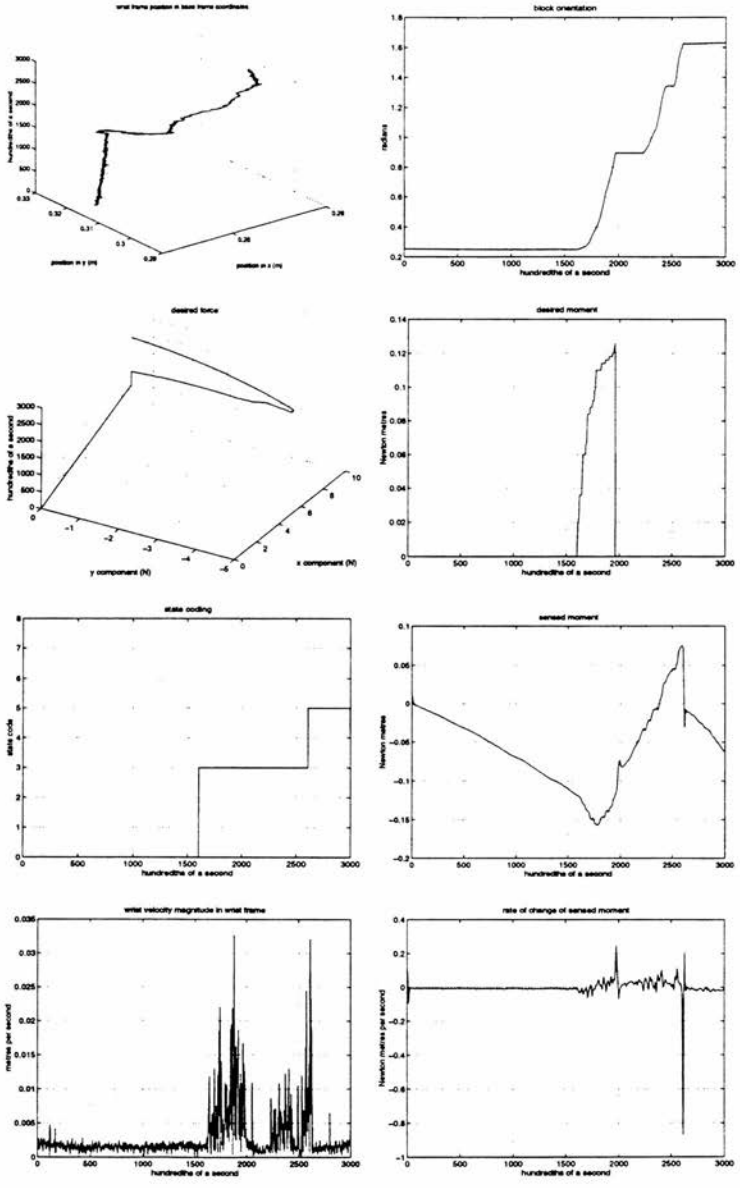


Figure 6.18: Example data generated rotating a block in a corner. Parameters: block, $1" \times \frac{3}{4}"$ aluminium; arm configuration, right handed; reorientation sense, anticlockwise; location 1; initial orientation, 0.25 rads; goal orientation, 1.63 rads; nominal desired force, 8N.

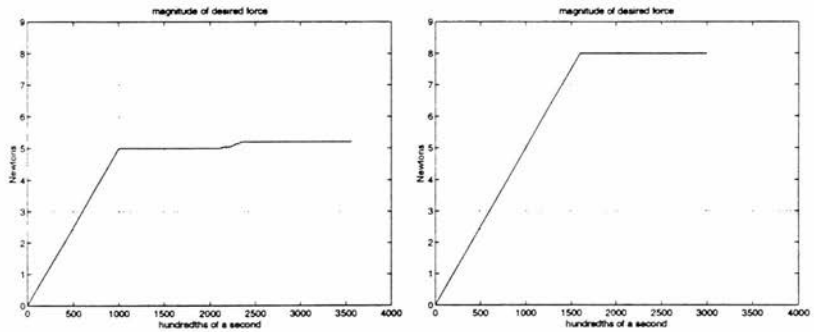


Figure 6.19: Magnitude of the desired force versus time for the example tasks illustrated in figures 6.15 (left plot) and 6.18 (right plot).

At the moment it is not clear why the block leaves two point contact. It may be that the assumptions required in order to apply equation D.38 are not satisfied to the extent that there is a noticeable affect on the performance of the arm. It may just be due to a poor estimate of the block orientation. This is something we intend to investigate further in the future. It is fortunate for our purposes that as the block slides away from the corner constraint that its reorientation during sliding is in the desired sense and tends to take the applied force closer to the surface normal rather than further away.

Figure 6.19 shows the magnitudes of the desired force versus time for the examples of figure 6.15 and 6.18. It is interesting to note that to guarantee that the 1" block will maintain two point contact we need to use a nominal desired force of 8N, whereas if we allow a possible loss of contact, success can be achieved with a significantly smaller desired force, although the task takes longer to complete.

Although in Chapter 5 we show that the right or left handedness of the arm affects the performance of the crank turning task, there is no noticeable affect on the performance of the block in corner task. We would expect this to be the case providing our assumptions concerning the quasi-static nature of the arm's operation are valid. Basically, providing the arm moves slowly enough we can get away without having to compensate for dynamic forces.

In section 4.5.4 we consider the constraints on where the centre of compliance can be in order to be able to induce a torque of the desired sense. There is also a constraint due to the inducement of unwanted torques. In section 4.6 we consider the rotating of a block and how the sensed wrench appears in wrench space as the contact configuration changes. At the point at which c_2 touches the horizontal constraint (see figure 4.7) there is a discontinuity in the induced torque which can be sensed, thus signalling the change in kinematic relationship. However, it is possible when this change of kinematic relationship occurs that the product $s_x {}^w f_{a_y}$ in equation 4.14 is still large enough to produce an induced moment that will carry on turning the block. For the task of placing a block in the corner this is undesirable since it allows the possibility of the block overshooting the goal. In order to avoid this situation we placed the origin of the frame w coincident with the point at which the robot arm ends and the block begins. In terms of figure 4.7 this corresponds with mid-way between c_2 and c_3. By doing this we effectively ruled out the need for any action procedure to be associated with state $C10C12$. Notice, however, that this choice, in conjunction with the aspect ratio constraint on the location of the centre of compliance described in section 4.5.4 effectively places a restriction on the length of the block that this strategy will work for. In order to avoid this we could make the location of the centre of compliance along the x-axis of frame w a function of the orientation of the block with respect to the goal, or we could allow the possibility of the block overshooting the goal and encode a strategy to cope with it.

6.3 Inserting a Peg into a Hole

The insertion of a peg into a hole is a classic problem in the automatic assembly literature that arises time and again because of its fundamental practical importance as the exemplar of the type of operation used to fit two parts together. Its significance was shown by an analysis of the operations commonly occurring in real assemblies [Nevins & Whitney 78]. Some of the works that perform an analysis of the mechanics of this operation include [Simunovic 75], [McCallion & Wong 75], [Ohwovoriole 80], [Arai & Kinoshita 81], [Whitney 82], [Erdmann 84], [Rajan *et al.* 87], [Caine *et al.* 89],

[Brost & Mason 90], [Mason 91], and this list is by no means exhaustive![4] Most of this work tends to be planar in nature, but [Ohwovoriole 80] showed that for a cylindrical peg and a cylindrical hole this results in no loss of generality. There has also been work on analysing the insertion of three dimensional rectangular blocks into rectangular holes [Sturges 88], [Caine *et al.* 89], [Sturges & Laowattana 92].

The program that we implemented is based on the analysis of [Caine *et al.* 89] as described in section 4.5.2. In particular we exploit the constraints implied by inequalities 4.29 to 4.31.

In order to facilitate an explanation of our peg in hole strategy we will find it useful to refer to the iconic representations of some of the contact configurations of the task given in figure 6.20. We only show these for one sense of tilt of the peg. There is of course another set of contact configurations where the peg is oriented in the opposite sense with respect to the central axis of the hole. The symmetry of the task means that the strategy that we are about to describe applies for both of these sets of contact configurations. We tell the system the goal orientation, so that it can determine for itself the reorientation initially required, so that the same program code applies regardless of the two mirror-image starting conditions.

As for the block in corner program the first thing that the program does is to interrogate the user for the goal orientation of the peg. This is input to the program by placing the peg in the hole and telling the program to remember the orientation. Next an initial push is applied, which is built up in increments of 0.005N per sample instance, until 3N is reached. The system expects to be in a state of static equilibrium upon completion of this push, and if it isn't the task terminates. The program assumes that either a corner of the hole entrance will be touching the bottom of the peg once this equilibrium is achieved, or that the peg has just entered the hole but is jammed, *i.e.*, with reference to figure 6.20, that either contact configurations $C3$, $C4$ or $C5$ hold.

As for the block in corner strategy changes to the applied wrench tend to get made in 'small' increments, after which flow of control passes to the state interpreter. There are exceptions to this rule however, in the peg in hole strategy, as will become apparent.

[4] A literature survey is given in [Bland 86], but this is rather dated now.

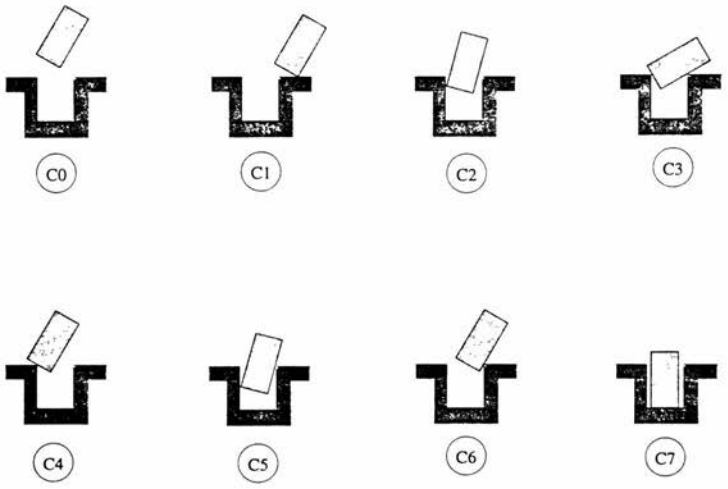


Figure 6.20: Some of the contact configurations possible in the peg in hole problem.

There is some compliance added to the robot's distal link, but for this task we use a softer material than in the block in corner task. This can be seen in figure 6.21. We are not sure what the actual material is (it was bought from a toy shop). The piece of material that we use is naturally a ball shape, about the size of a tennis ball. If it is squeezed it slowly regains its shape, once released, after a few seconds. We were initially interested in it as a way of mechanically filtering out transients after impact in order to simplify the signal processing that the software needs to perform. This is why we included the ability to adjust the amount it is compressed when *in situ*. However, dealing with transients didn't turn out to be the major problem we expected it to be. Despite this we continued to use this material as a means of protecting the force/torque sensor from impacts. In the experimental data presented below the wrist orientation sometimes overshoots alignment with the hole: this is due to this compliant material.

The state action procedures that we use in this task are described below. We will assume, for the sake of simplicity of explanation, that the block is initially oriented in the sense shown in figure 6.20.

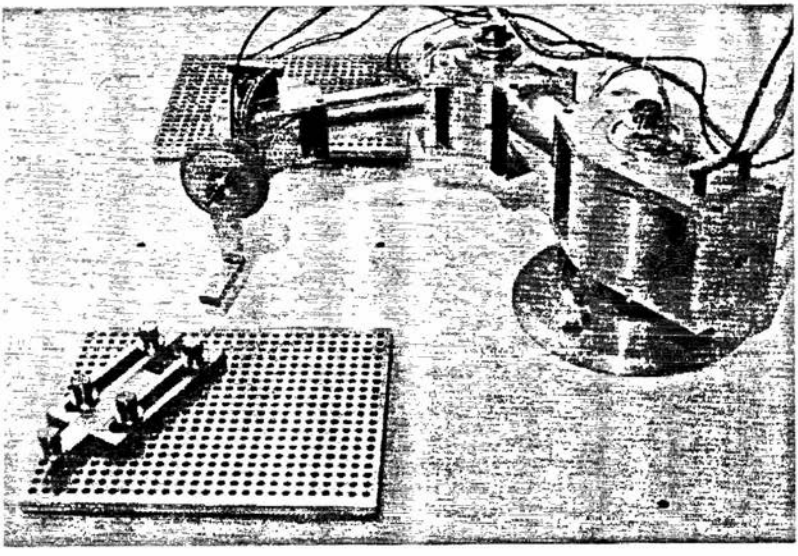


Figure 6.21: A view of our robot arm showing the (potentially) adjustable compliance used in the peg in hole task.

C0 There is no contact. For safety purposes if no contact is experienced the arm is powered down.

- The task terminates.

C3C4C5 The peg is somewhere near the entrance to the hole. The general aim of this state is to rule out the possibility of being in contact configuration $C4$.

- A flag is set so that any sensed moment, *i.e.* induced torque, is balanced.
- If there is no contact then
  - the current state is set to $C0$
- If there is no motion then
  - if the desired force is less than 60^o from the x-axis of frame w then
    - ⋆ rotate the applied force clockwise

    else
    - ⋆ the current drive wrench is stored as the current desired wrench (so

that any induced torque being balanced continues to be balanced)

- ⋆ the current state is set to $C3C5$

- If there is motion then
  - if a discontinuity is detected in the sensed moment then
    - ⋆ the current drive wrench is stored as the current desired wrench (so that any induced torque being balanced continues to be balanced)
    - ⋆ the current state is set to $C3C5$

C3C5 The peg is in two point contact somewhere near the hole entrance. The general aim of this state is to rotate the peg so that it can slide into the hole shaft.

- If there is no contact then
  - the current state is set to $C0$
- If there is no motion then
  - increment the moment component of the desired wrench
- If there is motion then
  - if within $\pm\varepsilon$ of the goal orientation then
    - ⋆ decrement the moment component of the desired wrench
  - if a discontinuity is detected in the direction of the sensed force then
    - ⋆ the desired force is aligned with the positive x-axis of frame w, and the desired moment is set to zero
    - ⋆ the current state is set to $C2C5$

C2C5 The peg is somewhere in the hole. The general aim of this state is to get the peg to slide along the hole shaft.

- If there is no contact then
  - the current state is set to $C0$
- If there is no motion then
  - if the current orientation is less than the goal orientation then
    - ⋆ increment the moment component of the desired wrench
  - if the current orientation is greater than the goal orientation then
    - ⋆ decrement the moment component of the desired wrench
- If there is motion then
  - if a discontinuity is detected in the sensed force magnitude then

- ⋆ the current state is set $C7$

C7 The goal configuration.

- The task terminates.

The reader will note that there is no branching in this task. A number of other features of these descriptions warrant some discussion.

Firstly, note that state $C3C4C5$ balances the induced torque in order that the bottom of the peg might slide across the corner of the hole (should the relationship between the parts be such that motion is allowed), whilst maintaining a (roughly) constant orientation with respect to the hole (see section 4.5.1).

We described earlier, in section 4.5.2, how we could perform a search in wrench space to find the point at which inequality 4.31 was satisfied. The analysis of [Caine *et al.* 89] also gave inequalities which constitute constraints on the value of α shown in figure 4.9. Since for all (positive) values of the coefficient of friction, μ, inequality 4.29 will always be less than inequality 4.30, it is the latter that is the bounding constraint. From equation 4.4 it is clear that inequality 4.30 can be rewritten as

$$\alpha \geq \psi \tag{6.1}$$

In other words the angle that the applied force makes with the x-axis of the frame of reference w has got to be greater than, or equal to, the friction cone half-angle. Now, as the desired force is rotated about the centre of compliance (the origin of frame w) in order to precipitate motion of the bottom of the peg relative to the corner of the hole (should the relationship between the parts be such that motion is allowed), because the orientation of the centre line of the friction cone due to the point contact of the hole's corner is fixed with respect to the peg, motion will not occur until the direction of the applied force has rotated past the 'right' edge of the friction cone at this point. In other words, providing sliding motion has been precipitated inequality 6.1 must be satisfied. Consequently, we maintain the desired force in this direction when the flow of control is passed to $C3C5$. It is then the job of this state to incrementally change the desired moment until inequality 4.31 is satisfied. In this way all of the inequalities from 4.29

to 4.31 are satisfied simultaneously, as the result of searching for their limiting values on-line.

Due to the fact that the parts may not be in a relative pose where the bottom of the peg can slide across the corner of the hole, we allow the desired force to be rotated up to 60^o from the x axis of frame w, on the assumption that the friction cone half-angle must be less than this[5], so that if motion were possible it would occur by the time the direction of the desired force reached this direction. Since we are assuming that 60^o is greater than any friction cone half-angle that we might encounter, inequality 6.1 must again be satisfied by the time that $C3C4C5$ is exited, as in the case where motion was precipitated.

For the work described in [Krishnaswamy & Seering 94] the analysis of [Caine *et al.* 89] was used as a test-bed to evaluate the force controllability of a planar manipulator with harmonic drives at the joints. The purpose was to ascertain how sensitive the performance of the arm was to deficiencies in its calibration. It is shown theoretically that certain areas of the manipulator's workspace are more problematic than others, and this is verified by experiment. For a particular pair of peg and hole objects, inequalities 4.29 to 4.31 were evaluated and a desired wrench consistent with them was chosen to perform some experiments. This desired wrench met with varied degrees of success. The strategy that we describe here, which is also based on inequalities 4.29 to 4.31, is invariably successful. It is not clear what this difference in performance is attributable to. It may be due to the fact that our system performs an on-line 'search' to find the limiting values of these inequalities for the actual instance of the task that is under way. The advantage of this being that because we find the conditions where motion is just possible, it tends to maintain the assumption of the task being quasi-static. Alternatively, the crucial factor may be the nature of the arm construction.

One problem that we did encounter is that for large initial angular misalignments, the velocity of the peg, by the time it reaches alignment with the hole, may be so high that the discontinuity sensed as it gets into the hole could be misinterpreted as the peg being in the goal configuration. This is in part due to the fact that we use no positional

[5] We could of course have set this angle to $90°$, but $60°$ seemed more than adequate for most practical purposes.

information to establish achievement of the goal. In order to minimise this problem we introduced an interval about the goal orientation given by $\pm\varepsilon$ in state $C3C5$ where the magnitude of the desired moment is decremented towards zero. The effects of this can be seen in a number of the experiment plots as a jagged, sawtooth characteristic in the desired moment. For the experiments shown here $\varepsilon = 10^o$. This is an *ad hoc* solution to the problem that seems to work: what we really require is a tighter control over the arm's velocity by the velocity limiting loop described in section 5.3. This is something we intend to address once we have changed to a real-time operating system.

The potentially 'violent' nature of the entry of the peg into the hole played a role in the value of the rate of change of sensed force magnitude that we set as signalling the peg reaching the bottom of the hole: we basically required that this threshold be set as low as possible for reliability in detecting the goal state, but it also had to be set high enough to avoid misinterpretation of the peg entering the hole as being the making of contact with the goal. Since the magnitude of the discontinuity sensed will depend on the velocity that the peg achieves by the time it hits the bottom of the hole, tasks with short holes, thereby giving little time for the velocity to build up, might prove problematic. For the other thresholds, similar trade-offs to those discussed for the block in corner task applied: we required the lowest threshold that would detect the expected transition despite the noise present on the sensor signals. The threshold values that we used were: detection of the block sliding into the corner of the hole, 0.1Nm/sec; detection of the peg entering the hole, 250^o/sec; detection of the peg touching the bottom of the hole, 20N/sec.

As for the block in corner task, each state is assigned an integer code so that we have a record of what state the system is in at any particular time. The codes used are as follows:–

0 corresponds to the initial push that the arm exerts

1 corresponds to the initialisation being successful

2 corresponds to state $C0$, *i.e.* no contact

3 corresponds to state $C3C4C5$

4 corresponds to state $C3C5$

5 corresponds to state $C2C5$

6 corresponds to state $C7$, *i.e.* the goal state

As for the block in corner task we didn't detect any noticeable affects on the performance of the peg in hole strategy as a result of the left or right handedness of the arm. The kinematics of the arm did however affect the maximum angular misalignment that the strategy could cope with for the experimental set up of figures 6.35 and 6.36. For these situations the angular misalignment in a clockwise sense was limited by the fact that the proximal and distal joint tended towards being aligned as the angular misalignment was increased. There was a point beyond which the maximum torque that we allow the arm to exert (0.6Nm) was insufficient to move these links so that the distal link could align itself with the hole.

So far the experiments that we have performed have all been on chamferless parts with fairly close tolerances. Chamfers are usually considered beneficial to getting pegs into holes [Whitney *et al.* 83], [Caine 90]. Whether they would be of any major advantage when using EDDIE is something that we intend to investigate in the near future.

Each of the single peg insertion experiments reported here had tolerances of a few thousandths of an inch. It is difficult to be more precise than this because, although spacers were fabricated that were just two to three thousandths of an inch wider than the pegs used, the constraints that formed the holes that the pegs were inserted into were often displaced slightly by the efforts of the arm. Usually such close tolerances are problematic. We suspect however, that for our case it simplifies things, since the likelihood of jamming [Whitney 82] is reduced because it is difficult for the peg to be in the hole with a sufficient misalignment for this to occur. It is likely that we would be able to perform a search in wrench space for the point at which the jamming could be broken, but we would have to deal with the problem of distinguishing between being jammed in the hole and being in the goal configuration.

The fact that our strategy for inserting a peg in works with the dual peg in a hole (for the objects that we have at our disposal, at least), we discovered by experimenting with our robot arm under joystick control. The tolerance between the parts is no more than three thousandths of an inch. (This time we can be more specific about the tolerances since the part that the dual peg is inserted into is just one piece.) Again, we suspect that this close tolerance aids the performance of the task since if the strategy gets one

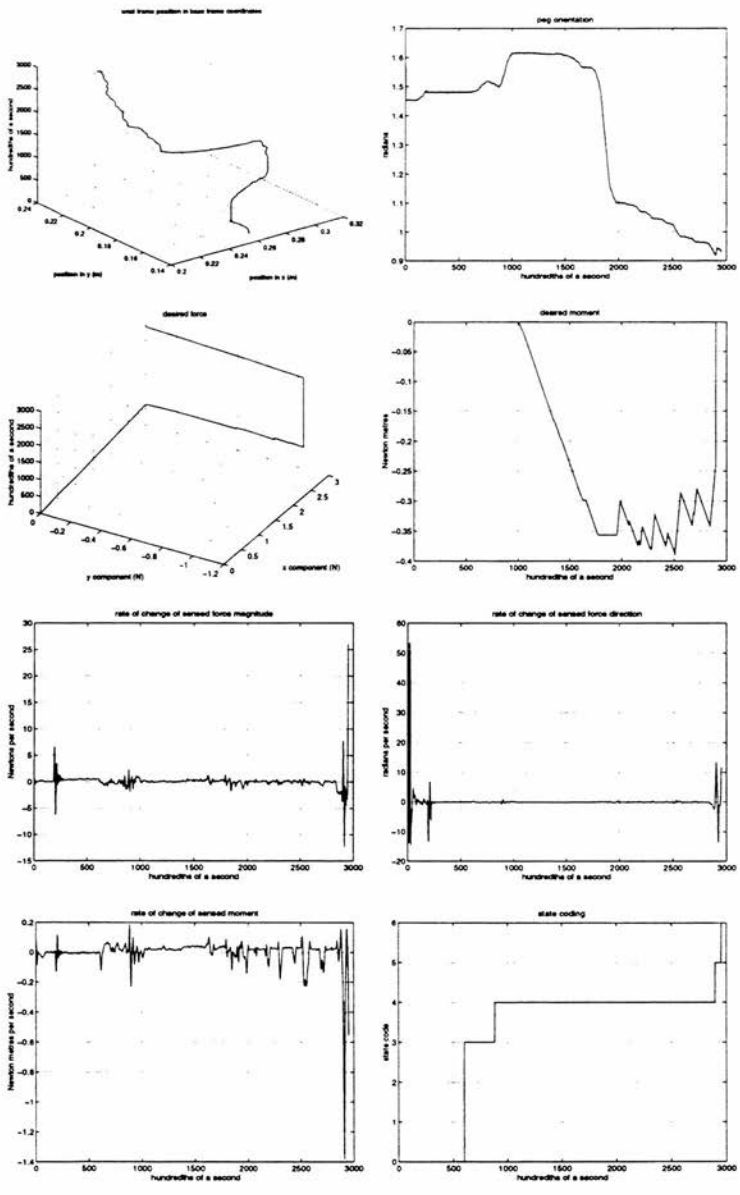


Figure 6.22: Example data generated inserting a peg into a hole. Parameters: peg, 2" $\times$1" aluminium; arm configuration, right-handed; reorientation sense, clockwise; goal position (0.223, 0.218) metres; initial orientation, 1.48 rads; goal orientation, 0.94 rads.

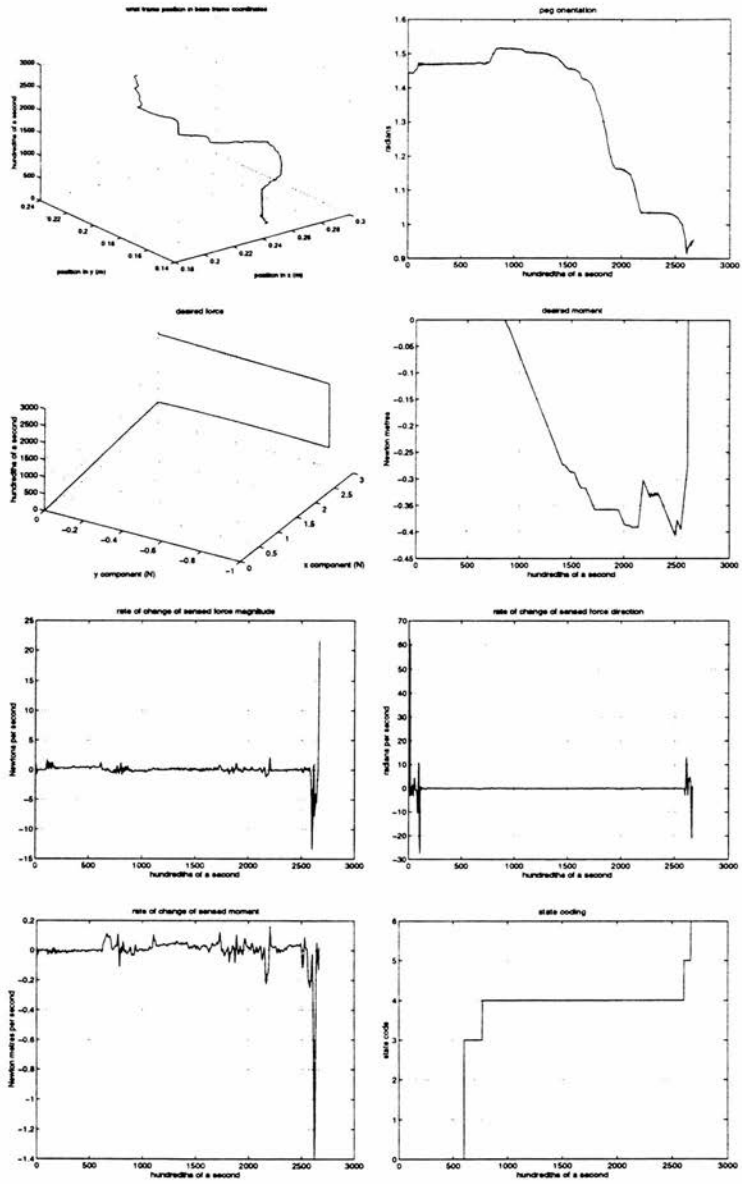


Figure 6.23: Example data generated inserting a peg into a hole. Parameters: peg, 2" × 1" aluminium; arm configuration, left-handed; reorientation sense, clockwise; goal position (0.223, 0.218) metres; initial orientation, 1.47 rads; goal orientation, 0.91 rads.

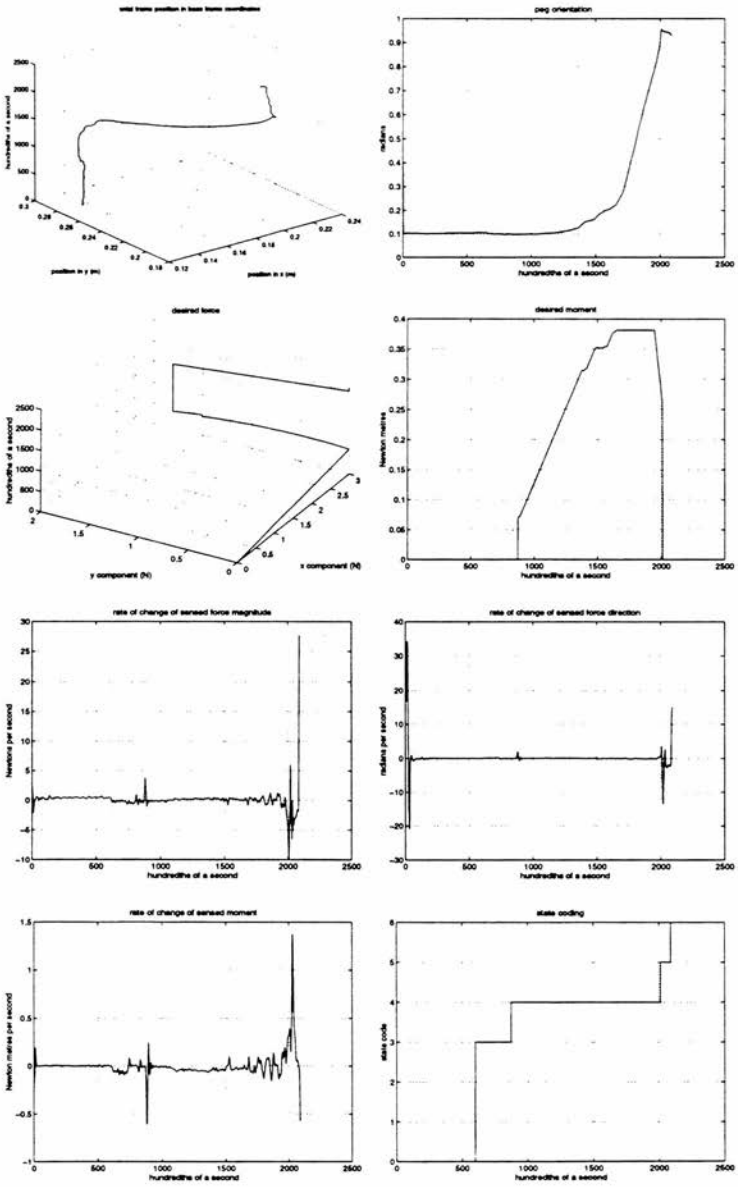


Figure 6.24: Example data generated inserting a peg into a hole. Parameters: peg, $2" \times \frac{3}{4}"$ aluminium; arm configuration, right-handed; reorientation sense, anticlockwise; goal position (0.219, 0.221) metres; initial orientation, 0.11 rads; goal orientation, 0.95 rads.

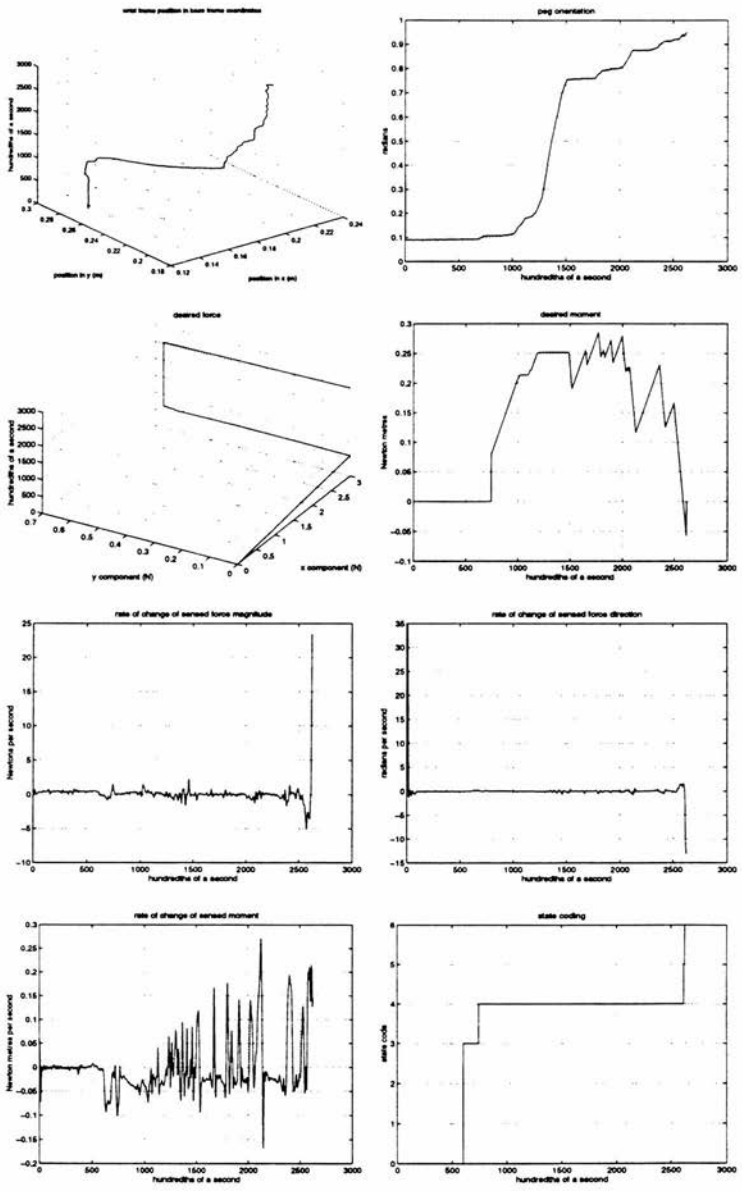


Figure 6.25: Example data generated inserting a peg into a hole. Parameters: peg, 2" $\times \frac{3}{4}$" aluminium; arm configuration, left-handed; reorientation sense, anticlockwise; goal position (0.219, 0.221); initial orientation, 0.094 rads; goal orientation, 0.91 rads.

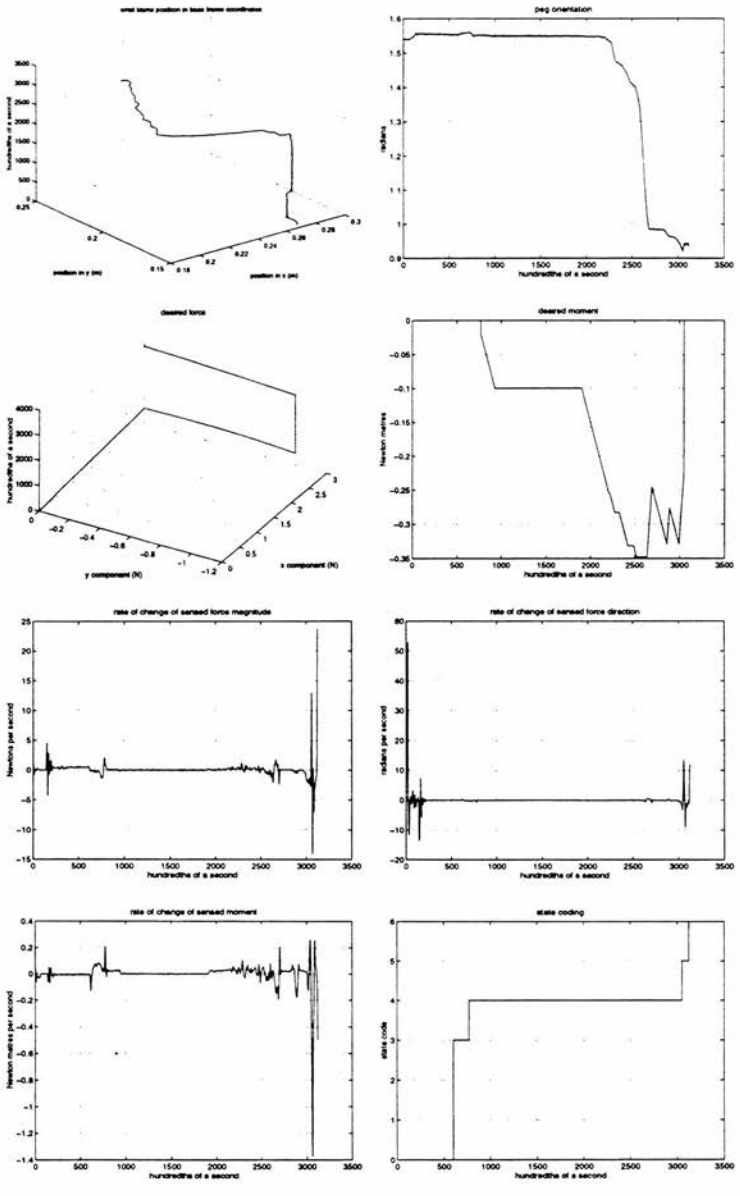


Figure 6.26: Example data generated inserting a peg into a hole. Parameters: peg, 2" $\times$ $\frac{1}{2}$" aluminium; arm configuration, right-handed; reorientation sense, clockwise; goal position (0.212, 0.222) metres; initial orientation, 1.55 rads; goal orientation, 0.95 rads.

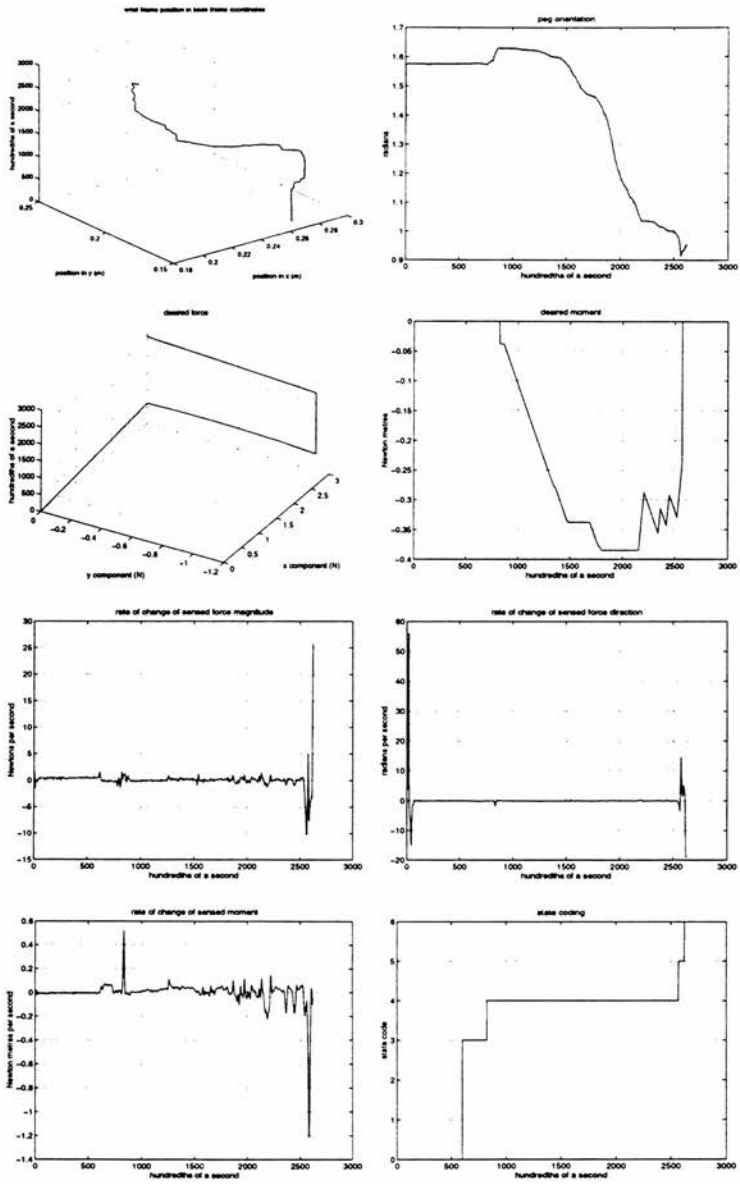


Figure 6.27: Example data generated inserting a peg into a hole. Parameters: peg, 2" $\times \frac{1}{2}$" aluminium; arm configuration, left-handed; reorientation sense, clockwise; goal position (0.212, 0.222) metres; initial orientation, 1.58 rads; goal orientation, 0.90 rads.

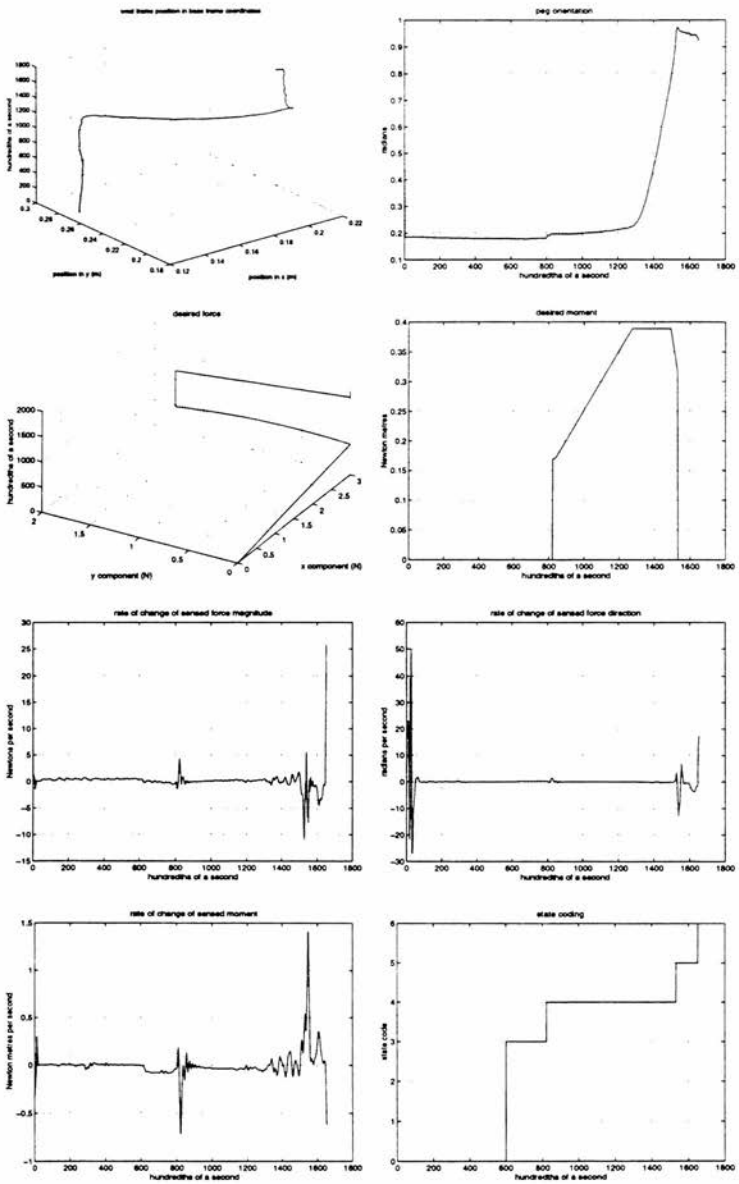


Figure 6.28: Example data generated inserting a peg into a hole. Parameters: peg, $2" \times \frac{1}{4}"$ aluminium; arm configuration, right-handed; reorientation sense, anticlockwise; goal position (0.215, 0.224) metres; initial orientation, 0.18 rads; goal orientation, 0.94 rads.

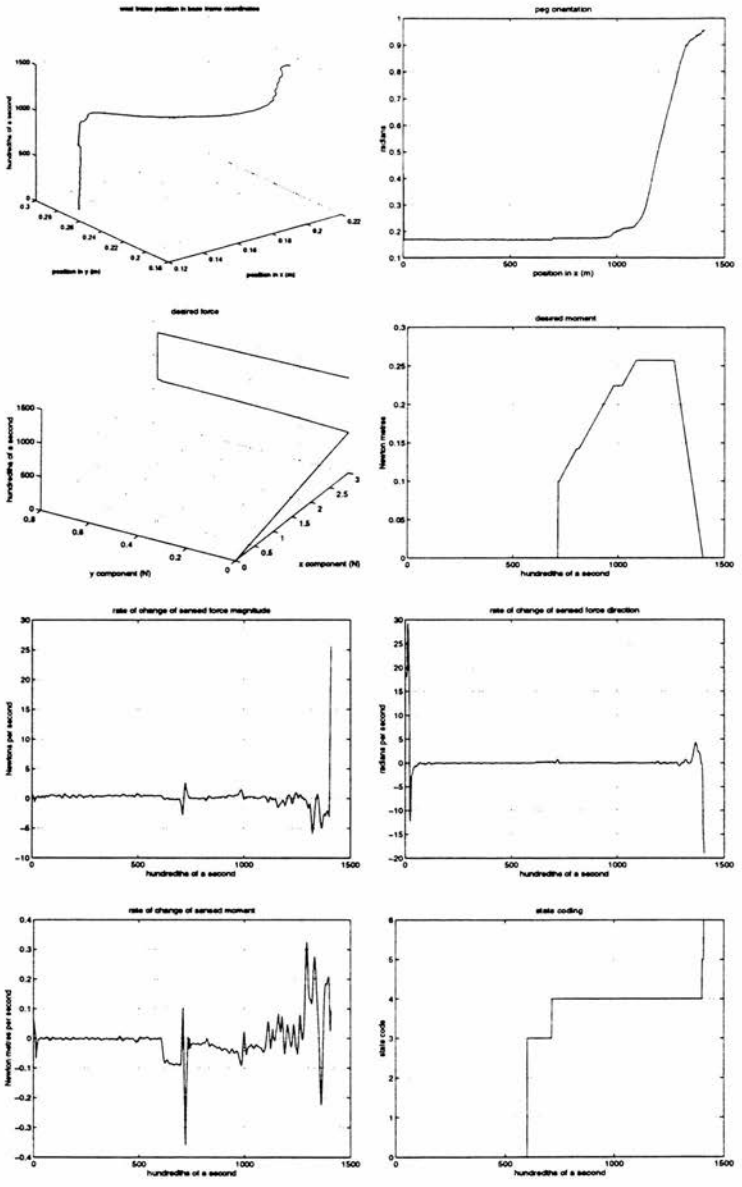


Figure 6.29: Example data generated inserting a peg into a hole. Parameters: peg, 2" $\times \frac{1}{4}$" aluminium; arm configuration, left-handed; reorientation sense, anticlockwise; goal position (0.215, 0.224) metres; initial orientation, 0.17 rads; goal orientation, 0.91 rads.

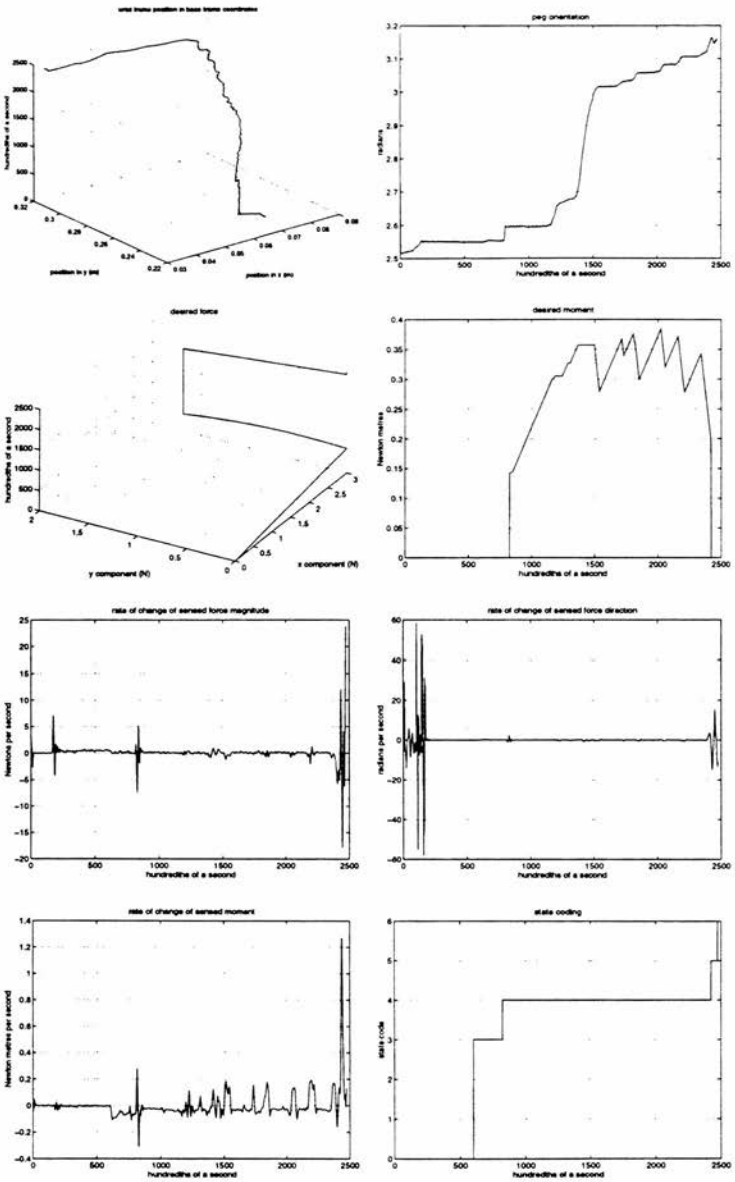


Figure 6.30: Example data generated inserting a peg into a hole. Parameters: peg, $2" \times \frac{3}{4}"$ aluminium; arm configuration, right-handed; reorientation sense, anticlockwise; goal position (0.033, 0.314) metres; initial orientation, 2.55 rads; goal orientation, 3.16 rads.

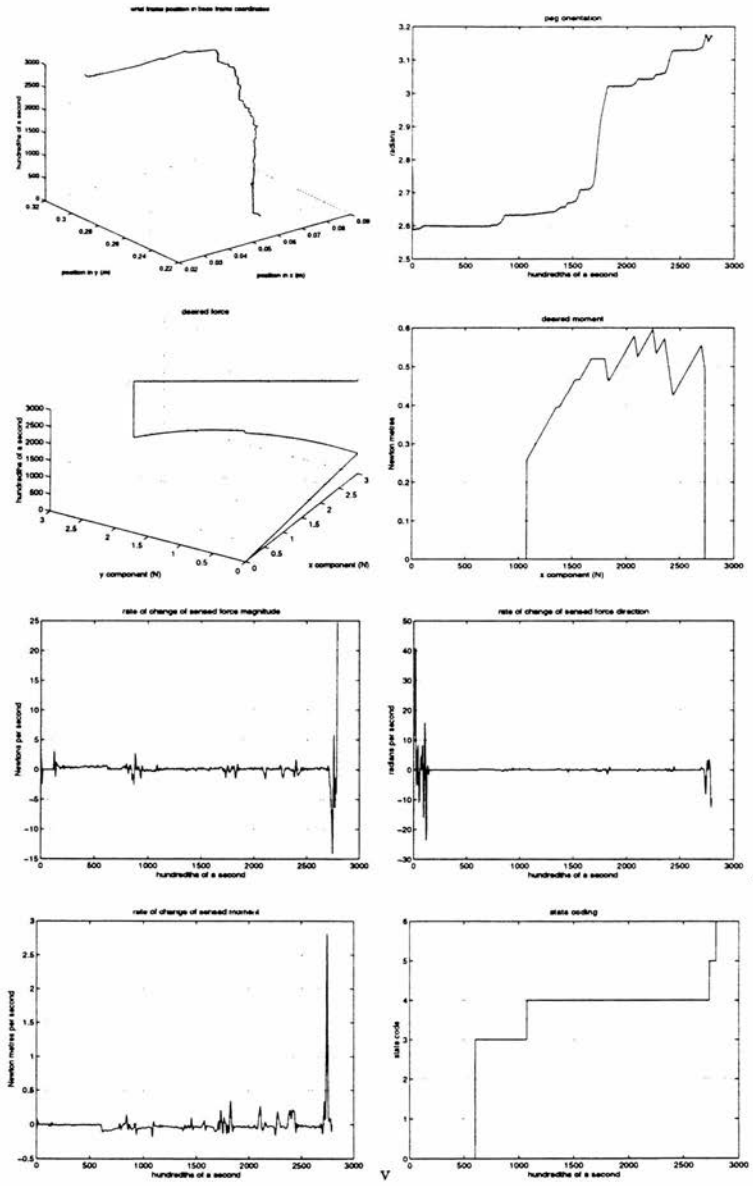


Figure 6.31: Example data generated inserting a peg into a hole. Parameters: peg, 2" × $\frac{3}{4}$" bright mild steel; arm configuration, right-handed; reorientation sense, clockwise; goal position (0.033, 0.314) metres; initial orientation, 2.60 rads; goal orientation, 3.16 rads.

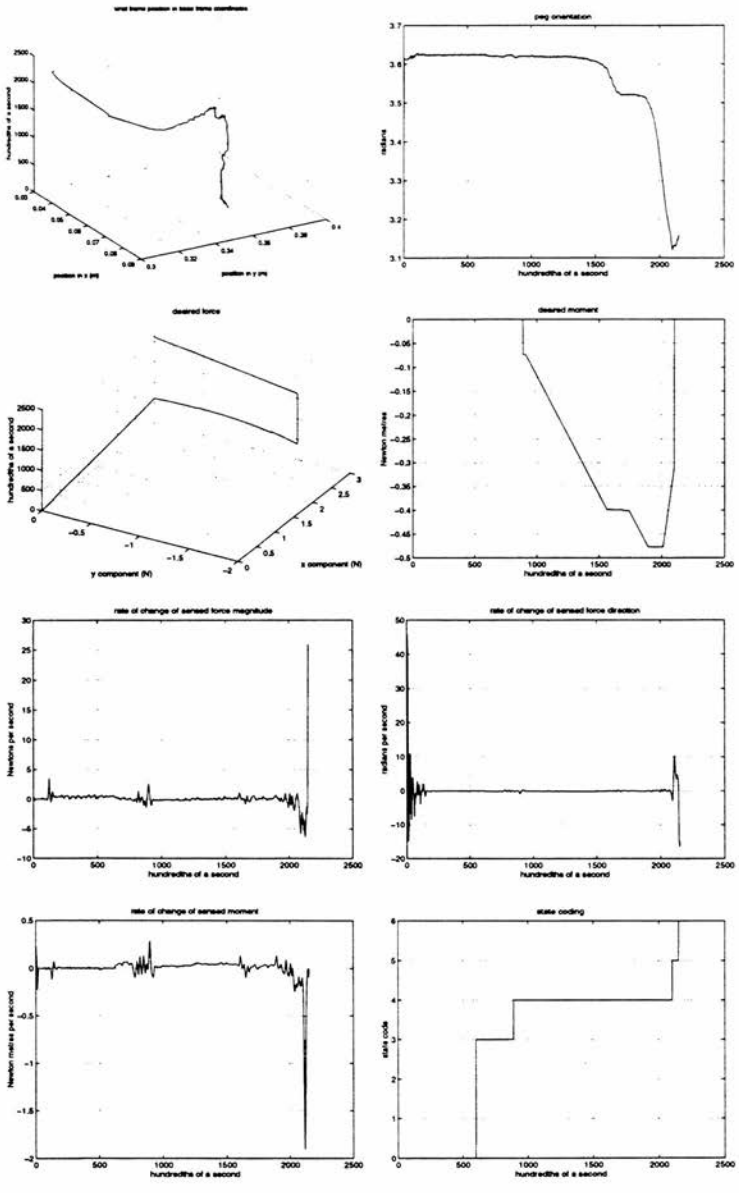


Figure 6.32: Example data generated inserting a peg into a hole. Parameters: peg, $2" \times \frac{3}{4}"$ brass; arm configuration, right-handed; reorientation sense, clockwise; goal position (0.033, 0.314) metres; initial orientation, 3.63 rads; goal orientation, 3.16 rads.

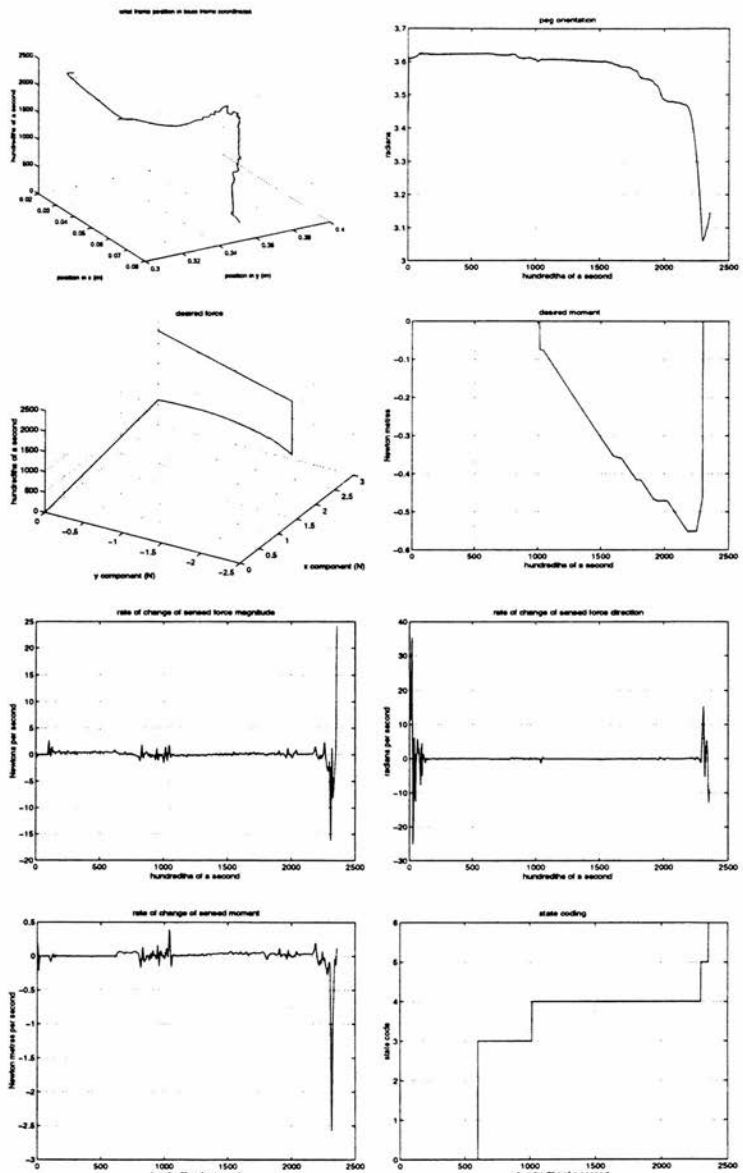


Figure 6.33: Example data generated inserting a peg into a hole. Parameters: peg, 2" $\times$ $\frac{3}{4}$" ramon wood; arm configuration, right handed; reorientation sense, clockwise; goal position (0.033, 0.314) metres; initial orientation, 3.63 rads; goal orientation, 3.14 rads.

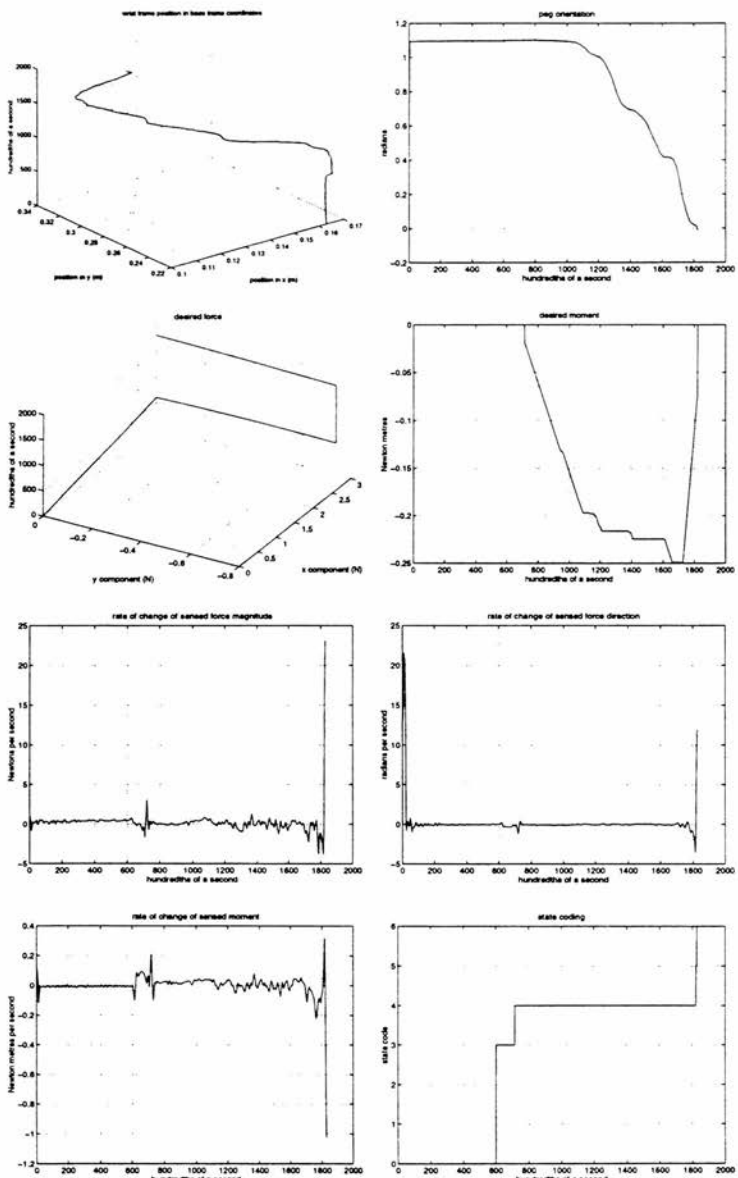


Figure 6.34: Example data generated inserting a peg into a hole. Parameters: peg, $1" \times \frac{3}{4}"$ aluminium; arm configuration, left-handed; reorientation sense, clockwise; goal position (0.125, 0.320) metres; initial orientation, 1.10 rads; goal orientation, 0.01 rads.

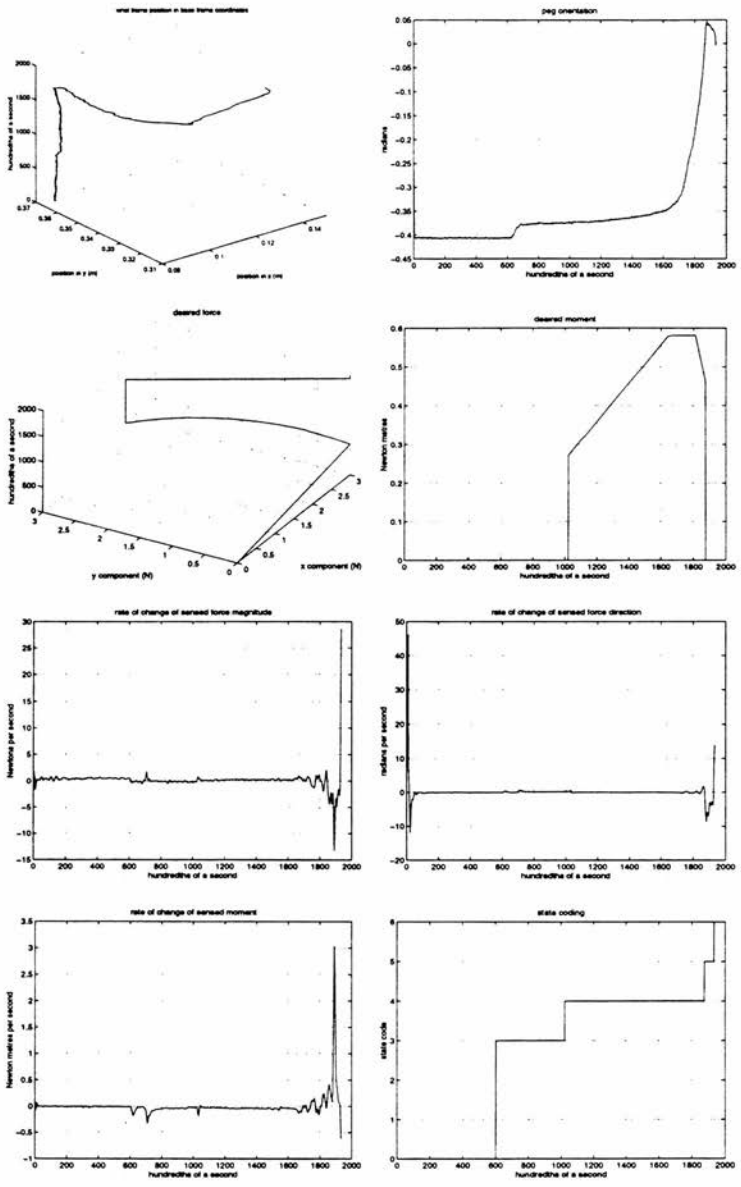


Figure 6.35: Example data generated inserting a peg into a hole. Parameters: peg, 2" $\times \frac{3}{4}$" aluminium; arm configuration, left-handed; reorientation sense, anticlockwise; goal position (0.130, 0.315) metres; initial orientation, 5.88 rads; goal orientation, 0.02 rads.

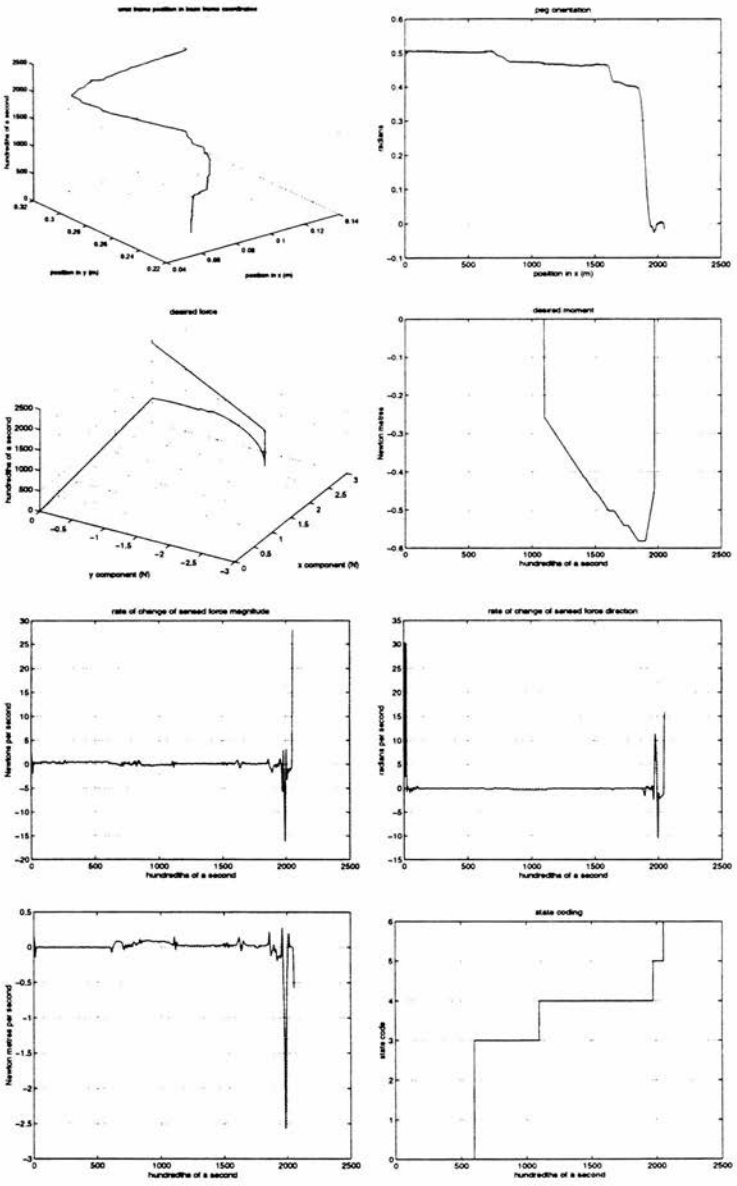


Figure 6.36: Example data generated inserting a peg into a hole. Parameters: peg, $3" \times \frac{3}{4}"$ aluminium; arm configuration, left-handed; reorientation sense, clockwise; goal position (0.122, 0.316) metres; initial orientation, 0.50 rads; goal orientation, 0.00 rads.

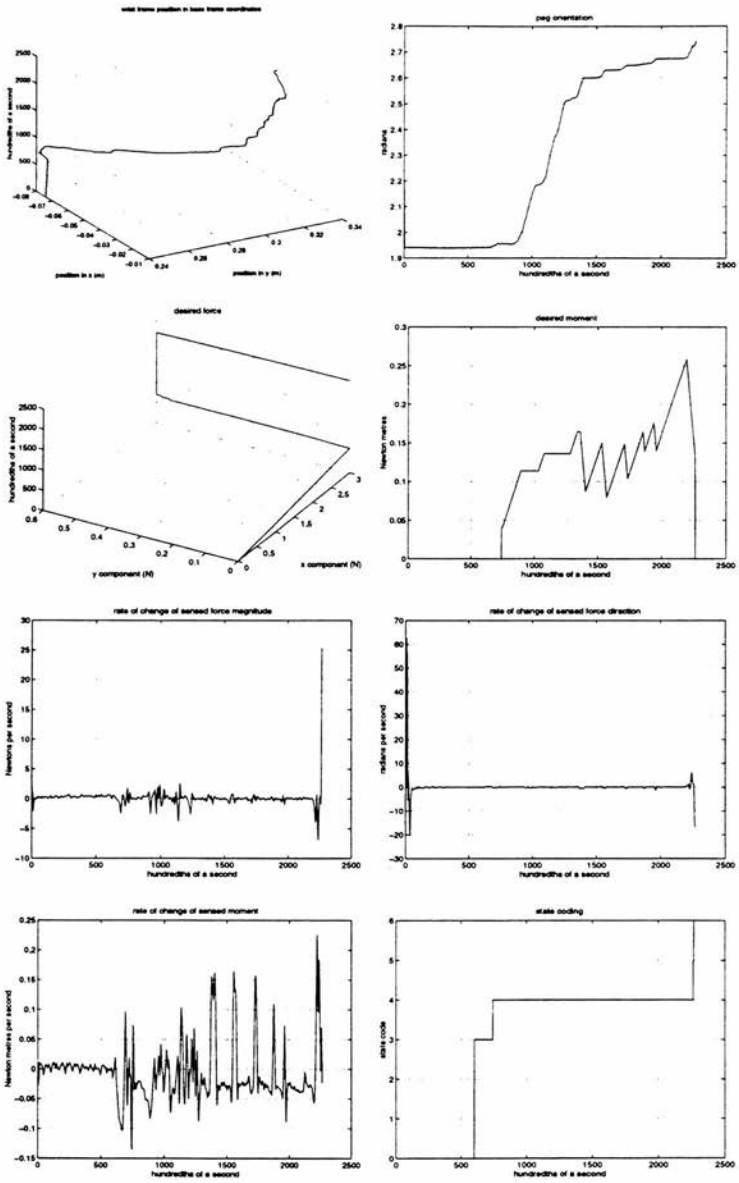


Figure 6.37: Example data generated during a dual peg in hole insertion. Parameters: two aluminium pegs $\frac{1}{4}$" wide by 1" long, with a 1" gap between them; arm configuration, right-handed; reorientation sense, anticlockwise; goal position (0.039, 0.330) metres; initial orientation, 1.94 rads; goal orientation, 2.73 rads.

of the pegs into its hole, the other one must be entering its hole at (approximately) the same time, because the tight tolerances mean that there is nowhere else it could be. Also for a tightly toleranced part, the CSO for a dual peg will look very similar to the CSO for a single peg, apart from an offset because the pegs in the dual peg case are offset from the x-axis of frame w.

In order to demonstrate that it is the inherently accommodating nature of our arm rather than the passive compliance in the wrist that enables our system to exhibit the robust behaviour that it does, we decided to run a further dual peg in hole experiment without any compliant material included, using instead an attachment made entirely of aluminium. The results are shown in figure 6.38.

6.4 Discussion

The behaviour of the strategies described above are quite robust. The experimental data presented here show that they are applicable over a wide range of circumstances. We haven't formally quantified how often that they fail, but a 1% failure rate for the block in corner task, and a 2% failure rate for the peg in hole task are probably reasonable estimates. Nearly all of these failures can be attributed to ambiguity in sensor signal interpretation rather than failure to perform mechanically. This is probably due, in part, to the fact that we empirically determined the thresholds that were used. It would be desirable to be able to automatically set these, or even better, somehow assess the characteristics of the sensor signals in real-time, so that what constitutes a discontinuity can be decided in the context of how the signal has performed in an immediately prior time interval. We intend to investigate the use of convolution bar masks for this purpose in the future [Ballard & Brown 82], [Haralick & Shapiro 93].

From the experiments that we have performed so far, it seems to be the case that tasks that traditional manipulators find difficult, our arm finds relatively straightforward. We suspect that the converse is also true. Putting a peg in a hole and turning a crank are relatively straightforward for our manipulator because there is a high degree of constraint to guide its motion. This is less true for the block in corner task, and as we saw, it is possible for it to lose a desired contact. We suspect that its performance

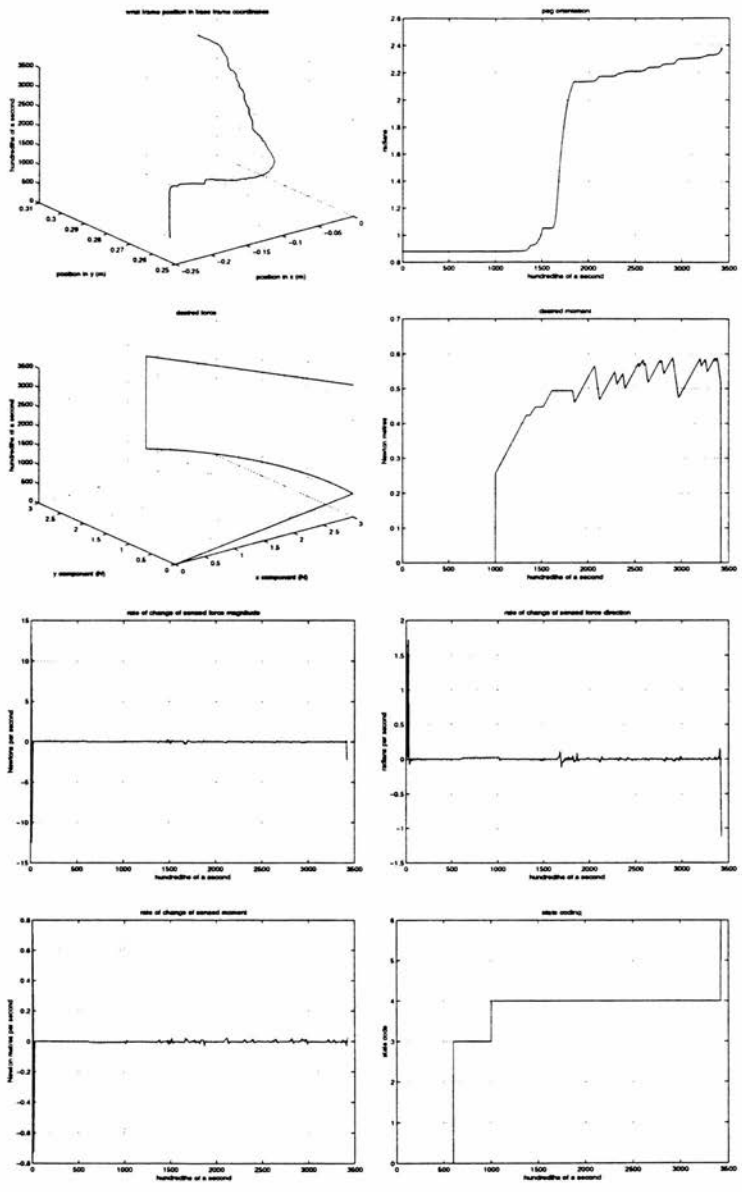


Figure 6.38: Example data generated during a dual peg in hole insertion. Parameters: two aluminium pegs $\frac{1}{4}$" wide by 1" long, with a 1" gap between them; arm configuration, right-handed; reorientation sense, anticlockwise; goal position (0.058, 0.314) metres; initial orientation, 0.88 rads; goal orientation, 2.30 rads.

at paint spraying, something that position-controlled robots are very good at, would leave something to be desired. Of course we could always add position servos to our arm if we wanted to (see for example [Asada *et al.* 83]), whereas it is difficult to get the inherently accommodating behaviour that our arm exhibits from an arm designed to be position controlled, as we discussed in Chapter 5.

Chapter 7

Conclusions and Further Work

7.1 What Have We Achieved?

Our motivation for embarking on this work was to try and make robots the flexible manufacturing devices they were always meant to be. We identified the limiting factor as being the critical susceptibility to spatial uncertainty of traditionally constructed industrial manipulators, and argued that this susceptibility was due to the fact that they were designed to be position controlled. Part of this problem arises due to the inertial properties of standard industrial manipulators making them unable to accommodate environmental constraints.

We noted that the object level programming language RAPT was constructed at a level of abstraction that could have provided a means for escaping from the spatial uncertainty problem, but because it was translated into statements executable by a position controlled robot the end behaviour was no better than any other position controlled system. We then noted that by moving to a situation where sliding is the basic motion primitive that a number of benefits could be derived.

The first of these is that the exact locations of the parts involved in the assembly no longer need to be known, because sliding of one part over another means that you implicitly know their relative locations, and after all, assembly is just an attempt to achieve a desired set of relative locations of the parts rather than some desired 'absolute' positioning in space. The other major benefit is that the assembly proceeds in terms of contact configurations, *i.e.* at the level of abstraction that RAPT programs

describe a task. The advantage of this being that these configurations are intrinsic properties of the task that are invariant with respect to part size and location. By generating behaviour in terms of transitions between contact configurations we arrived at force application strategies that do not require precise metric spatial information. Note that we were able to do this despite the fact our states were defined in terms of the set of contact configurations that could possibly hold at a particular point in a task. An essential tool in achieving this was the ability to be able to perform a search in wrench space to find the boundary of the region defining static equilibrium.

Because we were able to realise reliable behaviour at the level of abstraction of contact configurations we were then in a position to attempt to build on this and raise the level of programming abstraction further by stringing together a series of state transitions. As a consequence we were able to deduce assembly strategies that are task-invariant, thereby forming the basis of a task-level interface to the user. The end result is that we have potentially simplified the programming interface to assembly robots.

We claim that the positional and dimensional invariance of our technique that arises because we work in terms of shape and singularities, is an essential property for realising task-level robotic assembly. Moreover, we believe that it is the fact that there are no prespecified trajectories in either sensory or motor space that allows us to achieve reliable behaviour whilst moving to a higher level of abstraction, whereas the reason that other attempts have failed is because they have tried to be too precise in dealing with metric spatial properties.

A possible explanation for this situation is that historically, assembly planning work has tended to be performed by computer scientists, and robot control by control engineers. Position control has traditionally formed the 'bridge' or 'interface' between the two groups. The assumption seems to have been that the computer scientists would provide plans in geometric terms that the robots designed by control engineers would execute as sequences of desired positions, *i.e.* trajectories.

As a result of Mason's work on constraint frames, and since modelling friction in terms of the friction cone allows a number of rigid body mechanics problems to be expressed as problems in computational geometry, so that the design of admittance control matrices has been attacked by computer scientists, the assumption appears to

have become that the control engineer would deliver a machine that could be either force- or position-controllable in specified degrees of freedom.

But, as we have seen, position control creates the spatial uncertainty problem, and force control implemented on a manipulator designed to be position controlled is fraught with instability problems: hybrid force/position control inherits both of these limitations, as well as having limitations of its own concerning the independence of the force and position controlled sub-spaces when mapped into joint space. The approach described here circumvents these problems by starting with a manipulator designed for contact tasks. Moreover, because our control regime allows the possibility of an uncontrolled degree of freedom we are able to perform an on-line search for the 'decision boundaries' in the mechanics of contact, and so allow the operation of our arm to be guided by environmental constraints.

In the approach espoused here both sensing and motion are inextricably intertwined as constituents of the process of performing the task, so that when the user requires the execution of a peg-in-hole program (say), what they get is an indivisible package of sensorimotor behaviour appropriate to the task in hand. Moreover, the sensing used is dominated by an exteroceptive mode resulting in the actual behaviour that the robot exhibits for an instantiation of a task being shaped by the actual conditions between the robot and the environment. In fact it is not only sensing of the environment that shapes our robot's behaviour, the environment itself has a direct influence. Consequently this work could be viewed as a logical development of the work of Mason *et al.* on the mechanics of manipulation, but in our case we let the mechanics of a situation not only decide the object behaviour, but the robot behaviour as well. The following quote is apposite:

> At present the *trajectory* is the de facto model of manipulator action. For instance, the typical programming language takes a goal trajectory specified by a sequence of configurations and an interpolating function. Even in automatic planning research, the trajectory is the most usual model of action. The trajectory model of action is also reflected in the design of robot control systems, which usually attempt to control the absolute position of the manipulator.

> Funnel operations are profoundly different from positioning operations, requiring a different model of action. When interactions with the task environment are allowed to alter the motion of the manipulator, the trajectory becomes a priori indeterminate. Hence the trajectory is not a suitable representation for funnel operations. [Mason 85, p.547]

Nor, we maintain, for rigid body parts mating. In recent years the work of Mason and others has provided a firm foundation for the planning of pushing operations. But it is not just pushing that is a useful motion primitive, it is *sliding* in general.

Another way of viewing the work described here would be as complementary to Erdmann's work on the design of sensors around the information necessary to achieve a task [Erdmann 92]: what we have done here is designed a *sensitive actuator* (rather than a sensor) around the requirements of task-level assembly.

7.2 Further Work

Although we consider the work described in this thesis as progress towards a task-level programming interface, at the moment there is one major drawback: in order to generate an invariant assembly strategy for some previously unseen task it requires someone with a keen insight into both the robot's operation and the mechanics of contact.

We found the three degree of freedom joystick invaluable as a way of evaluating force application strategies before coding them in Turbo Pascal for the arm to act autonomously. In the case of rotating a block in a corner it was the ability to experiment that showed us how to go about analysing the task. As a consequence we consider that it would be useful to include as part of the system a motorised but backdrivable joystick. The idea is that the joystick motors will exert wrenches on the user's hand that are proportional to the wrenches that the robot arm experiences as it progresses through a task. In this way users can become experts at some particular task by using their intuitive manipulation skills, and once they have become competent at the task the system will endeavour to acquire their skill from them. We believe that this skill acquisition process will be aided by the state/action model used in our system, as well

as the division into isotonic and isometric phases.

In fact work on this is already under way under EPSRC grant GR/K68271. We have chosen to use limited angle direct-drive motors (controlled using motor current regulating feedback loops) as the technology for driving this joystick. This provides an attractive symmetry between the joystick and the robot arm.

We are hoping that the insight that we gain working with our force-reflecting joystick may one day put us in a position where we can let the system explore a task for itself and construct an assembly strategy by trial and error. The fact that our model of assembly tasks discretises the world into contact states means that it potentially lends itself to reinforcement learning techniques [Sutton 92].

At the moment we have only looked at manipulating rigid body components using the approach described here. We intend to remove this self-imposed restriction in the near future and consider objects with some compliance. One task we intend to consider is a planar analogue of inserting a bayonet fit light-bulb. This task consists of an 'L'-shaped part with a spring-loaded plunger on its 'underside', that is to snap-fit into a 'L'-shaped cavity.

So far all of our experiments have been performed in the horizontal plane despite the fact that EDDIE was designed to be operated under a certain amount of gravitational influence (see Appendix D). If left uncompensated gravitational forces are likely to detrimentally affect both the control and the sensing that our system performs. We anticipate dealing with this by using feedforward gravity compensation, *cf.* [Asada *et al.* 83], [Khosla & Kanade 88], [An *et al.* 88], *i.e.* we would need to continuously evaluate the following to drive our arm:

$$\boldsymbol{\tau} = {}^{w}\mathbf{J}^{T}(\mathbf{q})\,{}^{w}\mathbf{w} + \mathbf{G}_{e}(\mathbf{q}) \tag{7.1}$$

which is a version of equation B.23 using the Jacobian attached to the wrist frame of reference as defined in B.34, ${}^{w}\mathbf{w}$ is the desired wrench in the wrist frame of reference, and $\mathbf{G}_{e}(\mathbf{q})$ is a vector of gravity compensation terms given by equations D.3 to D.5. Even if this arrangement did not achieve perfect compensation, the fact that our strategies rely on a control regime that searches for the freedoms afforded by the environmental configuration, and a perceptual regime that looks for sensory discontinuities, rather

than trying to achieve any prespecified trajectory in either the control or perceptual domains, we expect to work in our favour. Moreover, our control regime does not rule out the possibility of using closed loop force control should it be necessary to rely on the action of a servo to 'iron out' any errors or unknowns in our gravity compensation. The construction of our arm would be favourable towards stable closed loop force control since the competition between intrinsic and feedback generated impedances that Colgate [Colgate 89] reports would be avoided. What remains to be seen is whether closed loop force control could work in tandem with our velocity limiting loop.

We are also interested in extending our system to include other sensory modalities. We believe that the technique of using discontinuities in a sensory field to signal significant events is an approach with general applicability. The most obvious next sense to consider is that of vision. Here we expect the aspect graph of Koenderink and Van Doorn discussed in Chapter 3 to be directly applicable to visual servoing tasks [Hashimoto 93]. Since at the moment our arm's behavioural repertoire is limited to tasks in which it can be informed by its force/torque sensor, *i.e.* contact tasks, addition of vision would provide the complementary capability of being able to navigate free space. This would serve to get the arm into a situation where tactile sensing is informative. The question arises, however, as to how you use position information with an arm that is designed to be force controlled. One possible answer is to emulate what it appears that animals do when controlling biological arms. It appears that muscles can be modelled as tunable springs; that is, the muscle length that corresponds to an equilibrium can be actively controlled by the central nervous system [Bizzi *et al.* 82], [Bizzi *et al.* 84]. It is possible to imagine a virtual equilibrium point that would determine where a muscle would be should it be in equilibrium. If this equilibrium point is controlled over time it gives rise to a virtual trajectory [Hogan 84a] which controls the progress of a joint's motion. This can be generalised to groups of muscles, thus determining the behaviour of a whole limb [Hogan 85b], [Flash 87]. This model is very similar to the notion of stiffness control [Salisbury 80] which effectively results in the end-effector of a robot arm emulating a collection of springs in equilibrium. If we used a virtual trajectory in combination with stiffness control it would be possible to use positional information (such as that from a vision system) to guide the motion of our arm.

We also have a suspicion that the qualitative/state-based approach described here might simplify the complexity of multi-actuator situations, *e.g.* dextrous manipulation using a multi-fingered hand [Venkataraman & Iberall 90]. Being able to decompose a dextrous manipulation task into qualitatively similar states would significantly reduce the complexity of the problem.

Bibliography

[Abbati-Marescotti *et al.* 90] A. Abbati-Marescotti, C. Bonivento, and C. Melchiorri. On the invariance of the hybrid position/force control. *J. of Intelligent and Robotic Systems*, 3:233–250, 1990.

[Akella & Mason 95] S. Akella and M.T. Mason. Parts orienting by push-aligning. In *IEEE Int. Conf. on Robotics and Automation*, pages 414–420, Nagoya, Japan, May 1995.

[Akella 95] S. Akella. Personal communication, 17^{th} April, 1995.

[Ambler & Corner 82] A.P. Ambler and D.F. Corner. RAPT1 users manual. Department of Artificial Intelligence, University of Edinburgh, Edinburgh, Scotland, 1982.

[Ambler & Popplestone 75] A.P. Ambler and R.J. Popplestone. Inferring the position of bodies from specified spatial relationships. *Artificial Intelligence*, 6:157–174, 1975.

[An *et al.* 88] C.H. An, C.G. Atkeson, and J.M. Hollerbach. *Model-Based Control of a Robot Manipulator*. MIT Press, Cambridge, MA, 1988.

[Arai & Kinoshita 81] T. Arai and N. Kinoshita. The part mating forces that arise when using a worktable with compliance. *Assembly Automation*, pages 204–210, August 1981.

[Arai 85] T. Arai. Application of knowledge engineering on automatic assembly of parts with complicated shapes. In *Proc.* 6^{th} *Int. Conf. Assembly Automation*, pages 67–76, Birmingham, England, May 1985.

[Armstrong 88] B.S.A. Armstrong. *Dynamics for Robot Control: Friction Modeling and Ensuring Excitation During Parameter Identification*. Unpublished PhD thesis, Department of Computer Science, Stanford University, Stanford, CA, May 1988. (Also: Technical Report STAN-CS-88-1205, Department of Computer Science, Stanford University).

[Asada & Hirai 90] H. Asada and S. Hirai. Towards a symbolic-level force feedback: Recognition of assembly process states. In H. Miura and S. Arimoto, editors, *Proc. 5th Int. Symp. Robotics Research*, pages 341–346. MIT Press, Cambridge, MA, 1990. (Held in Tokyo, Japan, August 1989).

[Asada & Izumi 89] H. Asada and H. Izumi. Automatic program generation from teaching data for the hybrid control of robots. *IEEE Trans. on Robotics and Automation*, 5(2):166–173, 1989.

[Asada & Kanade 83] H. Asada and T. Kanade. Design of direct-drive mechanical arms. *Trans. ASME, J. of Vibration, Acoustics, Stress, and Reliability in Design*, 105:312–316, 1983. (Also: Carnegie-Mellon University Robotics Institute Technical Report CMU-RI-TR-81-1).

[Asada & Slotine 86] H. Asada and J.-J.E. Slotine. *Robot Analysis and Control*. John Wiley & Sons, New York, NY, 1986.

[Asada & Youcef-Toumi 87] H. Asada and K. Youcef-Toumi. *Direct-Drive Robots: Theory and Practice*. MIT Press, Cambridge, MA, 1987.

[Asada 83] H. Asada. A geometrical representation of manipulator dynamics and its application to arm design. *Trans. ASME, J. Dynamic Systems, Measurement, and Control*, 105:131–135, 142, 1983.

[Asada 90] H. Asada. Teaching and learning of compliance using neural nets: Representation and generation of nonlinear compliance. In *IEEE Int. Conf. on Robotics and Automation*, pages 1237–1244, Cincinnati, OH, May 1990.

[Asada *et al.* 83] H. Asada, T. Kanade, and I. Takeyama. Control of a direct-drive arm. *Trans. ASME, J. Dynamic Systems, Measurement, and Control*, 105:136–142, 1983. (Also: Carnegie-Mellon University Robotics Institute Technical Report CMU-RI-TR-82-4).

[Ball 00] R.S. Ball. *A Treatise on the Theory of Screws*. Cambridge University Press, Cambridge, England, 1900.

[Ballard & Brown 82] D.H. Ballard and C.M. Brown. *Computer Vision*. Prentice-Hall, Inc., Englewood Cliffs, NJ, 1982.

[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. 3rd Int. Conf. on Robot Vision and*

Sensory Controls, pages 471–479, Cambridge, MA, November 1983.

[Barto 87] A.G. Barto. An approach to learning control surfaces by connectionist systems. In M.A. Arbib and A.R. Hanson, editors, *Vision, Brain, and Cooperative Computation*, pages 665–701. MIT Press, Cambridge, MA, 1987.

[Bizzi *et al.* 82] E. Bizzi, W. Chapple, and N. Hogan. Mechanical properties of muscles: Implications for motor control. *Trends in Neurosciences*, 5:395–398, November 1982.

[Bizzi *et al.* 84] E. Bizzi, N. Accornero, W. Chapple, and N. Hogan. Posture control and trajectory formation during arm movement. *J. of Neoroscience*, 4(11):2738–2744, November 1984.

[Bland 86] C.J. Bland. Peg-hole assembly: A literature survey. Technical Report ARC 10, Department of Mechanical and Manufacturing Systems Engineering, University of Wales Institute of Science and Technology, Cardiff, Wales, January 1986.

[Bonner & Shin 82] S. Bonner and K.G. Shin. A comparative study of robot languages. *IEEE Computer*, 15(12):82–96, 1982.

[Boothroyd 80] G. Boothroyd. Design for assembly handbook. Department of Mechanical Engineering, University of Massachusetts, Amherst, MA, 1980.

[Brooks 82] R.A. Brooks. Symbolic error analysis and robot planning. *Int. J. of Robotics Research*, 1(4):29–68, 1982.

[Brost & Christiansen 93] R.C. Brost and A.D. Christiansen. Probabilistic analysis of manipulation tasks: A research agenda. In *IEEE Int. Conf. on Robotics and Automation*, pages 549–556, Atlanta, GA, May 1993.

[Brost & Mason 90] R.C. Brost and M.T. Mason. Graphical analysis of planar rigid body dynamics with multiple frictional contacts. In H. Miura and S. Arimoto, editors, *Proc. 5^{th} Int. Symp. Robotics Research*, pages 293–300. MIT Press, Cambridge, MA, 1990. (Held in Tokyo, Japan, August 1989).

[Brost 85] R.C. Brost. Planning robot grasping motions in the presence of uncertainty. Technical Report CMU-RI-TR-85-12, The Robotics Institute, Carnegie-Mellon University, Pittsburgh, PA, July 1985.

[Brost 89] R.C. Brost. Computing metric and topological properties of configuration space obstacles. In *IEEE Int. Conf. on Robotics and Automation*, pages 170–176, Scottsdale, AZ, May 1989.

[Brost 91a] R.C. Brost. *Analysis and Planning of Planar Manipulation Tasks.* Unpublished PhD thesis, Department of Computer Science, Carnegie-Mellon University, Pittsburgh, PA, January 1991. (Also: Technical Report CMU-CS-91-149, Department of Computer Science, Carnegie-Mellon University).

[Brost 91b] R.C. Brost. Computing the possible rest configurations of two interacting polygons. In *IEEE Int. Conf. on Robotics and Automation*, pages 686–693, Sacramento, CA, April 1991.

[Brost 95] R.C. Brost. Personal communication, 16^{th} May, 1995.

[Buckley 87] S.J. Buckley. *Planning and Teaching Compliant Motion Strategies.* Unpublished PhD thesis, Department of Electrical Engineering and Computer Science, Massachusetts Institute of Technology, Cambridge, MA, 1987. (Also: MIT AI Lab. Technical Report 936).

[Buckley 89] S.J. Buckley. Teaching compliant motion strategies. *IEEE Trans. on Robotics and Automation*, 5(1):112–118, 1989.

[Byrne & Hopkins 91] C.B. Byrne and S.H. Hopkins. An expert system for the identification of assembly tasks. In *Proc. Eurotech '91*, Birmingham, England 1991. Paper C415/078.

[Byrne 87] C. Byrne. A literature survey of classification systems for assembly processes. Technical Report ARC 17, Department of Mechanical and Manufacturing Systems Engineering, University of Wales Institute of Science and Technology, Cardiff, Wales, October 1987.

[Byrne 88] C. Byrne. A description of the methodology for the survey and classification of assembly tasks. Technical Report ARC 23, Department of Mechanical and Manufacturing Systems Engineering, University of Wales Institute of Science and Technology, Cardiff, Wales, August 1988.

[Byrne 89] C. Byrne. A survey of mechanical and electromechanical assembly tasks. Technical Report ARC 32,

Department of Mechanical and Manufacturing Systems Engineering, University of Wales Institute of Science and Technology, Cardiff, Wales, June 1989.

[Caine 90] M.E. Caine. The effect of part shape on planar insertion operations. In *Proc. 2nd ASME Conf. on Flexible Assembly Systems*, pages 133–138, Chicago, IL, September 1990.

[Caine 93] M.E. Caine. *The Design of Shape from Motion Constraints.* Unpublished PhD thesis, Department of Mechanical Engineering, Massachusetts Institute of Technology, Cambridge, MA, January 1993. (Also: MIT AI Lab. Technical Report 1425).

[Caine *et al.* 89] M.E. Caine, T. Lozano-Pérez, and W.P. Seering. Assembly strategies for chamferless parts. In *IEEE Int. Conf. on Robotics and Automation*, pages 472–477, Scottsdale, AZ, May 1989.

[Canny & Goldberg 94] J.F. Canny and K.Y. Goldberg. RISC for industrial robotics: Recent results and open problems. In *IEEE Int. Conf. on Robotics and Automation*, pages 1951–1958, San Diego, CA, May 1994.

[Canudas *et al.* 87] C. Canudas, K.J. Åström, and K. Braun. Adaptive friction compensation in dc-motor drives. *IEEE J. of Robotics and Automation*, RA-3(6):681–685, 1987.

[Chang *et al.* 89] S.J. Chang, F. DiCesare, and G. Goldbogen. The generation of diagnostic heuristics for automated error recovery in manufacturing workstations. In *IEEE Int. Conf. on Robotics and Automation*, pages 522–527, Scottsdale, AZ, May 1989.

[Christiansen 91] A.D. Christiansen. *Automatic Acquisition of Task Theories for Robotic Manipulation.* Unpublished PhD thesis, Department of Computer Science, Carnegie-Mellon University, Pittsburgh, PA, May 1991. (Also: Technical Report CMU-CS-92-111, Department of Computer Science, Carnegie-Mellon University).

[Coiffet 83] P. Coiffet. *Robot Technology Vol. 2: Interaction with the Environment.* Kogan Page, London, England, 1983.

[Colgate & Hogan 89] E. Colgate and N. Hogan. An analysis of contact instability in terms of passive physical equivalents. In *IEEE Int. Conf. on Robotics and Automation*, pages 404–409, Scottsdale, AZ, May 1989.

[Colgate 89] E. Colgate. On the intrinsic limitations of force feedback compliance controllers. In Y. Youcef-Toumi and H. Kazerooni, editors, *Proc. ASME Winter Annual Meeting*, pages 23–30. American Society of Mechanical Engineers, New York, NY, San Fransisco, CA, December 1989. (Published as: *Robotics Research—1989*, DSC-Vol. 14).

[Cooper 87] C.R. Cooper. ASEA Industrial Robot Systems. Personal communication regarding the Soft Servo Control option of the IRB series of industrial robots, 25^{th} November, 1987.

[Corner *et al.* 83] D.F. Corner, A.P. Ambler, and R.J. Popplestone. Reasoning about the spatial relationships derived from a RAPT program for describing assembly by robot. In *Proc. 8th Int. Joint Conf. on Artificial Intelligence*, pages 842–844, Karlsruhe, Germany, August 1983.

[Craig 89] J.J. Craig. *Introduction to Robotics, 2nd edn.* Addison Wesley, Reading, MA, 1989.

[Davey & Selke 85] S.N. Davey and K.K.W. Selke. Error report on Flymo E25 assembly. Internal Report No. 24/85, Department of Electronic Engineering, University of Hull, Hull, England, 1985.

[De Schutter & Van Brussel 88a] J. De Schutter and H. Van Brussel. Compliant robot motion I: A formalism for specifying compliant motion tasks. *Int. J. of Robotics Research*, 7(4):3–17, 1988.

[De Schutter & Van Brussel 88b] J. De Schutter and H. Van Brussel. Compliant robot motion II: A control approach based on external loops. *Int. J. of Robotics Research*, 7(4):18–33, 1988.

[Deacon & Malcolm 94] G.E. Deacon and C. Malcolm. A robot system designed for task-level assembly. In *Int. Workshop on Advanced Robotics and Intelligent Machines*, Salford, England, March 1994.

[Deacon 90] G.E. Deacon. Achieving robotic assembly by sensing and controlling spatial relationships. In O. Kaynak, editor, *Proc. IEEE Int. Workshop on Intelligent Motion Control*, pages 159–164, Istanbul, Turkey, August 1990.

[DeFazio *et al.* 84] T.L. DeFazio, D.S. Seltzer, and D.E. Whitney. The instrumented remote centre compliance. *The Industrial Robot*, December:238–242, 1984.

[Desai & Volz 89] R.S. Desai and R.A. Volz. Identification and verification of termination conditions in fine motion in presence of sensor errors and geometric uncertainties. In *IEEE Int. Conf. on Robotics and Automation*, pages 800–807, Scottsdale, AZ, May 1989.

[Donald 86] B. Donald. Robot motion planning with uncertainty in the geometric models of the robot and environment: A formal framework for error detection and recovery. In *IEEE Int. Conf. on Robotics and Automation*, pages 1588–1593, San Fransisco, CA, April 1986.

[Donald 87] B.R. Donald. *Error Detection and Recovery for Robot Motion Planning with Uncertainty.* Unpublished PhD thesis, Department of Electrical Engineering and Computer Science, Massachusetts Institute of Technology, Cambridge, MA, July 1987. (Also: MIT AI Lab. Technical Report 982).

[Donald 88] B. Donald. Planning multi-step error detection and recovery strategies. In *IEEE Int. Conf. on Robotics and Automation*, pages 892–897, Philadelphia, PA, April 1988.

[Drake 77] S.H. Drake. *Using Compliance in Lieu of Sensory Feedback for Automatic Assembly.* Unpublished PhD thesis, Department of Mechanical Engineering, Massachusetts Institute of Technology, Cambridge, MA, September 1977. (Also: Technical Report T-657, Charles Stark Draper Laboratory, Cambridge, MA.).

[Dufay & Latombe 84] B. Dufay and J.-C. Latombe. An approach to automatic robot programming based on inductive learning. *Int. J. of Robotics Research*, 3(4):3–20, 1984.

[Duffy 90] J. Duffy. The fallacy of modern hybrid control theory that is based on "orthogonal complements" of twist and wrench spaces. *J. of Robotic Systems*, 7(2):139–144, 1990.

[Dupont 90] P.E. Dupont. Friction modeling in dynamic robot simulation. In *IEEE Int. Conf. on Robotics and Automation*, pages 1370–1376, Cincinnati, OH, May 1990.

[Durrant-Whyte 87] H.F. Durrant-Whyte. Sensor models and multisensor integration. In *Proc. Workshop on Spatial Reasoning and Multi-Sensor Fusion*, pages 303–312, Los Altos, CA, St. Charles, IL 1987. Morgan Kaufmann Publishers Inc.

[Durrant-Whyte 88] H.F. Durrant-Whyte. Uncertain geometry in robotics. *IEEE J. of Robotics and Automation*, 4(1):23–31, 1988.

[Electro-Craft Corporation 80] Electro-Craft Corporation. *DC Motors, Speed Controls, Servo Systems, 5^{th} Edition.* Robbins and Meyers, Eden Prairie, MN, 1980.

[Elosegui *et al.* 90] P. Elosegui, R.W. Daniel, and P.M. Sharkey. Joint servoing for robust manipulator force control. In *IEEE Int. Conf. on Robotics and Automation*, pages 246–251, Cincinatti, OH, May 1990.

[Engelberger 80] J.F. Engelberger. *Robots in Practice: Management and Applications of Industrial Robots.* Kogan Page, London, England, 1980.

[Eppinger & Seering 86] S.D. Eppinger and W.P Seering. On dynamic models of robot force control. In *IEEE Int. Conf. on Robotics and Automation*, pages 29–34, San Fransisco, CA, April 1986.

[Eppinger 88] S.D. Eppinger. *Modeling Robot Dynamic Performance for Endpoint Force Control.* Unpublished PhD thesis, Department of Mechanical Engineering, Massachusetts Institute of Technology, Cambridge, MA, September 1988. (Also: MIT AI Lab. Technical Report 1072).

[Erdmann & Mason 88] M.A. Erdmann and M.T. Mason. An exploration of sensorless manipulation. *IEEE J. of Robotics and Automation*, 4(4):369–379, 1988.

[Erdmann 84] M.A. Erdmann. On motion planning with uncertainty. Unpublished M.Sc. thesis, Department of Electrical Engineering and Computer Science, Massachusetts Institute of Technology, Cambridge, MA, August 1984. (Also: MIT AI Lab. Technical Report 810).

[Erdmann 92] M. Erdmann. Towards task-level planning: Action-based sensor design. Technical Report CMU-CS-92-116, Department of Computer Science, Carnegie-Mellon University, Pittsburgh, PA, February 1992. (Also: Carnegie-Mellon University Robotics Institute Technical Report CMU-RI-TR-92-03).

[Erdmann 94] M. Erdmann. On a representation of friction in configuration space. *Int. J. of Robotics Research*, 13(3):240–271, 1994.

[Featherstone 85] R. Featherstone. The simulator verification experiment. Working Paper No. 178, Department of Artificial Intelligence, University of Edinburgh, Edinburgh, Scotland, April 1985.

[Featherstone 87] R. Featherstone. *Robot Dynamics Algorithms.* Kluwer Academic Publishers, Boston, MA, 1987.

[Fisher & Mujtaba 92] W.D. Fisher and M.S. Mujtaba. Hybrid position/force control: A correct formulation. *Int. J. of Robotics Research*, 11(4):299–311, 1992.

[Fitzgibbon & Fisher 92] A.W. Fitzgibbon and R.B. Fisher. Practical aspect-graph derivation incorporating feature segmentation performance. In *British Machine Vision Association Conf.*, pages 580–589, Leeds, England 1992.

[Flash 87] T. Flash. The control of hand equilibrium trajectories in multi-joint arm movements. *Biological Cybernetics*, 57:257–274, 1987.

[Fleisher 88] W.A. Fleisher. How to select d.c. motors. *Machine Design*, November 1988.

[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, Edinburgh, Scotland, 1987.

[Fowler & Eppinger 91] H.C. Fowler and S.D. Eppinger. Bandwidth performance of a direct drive manipulator under joint torque and endpoint force control. In *IEEE Int. Conf. on Robotics and Automation*, pages 230–237, Sacramento, CA, April 1991.

[Gogoussis & Donath 87] A. Gogoussis and M. Donath. Coulomb friction joint and drive effects in robot mechanisms. In *IEEE Int. Conf. on Robotics and Automation*, pages 828–836, Raleigh, N. Carolina, March/April 1987.

[Gogoussis & Donath 88] A. Gogoussis and M. Donath. Coulomb friction effects on the dynamics of bearings and transmissions in precision robot mechanisms. In *IEEE Int. Conf. on Robotics and Automation*, pages 1440–1446, Philadelphia, PA, April 1988.

[Goldberg 90] K.Y. Goldberg. *Stochastic Plans for Robotic Manipulation.* Unpublished PhD thesis, Department of Computer Science, Carnegie-Mellon University, Pittsburgh, PA, August 1990.

[Goldenberg & Song 96] A.A. Goldenberg and P. Song. Analysis and design of position/force controllers. Part I: decomposition. Workshop WS3, *IEEE Int. Conf. on Robotics and Automation*, Minneapolis, MN, April 1996. To appear in the IEEE Trans. on Robotics and Automation.

[Goldman & Tucker 56] A.J. Goldman and A.W. Tucker. Polyhedral convex cones. In H.W. Kuhn and A.W. Tucker, editors, *Linear Inequalities and Related Systems*, pages 19–40. Princeton University Press, Princeton, NJ, 1956. Annals of Mathematical Studies, Number 38.

[Goswami & Peshkin 93] A. Goswami and M.A. Peshkin. Task-space/joint-space damping transformations for passive redundant manipulators. In *IEEE Int. Conf. on Robotics and Automation*, pages 642–647, Atlanta, GA, May 1993.

[Goyal *et al.* 91] S. Goyal, A. Ruina, and J. Papadopoulos. Planar sliding with dry friction. Part 1. Limit surface and moment function. *Wear*, 143:307–330, 1991.

[Greer 84] K. Greer. Physiology of motor control. In M.M. Smyth and A.M. Wing, editors, *The Physiology of Human Movement*, pages 17–46. Academic Press, Inc., London, England, 1984.

[Hager & Durrant-Whyte 88] G. Hager and H.F. Durrant-Whyte. Information and multi-sensor coordination. In J.F. Lemmer and L.N. Kanal, editors, *Uncertainty in Artificial Intelligence 2*, pages 381–394. Elsevier Science Publishers B.V. (North-Holland), 1988.

[Haralick & Shapiro 93] R. M. Haralick and L. G. Shapiro. *Computer and Robot Vision*, volume 1. Addison-Wesley, 1993.

[Hardy *et al.* 89] N.W. Hardy, D.P. Barnes, and M.H. Lee. Automatic diagnosis of task faults in flexible manufacturing systems. *Robotica*, 7(1):25–35, 1989.

[Hashimoto 93] K. Hashimoto, editor. *Visual Servoing: Real-Time Control of Robot Manipulators Based on Visual Sensory Feedback*. World Scientific Publishing Co. Ltd., River Edge, NJ, 1993.

[Hirai & Asada 93] S. Hirai and H. Asada. Kinematics and statics of manipulation using the theory of polyhedral convex cones. *Int. J. of Robotics Research*, 12(5):434–447, 1993.

[Hirai & Iwata 92] S. Hirai and K. Iwata. A model-based generation of damping control law for part-mating. In *Proc. IEEE/RSJ Int. Conf. on Intelligent Robots and Systems*, pages 494–499, Raleigh, NC, July 1992.

[Hirai 94] S. Hirai. Identification of contact states based on a geometric model for manipulative operations. *Advanced Robotics*, 8(2):139–155, 1994.

[Hogan 84a] N. Hogan. An organizing principle for a class of voluntary movements. *J. of Neoroscience*, 4(11):2745–2754, November 1984.

[Hogan 84b] N. Hogan. Some computational problems simplified by impedance control. In *Proc. ASME Conf. Comput. Eng.*, pages 203–209, 1984.

[Hogan 85a] N. Hogan. Impedance control: An approach to manipulation. *Trans. ASME, J. Dynamic Systems, Measurement, and Control*, 107:1–24, March 1985.

[Hogan 85b] N. Hogan. The mechanics of multi-joint posture and movement. *Biological Cybernetics*, 52:315–331, 1985.

[Hopkins *et al.* 88] S.H. Hopkins, C.J. Bland, and C.B. Byrne. A toolbox of assembly strategies. In *Proc.* 18^{th} *Int. Symp. on Industrial Robots*, pages 145–156, Lausanne, Switzerland, April 1988.

[Hunt 78] K.H. Hunt. *Kinematic Geometry of Mechanisms*. Clarendon Press, Oxford, England, 1978.

[Inoue 79] H. Inoue. Force feedback in precise assembly tasks. In P.H. Winston and R.H. Brown, editors, *Artificial Intelligence: An MIT Perspective*. MIT Press, Cambridge, MA, 1979. (Also: Memo No. 308, Artificial Intelligence Laboratory, MIT, Cambridge, MA, August 1974).

[Johnson & Hill 85] D.G. Johnson and J.J. Hill. A Kalman filter approach to sensor-based robot control. *IEEE J. of Robotics and Automation*, RA-1(3):159–162, 1985.

[Jones 95] E. Jones. *Learning in Behavioural Robotics*. Unpublished PhD thesis, Department of Artificial Intelligence, University of Edinburgh, Edinburgh, Scotland, 1995.

[Kanade & Schmitz 85] T. Kanade and D. Schmitz. Development of CMU direct-drive arm II. In *Proc. 1985 American Control Conference*, pages 703–709, Boston, MA, June 1985.

[Khatib 87] O. Khatib. A unified approach for motion and force control of robot manipulators: The operational space formulation. *IEEE J. of Robotics and Automation*, RA-3(1):43–53, 1987.

[Khosla & Kanade 88] P.K. Khosla and T. Kanade. Experimental evaluation of nonlinear feedback and feedforward control schemes. *Int. J. of Robotics Research*, 7(1):18–28, 1988.

[Klafter *et al.* 89] R.D. Klafter, T.A. Chmielewski, and M. Negin. *Robotic Engineering: An Integrated Approach.* Prentice-Hall International, Englewood Cliffs, NJ, 1989.

[Koenderink & Van Doorn 77] J.J. Koenderink and A.J. Van Doorn. How an ambulant observer can construct a model of the environment from the geometrical structure of the visual inflow. In G. Hauske and E. Butenandt, editors, *Kybernetik*, pages 224–247. Oldenburg, Munich, Germany, 1977.

[Koenderink & Van Doorn 79] J.J. Koenderink and A.J. Van Doorn. The internal representation of solid shape with respect to vision. *Biological Cynernetics*, 32:211–216, 1979.

[Koenderink 86] J.J. Koenderink. Optic flow. *Vision Research*, 26(1):161–180, 1986.

[Koutsou 81] A. Koutsou. A survey of model-based robot programming languages. Working Paper No. 108, Department of Artificial Intelligence, University of Edinburgh, Edinburgh, Scotland, December 1981.

[Koutsou 86] A. Koutsou. *Planning Motion in Contact to Achieve Parts Mating.* Unpublished PhD thesis, Department of Artificial Intelligence, University of Edinburgh, Edinburgh, Scotland, 1986.

[Krishnaswamy & Seering 94] S. Krishnaswamy and W.P. Seering. Effects of kinematic force errors on robot task performance. In *IEEE Int. Conf. on Robotics and Automation*, pages 2727–2732, San Diego, CA, May 1994.

[Latombe & Mazer 81] J-C. Latombe and E. Mazer. LM: a high-level programming language for controlling assembly robots. In *Proc.* 11^{th} *Int. Symp. on Industrial Robots*, pages 683–690, Tokyo, Japan, October 1981.

[Latombe 83] J-C. Latombe. Toward automatic robot programming. In *Int. Conf. on Advanced Robotics*, pages 203–213, Tokyo, Japan, September 1983.

[Laugier 88] C. Laugier. Planning robot motions in the SHARP system. In B. Ravani, editor, *CAD Based Programming for Sensory Robots*, pages 151–187. Springer-Verlag, Berlin, Germany, 1988. NATO ASI Series, Vol. F50.

[Laugier 89] C. Laugier. Planning fine motion by reasoning in the contact space. In *IEEE Int. Conf. on Robotics and Automation*, pages 653–659, Scottsdale, AZ, May 1989.

[Lee *et al.* 84] M.H. Lee, N.W. Hardy, and D.P. Barnes. Research into automatic error recovery. In *Proc. IMechE Conf. on UK Robotics Research*, pages 65–69, London, England, December 1984.

[Lieberman & Wesley 77] L.I. Lieberman and M.A. Wesley. AUTOPASS: an automatic programming system for computer controlled mechanical assembly. *IBM J. of Research and Development*, 21(4):321–333, 1977.

[Lipkin & Duffy 88] H. Lipkin and J. Duffy. Hybrid twist and wrench control for a robotic manipulator. *Trans. ASME, J. of Mechanisms, Transmissions, and Automation in Design*, 110:138–144, June 1988.

[Little 92] R. Little. Force/torque sensing in robotic manufacturing. *Sensors*, 9(11), 1992.

[Lončarić 87] J. Lončarić. Normal forms of stiffness and compliance matrices. *IEEE J. of Robotics and Automation*, RA-3(6):567–572, 1987.

[Lozano-Pérez & Wesley 79] T. Lozano-Pérez and M.A. Wesley. An algorithm for planning collision free paths among polyhedral obstacles. *Communications of the ACM*, 22(10):560–570, 1979.

[Lozano-Pérez 76] T. Lozano-Pérez. The design of a mechanical assembly system. Unpublished M.Sc. thesis, Department of Electrical Engineering, Massachusetts Institute of Technology, Cambridge, MA, December 1976. (Also: MIT AI Lab. Technical Report 397.).

[Lozano-Pérez 81] T. Lozano-Pérez. Automatic planning of manipulator transfer movements. *IEEE Trans. on Systems, Man, and Cybernetics*, SMC-11(10):681–698, 1981.

[Lozano-Pérez 82] T. Lozano-Pérez. Task planning. In M. Brady, J.M. Hollerbach, T.L. Johnson, T. Lozano-Pérez, and M.T. Mason, editors, *Robot Motion: Planning*

and Control, pages 473–498. MIT Press, Cambridge, MA, 1982.

[Lozano-Pérez 83a] T. Lozano-Pérez. Robot programming. *Proc. IEEE*, 71(7):821–841, 1983.

[Lozano-Pérez 83b] T. Lozano-Pérez. Spatial planning: A configuration space approach. *IEEE Trans. on Computers*, C-32(2):108–120, 1983.

[Lozano-Pérez *et al.* 84] T. Lozano-Pérez, M.T. Mason, and R.H. Taylor. Automatic synthesis of fine-motion strategies for robots. *Int. J. of Robotics Research*, 3(1):3–24, 1984.

[Lozano-Pérez *et al.* 92] T. Lozano-Pérez, J.L. Jones, E. Mazer, and P.A. O'Donnell. *Handey: A Robot Task Planner*. MIT Press, Cambridge, MA, 1992.

[Luh 83] J.Y.S. Luh. Conventional controller design for industrial robots—a tutorial. *IEEE Trans. on Systems, Man, and Cybernetics*, SMC-13(3):298–316, 1983.

[Luh *et al.* 83] J.Y.S. Luh, W.D. Fisher, and R.P.C. Paul. Joint torque control by a direct feedback for industrial robots. *IEEE Trans. on Automatic Control*, AC-28(2):153–161, 1983.

[Lynch & Mason 93] K.M. Lynch and M.T. Mason. Pulling by pushing, slip with infinite friction, and perfectly rough surfaces. In *IEEE Int. Conf. on Robotics and Automation*, pages 745–751, Atlanta, GA, May 1993.

[Lynch 92] K.M. Lynch. The mechanics of fine manipulation by pushing. In *IEEE Int. Conf. on Robotics and Automation*, pages 2269–2276, Nice, France, May 1992.

[Lynch 96] K.M. Lynch. *Nonprehensile Robotic Manipulation: Controllability and Planning*. Unpublished PhD thesis, Robotics Institute, Carnegie-Mellon University, Pittsburgh, PA, March 1996. (Also: Carnegie-Mellon University Robotics Institute Technical Report CMU-RI-TR-96-05).

[MacMillan 36] W.D. MacMillan. *Dynamics of Rigid Bodies*. McGraw-Hill, New York, NY, 1936.

[Malcolm & Fothergill 87] C.A. Malcolm and A.P. Fothergill. Some architectural implications of the use of sensors. In U. Rembold and K. Hörman, editors, *Languages for Sensor-Based Control in Robotics*, pages 101–122. Springer-Verlag, Berlin, Germany, 1987. NATO ASI Series, Vol. F29.

[Mason & Wang 88] M.T. Mason and Y. Wang. On the inconsistency of rigid-body frictional planar mechanics. In *IEEE Int. Conf. on Robotics and Automation*, pages 524–528, Philadelphia, PA, April 1988.

[Mason 78] M.T. Mason. Compliance and force control for computer controlled manipulators. Unpublished M.Sc. thesis, Department of Electrical Engineering and Computer Science, Massachusetts Institute of Technology, Cambridge, MA, May 1978. (Also: MIT AI Lab. Technical Report 515).

[Mason 81] M.T. Mason. Compliance and force control for computer controlled manipulators. *IEEE Trans. on Systems, Man, and Cybernetics*, SMC-11(6):418–432, 1981.

[Mason 82] M.T. Mason. *Manipulator Grasping and Pushing Operations.* Unpublished PhD thesis, Department of Electrical Engineering and Computer Science, Massachusetts Institute of Technology, Cambridge, MA, June 1982. (Also: MIT AI Lab. Technical Report 690, and reprinted in: M.T. Mason and J.K. Salisbury, editors, *Robot Hands and the Mechanics of Manipulation*, MIT Press, Cambridge, MA, 1985).

[Mason 84] M.T. Mason. Automatic planning of fine motions: Correctness and completeness. In *IEEE Int. Conf. on Robotics and Automation*, pages 492–503, Atlanta, GA, March 1984. (Also: Carnegie Mellon University Robotics Institute Technical Report CMU-RI-TR-83-18).

[Mason 85] M.T. Mason. The mechanics of manipulation. In *IEEE Int. Conf. on Robotics and Automation*, pages 544–548, St. Louis, MO, March 1985.

[Mason 86] M.T. Mason. Mechanics and planning of manipulator pushing operations. *Int. J. of Robotics Research*, 5(3):53–71, 1986.

[Mason 89] M.T. Mason. How to push a block along a wall. In *Proc. NASA Conf. Space Telerobotics*, pages 173–182, Pasadena, CA, January 1989.

[Mason 91] M.T. Mason. Two graphical methods for planar contact problems. In *Proc. IEEE/RSJ Int. Workshop on Intelligent Robots and Systems, IROS '91*, pages 443–448, Osaka, Japan, November 1991.

[Mathewson & Newman 95] B.B. Mathewson and W.S. Newman. Integration of force strategies and natural admittance control. In *Video Proceedings of the IEEE Int. Conf. on Robotics and Automation*, Nagoya, Japan, May 1995.

[Mazer 82] E. Mazer. An algorithm for computing the relative position between two objects from symbolical specifications. IMAG, Research Report No. 297, March 1982.

[Mazer 84] E. Mazer. LM-GEO: Geometric programming of assembly robots. In A. Danthine and M. Geradin, editors, *Advanced Software in Robotics*, pages 99–110. Elsevier Science Publishers B.V. (North-Holland), 1984.

[McCallion & Wong 75] H. McCallion and P.C. Wong. Some thoughts on the automatic assembly of a peg and a hole. *The Industrial Robot*, 2:141–146, December 1975.

[McCarragher & Asada 93] B.J. McCarragher and H. Asada. Qualitative template matching using dynamic process models for state transition recognition of robotic assembly. *Trans. ASME, J. Dynamic Systems, Measurement, and Control*, 115:261–269, June 1993.

[McCarragher 93] B.J. McCarragher. Robotic assembly using trajectory planning and discrete event modelling. In 2^{nd} *IEEE Conf. on Emerging Technologies and Factory Automation*, pages 188–196, Cairns, Australia, September 1993.

[Meriam 75] J.L. Meriam. *Statics, 2^{nd} edition.* John Wiley and Sons, Inc., New York, NY, 1975.

[Morris 92] K.A. Morris. Has force/torque sensing gained factory acceptance? In *Proc. IEEE/RSJ Int. Conf. on Intelligent Robots and Systems*, Raleigh, NC, July 1992.

[Mundy & Zisserman 92] J.L. Mundy and A. Zisserman, editors. *Geometric Invariants in Computer Vision.* MIT Press, Cambridge, MA, 1992. (Proceedings of a Workshop on Applications of Invariance in Computer Vision, Reykjavik, Iceland, March 25-28, 1991).

[Nakamura 91] Y. Nakamura. *Advanced Robotics: Redundancy and Optimization.* Addison-Wesley Publishing Company, Reading, MA, 1991.

[Nevins & Whitney 73] J.L. Nevins and D.E. Whitney. The force vector assembly concept. In *Proc. 1st Int. Conf. Robots and Manipulator Systems*, pages 273–288, Udine, Italy 1973.

[Nevins & Whitney 78] J.L. Nevins and D.E. Whitney. Computer controlled assembly. *Scientific American*, pages 62–74, February 1978.

[Nguyen 86] V.-D. Nguyen. The synthesis of stable force-closure grasps. Unpublished M.Sc. thesis, Department of Electrical Engineering and Computer Science, Massachusetts Institute of Technology, Cambridge, MA, May 1986. (Also: MIT AI Lab. Technical Report 905).

[Ohwovoriole & Roth 81] M.S. Ohwovoriole and B. Roth. An extension of screw theory. *Trans. ASME, J. of Mechanical Design*, 103:138–144, October 1981.

[Ohwovoriole 80] M.S. Ohwovoriole. *An Extension of Screw Theory and its Application to the Automation of Industrial Assemblies.* Unpublished PhD thesis, Department of Mechanical Engineering, Stanford University, Stanford, CA, April 1980. (Also: Memo 338, AI Lab., Stanford University).

[Owen 85] T. Owen. *Assembly with Robots.* Kogan Page, London, England, 1985.

[Paul 81] R.P. Paul. *Robot Manipulators: Mathematics, Programming and Control.* MIT Press, Cambridge, MA, 1981.

[Paul 87] R.P. Paul. Problems and research issues associated with the hybrid control of force and displacement. In *IEEE Int. Conf. on Robotics and Automation*, pages 1966–1971, Raleigh, NC, March/April 1987.

[Peshkin 86] M.A. Peshkin. *Planning Robotic Manipulation Strategies for Sliding Objects.* Unpublished PhD thesis, Department of Physics and the Robotics Institute, Carnegie-Mellon University, Pittsburgh, PA, November 1986.

[Peshkin 90] M.A. Peshkin. Programmed compliance for error corrective assembly. *IEEE Trans. on Robotics and Automation*, 6(4):473–482, 1990.

[Peshkin 94] M.A. Peshkin. Personal communcation, 15th February, 1994.

[Popplestone & Ambler 83] R.J. Popplestone and A.P. Ambler. A language for specifying robot manipulations. In A. Pugh, editor, *Robotic Technology*, pages 125–141. Peter Peregrinus, London, England, 1983.

[Popplestone *et al.* 78] R.J. Popplestone, A.P. Ambler, and I.M. Bellos. RAPT: A language for describing assemblies. *The Industrial Robot*, 5(3):131–137, September 1978.

[Popplestone *et al.* 80] R.J. Popplestone, A.P. Ambler, and I.M. Bellos. An interpreter for a language for describing assemblies. *Artificial Intelligence*, 14:79–107, 1980.

[Poston & Stewart 78] T. Poston and I. Stewart. *Catastrophe Theory and its Applications.* Pitman Publishing Limited, London, England, 1978.

[Prentis 79] J.M. Prentis. *Engineering Mechanics.* Oxford University Press, Oxford, England, 1979.

[Prentis 80] J.M. Prentis. *Dynamics of Mechanical Systems, 2^{nd} edition.* Ellis Horwood Limited, Chichester, England, 1980.

[Raibert & Craig 81] M.H. Raibert and J.J. Craig. Hybrid position/force control of manipulators. *Trans. ASME, J. Dynamic Systems, Measurement, and Control*, 102:126–133, June 1981.

[Rajan *et al.* 87] V.T. Rajan, R. Burridge, and J.T. Schwartz. Dynamics of a rigid body in frictional contact with rigid walls. In *IEEE Int. Conf. on Robotics and Automation*, pages 671–677, Raleigh, NC, March/April 1987.

[Redford & Lo 86] A. Redford and E. Lo. *Robots in Assembly.* Open University Press, Milton Keynes, England, 1986.

[Requicha 83] A.A.G. Requicha. Towards a theory of geometric tolerancing. *Int. J. of Robotics Research*, 2(4):45–60, 1983.

[Salisbury 80] J.K. Salisbury. Active stiffness control of a manipulator in cartesian coordinates. In *Proc. 19^{th} IEEE Conf. on Decision and Control*, pages 95–100, Albuquerque, NM, December 1980.

[Sanderson 84] A.C. Sanderson. Parts entropy methods for robotic assembly system design. In *IEEE Int. Conf. on Robotics and Automation*, pages 600–608, Atlanta, GA, March 1984.

[Schimmels & Peshkin 92] J.M. Schimmels and M.A. Peshkin. Admittance matrix design for force-guided assembly. *IEEE Trans. on Robotics and Automation*, 8(2):213–227, 1992.

[Schmitz *et al.* 85] D. Schmitz, P. Khosla, and T Kanade. Development of CMU direct-drive arm II. In *Proc.* 15^{th} *Int. Symp. on Industrial Robots*, pages 471–478, Tokyo, Japan, September 1985.

[Shen *et al.* 90] H.C. Shen, K. Selke, G.E. Deacon, and A. Pugh. A sensory data-driven rule-based strategy for error recovery in robotic assembly. In G.E. Taylor, editor, *Kinematic and Dynamic Issues in Sensor-Based Control*, pages 97–110. Springer-Verlag, Berlin, Germany, 1990. NATO ASI Series F, Vol. No. 57.

[Shimano *et al.* 84] B. Shimano, C.C. Geschke, and C. Spalding. VAL-II: a robot programming language and control system. In *SME Robots VIII Conf.*, pages 20:103–20:119, Detroit, MI, June 1984.

[Simunovic 75] S. Simunovic. Force information in assembly processes. In *Proc.* 5^{th} *Int. Symp. on Industrial Robots*, pages 415–431, Chicago, IL, September 1975.

[Smith & Cheesman 86] R.C. Smith and P. Cheesman. On the representation and estimation of spatial uncertainty. *Int. J. of Robotics Research*, 5(4):56–68, 1986.

[Smith & Gini 86] R.E. Smith and M. Gini. Reliable real-time robot operation employing intelligent forward recovery. *J. of Robotic Systems*, 3(3):281–300, 1986.

[Smith *et al.* 88] R. Smith, M. Self, and P. Cheesman. Estimating uncertain spatial relationships in robotics. In J.F. Lemmer and L.N. Kanal, editors, *Uncertainty in Artificial Intelligence 2*, pages 435–461. Elsevier Science Publishers B.V. (North-Holland), Amsterdam, Holland, 1988.

[Snyder 85] W.E. Snyder. *Industrial Robots: Computer Interfacing and Control.* Prentice-Hall International, Englewood Cliffs, NJ, 1985.

[Song & Goldenberg 96] P. Song and A.A. Goldenberg. Analysis and design of position/force controllers. Part II: controller design. Workshop WS3, *IEEE Int. Conf. on Robotics and Automation*, Minneapolis, MN, April 1996. To appear in the IEEE Trans. on Robotics and Automation.

[Srinivas 78] S. Srinivas. Error recovery in robots through failure reason analysis. In *Proc. AFIPS National Computer Conf.*, pages 275–282, 1978.

[Stokić & Vukobratović 93] D. Stokić and M. Vukobratović. Historical perspective and state of the art in joint force sensory feedback control of manipulation robots. *Robotica*, 11:149–157, 1993.

[Sturges & Laowattana 92] R.H. Sturges and S. Laowattana. Virtual wedging in three dimensional peg insertion tasks. In *Proc. IEEE/RSJ Int. Conf. on Intelligent Robots and Systems*, pages 494–499, Raleigh, NC, July 1992.

[Sturges 88] R.H. Sturges. A three-dimensional assembly task quantification with application to machine dexterity. *Int. J. of Robotics Research*, 7(4):34–78, 1988.

[Sutton 92] R.S. Sutton, editor. *Reinforcement Learning*. Kluwer Academic Publishers, Boston, MA, 1992. (Reprinted from *Machine Learning*, 8(3–4), 1992).

[Swift 80] K.G. Swift. A system of classification for automatic assembly. Unpublished M.Sc. thesis, Department of Aeronautical and Mechanical Engineering, University of Salford, Salford, England, 1980.

[Tal 81] J. Tal. Modeling motors for control applications. In *Proc. 1st Int. Motorcon Conf.*, pages 2A1–1–2A1–12, June 1981.

[Taylor 76] R.H. Taylor. *A Synthesis of Manipulator Control Programs from Task-Level Specifications*. Unpublished PhD thesis, Department of Computer Science, Stanford University, Stanford, CA, July 1976. (Also: Memo 282, AI Lab., Stanford University).

[Taylor 90] P.M. Taylor. Multisensory assembly and error recovery. In G.E. Taylor, editor, *Kinematic and Dynamic Issues in Sensor-Based Control*. Springer-Verlag, Berlin, Germany, 1990. NATO ASI Series F, Vol. No. 57.

[Trevelyan & Nelson 87] J. Trevelyan and M. Nelson. Adaptive robot control incorporating automatic error recovery. In *Proc. 3rd Int. Conf. on Advanced Robotics*, pages 385–398, Versailles, France, October 1987.

[Trevelyan 94] J. Trevelyan. Robot force control without stability problems. In T. Yoshikawa and F. Miyazaki, editors, *Proc. Experimental Robotics 3: The 3rd Int.*

Symp. Springer Verlag, Berlin, 1994. (Held in Kyoto, Japan, October 1993).

[Udupa 77] S.M. Udupa. *Collision Detection and Avoidance in Computer Controlled Manipulators.* Unpublished PhD thesis, Department of Electrical Engineering, California Institute of Technology, Pasadena, CA, 1977.

[Venkataraman & Iberall 90] S.T. Venkataraman and T. Iberall, editors. *Dextrous Robot Hands.* Springer-Verlag, New York, NY, 1990.

[Walker & Orin 82] M.W. Walker and D.E. Orin. Efficient dynamic computer simulation of robotic mechanisms. *Trans. ASME, J. Dynamic Systems, Measurement, and Control*, 104:205–211, 1982.

[Wang 89] Y. Wang. *Dynamic Analysis and Simulation of Mechanical Systems with Intermittent Constraints.* Unpublished PhD thesis, Department of Physics and the Robotics Institute, Carnegie-Mellon University, Pittsburgh, PA, May 1989.

[Watson & Drake 75] P.C. Watson and S.H. Drake. Pedestal and wrist force sensors for automatic assembly. In *Proc.* 5^{th} *Int. Symp. on Industrial Robots*, pages 501–511, Chicago, IL, September 1975.

[West & Asada 85] H. West and H. Asada. A method for the design of hybrid position/force controllers for manipulators constrained by contact with the environment. In *IEEE Int. Conf. on Robotics and Automation*, pages 602–607, St. Louis, MO, March 1985.

[Whitney & Junkel 82] D.E. Whitney and E.F. Junkel. Applying stochstic control theory to robot sensing, teaching, and long term control. In D.E. Hardt, editor, *IFAC Conf. on Information Control Problems in Manufacturing Technology*, pages 109–117, Maryland 1982.

[Whitney 69] D.E. Whitney. Resolved motion rate control of manipulators and human prosthesis. *IEEE Trans. on Man-Machine Systems*, MMS-10(2):47–53, June 1969.

[Whitney 77] D.E. Whitney. Force feedback control of manipulator fine motions. *Trans. ASME, J. Dynamic Systems, Measurement, and Control*, 99:91–97, June 1977.

[Whitney 82] D.E. Whitney. Quasi-static assembly of compliantly supported rigid parts. *Trans. ASME, J. Dynamic Systems, Measurement, and Control*, 104:65–77, March 1982.

[Whitney 87] D.E. Whitney. Historical perspective and state of the art in robot force control. *Int. J. of Robotics Research*, 6(1):3–14, 1987.

[Whitney *et al.* 83] D.E. Whitney, R.E. Gustavson, and M.P. Hennessey. Designing chamfers. *Int. J. of Robotics Research*, 2(4):3–18, 1983.

[Wilson 92] M.S. Wilson. *Achieving Reliability using Behavioural Modules in a Robotic Assembly System.* Unpublished PhD thesis, Department of Artificial Intelligence, University of Edinburgh, Edinburgh, Scotland, 1992.

[Wright & Deacon 96] M. Wright and G.E. Deacon. Orienting 2.5D objects of arbitrary shape: A computational theory of planar orientation. Working Paper No. 257, Department of Artificial Intelligence, University of Edinburgh, Edinburgh, Scotland, 1996.

[Yin 84] B. Yin. *Combining Vision Verification with a High Level Robot Programming Language.* Unpublished PhD thesis, Department of Artificial Intelligence, University of Edinburgh, Edinburgh, Scotland, 1984.

[Zhang & Paul 85] H. Zhang and R.P. Paul. Hybrid control of robot manipulators. In *IEEE Int. Conf. on Robotics and Automation*, pages 602–607, St. Louis, MO, March 1985.

Appendix A

Simplified Manipulator Dynamics

In this appendix we will derive a set of differential equations that are a simplified version of the manipulator's dynamics equations. We will follow an example given in [Paul 81] and use the Lagrangian formulation with the simplification of modelling the weight of the arm as a collection of point masses located at the joints. This is a reasonable approximation since, as noted in [Asada & Kanade 83], the weight of connecting links is light in comparison with that of the joints. We also include another point mass along the manipulator's distal link in order to model the force/torque sensor. A more accurate representation of the arm's dynamics would be more precise about the mass distribution by using inertia tensors to model each of the arm's mechanical components, but the purpose of this appendix is merely to illustrate the structure of the dynamics equations.

The Lagrangian, L, is a scalar function that is defined as the difference between a system's kinetic energy K, and its potential energy P, *i.e.*

$$L = K - P \tag{A.1}$$

We will use the manipulator's joint angles $q_i, (i = 1 \ldots 3)$ to define the coordinate system in which the potential and kinetic energy is expressed in. See figure A.1. The dynamics equations in terms of these coordinates are derived from

$$\tau_i = \frac{d}{dt}\frac{\partial L}{\partial \dot{q}_i} - \frac{\partial L}{\partial q_i} \tag{A.2}$$

where τ_i is the torque at joint i, q_i is the position of joint i, and $\dot{q}_i$ its angular velocity.

The coordinates of the various point masses modelled follow from the arm kinematics:

$$x_{m_0} = l_3\cos(q_3) - l_2\cos(q_2 + q_3) + l_0\cos(q_1 + q_2 + q_3) \tag{A.3}$$

$$y_{m_0} = l_3\sin(q_3) - l_2\sin(q_2 + q_3) + l_0\sin(q_1 + q_2 + q_3) \tag{A.4}$$

$$x_{m_1} = l_3\cos(q_3) - l_2\cos(q_2 + q_3) \tag{A.5}$$

$$y_{m_1} = l_3\sin(q_3) - l_2\sin(q_2 + q_3) \tag{A.6}$$

$$x_{m_2} = l_3\cos(q_3) \tag{A.7}$$

$$y_{m_2} = l_3\sin(q_3) \tag{A.8}$$

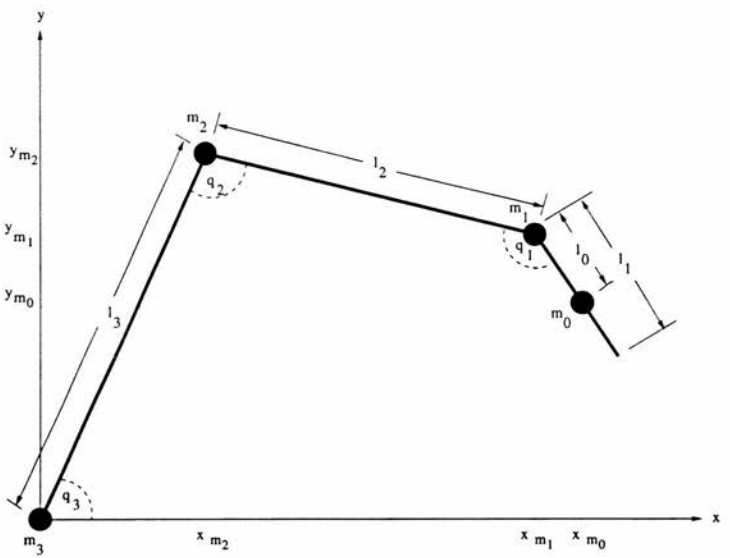


Figure A.1: Variables used in deriving the manipulator's simplified dynamics equations.

The kinetic energy of the mass at joint i is given by

$$K_i = \frac{1}{2} m_i v_i^2 \tag{A.9}$$

where v_i is its Cartesian space velocity. If we differentiate equations A.3 to A.8 with respect to time we arrive at the x and y components of the joints' Cartesian space velocities:

$$\begin{aligned}
\dot{x}_{m_0} &= -l_3\dot{q}_3\sin(q_3) + l_2(\dot{q}_2+\dot{q}_3)\sin(q_2+q_3) \\
&\quad - l_0(\dot{q}_1+\dot{q}_2+\dot{q}_3)\sin(q_1+q_2+q_3) && \text{(A.10)} \\
\dot{y}_{m_0} &= l_3\dot{q}_3\cos(q_3) - l_2(\dot{q}_2+\dot{q}_3)\cos(q_2+q_3) \\
&\quad + l_0(\dot{q}_1+\dot{q}_2+\dot{q}_3)\cos(q_1+q_2+q_3) && \text{(A.11)} \\
\dot{x}_{m_1} &= -l_3\dot{q}_3\sin(q_3) + l_2(\dot{q}_2+\dot{q}_3)\sin(q_2+q_3) && \text{(A.12)} \\
\dot{y}_{m_1} &= l_3\dot{q}_3\cos(q_3) - l_2(\dot{q}_2+\dot{q}_3)\cos(q_2+q_3) && \text{(A.13)} \\
\dot{x}_{m_2} &= -l_3\dot{q}_3\sin(q_3) && \text{(A.14)} \\
\dot{y}_{m_2} &= l_3\dot{q}_3\cos(q_3) && \text{(A.15)}
\end{aligned}$$

We can now find the square of the joint velocities from

$$v_i^2 = \dot{x}_{m_i}^2 + \dot{y}_{m_i}^2 \tag{A.16}$$

resulting in the following kinetic energy expressions

$$\begin{aligned} K_0 &= \frac{1}{2}m_0[l_0^2(\dot{q}_1+\dot{q}_2+\dot{q}_3)^2 + l_2^2(\dot{q}_2+\dot{q}_3)^2 + l_3^2\dot{q}_3^2 \\ &\quad - 2l_0l_2(\dot{q}_1+\dot{q}_2+\dot{q}_3)(\dot{q}_2+\dot{q}_3)\cos(q_1) \\ &\quad + 2l_0l_3(\dot{q}_1+\dot{q}_2+\dot{q}_3)\dot{q}_3\cos(q_1+q_2) \\ &\quad - 2l_2l_3(\dot{q}_2+\dot{q}_3)\dot{q}_3\cos(q_2)] \end{aligned} \tag{A.17}$$

$$K_1 = \frac{1}{2}m_1[l_3^2\dot{q}_3^2 + l_2^2(\dot{q}_2+\dot{q}_3)^2 - 2l_2l_3\dot{q}_3(\dot{q}_2+\dot{q}_3)\cos(q_2)] \tag{A.18}$$

$$K_2 = \frac{1}{2}m_2l_3^2\dot{q}_3^2 \tag{A.19}$$

The potential energy of the mass at joint i is given by

$$P_i = m_igh_i \tag{A.20}$$

where g is gravitational acceleration and h_i is the joint's height above some datum; in our case we will assume that gravity acts parallel to the y-axis and use the x-axis as our datum giving $h_i = y_{m_i}$ which results in the following potential energy expressions

$$P_0 = m_0g[l_0\sin(q_1+q_2+q_3) - l_2\sin(q_2+q_3)l_3\sin(q_3)] \tag{A.21}$$

$$P_1 = m_1g[-l_2\sin(q_2+q_3) + l_3\sin(q_3)] \tag{A.22}$$

$$P_2 = m_2gl_3\sin(q_3) \tag{A.23}$$

The Lagrangian for the system as a whole is given by the difference in kinetic and potential energies of the individual component parts:

$$L = K_0 + K_1 + K_2 - (P_0 + P_1 + P_2) \tag{A.24}$$

The resulting equation is what we require in order to evaluate expression A.2 for each of the joints. After performing the necessary differentiations we find that the constituent parts of these equations are as follows:-

$$\begin{aligned} \frac{d}{dt}\frac{\partial L}{\partial \dot{q}_1} &= m_0\{l_0^2(\ddot{q}_1+\ddot{q}_2+\ddot{q}_3) \\ &\quad + l_0l_2[(\dot{q}_2+\dot{q}_3)\dot{q}_1\sin(q_1) - (\ddot{q}_2+\ddot{q}_3)\cos(q_1)] \\ &\quad + l_0l_3[\ddot{q}_3\cos(q_1+q_2) - \dot{q}_3(\dot{q}_1+\dot{q}_2)\sin(q_1+q_2)]\} \end{aligned} \tag{A.25}$$

$$\begin{aligned} \frac{\partial L}{\partial q_1} &= m_0[l_0l_2(\dot{q}_2^2+\dot{q}_3^2+\dot{q}_1\dot{q}_2+\dot{q}_1\dot{q}_3+2\dot{q}_2\dot{q}_3)\sin(q_1) \\ &\quad - l_0l_3(\dot{q}_1\dot{q}_3+\dot{q}_2\dot{q}_3+\dot{q}_3^2)\sin(q_1+q_2) \\ &\quad - gl_0\cos(q_1+q_2+q_3)] \end{aligned} \tag{A.26}$$

$$\begin{aligned} \frac{d}{dt}\frac{\partial L}{\partial \dot{q}_2} &= m_0\{l_0^2(\ddot{q}_1+\ddot{q}_2+\ddot{q}_3) + l_2^2(\ddot{q}_2+\ddot{q}_3) \\ &\quad + l_0l_2[(\dot{q}_1+2\dot{q}_2+2\dot{q}_3)\dot{q}_1\sin(q_1) - (\ddot{q}_1+2\ddot{q}_2+2\ddot{q}_3)\cos(q_1)] \\ &\quad + l_0l_3[\ddot{q}_3\cos(q_1+q_2) - \dot{q}_3(\dot{q}_1+\dot{q}_2)\sin(q_1+q_2)] \\ &\quad + l_2l_3[\dot{q}_3\dot{q}_2\sin(q_2) - \ddot{q}_3\cos(q_2)]\} \\ &\quad + m_1\{l_2^2(\ddot{q}_2+\ddot{q}_3) + l_2l_3[\dot{q}_3\dot{q}_2\sin(q_2) - \ddot{q}_3\cos(q_2)]\} \end{aligned} \tag{A.27}$$

$$\begin{aligned}
\frac{\partial L}{\partial q_2} &= m_0\{[-l_0 l_3(\dot{q}_1\dot{q}_3 + \dot{q}_2\dot{q}_3 + \dot{q}_3^2)\sin(q_1+q_2) + l_2 l_3(\dot{q}_2\dot{q}_3 + \dot{q}_3^2)\sin(q_2)] \\
&\quad - g[l_0\cos(q_1+q_2+q_3) - l_2\cos(q_2+q_3)]\} \\
&\quad + m_1\{[l_2 l_3(\dot{q}_2\dot{q}_3 + \dot{q}_3^2)\sin(q_2)] \\
&\quad + g[l_2\cos(q_2+q_3)]\} \qquad \text{(A.28)} \\
\frac{d}{dt}\frac{\partial L}{\partial \dot{q}_3} &= m_0\{l_0^2(\ddot{q}_1+\ddot{q}_2+\ddot{q}_3) + l_2^2(\ddot{q}_2+\ddot{q}_3) + l_3^2\ddot{q}_3 \\
&\quad + l_0 l_2[(\dot{q}_1+2\dot{q}_2+2\dot{q}_3)\dot{q}_1\sin(q_1) - (\ddot{q}_1+2\ddot{q}_2+2\ddot{q}_3)\cos(q_1)] \\
&\quad + l_0 l_3[(\ddot{q}_1+\ddot{q}_2+2\ddot{q}_3)\cos(q_1+q_2) - (\dot{q}_1+\dot{q}_2+2\dot{q}_3)(\dot{q}_1+\dot{q}_2)\sin(q_1+q_2)] \\
&\quad + l_2 l_3[(\dot{q}_2+2\dot{q}_3)\dot{q}_2\sin(q_2) - (\ddot{q}_2+2\ddot{q}_3)\cos(q_2)]\} \\
&\quad + m_1\{l_2^2(\ddot{q}_2+\ddot{q}_3) + l_3^2\ddot{q}_3 + l_2 l_3[(\dot{q}_2+2\dot{q}_3)\dot{q}_2\sin(q_2) - (\ddot{q}_2+2\ddot{q}_3)\cos(q_2)]\} \\
&\quad + m_2 l_3^2\ddot{q}_3 \qquad \text{(A.29)} \\
\frac{\partial L}{\partial q_3} &= -m_0 g[l_0\cos(q_1+q_2+q_3) - l_2\cos(q_2+q_3) + l_3\cos(q_3)] \\
&\quad - m_1 g[-l_2\cos(q_2+q_3) + l_3\cos(q_3)] \\
&\quad - m_2 g[l_3\cos(q_3)] \qquad \text{(A.30)}
\end{aligned}$$

These expressions can be used to derive the manipulator's dynamics equations:

$$\begin{aligned}
\tau_1 &= m_0 l_0^2\ddot{q}_1 + m_0[l_0^2 - l_0 l_2\cos(q_1)]\ddot{q}_2 + m_0[l_0^2 + l_0 l_3\cos(q_1+q_2)]\ddot{q}_3 \\
&\quad - m_0[l_0 l_2\sin(q_1)]\dot{q}_2^2 - m_0[l_0 l_2\sin(q_1) + l_0 l_3\sin(q_1+q_2)]\dot{q}_3^2 \\
&\quad + m_0 l_0 l_2\sin(q_1)\dot{q}_1\dot{q}_3 - 2m_0 l_0 l_2\sin(q_1)\dot{q}_2\dot{q}_3 \\
&\quad + g m_0 l_0\cos(q_1+q_2+q_3) \qquad \text{(A.31)} \\
\tau_2 &= m_0[l_0^2 - l_0 l_2\cos(q_1)]\ddot{q}_1 + \{m_0[l_0^2 + l_2^2 - 2l_0 l_2\cos(q_1)] + m_1 l_2^2\}\ddot{q}_2 \\
&\quad + \{m_0[l_0^2 + l_2^2 - 2l_0 l_2\cos(q_1) + l_0 l_3\cos(q_1+q_2) - l_2 l_3\cos(q_2)] \\
&\quad + m_1[l_2^2 - l_2 l_3\cos(q_2)]\}\ddot{q}_3 \\
&\quad + m_0 l_0 l_2\sin(q_1)\dot{q}_1^2 + \{m_0[l_0 l_3\sin(q_1+q_2) - l_2 l_3\sin(q_2)] - m_1 l_2 l_3\sin(q_2)\}\dot{q}_3^2 \\
&\quad + 2m_0 l_0 l_2\sin(q_1)\dot{q}_1\dot{q}_2 + 2m_0 l_0 l_2\sin(q_1)\dot{q}_1\dot{q}_3 \\
&\quad + g\{m_0[l_0\cos(q_1+q_2+q_3) - l_2\cos(q_2+q_3)] - m_1 l_2\cos(q_2+q_3)\} \qquad \text{(A.32)} \\
\tau_3 &= m_0[l_0^2 - l_0 l_2\cos(q_1) + l_0 l_3\cos(q_1+q_2)]\ddot{q}_1 \\
&\quad + \{m_0[l_0^2 + l_2^2 - 2l_0 l_2\cos(q_1) + l_0 l_3\cos(q_1+q_2) - l_0 l_3\cos(q_2)] \\
&\quad + m_1[l_2^2 l_3\cos(q_2)]\}\ddot{q}_2 \\
&\quad + \{m_0[l_0^2 + l_2^2 + l_3^2 - 2l_0 l_2\cos(q_1) + 2l_0 l_3\cos(q_1+q_2) - 2l_2 l_3\cos(q_2)] \\
&\quad + m_1[l_2^2 + l_3^2 - 2l_2 l_3\cos(q_2)] + m_2 l_3^2\}\ddot{q}_3 \\
&\quad + m_0[l_0 l_2\sin(q_1) - l_0 l_3\sin(q_1+q_2)]\dot{q}_1^2 \\
&\quad + \{m_0[l_2 l_3\sin(q_2) - l_0 l_3 sin(q_1+q_2)] + m_1 l_2 l_3\sin(q_2)\}\dot{q}_2^2 \\
&\quad + m_0[2l_0 l_2\sin(q_1) - 2l_0 l_3\sin(q_1+q_2)]\dot{q}_1\dot{q}_2 \\
&\quad + m_0[2l_0 l_2\sin(q_1) - 2l_0 l_3\sin(q_1+q_2)]\dot{q}_1\dot{q}_3 \\
&\quad + \{m_0[2l_2 l_3\sin(q_2) - 2l_0 l_3\sin(q_1+q_2)] + 2m_1 l_2 l_3\sin(q_2)\}\dot{q}_2\dot{q}_3 \\
&\quad + g\{m_0[l_0\cos(q_1+q_2+q_3) - l_2\cos(q_2+q_3) + l_3\cos(q_3)] \\
&\quad - m_1[l_2\cos(q_2+q_3) - l_3\cos(q_3)] + m_2 l_3\cos(q_3)\} \qquad \text{(A.33)}
\end{aligned}$$

Inspection of the above equations will show that they can be rewritten in the following form:

$$\begin{aligned}
\tau_1 &= M_{11}\ddot{q}_1 + M_{12}\ddot{q}_2 b + M_{13}\ddot{q}_3 \\
&\quad + C_{111}\dot{q}_1^2 + C_{122}\dot{q}_2^2 + C_{133}\dot{q}_3^2 \\
&\quad + C_{112}\dot{q}_1\dot{q}_2 + C_{121}\dot{q}_2\dot{q}_1 + C_{113}\dot{q}_1\dot{q}_3 + C_{131}\dot{q}_3\dot{q}_1 + C_{123}\dot{q}_2\dot{q}_3 + C_{132}\dot{q}_3\dot{q}_2 \\
&\quad + G_1 \qquad \text{(A.34)} \\
\tau_2 &= M_{21}\ddot{q}_1 + M_{22}\ddot{q}_2 b + M_{23}\ddot{q}_3 \\
&\quad + C_{211}\dot{q}_1^2 + C_{222}\dot{q}_2^2 + C_{233}\dot{q}_3^2 \\
&\quad + C_{212}\dot{q}_1\dot{q}_2 + C_{221}\dot{q}_2\dot{q}_1 + C_{213}\dot{q}_1\dot{q}_3 + C_{231}\dot{q}_3\dot{q}_1 + C_{223}\dot{q}_2\dot{q}_3 + C_{232}\dot{q}_3\dot{q}_2 \\
&\quad + G_2 \qquad \text{(A.35)} \\
\tau_3 &= M_{31}\ddot{q}_1 + M_{32}\ddot{q}_2 b + M_{33}\ddot{q}_3 \\
&\quad + C_{311}\dot{q}_1^2 + C_{322}\dot{q}_2^2 + C_{333}\dot{q}_3^2 \\
&\quad + C_{312}\dot{q}_1\dot{q}_2 + C_{321}\dot{q}_2\dot{q}_1 + C_{313}\dot{q}_1\dot{q}_3 + C_{331}\dot{q}_3\dot{q}_1 + C_{323}\dot{q}_2\dot{q}_3 + C_{332}\dot{q}_3\dot{q}_2 \\
&\quad + G_3 \qquad \text{(A.36)}
\end{aligned}$$

which can be collected together and expressed as:

$$\tau_i = \sum_{j=1}^{3} M_{ij}\ddot{q}_j + \sum_{j=1}^{3}\sum_{k=1}^{3} C_{ijk}\dot{q}_j\dot{q}_k + G_i \qquad \text{(A.37)}$$

where the M_{ij} are the inertial terms (a coefficient of the form M_{ii} is known as the effective inertia at joint i, since an acceleration at joint i causes a torque at joint i equal to $M_{ii}\ddot{q}_i$; a coefficient of the form $M_{ij}(i \neq j)$ is known as the coupling inertia between joints i and j since an acceleration at joint i causes a torque at joint j of $M_{ij}\ddot{q}_i$), C_{ijk} represent the centripetal and Coriolis effects[1] (a term of the form $C_{iij}\dot{q}_j^2$ is the centripetal force acting at joint i due to a velocity at joint j; a combination of terms of the form $C_{ijk}\dot{q}_j\dot{q}_k + C_{ikj}\dot{q}_k\dot{q}_j$ is known as the Coriolis force acting at joint i due to velocities at joints j and k), and the G_i are the gravitational loadings.

Alternatively these equations can be written in vector/matrix notation as:

$$\boldsymbol{\tau} = \mathbf{M}(\mathbf{q})\,\ddot{\mathbf{q}} + \mathbf{C}(\mathbf{q}, \dot{\mathbf{q}})\,\dot{\mathbf{q}} + \mathbf{G}(\mathbf{q}) \qquad \text{(A.38)}$$

where, for an n degree of freedom manipulator:–
$\mathbf{q}, \dot{\mathbf{q}}, \ddot{\mathbf{q}}$ are $n \times 1$ vectors of joint positions, velocities and accelerations, respectively,
$\mathbf{M}(\mathbf{q})$ is an $n \times n$ moment of inertia matrix,
$\mathbf{C}(\mathbf{q}, \dot{\mathbf{q}})$ is an $n \times n$ matrix of centripetal and Coriolis effects,
$\mathbf{G}(\mathbf{q})$ is an $n \times 1$ vector specifying the effects of gravity,
$\boldsymbol{\tau}$ is the $n \times 1$ vector of joint torques.

[1] These are also known as the velocity product terms [Featherstone 87] for obvious reasons.

Appendix B

The Manipulator Jacobian

In this appendix we will derive the Jacobian matrix that is central to the control of our direct-drive arm. There are a number of text books which cover this material, see for example [Paul 81], [Snyder 85], [Asada & Slotine 86], [Craig 89]. We will proceed by deriving what might be called the standard manipulator Jacobian, which is defined in terms of a frame of reference attached to the manipulator's distal link but aligned with the robot's base coordinates. We then go on to derive the actual Jacobian that we use during our experiments as a function of this. This Jacobian is defined with respect to a coordinate frame aligned with the manipulator's distal link. Wherever we refer to the "base jacobian" we mean the former, and where we refer to the "wrist Jacobian" we mean the latter.

We define the base Jacobian as well as the wrist Jacobian here because the base Jacobian is useful if it is desired to generate force commands in a frame of reference aligned with world coordinates. We used this facility occasionally when driving the arm from the joystick, and in many situations this appears to be a more natural frame of reference for a human operator to work in.

B.1 Basic Definition

In the general case the position of a robot's end-effector in some frame of reference is specified by six variables; three translational and three rotational. We shall call them $p_1, \ldots, p_6$, and let $\mathbf{p} = (p_1 \ \ldots \ p_6)^T$. The value of each of these variables is a function of the manipulator's joint variables, $q_1, \ldots, q_n$ (for a manipulator with n joints), *i.e.*

$$\begin{aligned} p_1 &= f_1(q_1, q_2, \ldots, q_n), \\ p_2 &= f_2(q_1, q_2, \ldots, q_n), \\ &\vdots \\ p_6 &= f_6(q_1, q_2, \ldots, q_n). \end{aligned} \tag{B.1}$$

Using the chain rule we can calculate the differentials of the p_i as a function of the differentials of the q_i:

$$\begin{aligned}
\delta p_1 &= \frac{\partial f_1}{\partial q_1}\delta q_1 + \frac{\partial f_1}{\partial q_2}\delta q_2 + \cdots + \frac{\partial f_1}{\partial q_n}\delta q_n, \\
\delta p_2 &= \frac{\partial f_2}{\partial q_1}\delta q_1 + \frac{\partial f_2}{\partial q_2}\delta q_2 + \cdots + \frac{\partial f_2}{\partial q_n}\delta q_n, \\
&\vdots \\
\delta p_6 &= \frac{\partial f_6}{\partial q_1}\delta q_1 + \frac{\partial f_6}{\partial q_2}\delta q_2 + \cdots + \frac{\partial f_6}{\partial q_n}\delta q_n.
\end{aligned} \tag{B.2}$$

This can be rewritten in vector/matrix notation as:

$$\delta \mathbf{p} = \mathbf{J}\,\delta \mathbf{q} \tag{B.3}$$

where $\mathbf{q} = (q_1 \ \ldots \ q_n)^T$ and $\mathbf{J}$ is the matrix of differential coefficients known as the manipulator Jacobian. For most manipulators the functions $f_i(\mathbf{q})$ are nonlinear, so the values of the differential coefficients are functions of the q_i, *i.e.*

$$\delta \mathbf{p} = \mathbf{J}(\mathbf{q})\,\delta \mathbf{q} \tag{B.4}$$

By dividing through by the differential time element we can consider the Jacobian as a mapping from joint velocities to end-effector velocities, *i.e.*

$$\dot{\mathbf{p}} = \mathbf{J}(\mathbf{q})\,\dot{\mathbf{q}} \tag{B.5}$$

Usually in robotics the frame of reference that the position of the end-effector is measured in is a Cartesian one that is fixed with respect to the base. As an example, for our three degree of freedom, planar arm we could choose a frame of reference attached to the base of the robot as shown in figure B.1.

Using the variables illustrated in figure B.1 we can express the coordinates of the arm's end-point in terms of its joint variables as follows:

$$x_0 = l_3\cos(q_3) - l_2\cos(q_2+q_3) + l_1\cos(q_1+q_2+q_3) \tag{B.6}$$

$$y_0 = l_3\sin(q_3) - l_2\sin(q_2+q_3) + l_1\sin(q_1+q_2+q_3) \tag{B.7}$$

In addition, the orientation of the end-point is given by:

$$\theta_0 = q_1 + q_2 + q_3 \tag{B.8}$$

The standard Jacobian for our planar, three degree of freedom, revolute manipulator is given by:

$$\mathbf{J}(\mathbf{q}) = \begin{pmatrix} \frac{\partial x_0}{\partial q_1} & \frac{\partial y_0}{\partial q_1} & \frac{\partial \theta_0}{\partial q_1} \\ \frac{\partial x_0}{\partial q_2} & \frac{\partial y_0}{\partial q_2} & \frac{\partial \theta_0}{\partial q_2} \\ \frac{\partial x_0}{\partial q_3} & \frac{\partial y_0}{\partial q_3} & \frac{\partial \theta_0}{\partial q_3} \end{pmatrix} \tag{B.9}$$

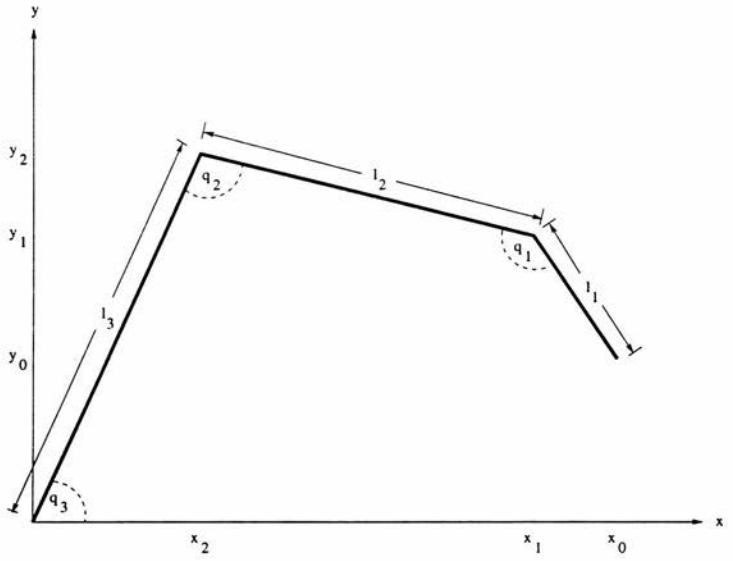


Figure B.1: Diagram of the parameters used in deriving the manipulator Jacobian.

the individual components of which, from equations B.6, B.7 and B.8, evaluate to:

$$\frac{\partial x_0}{\partial q_1} = -l_1 \sin(q_1 + q_2 + q_3) \tag{B.10}$$

$$\frac{\partial y_0}{\partial q_1} = l_1 \cos(q_1 + q_2 + q_3) \tag{B.11}$$

$$\frac{\partial \theta_0}{\partial q_1} = 1 \tag{B.12}$$

$$\frac{\partial x_0}{\partial q_2} = l_2 \sin(q_2 + q_3) - l_1 \sin(q_1 + q_2 + q_3) \tag{B.13}$$

$$\frac{\partial y_0}{\partial q_2} = -l_2 \cos(q_2 + q_3) + l_1 \cos(q_1 + q_2 + q_3) \tag{B.14}$$

$$\frac{\partial \theta_0}{\partial q_2} = 1 \tag{B.15}$$

$$\frac{\partial x_0}{\partial q_3} = -l_3 \sin(q_3) + l_2 \sin(q_2 + q_3) - l_1 \sin(q_1 + q_2 + q_3) \tag{B.16}$$

$$\frac{\partial y_0}{\partial q_3} = l_3 \cos(q_3) - l_2 \cos(q_2 + q_3) + l_1 \cos(q_1 + q_2 + q_3) \tag{B.17}$$

$$\frac{\partial \theta_0}{\partial q_3} = 1 \tag{B.18}$$

B.2 Relating Joint Torques and Cartesian Space Wrenches via the Jacobian

For a constrained manipulator we can use the principle of virtual work to relate the forces and torques experienced at its wrist to those experienced at its joints via its Jacobian. Work is defined as force acting through some distance. Now if the manipulator is constrained it can't *actually* move, but we can consider what would happen if infinitesimal displacements were possible. In the multidimensional case work is the dot product of a vector force or torque and a vector displacement. So we have

$$\mathbf{w} \cdot \delta\mathbf{p} = \boldsymbol{\tau} \cdot \delta\mathbf{q} - \mathbf{G}(\mathbf{q}) \cdot \delta\mathbf{q} \tag{B.19}$$

where $\mathbf{w}$ is a Cartesian space wrench, $\boldsymbol{\tau}$ is a vector of joint torques, $\mathbf{G}(\mathbf{q})$ is a vector of gravitational loadings (as in equation A.38), and $\delta\mathbf{q}$ and $\delta\mathbf{p}$ are infinitesimal joint space and Cartesian space displacements (as in section B.1). In other words the virtual work done by the hand in Cartesian space will be equal to that done by the actuators in the space defined by the robot's joint variables. This relationship can be rewritten in terms of inner products as

$$\mathbf{w}^T\delta\mathbf{p} = [\boldsymbol{\tau} - \mathbf{G}(\mathbf{q})]^T\,\delta\mathbf{q} \tag{B.20}$$

Substituting for $\delta\mathbf{p}$ from equation B.4 we get

$$\mathbf{w}^T\mathbf{J}(\mathbf{q})\delta\mathbf{q} = [\boldsymbol{\tau} - \mathbf{G}(\mathbf{q})]^T\,\delta\mathbf{q} \tag{B.21}$$

And since this must hold for any $\delta\mathbf{q}$ we have

$$\mathbf{w}^T\mathbf{J}(\mathbf{q}) = [\boldsymbol{\tau} - \mathbf{G}(\mathbf{q})]^T \tag{B.22}$$

Transposing both sides and rearranging yields

$$\boldsymbol{\tau} = \mathbf{J}^T(\mathbf{q})\,\mathbf{w} + \mathbf{G}(\mathbf{q}) \tag{B.23}$$

For the experiments performed in this thesis the arm was working in a horizontal plane orthogonal to the gravitational field so:

$$\mathbf{G}(\mathbf{q}) = \mathbf{0} \tag{B.24}$$

so, for us, equation B.23 reduces to

$$\boldsymbol{\tau} = \mathbf{J}^T(\mathbf{q})\,\mathbf{w} \tag{B.25}$$

It is worth pointing out that equations B.23 and B.25 are statics relationships since they are only genuinely realisable if the last link of the robot is rigidly clamped. When the arm is moving static conditions do not apply and some of the joint torques that are being developed to realise the desired wrist wrench will be 'spent' on accelerating the masses of the arm. See Appendix A for the full dynamics relationships of a robot arm. We assume (throughout this thesis) that inertial and velocity product terms can be neglected during an assembly process. Consequently our robot arm might be considered a quasi-static manipulator, performing quasi-static manipulation.

B.3 A Jacobian Aligned with the Distal Link

Equation B.25 expresses the (statics) relationship between the joint torques and the wrench necessary to balance them in a frame of reference at the wrist aligned with the base frame. However, we want to control our robot arm by specifying what forces the robot should be applying in a frame of reference attached to the wrist, because then we will be specifying them relative to the 'held' object. We can rewrite equation B.25 to take this fact into account

$$\boldsymbol{\tau} = {}^{w}\mathbf{J}^{T}(\mathbf{q})\,{}^{w}\mathbf{w} \tag{B.26}$$

where we use leading w superscripts to denote quantities measured at the wrist in a frame of reference that remains constant with respect to the wrist. We will also find it useful to use leading b superscripts to denote quantities measured at the wrist in a frame of reference aligned with the base frame.

Since we have a symbolic solution for ${}^{b}\mathbf{J}(\mathbf{q})$ (equations B.10 to B.18) we will derive ${}^{w}\mathbf{J}(\mathbf{q})$ in terms of it. In order to do this first we note that from the definition of a Jacobian given above, the joint velocities can be used to find the wrist velocity in either the base or the wrist frame of reference by using the appropriate Jacobian:

$${}^{b}\dot{\mathbf{p}} = {}^{b}\mathbf{J}(\mathbf{q})\,\dot{\mathbf{q}} \tag{B.27}$$

$${}^{w}\dot{\mathbf{p}} = {}^{w}\mathbf{J}(\mathbf{q})\,\dot{\mathbf{q}} \tag{B.28}$$

With reference to figure B.2 we can see that because the z-axis of the wrist frame of reference attached to the wrist and the wrist frame of reference aligned with the base coincide, we can just apply the transformation of C.11 with the distances between the frames set to zero, *i.e.*

$${}^{b}\dot{\mathbf{p}} = \begin{pmatrix} \cos(\theta) & -\sin(\theta) & 0 \\ \sin(\theta) & \cos(\theta) & 0 \\ 0 & 0 & 1 \end{pmatrix} {}^{w}\dot{\mathbf{p}} \tag{B.29}$$

or, since the angle between the two frames, because of the way that we have chosen to label the joint angles, is given by θ_0 in equation B.8, we have

$${}^{b}\dot{\mathbf{p}} = {}^{b}_{w}\mathbf{R}_{z}(\theta_0)\,{}^{w}\dot{\mathbf{p}} \tag{B.30}$$

where $\mathbf{R}_z(\theta)$ is the rotation matrix that rotates a vector by an angle θ about the z-axis (equation C.6).

Substituting for ${}^{b}\dot{\mathbf{p}}$ and ${}^{w}\dot{\mathbf{p}}$ from equations B.27 and B.28 we arrive at

$${}^{b}\mathbf{J}(\mathbf{q})\,\dot{\mathbf{q}} = {}^{b}_{w}\mathbf{R}_{z}(\theta_0)\,{}^{w}\mathbf{J}(\mathbf{q})\,\dot{\mathbf{q}} \tag{B.31}$$

Since this must hold for all $\dot{\mathbf{q}}$, then

$${}^{b}\mathbf{J}(\mathbf{q}) = {}^{b}_{w}\mathbf{R}_{z}(\theta_0)\,{}^{w}\mathbf{J}(\mathbf{q}) \tag{B.32}$$

We can rearrange this, taking into account the fact that the inverse of a rotation matrix is equal to its transpose because of its orthonormality [Craig 89, p.23], to give

$${}^{w}\mathbf{J}^{T}(\mathbf{q}) = {}^{b}\mathbf{J}^{T}(\mathbf{q})\,{}^{b}_{w}\mathbf{R}_{z}(\theta_0) \tag{B.33}$$

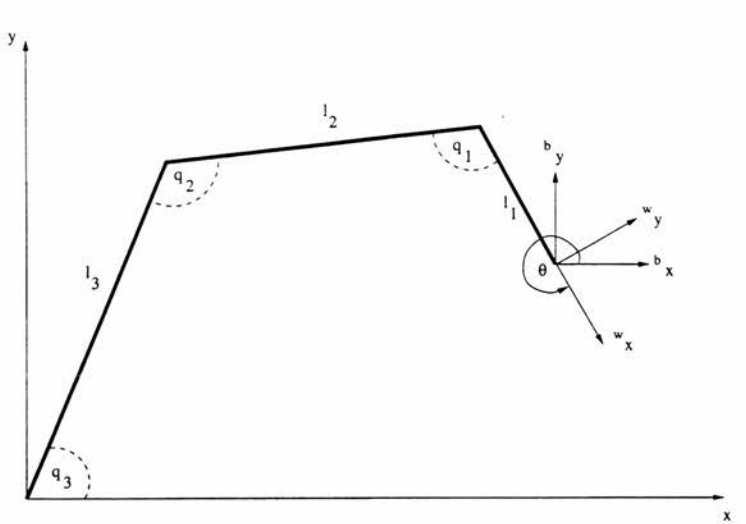


Figure B.2: Diagram of the relationship between the frame of reference aligned with the distal link and the frame of reference aligned with the base.

This evaluates to:

$$ {}^{w}\mathbf{J}^{T}(\mathbf{q}) = \begin{pmatrix} 0 & l_1 & 1 \\ -l_2\sin(q_1) & l_1 - l_2\cos(q_1) & 1 \\ -l_2\sin(q_1) + l_3\sin(q_1+q_2) & l_1 - l_2\cos(q_1) + l_3\cos(q_1+q_2) & 1 \end{pmatrix} \tag{B.34} $$

which is the mapping between the desired wrist wrench and the joint torques needed to realise it, that we continuously evaluate in order to drive the arm. In fact, as can be seen, only four of the elements of the wrist Jacobian need to be continuously evaluated, and the remainder are constants.

Appendix C

Mapping Screws Between Frames of Reference

In this appendix we will state the mathematics for transforming a screw from one frame of reference to another. This is a summary of some of the material in [Craig 89, §5.11] adapted for our purposes. The details of the derivations of these transformations can be found in [Craig 89, Ch5]

We assume that the two frames of reference a and b are rigidly connected. For the general spatial case wrenches are composed of a 3×1 force vector, $\mathbf{f} = (f_x\, f_y\, f_z)^T$, and a 3×1 moment vector, $\boldsymbol{\tau} = (\tau_x\, \tau_y\, \tau_z)^T$, which we can write as the wrench $\mathbf{w}$ as follows

$$\mathbf{w} = \begin{pmatrix} \mathbf{f} \\ \boldsymbol{\tau} \end{pmatrix} \tag{C.1}$$

A twist is composed of a 3×1 linear velocity vector, $\mathbf{v} = (v_x\, v_y\, v_z)^T$, and a 3×1 angular velocity vector, $\boldsymbol{\omega} = (\omega_x\, \omega_y\, \omega_z)^T$, which we can write as the twist $\mathbf{t}$ as follows

$$\mathbf{t} = \begin{pmatrix} \mathbf{v} \\ \boldsymbol{\omega} \end{pmatrix} \tag{C.2}$$

The mapping of a wrench in frame b to a wrench in frame a is given by

$$\begin{pmatrix} {}^a\mathbf{f} \\ {}^a\boldsymbol{\tau} \end{pmatrix} = \begin{pmatrix} {}^a_b\mathbf{R} & \mathbf{0} \\ {}^a\mathbf{P}_b \times {}^a_b\mathbf{R} & {}^a_b\mathbf{R} \end{pmatrix} \begin{pmatrix} {}^b\mathbf{f} \\ {}^b\boldsymbol{\tau} \end{pmatrix} \tag{C.3}$$

and the mapping of a twist in frame b to a twist in frame a is given by

$$\begin{pmatrix} {}^a\mathbf{v} \\ {}^a\boldsymbol{\omega} \end{pmatrix} = \begin{pmatrix} {}^a_b\mathbf{R} & {}^a\mathbf{P}_b \times {}^a_b\mathbf{R} \\ \mathbf{0} & {}^a_b\mathbf{R} \end{pmatrix} \begin{pmatrix} {}^b\mathbf{v} \\ {}^b\boldsymbol{\omega} \end{pmatrix} \tag{C.4}$$

where ${}^a_b\mathbf{R}$ is the rotation operator that maps frame b into frame a, and the cross product can be implemented as the matrix operator

$${}^a\mathbf{P}_b \times = \begin{pmatrix} 0 & -p_z & p_y \\ p_z & 0 & -p_x \\ -p_y & p_x & 0 \end{pmatrix} \tag{C.5}$$

where the p_j are the displacements of the origin of the frame of reference b along the j-axis of frame a coordinates.

Since our robot arm operates in the x-y plane the rotation mapping is the matrix operator given by

$$ {}^a_b\mathbf{R}_z(\theta) = \begin{pmatrix} \cos(\theta) & -\sin(\theta) & 0 \\ \sin(\theta) & \cos(\theta) & 0 \\ 0 & 0 & 1 \end{pmatrix} \tag{C.6} $$

where θ is the angle between the frames of reference a and b (measured in an anti-clockwise sense from a to b). For operation in the x-y plane equation C.3 reduces to

$$ {}^a\mathbf{w} = {}^a_b\mathbf{W}\,{}^b\mathbf{w} \tag{C.7} $$

where the ${}^j\mathbf{w}$ are planar wrenches as defined in equation 4.1 and the wrench transformation ${}^a_b\mathbf{W}$ is given by

$$ {}^a_b\mathbf{W} = \begin{pmatrix} \cos(\theta) & -\sin(\theta) & 0 \\ \sin(\theta) & \cos(\theta) & 0 \\ p_x\sin(\theta) - p_y\cos(\theta) & p_x\cos(\theta) + p_y\sin(\theta) & 1 \end{pmatrix} \tag{C.8} $$

Similarly, for the twist case we have

$$ {}^a\mathbf{t} = {}^a_b\mathbf{T}\,{}^b\mathbf{t} \tag{C.9} $$

where an x-y planar twist, $\mathbf{t}$, is given by:

$$ \mathbf{t} = \begin{pmatrix} v_x \\ v_y \\ \omega_z \end{pmatrix} \tag{C.10} $$

and the twist transformation ${}^a_b\mathbf{T}$ is

$$ {}^a_b\mathbf{T} = \begin{pmatrix} \cos(\theta) & -\sin(\theta) & p_y \\ \sin(\theta) & \cos(\theta) & -p_x \\ 0 & 0 & 1 \end{pmatrix} \tag{C.11} $$

Appendix D

Arm Design Details

In this appendix we will cover the design of our robot manipulator in more detail.

D.1 Choosing the Motors

D.1.1 Torque Requirements

In [Asada & Kanade 83] Asada and Kanade describe a method for designing direct-drive mechanical arms. This is undoubtedly a prime reference on the subject, and in fact is probably the first article to cover the matter, so we will describe their recommended technique. It does however possess a shortcoming for our purposes which we will identify before going on to describe how we designed our arm.

Appendix A describes a simplified derivation of the dynamics equations for a robot arm. In the design method described in [Asada & Kanade 83] the authors decide to ignore the velocity product terms, *i.e.* $\mathbf{C}(\mathbf{q}, \dot{\mathbf{q}})$ in equation A.38. This is in fact quite often done because their effects tend to be minimal at anything other than very high joint velocities (see for example [Paul 81]). Instead they calculate the maximum torque that will be required of a motor as the sum of the inertia and gravity terms of the dynamics equation for a particular motor. They model the mass of an arm as a series of point masses at the joints, much the same as we do in Appendix A, with an extra mass beyond the last joint to model the end-effector. Their procedure is to calculate the maximum torque required of the distal motor due to inertial and gravitational loading (for a maximum desired acceleration), choose a motor that can deliver this torque, estimate its weight when housed, and use this value in determining the torque required of the next motor along the kinematic chain (they assume a serial link manipulator rather than a parallel link geometry). This calculation is then repeated, taking into account all point masses distal to a motor, until the required number of degrees of freedom have been designed. This design procedure highlights how the weight of distal motors are loads for more proximal ones, and so Asada and Kanade investigate designing the manipulator link structure in order to minimise this loading. This is an important issue because demanding high torques from direct-drive motors can result in large amounts of power dissipation causing significant temperature rises. We deal with this further

in sections D.1.2 and D.3.

The dynamics equations for a manipulator arm are usually written in the vector/matrix form of equation A.38 (see for example [Craig 89]). A more general relationship however is [Walker & Orin 82][1]:

$$\tau = \mathbf{M}(\mathbf{q})\,\ddot{\mathbf{q}} + \mathbf{C}(\mathbf{q},\dot{\mathbf{q}})\,\dot{\mathbf{q}} + \mathbf{G}(\mathbf{q}) + \mathbf{J}^T(\mathbf{q})\,\mathbf{w} \tag{D.1}$$

where $\mathbf{J}^T(\mathbf{q})$ is the transpose of the manipulator Jacobian as described in Appendix B, and $\mathbf{w}$ represents the external forces and moments acting at the point that the Jacobian is defined for. We show in section 5.1.3 that for a geared manipulator it is the invariant motor inertia that dominates a joint's dynamics. For a direct-drive manipulator the significance of this inertia is reduced in comparison with the other terms in equation D.1 making it worthwhile to consider what this equation says about arm behaviour.

Consider the motion of an arm in free space; in this situation the wrench at the endpoint will be zero, *i.e.* $\mathbf{w} = \mathbf{0}$, and the last term on the right hand side of equation D.1 disappear, leaving equation A.38. In the static case, when an arm is not moving, we have $\ddot{\mathbf{q}} = \dot{\mathbf{q}} = \mathbf{0}$, and the first two terms on the right hand side of equation D.1 disappears leaving equation B.23. The intermediate case is when the arm is moving but the forces at the wrist are not zero. This is the kind of situation prevalent in the mating of rigid body components when the parts spend a large amount of time sliding over each other, so it is the situation that we are particularly interested in.

Asada and Kanades' design procedure appears to be best suited to the design of direct-drive manipulators that are required to perform motions in free space where significant accelerations are expected. Our primary interest is in assembly, a situation that requires slow motion in contact without any high accelerations. Moreover, because we are designing a direct-drive manipulator, the forces that it is to exert on the environment ought to be taken into account when calculating the maximum torques required of a motor since, due to the fact that there is no velocity to torque conversion via a gear train, these might easily form a significant part of the motors' loads. In fact for our manipulator this is the case.[2]

Ideally, in order to test the approach to assembly described in Chapter 3 we would have constructed a manipulator with six degrees of freedom. Unfortunately this would have proved very expensive. It is possible, however, to perform planar assemblies that possess a sufficient degree of difficulty to be interesting, so we decided to build a three degree of freedom, planar, serial link manipulator. In order to further reduce the cost we decided to restrict the maximum amount of gravity that the arm would ever be expected to support itself against. This can be arranged by situating the arm such that the plane that it operates in is at an angle ν from the horizontal. As a consequence we will define the effective gravitational acceleration g_e as

$$g_e = g\sin(\nu) \tag{D.2}$$

[1] In section 5.6, due to empirical observations, we argue that this equation may need to be further augmented in order to model the non-linear effects of joint friction.

[2] Asada and Kanades' method was used to design the Carnegie-Mellon University Direct-Drive Arm I (CMU DD Arm I) [Asada *et al.* 83]. This was soon succeeded by the CMU DD Arm II because the motors were "underspecified" [Schmitz *et al.* 85].

Since the *raison d'être* for our robot arm is to perform assembly tasks we will assume that during its mode of operation of interest to us, both the velocity product terms, and the inertial terms in equation D.1 are negligible. This is because the arm will be operating at very 'slow' speeds, making their effects small anyway, and also because their perturbatory effects on the arm's motion are likely to be swamped by those of the environment as it resists the arm's pushing. This leaves us with equation B.23 except that gravity g is replaced with g_e.

In order to evaluate the maximum torques expected of the motors we need to specify worst case values for the components of the wrist wrench ${}^w\mathbf{w}$. The insertion of a peg into a hole is the archetypical planar assembly task: we will require that the arm be able to exert up to 10N along the axis of insertion, and be able to balance a misalignment moment of up to 0.2Nm. Both these figures are quite high estimates to compensate for the fact that we are not explicitly including inertial terms in our calculations[3]. To eliminate peripheral problems concerned with the behaviour of a part when held in a gripper we decided to construct the arm such that the part which would normally be held by a gripper can be screwed onto the arm[4]. This arrangement means that the 10N that we require along the axis of insertion will always be expected to act along the x-axis of the wrist frame of reference, *i.e.* ${}^w f_{x_{max}} = 10\text{N}$.

We can separate the gravitational loading terms from equations A.34 to A.36 and substitute for g from equation D.2 to give:

$$G_1 = g_e m_0 l_0 \cos(q_1 + q_2 + q_3) \tag{D.3}$$

$$G_2 = g_e\{m_0[l_0 \cos(q_1 + q_2 + q_3) - l_2 \cos(q_2 + q_3)] - m_1 l_2 \cos(q_2 + q_3)\} \tag{D.4}$$

$$\begin{aligned} G_3 = {} & g_e\{m_0[l_0 \cos(q_1 + q_2 + q_3) - l_2 \cos(q_2 + q_3) + l_3 \cos(q_3)] \\ & - m_1[l_2 \cos(q_2 + q_3) - l_3 \cos(q_3)] + m_2 l_3 \cos(q_3)\} \end{aligned} \tag{D.5}$$

Combining these equations with the expression for the wrist Jacobian of B.34 to evaluate equation B.23 we get the following:

$$\tau_1 = l_1 {}^w f_y + \tau_z + g_e m_0 l_0 \cos(q_1 + q_2 + q_3) \tag{D.6}$$

$$\begin{aligned} \tau_2 = {} & -l_2 \sin(q_1) {}^w f_x + [l_1 - l_2 \cos(q_1)] {}^w f_y + \tau_z \\ & + g_e\{m_0[l_0 \cos(q_1 + q_2 + q_3) - l_2 \cos(q_2 + q_3)] - m_1 l_2 \cos(q_2 + q_3)\} \end{aligned} \tag{D.7}$$

$$\begin{aligned} \tau_3 = {} & [-l_2 \sin(q_1) + l_3 \sin(q_1 + q_2)] {}^w f_x + [l_1 - l_2 \cos(q_1) + l_3 \cos(q_1 + q_2)] {}^w f_y + \tau_z \\ & + g_e\{m_0[l_0 \cos(q_1 + q_2 + q_3) - l_2 \cos(q_2 + q_3) + l_3 \cos(q_3)] \\ & - m_1[l_2 \cos(q_2 + q_3) - l_3 \cos(q_3)] + m_2 l_3 \cos(q_3)\} \end{aligned} \tag{D.8}$$

τ_z represents the misalignment moment at the wrist (which is the same in either the wrist or base frames).

From inspection of equation D.6 it is clear that the insertion force will have no effect on the torque required of the wrist motor: this is to be expected because the x-axis of the wrist frame of reference goes through the wrist joint axis. Consequently the maximum torque required of the wrist motor will occur when it has to balance a

[3] The velocity product terms will be minute for the joint velocities that we expect during an assembly task.

[4] We are, after all, just trying to demonstrate the viability of our approach at this stage.

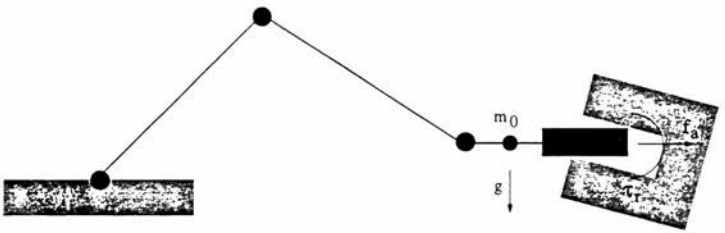


Figure D.1: A worst-case loading situation for the distal link.

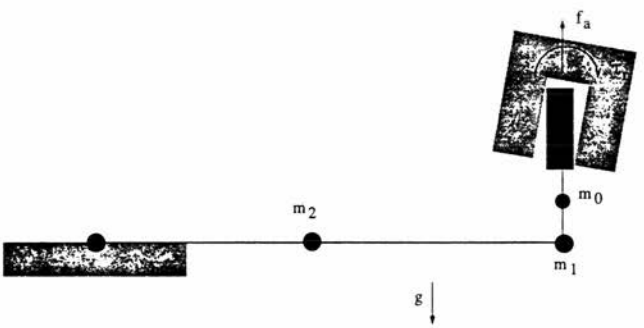


Figure D.2: A worst-case loading situation for the proximal and medial links.

maximum misalignment moment that acts in the same sense as a maximum (effective) gravitational loading. Maximum gravitational loading for the wrist motor occurs when the orientation of the distal link, θ_0, as given in equation B.8, is $n180^o$ where n is an integer. One such configuration is shown in figure D.1.

Worst case loadings for the motors at the shoulder and elbow joints occur when there is a maximum (effective) gravitational loading, the maximum insertion force is required in the opposite sense to the effects of the gravitational field, and there is a misalignment moment to be balanced that is acting about the joints in the same sense as the gravitational loading. These situations occur when $q_3 = n180^o$, $q_2 = 180^o$ and $\theta_0 = 90^o$ (n is an integer). One such configuration is shown in figure D.2.

The general design procedure that we followed was to find the maximum torque required of the wrist motor from equation D.6 evaluated under a worst case loading situation such as that illustrated in figure D.1. We then checked the likely running temperature of the motor (see section D.1.2) to make sure an acceptable temperature range would not be exceeded. We next checked that the motor was in fact capable of a 'respectable' degree of acceleration when not loaded in order to make sure that our

ignoring inertial terms in the arm design was not problematic. Once a motor had been chosen we estimated the weight of a housing for it and repeated the process for the elbow and shoulder joints in turn.

The lengths of the arm links could have been specified *a priori* on the basis of some desired workspace volume. Because of cost considerations however, we were prepared to compromise on the values of these parameters (the longer the links, the greater the effects of gravity, and the larger and more expensive the motor required), so their values, along with the value of ν, became part of an iterative process whereby we would design an arm, check the cost, and if the result was too expensive the motor loadings would be reduced by reducing some or all of these parameters.

D.1.2 Temperature Rise Calculations

Finding a set of direct-drive motors that can deliver the worst-case torques ever likely to be required by a robot manipulator is a necessary activity to a successful arm design but it is not sufficient in itself. In order to realise the torques demanded of them, direct-drive motors are likely to dissipate significant amounts of power. This power dissipation can manifest itself as temperature rises in the motors. The following calculations show how to calculate the steady-state temperature of a direct-drive motor given a particular loading. They are adapted from the article by [Fleisher 88]. We assume throughout this analysis that the index j ranges over $1 \ldots 3$.

The power dissipated in motor j, $P_{d,j}$, is equal to the power into motor j, $P_{i,j}$, minus the power output of motor j, $P_{o,j}$, *i.e.*

$$P_{d,j} = P_{i,j} - P_{o,j} \tag{D.9}$$

The power output from a motor is the product of the load torque and the load speed. The situation of interest to us is what temperature the motors will reach during an assembly task. We will be interested in a worst case arm configuration, and will assume that the arm is exerting a constant wrench on the environment with very little or no motion. This means that $\dot{q}_j \approx 0, \forall j$, giving us

$$P_{o,j} \approx 0, \ \forall j \tag{D.10}$$

In other words we are assuming that for our application, due to the fact that we will always be operating in or near stall conditions, all the power dissipated by a motor is dissipated as heat. The input power to a motor can be calculated from the product of its running voltage and current, *i.e.*

$$P_{i,j} = i_{mot,j} V_{mot,j} \tag{D.11}$$

The expression for a motor's running voltage is given by equation 5.2, which we reproduce here for convenience

$$V_{mot,j} = i_{mot,j} R_{mot,j} + E_j \tag{D.12}$$

where E_j is motor j's back e.m.f., and $R_{mot,j}$ is its terminal resistance. In our case, since the back e.m.f. is the product of the motor's speed and back e.m.f. constant (equation 5.4), we have

$$E_j = 0, \ \forall j \tag{D.13}$$

Combining equations D.12 and D.13, and substituting for the voltage term in equation D.11 gives

$$P_{i,j} = i_{mot,j}^2 R_{mot,j} \tag{D.14}$$

The relationship between the torque produced by a motor and its current is given by equation 5.5 which we reproduce here:

$$\tau_{mot,j} = K_{t,j} i_{mot,j} \tag{D.15}$$

$K_{t,j}$ is the torque sensitivity coefficient or motor torque constant for motor j. Rearranging and substituting for $i_{mot,j}$ in equation D.14 gives

$$P_{i,j} = \left(\frac{\tau_{mot,j}}{K_{t,j}} \right)^2 R_{mot,j} \tag{D.16}$$

And since the motor constant for motor j is defined as:

$$K_{m,j} = \frac{K_{t,j}}{\sqrt{R_{mot,j}}} \tag{D.17}$$

it follows that,

$$P_{i,j} = \left(\frac{\tau_{mot,j}}{K_{m,j}} \right)^2 \tag{D.18}$$

Combining equations D.9, D.10, and D.18 gives

$$P_{d,j} = \left(\frac{\tau_{mot,j}}{K_{m,j}} \right)^2 \tag{D.19}$$

Once we have calculated the power dissipated by the motor, the rise in temperature above ambient is given by

$$T_{r,j} = P_{d,j} \times \mathrm{TPR}_j \tag{D.20}$$

where TPR_j is the temperature rise coefficient for motor j in degrees centigrade per watt (^{o}C/W).

For the application area that we are interested in it was very difficult to say *a priori* what the duty cycle for the motor torques would be like. We expected that, in all likelihood, the maximum values of the end-point wrench would only be required intermittently, if at all. So we decided to evaluate the temperatures that the motors would achieve under steady state conditions for worst case loading situations whilst the arm is being commanded to deliver a force of 5N and a torque of 0.2Nm (in the wrist frame of reference): the 5N being half of the maximum that we might require, and the 0.2Nm might easily be induced as a loading on the wrist motor whilst the shoulder and elbow motors combine to deliver the required 5N. The τ_j we used to calculate the temperature rise above ambient follow from equations D.6 to D.8.

D.2 The Motor Current Regulation Feedback Loops

In section 5.1.2 we describe the basic behaviour of a d.c. motor under current control and figure 5.2 illustrates the basic circuit configuration involved in order to realise this behaviour. In this section we describe the design rationale behind this circuitry.

The sense resistor R_s in this kind of application is a very low value; we used two 0.1Ω resistors in parallel giving us $R_s = 0.05\Omega$. This means that virtually all the current that flows through the motor flows through this resistance because the input impedance of the feedback scaling amplifier is of the order of megohms. Consequently the voltage across the sense resistor, V_s is given by

$$V_s = i_{mot} R_s \tag{D.21}$$

i.e. it is proportional to the current flowing through the motor and can therefore be used as a measurement of this current.

The reference values coming from the D/A card in the PC are voltages. We used a double ended D/A card, giving us output signals in the range $\pm V_{D/A_{max}}$. The maximum torque that a motor can generate, $\pm\tau_{mot_{max}}$, will occur when its maximum current $\pm i_{mot_{max}}$, is flowing through it (see equation 5.5). In order to make the most of the output range of the D/A converter we would like a current of $i_{mot_{max}}$ to correspond to a voltage of $V_{D/A_{max}}$ on the relevant channel of the D/A converter. The voltage across R_s when the motor is producing full torque will be (approximately) given by

$$V_{s_{max}} = i_{mot_{max}} R_s \tag{D.22}$$

Now, since the overall system is a feedback loop we require that, in ideal steady-state conditions, the value of the output voltage of the feedback amplifier, when subtracted from the control voltage coming from the D/A converter, will result in $0V$, *i.e.* there is zero error[5]

$$V_{D/A_{max}} - A_{fb} V_{s_{max}} = 0 \tag{D.23}$$

where A_{fb} is the gain of the feedback amplification stage in V/V. Substituting for $V_{s_{max}}$ from equation D.22 and rearranging we arrive at

$$A_{fb} = \frac{V_{D/A_{max}}}{i_{mot_{max}} R_s} \tag{D.24}$$

Note that because we are primarily interested in operating our arm to perform assembly tasks, that the motors will constantly be operating in stall or near stall conditions. This means that the motors' back e.m.f. in equation 5.2 will be approximately zero implying that the motors' behaviour can be modelled as[6]

$$V_{mot} = i_{mot} R_{mot} \tag{D.25}$$

In other words, electrically the motors can be considered to be behaving like pure resistances.

We will now show that the current regulation feedback loop designed as described above realises a scheme whereby the torque generated by a motor is proportional to the D/A voltage generated on the PC. This analysis assumes that the bandwidth of the

[5] In practice there is always an error since otherwise there is no command to the power amplifier and no power is delivered to the motor.

[6] Actually the p.d. across the motor should be $V_{mot} - V_s$ but we assume that the p.d. across the sense resistors are so small as to be negligible.

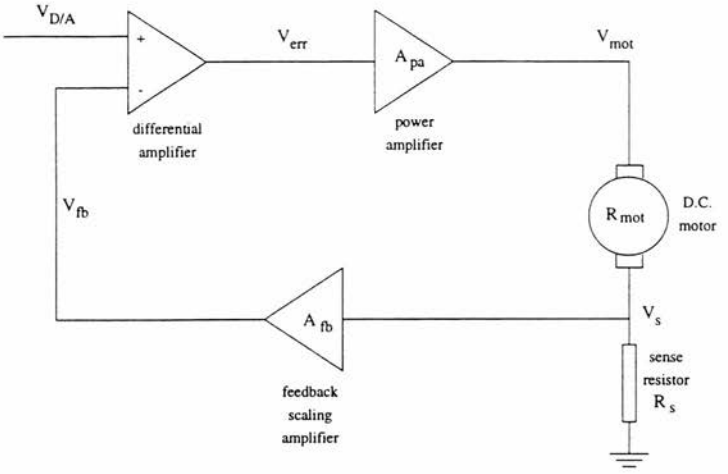


Figure D.3: The electrical parameters in the motor current regulation feedback loops.

closed loop is much greater than the frequency at which we change the reference signals to the loops so that we can ignore the frequency dependency of the variables. Since changes in the reference signals are of the order of a few Hertz[7] this is a reasonable assumption, and we can consider the loop to continuously be in a steady state.

With reference to the parameters illustrated in figure D.3 we can write down the following relationships between the voltages:

$$V_{err} = V_{D/A} - V_{fb} \quad \text{(D.26)}$$

$$V_{mot} = A_{pa} V_{err} \quad \text{(D.27)}$$

$$V_{fb} = A_{fb} V_s \quad \text{(D.28)}$$

where $V_{D/A}$ is the digital to analogue converter voltage, V_{fb} is the feedback voltage, V_{err} is the error voltage, A_{pa} is the transfer function of the power amplifier (including a pre-amplifier stage), and A_{fb} is the transfer function of the feedback amplifier. Because we are assuming we can ignore frequency dependencies the amplifier transfer functions can be considered to be pure gains.

If we substitute for V_{mot} in equation D.25 from equation D.27 we get

$$A_{pa} V_{err} = i_{mot} R_{mot} \quad \text{(D.29)}$$

If we now substitute for V_{err} from equation D.26 and rearrange we get

$$V_{D/A} = \frac{i_{mot} R_{mot}}{A_{pa}} + V_{fb} \quad \text{(D.30)}$$

[7] This in fact gives us a further justification for ignoring the inductance term in equation 5.1: not only is the inductance 'small', but so is $\frac{di_{mot}}{dt}$.

Next substitute for V_{fb} from equation D.28

$$V_{D/A} = \frac{i_{mot}\, R_{mot}}{A_{pa}} + A_{fb} V_s \tag{D.31}$$

then substitute for V_s from equation D.21

$$V_{D/A} = i_{mot} \left(\frac{R_{mot}}{A_{pa}} + A_{fb} R_s \right) \tag{D.32}$$

But as part of our loop design we chose that $R_s\, A_{fb} = \frac{V_{D/A_{max}}}{i_{mot_{max}}}$ (equation D.24). Taking this into account, substituting for i_{mot} from equation 5.5 and rearranging yields

$$\tau_{mot} = \frac{K_t\, i_{mot_{max}}\, V_{D/A}}{\left(\frac{R_{mot}\, i_{mot_{max}}}{A_{pa}} + V_{D/A_{max}} \right)} \tag{D.33}$$

But, from equation 5.5 we have

$$\tau_{mot_{max}} = K_t\, i_{mot_{max}} \tag{D.34}$$

and substituting $V_{mot_{max}}$ for $R_{mot}\, i_{mot_{max}}$ we arrive at

$$\tau_{mot} = \frac{\tau_{mot_{max}}}{\left(\frac{V_{mot_{max}}}{A_{pa}} + V_{D/A_{max}} \right)} V_{D/A} \tag{D.35}$$

or

$$\tau_{mot} = A_{loop} V_{D/A} \tag{D.36}$$

where

$$A_{loop} = \frac{\tau_{mot_{max}}}{\left(\frac{V_{mot_{max}}}{A_{pa}} + V_{D/A_{max}} \right)} \tag{D.37}$$

which is the (frequency independent) loop transfer function between the D/A voltage and the motor torque. If $A_{pa} \gg V_{mot_{max}}$ then $A_{loop} \approx \frac{\tau_{mot_{max}}}{V_{D/A_{max}}}$ and equation D.35 reduces to

$$\frac{\tau_{mot}}{\tau_{mot_{max}}} = \frac{V_{D/A}}{V_{D/A_{max}}} \tag{D.38}$$

showing that under this condition there is a direct correspondence between the specified D/A voltage as a fraction of the maximum possible and the motor torque as a fraction of the maximum possible. In our implementation we assumed the performance of our loops to approximate the behaviour implied by equation D.38.

D.3 The Actual Design

The final design of our manipulator was in part constrained by financial considerations, and in part by our desired to keep things simple, working on the principle that complex systems include more opportunities for things to go wrong. We chose to use brushed d.c. motors controlled by linear power amplifiers. We initially experimented with using pulse width modulation (PWM), but found it difficult to get a useful error signal in the current regulation loop without significant filtering of the feedback signal, which

arm joint	wrist	elbow	shoulder
motor model	BMU075S20A1	BMU100S45A1	BMU100S65A1
peak torque (Nm)	0.73	6.0	9.85
power input at peak torque (Watts)	63	290	450
volts at peak torque (Volts)	28	28	50
terminal resistance (Ohms)	12.4	2.7	5.5
motor inductance (mH)	12.4	5.67	13.2
motor constant (Nm/$\sqrt{\text{W}}$)	0.09	0.352	0.464
temperature rise/loss (°C/W)	8	2.0	1.4
weight (kg)	0.37	1.5	2.2
torque constant (Nm/A)	0.32	0.58	1.08

Table D.1: The characteristics of the motors used in our final design.

introduced a time lag. PWM has the advantage that the power transistors used spend a lot of their time in saturation where they dissipate little power. This makes them the preferred method of powering motors that require large currents. The power amplifier that we used for our shoulder motor is in fact the largest commercially available linear power amplifier that we could find. This means that it would have been difficult for us to build an arm any bigger than we did using commercially available linear power amplifiers.

Our final design uses three brushed direct-drive motors with samarium cobalt magnets from the metric d.c. torque motor range of Muirhead Vactric Components of Penge, Kent. The motors chosen and some of their electro-mechanical specifications are listed in table D.1

The details of the power amplifiers that we used are listed in table D.2

The final geometry of our direct-drive arm is as follows:-

$$l_0 = 0.06\text{m} \quad \text{(D.39)}$$

$$l_1 = 0.14\text{m} \quad \text{(D.40)}$$

$$l_2 = 0.212\text{m} \quad \text{(D.41)}$$

$$l_3 = 0.203\text{m} \quad \text{(D.42)}$$

(Equation D.40 assumes that there is a 5cm block attached to the arm.) When the

arm joint	manufacturer	model	output voltage (Volts)	maximum continuous current (Amps)	peak current (Amps)
wrist	PVP Controls Camberley, Surrey	PVP144	±35	4	8
elbow	PVP Controls Camberley, Surrey	PVP173	±24	12	24
shoulder	McLennan Servo Supplies Camberley, Surrey	EM200	±50	9	30

Table D.2: Electrical characteristics of the power amplifiers that we used.

weight of the motor housings is added to the weight of the motors the (point) masses are:–

$$m_0 = 0.1\text{kg} \quad \text{(D.43)}$$
$$m_1 = 1.0\text{kg} \quad \text{(D.44)}$$
$$m_2 = 3.2\text{kg} \quad \text{(D.45)}$$

If we set $\nu = 30^0$ then

$$g_e \approx 4.9\text{m/s}^2 \quad \text{(D.46)}$$

Given these parameters we can work out the torques required of our motors and the steady state temperature rises for our chosen worst case conditions.

The torque required of the wrist motor when in the worst case configuration represented by figure D.1, *i.e.* when $\theta_0 = 180^o$ and it is being commanded to deliver 0.2Nm (from equation D.6) is:

$$\tau_1 = 0.23\text{Nm} \quad \text{(D.47)}$$

The temperature rise above ambient (from equation D.20) is:

$$T_{r,1} = 52^o\text{C} \quad \text{(D.48)}$$

Performing the corresponding calculations for the elbow and shoulder motors for the worst case configuration shown in figure D.2 with ${}^w f_x = 5\text{N}$ and $\tau_z = 0.2\text{Nm}$ we get

$$\tau_2 = 2.4\text{Nm} \quad \text{(D.49)}$$
$$\tau_3 = 7.7\text{Nm} \quad \text{(D.50)}$$

and

$$T_{r,2} = 93^o\text{C} \quad \text{(D.51)}$$
$$T_{r,3} = 386^o\text{C} \quad \text{(D.52)}$$

Note that the temperature rise calculations give the rise above ambient, so we can add another 20°C for the ambient temperature to get the actual motor temperatures. The motors are rated to operate up to 180°C, but for safety purposes we have included

thermistors in the motor housings that switch at approximately 80°C. The above figures imply that the wrist motor should be no problem from a temperature rise point of view, but this cannot be said of the elbow and shoulder motors. Things are not as bad as they seem however. Fleisher notes [Fleisher 88] that the value of TPR quoted by manufacturers is for the unmounted motor: the TPR for a mounted motor can be 25% of that for an unmounted motor. In addition our temperature calculations don't tell the whole story: they give steady state values; it takes some time for a motor to reach these temperatures dependent on its thermal time constant. Muirhead Vactric possess some software that calculates the temperature rise of their motors over time. For the loadings that we require, with the TPR set at half of the manufacturers quoted value in order to take into account the motor mountings, the elbow motor would take in excess of five minutes to exceed 80°C, and the shoulder would take in excess of a minute. Because we were working to a budget we decided that this would be acceptable, because it was likely that the TPR would be reduced by more than one half of the quoted value, it was highly unlikely that a worst case configuration would have to be held for any length of time (in fact we could ensure that this was the case since we would be chosing the locations in which the experiments would be performed), and in the worst case we could arrange that the arm operate against a lower degree of gravity. It might mean that we would have to let the arm cool in between assembly trials, but we could live with that. We also included in the mechanical design of the arm the possibility of including some counterbalancing about the shoulder joint in order that we might, if necessary, reduce the gravitational moments that the shoulder motor would need to equal. So far temperature rises have not been an issue during the operation of our arm because all the experiments we have conducted have been in a horizontal plane.

One other consideration that we thought might cause us problems is that of torque ripple. This is a variation in the torque that a motor produces that is dependent on the relative location of its windings and magnets. It arises due to variation in the magnetic field strength as the magnets and windings vary their alignment, effectively causing a variation in the torque constant K_t. It occurs in all d.c. motors but is not usually noticeable because of the speed at which they are normally operated. We are not aware of experiencing any practical problems as a consequence of this phenomenon, possibly because of the fact that our assembly strategies are not dependent on achieving precise wrench values, and the wrench values adopted during a particular configuration transition are found by performing a search in wrench space until the desired conditions are met.

D.4 Position and Velocity Sensing

The current trend in robotics appears to be towards using very precise joint angle measurements, typically from optical shaft encoders. The accurate measurement of the position and velocity of a joint of a direct-drive manipulator is more difficult than for a geared robot because the motor is moving at the same rate as the arm link. With a geared robot the motor speed is much greater than that of the link it is driving making changes in position easier to detect. In [Kanade & Schmitz 85] the design of the CMU DD Arm II is described where pancake resolvers are used to achieve position measurements with a resolution of 2^{16} intervals per shaft resolution, and a worst case

error of ±2 LSB.

In order to sense the joint angles of our manipulator we chose to use 'hand calibrated' potentiometers. That is to say that each joint was taken to two reference positions that were measured by hand with a set square and the voltage that corresponded to these points was recorded as the corresponding bit pattern of a 12 bit A/D converter. A linear relationship was assumed between the joint position and the measured voltage so for each joint j the two measurements gave us two equations of the form

$$q_{ref1,j} = q_{bits1,j}\, a_j + b_j \tag{D.53}$$

$$q_{ref2,j} = q_{bits2,j}\, a_j + b_j \tag{D.54}$$

which we solved as simultaneous equations to find the values of a_j and b_j for each joint, j.

The potentiometers that we used were Radio Spares conductive plastic servo potentiometers rated at 10kΩ (stock number 173-580). Their linearity is listed as ±0.5%. Although in principle a potentiometer, being an analogue device, gives infinite resolution, in practice the resolution is limited to that of the A/D converter used (in our case 12 bits). More importantly the use of potentiometers in preference to optical methods tends to produce noisy results because optical encoders, being digital in nature, are significantly less prone to noise[8]. We chose this joint sensing equipment and method of calibration in order to demonstrate that it is possible to perform assembly tasks reliably and to tight tolerance without the use of accurate position sensing. In fact it is part of our thesis that our technique works *because* we don't require accurate position information.

We could have chosen to derive the joint velocities by digitally differentiating the joint positions. We chose however to perform the differentiation in analogue electronics in order to reduce the computational burden on our PC. The circuit we used is shown in figure D.4 which we include here because, as far as we are aware, it doesn't appear in any text book. It appears that the authors of electronics text books assume that the frequencies of interest for a differentiator will be at least in the order of kilohertz. The circuit shown, which we arrived at through trial and error by varying its constituent components, is in fact a bandpass filter with a resonant frequency of approximately 16Hz. We include it here as a useful technique that other designers might find of interest. It is our understanding of this circuit that for signals in the low frequency end of its pass band it behaves as a frequency to voltage converter, though we have not analysed this behaviour in detail. For the frequencies that we are working with, *i.e.* a few Hertz, the circuit works well (as calibrated against a digitally differentiated signal), but again it is relatively noisy compared to a velocity derived from a digital sensor. Nevertheless, (with the aid of some running average filters and appropriate thresholding) our assembly strategies are robust in the face of this noise.

[8] We do in fact perform some digital filtering of the signals in the form of running averages.

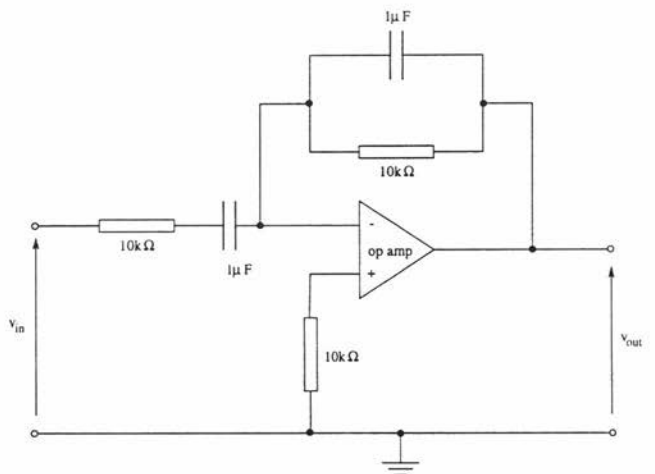


Figure D.4: An analogue differentiator circuit for low frequency signals.