Contact Sensing: A Sequential Decision Approach to Sensing Manipulation Contact Features

by

Brian Scott Eberman

Revised version of a thesis submitted to the Department of Mechanical Engineering in partial fulfillment of the requirements for the degree of Doctor of Philosophy in Mechanical Engineering at the Massachusetts Institute of Technology on February 10, 1995.

Abstract

This paper describes a new statistical, model-based approach to building a contact state observer. The observer uses measurements of the contact force and position, and prior information about the task encoded in a graph, to determine the current location of the robot in the task configuration space. Each node represents what the measurements will look like in a small region of configuration space by storing a predictive, statistical, measurement model. This approach assumes that the measurements are statistically block independent conditioned on knowledge of the model, which is a fairly good model of the actual process. Arcs in the graph represent possible transitions between models. Beam Viterbi search is used to match the measurement history against possible paths through the model graph in order to estimate the most likely path for the robot. The resulting approach provides a new decision process that can be used as an observer for event driven manipulation programming. The decision procedure is significantly more robust than simple threshold decisions because the measurement history is used to make decisions. The approach can be used to enhance the capabilities of autonomous assembly machines and in in quality control applications.

© Massachusetts Institute of Technology 1995

Acknowledgments

This research was performed at the MIT Artificial Intelligence Laboratory of Massachusetts Institute of Technology. Support for this research was provided by the Office of Naval Research University Research Initiative Program under Office of Naval Research contract N00014-92-J-1814.

I would like to thank the members of my thesis committee. My thesis advisor, and committee chairman, Dr. Kenneth Salisbury provided motivation for this work through many valuable discussions of sensing and manipulation. His support, patience, and reviews guided all my research at MIT. Discussions with Prof. Tomás Lozano-Pérez provided insight into the relationship of sensing to geometry and configuration space. The relationship of feature model space to configuration space presented in this thesis is an outgrowth of work done in his research group over many years. I would like to thank Prof. Wallace E. Vander Velde for assistance in developing the statistical theory, and his comprehensive review of the thesis. He often provided a critical reference or source when the need arose, and was available to review difficult statistical problems. I would like to thank Prof. Jean-Jacques Slotine for always reminding me that it is not always necessary to sense everything to perform a task.

Thanks to many friends and colleagues at MIT over the years. In particular I want to thank my closest peers David Brock, Mike Caine, Sundar Narasimhan, Gunter Niemeyer, and Jose Robles. Gunter Niemeyer's tireless work on the Hummingbird software system made it possible to actually do the experimental work in the thesis. I would like to thank Sundar Narasimhan and Jose Robles for the many discussions about the problem of manipulation programming and sensing, and their careful review of the thesis. I would like to thank David Brock for discussions about manipulation, and for building the force sensor I used in this thesis.

I would also like to thank Bill Goldenthal for many valuable discussions about statistical signal processing.

Thanks to my fellow members of Dr. Salisbury's research group: Catherine Anderson, Susanna Leveroni, Mike Levin, Akhil Madhani, Thomas Massie, John Morrell, Tom Moyer, Janis Stipins, Nick Swarup, and Craig Zilles. Thanks especially to Thomas Massie for the use of the PHANToM.

To my brothers Doug and Kevin and parents Kay and Paul, who supported me and kept telling me eventually I would finish.

Finally, to my wife Margaret who saw everyday of the effort, and without who's love and support this thesis would not have been possible.

Contents

Notation

The notation in this thesis tries to follow the conventions of the International Journal of Robotics Research. Scalar constants and variables are typeset in plain math italics. Vectors and matrices are typeset in bold math italics. Sets are typeset in calligraphic. Functions are typeset according to the type that they return.

This thesis uses concepts from statistics, differential geometry, and robotics so there is quite a bit of notation. This sheet provides a reference to the meaning of each symbol.

Below is the configuration space and planning notation. This notation is primarily introduced in chapter 2.

$\mathcal{C}$	Configuration space
$\mathcal{F}$	Free space
$\mathcal{O}$	The obstance space
ϕ	Empty set or space
$\mathcal{X}$	The robot's task state space
$\mathbf{x}$	The current robot state or configuration
$\mathbf{v}$	Generalized velocity of the robot
$\mathcal{G}$	A goal subset of $\mathcal{C}$
$\mathcal{M}$	The measurement function
$\mathcal{F}$	Forward projection operator
$\mathcal{B}$	Backward projection operator
$\mathcal{U}$	Control space
$\mathbf{u}$	Generalized control
$\mathcal{K}$	Interpretation set

Below is the differential geometry notation. This notation is primarily introduced in chapter 4.

$\mathcal{M}$	A manifold
$\mathcal{T}(\mathcal{M})$	The tangent bundle of a manifold $\mathcal{M}$
$\mathcal{T}_{\mathbf{x}}(\mathcal{M})$	The tangent space at $\mathbf{x}$ of $\mathcal{M}$
$\mathcal{T}^*(\mathcal{M})$	The cotangent bundle of a manifold $\mathcal{M}$
$\mathcal{T}^*_{\mathbf{x}}(\mathcal{M})$	The cotangent space at $\mathbf{x}$ of a manifold $\mathcal{M}$
$\mathcal{P}$	Phase space $= \mathcal{C} \times \mathcal{T}(\mathcal{C})$
$\mathbf{g}$	Geometric parameters of a constraint equation
$\mathbf{C}(\mathbf{g}, \mathbf{x})$	A contact constraint equation
$\mathbf{C}_{\mathbf{x}}$	Partial derivative of $\mathbf{C}$ with respect to $\mathbf{x}$
$\mathbf{C}^*_{\mathbf{x}}$	Complement of $\mathbf{C}_{\mathbf{x}}$ and a basis for the cotangent bundle

Below is notation about the robot joint and endpoint coordinates and dynamics. This notation is introduced primarily in chapter 4.

$\mathrm{d}\mathbf{x}$	Differential endpoint or state-space motion
$\dot{\mathbf{x}}$	Endpoint velocity of the robot
$\ddot{\mathbf{x}}$	Endpoint acceleration of the robot
$\mathbf{q}$	Robot joint coordinates
$\mathrm{d}\mathbf{q}$	Differential joint motion
$\dot{\mathbf{q}}$	Robot joint velocity
$\ddot{\mathbf{q}}$	Robot joint acceleration
$\mathbf{H}$	Generalized mass or inertia
$\mathbf{Cor}$	Coriolis acceleration
$\mathbf{K}$	Generalized spring
$\mathbf{B}$	Generalized damper
$\mathbf{p}$	Generalized momentum

Below is the measurement and statistical notation. A standard is that any symbol with a hat over it is an estimate of the value of the symbol, and any symbol with a tilde over it is the difference between the estimate and the true value of the symbol.

$\mathbf{y}$	Measurement vector
$\hat{\mathbf{y}}$	Predicted measurement
$\mathbf{w}_m$	Generalized force measurement
$\mathbf{s}$	The strain measurements
v	The vibration measurements
$\hat{\mathbf{n}}$	Estimated normal to a surface
$\mathbf{h}$	Observer function
$\mathrm{P}(x)$	Probability of an event
$\mathrm{p}(x)$	Probability density function
E	Expected value
$\mathbf{V}$	Variance or covariance of a variance
H_i	Hypothesis i
N	Standard normal distribution
U	Uniform distribution over a set
Exp	Exponential distribution
ME	Maximum entropy distribution over a set
χ	Chi-square distribution
$\mathcal{I}$	Measured empirical covariance
i.i.d.	Independent identically distributed
L	Likelihood of an event
l	Log-likelihood of an event
$\boldsymbol{\theta}$	Parameter vector of a distribution
$\boldsymbol{\nu}$	White residual process from an estimator
$\mathrm{d}\mathbf{v}$	White Wiener process

Below is the feature notation.

$\mathcal{N}$	Feature nodes
$\mathcal{E}$	Feature edges
$\mathcal{G}$	Feature graph $= (\mathcal{N}, \mathcal{G})$
$\mathcal{P}$	Feature partition

Introduction

Chapter 1

A basic problem of autonomous manipulation and semi-autonomous telerobotics is to perform a basic control primitive: move object until some condition is satisfied. In fine manipulation, the primitives may be defined in terms of contact constraints and forces on the object. For example, we might like to instruct a Mars rover to probe for hard rocks using a stick-like probe. In assembly, we would like to instruct a robot to mate object A with object B. As a last example, a maintenance robot might be instructed to tighten a fastener until "it is tight". The facility with which humans perform all these tasks hides the complexity and detail involved in these simple instructions.

Each of these tasks involves the interaction of a grasped part with the environment. Therefore, the contact forces between the part and the environment cannot be directly measured. Instead, the interaction state must be inferred from sensors in the robot wrist or end-effector. In addition, often in robotic grasping the end-effector cannot directly measure grasp forces. Instead force and torque sensors are placed in the wrist or fingertips which are used to infer the grasp force and thereby the current grasp state. Therefore, an understanding of perception which helps to track the progress of mating tasks using force sensing has wide applicability in robotic manipulation.

These manipulation tasks are usually accomplished by a sequence of steps. Generally each step reduces the relative positioning error between the manipulated parts by incrementally adding contact constraints. Each step is usually a compliant motion and the sequence of motions is indexed by some termination predicate. The sequence of steps may depend upon the effects of the interaction. Since there can be multiple outcomes in any given interaction, branching and context based decision making is always necessary.

For example, one strategy for peg-in-hole assembly is to bring the the peg into contact with the top surface. Then a new motion is used to place the tip of the peg in the hole. Then the peg is approximately aligned with the hole. Finally, an insertion step is used to push the peg into the hole.

Multi-step strategies also occur in grasping. As an example consider the experiment of grasping, lifting, and replacing an object of unknown mass given by [Howe *et al.*, 1990]. In their paper, the sensors measured fingertip forces, accelerations, and relative slip while a two finger grasper was used to grasp, lift, and replace a single test object. Six distinct control steps, or *phases*, were determined for this task: pre-contact, loading, manipulation, unloading, post-contact, and slip. Each phase required a different control algorithm, and termination of each phase was signaled by different signal predicates.

The idea is to connect a set of continuous control algorithms into a discrete graph. Transitions in the graph are controlled by a decision process that uses the measurements and possibly the controls. A collection of thresholds can serve as a simple, but not very robust, decision procedure. When the measurement cross these thresholds a *contact event* is declared to have occured. The resulting controller is a mixture of continuous controllers embedded within a nondeterministic finite state machine. The algorithm which observers the measurements to detect contact events, or to determine the contact state is a *contact state observer*.

This idea of event driven manipulation programming has been evolving under different names at many research centers. Brooks [Brooks, 1985, Brooks, 1987] argues for an implementation called the subsumption architecture. Howe and Cutkosky define the events by the measurements generated as the robot transitions between manipulation phases. They used these events to drive their grasping algorithm. Another work [McCarragher and Asada, 1993b] focused on the transitions to drive an assembly algorithm.

In most of the previous work, the contact state observer has taken the form of a set of thresholds. For unstructured manipulation this level of perception is not sufficient. Uncertainty, noise, and the range and frequency of contact events makes observing the contact state difficult. More powerful statistical tools are needed to formulate and solve the observer problem.

Hannaford's [Hannaford and Lee, 1991] use of Hidden Markov Models (HMM) is one of the few formulations of the observer as a statistical decision problem. They modeled the sequence of forces that occur during a given manipulation using a HMM for the force. Although highly effective for repetitive problems, such as assembly, the approach may not be able to deal with less structured problems because in general the forces depend upon the commands sent to the robot.

More recently [Delson, 1994] looked at programming a robot using human demonstration. As part of this work he developed some tests for segmenting the motion data into section that come from a single contact configuration. The tests were done either only on the force or on the velocity depending upon the contact condition. This thesis significantly extends his work by explicitly modeling the process noise, making decisions on all of the data, making all the decisions on both the position and the force information, and creating a framework that can incorporate more than just constraint models.

This thesis presents a new statistical, model-based approach to building contact state observers. The first part of the thesis shows how the measurements of forces and positions produced in a given task can be described as a graph of predictive, statistical, measurement models. Each model describes statistics of the measurements that are intrinsic to the task and which are not functions of the applied command.

The second part of the thesis shows how the graph and the measurement models can be used as basis for constructing a contact state observer. Because of the statistical formulation of the measurement models, we can formulate the observer as a statistical decision problem. The result is that the observer is simply a search procedure which attempts to determine a collection of statistically most likely paths for the measurements through the graph and thereby observers the contact state.

This approach can produce timely results and incorporate global context. The approach also makes explicit use of the complete measurement history. Decisions are made dynamically by comparing one possible model path for the measurements to other model paths. This eliminates the problems inherent in making fixed threshold decisions on a fixed, generally short, length of data. In addition, since decision tests are derived from models, the assumptions made in picking decision thresholds are explicit.

Figure 1.1 shows the essential idea of this thesis. The figure shows how the information available to a robot with an endpoint force sensor in a typical task can be represented by a collection of constraint models.

The upper-left figure shows a circular robot constrained by a maze with five legs. The position and orientation of the maze are only approximately known. This figure pictorially represents a spherical fingertip that was attached to a three degree-of-freedom robot, the PHANToM, that was used to experimentally verify the theory. The PHANToM is shown in the maze in figure 1.2. A six axis force-torque sensor, shaped like a fingertip, is attached at the end of the PHANToM. Encoders on the PHANToM's motors are used to measure the Cartesian position of the fingertip. The measured position was first order differenced and lowpass filtered to compute the fingertip velocity.

The constraints between the robot and the maze are best represented by the robot's *configuration space*. The configuration space, or *c-space*, represents all the locations the robot's reference frame can be placed without causing a collision. The c-space is shown in the lower-left of figure 1.1. The c-space representation shows that the maze is representative of may assemblies. Many real assemblies look like tubes or mazes when the c-space is represented in the correct coordinates.

Constant velocity commands can be applied to the robot to move it through the maze. The lower-left of figure 1.1 shows a motion from the bottom-left leg to the upper-middle leg resulting from the indicated velocity command. Part of the path is highlighted in the figure.

In all the experiments, the force sensor measured the three Cartesian contact forces between the robot and the maze, and the position sensor measured the Cartesian position of the robot relative to the robot's fixed ground. The upper-right of figure 1.1 shows a sequence of forces and velocities measured by the sensors for one trial for this example path.

The contact state observer problem is to determine from knowledge of the c-space and force and position measurements where the robot is in the maze. The force trace shows why this is a difficult problem. There is considerable noise due to vibration and friction stick-slip displayed in the force signal. Similar noise also appears on the velocity signal. Both signals are statistically non-stationary, and there are several short time impact events in the force signal.

The bottom-right figure of figure 1.1 shows the central representation idea of this thesis. Each contact in the c-space is represented as a constraint model. The robot starts out in the lower-left corner of the c-space and this is represented by the node showing the corner contact. The motion command causes the robot to move into free space and then against the opposite wall. This is shown by the empty node

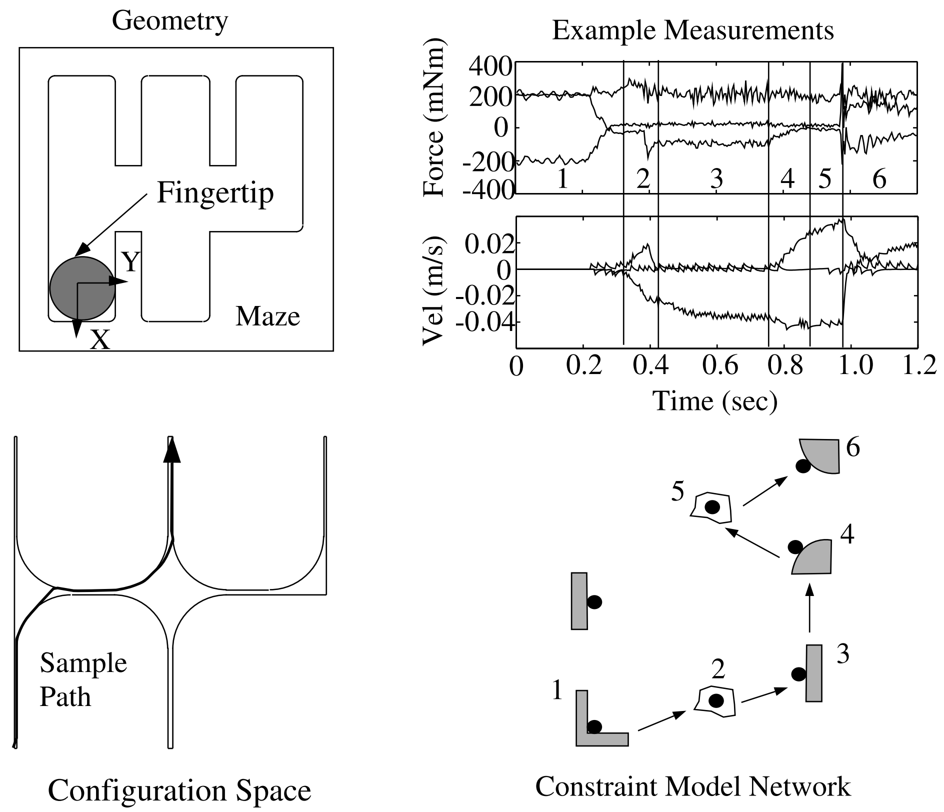


Figure 1.1: Example of information available in contact sensing for a circular robot in a maze, and the representation of the task in terms of a graph of constraint models.

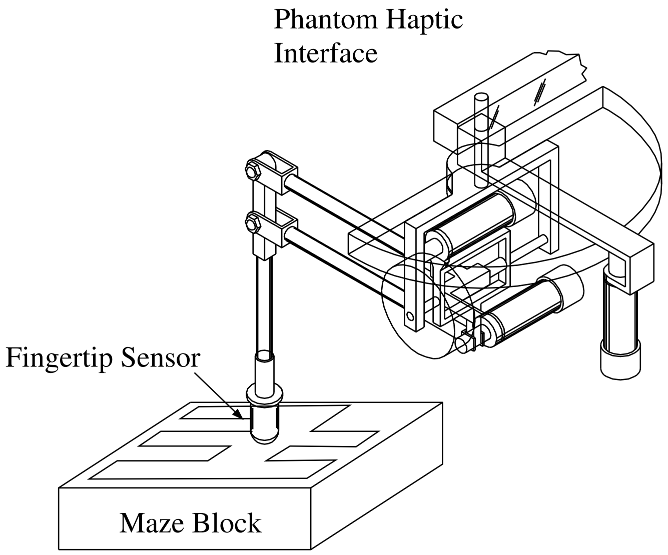


Figure 1.2: The PHANToM haptic interface with a force sensing fingertip. This was the experimental apparatus used in the experiments in this thesis.

followed by the point in contact with the single wall. In this way each of the nodes represents a single, continuous, constraint situation in the c-space. The connection of the nodes comes from the connections caused by the indicated command in the underlying c-space.

Each of these nodes represents a single contact situation for which a predictive statistical measurement model can be created. The predictions can be used by a search procedure to determine the best marking of the measurement sequence against possible paths through the graph of models. The best marking for this example path is shown in the upper-right figure. A procedure which finds the best marking is in effect a contact state observer.

Other models, such as models for the vibration level, spectral shape of the vibration, or impacts can also be added into the graph. Transitions from free space into a contact situation can give rise to impact events. These appear in the force trace, and detecting these events can help in tracking motion through c-space.

Figure 1.3 shows the basic components of the contact state observer. The forces and positions are input into a collection of candidate estimators. There is one estimator

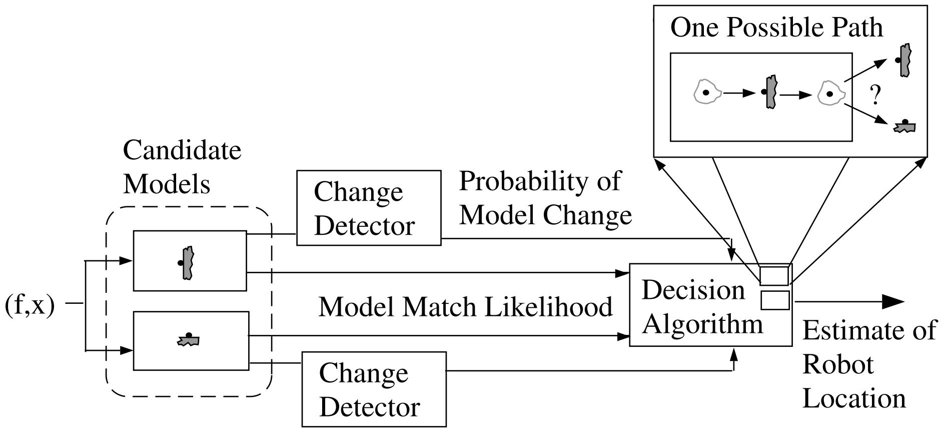


Figure 1.3: Contact state observer processing. The figure shows the basic architecture of the contact state observer processing force and position information measured by the robot as it explores the maze.

for each hypothesized contact model. The estimators output a measure of match between the underlying model and measurements called the log-likelihood.[1] The estimators also output a residual process, which conditioned on the model being true, is a white noise sequence. The residual processes gets fed into a change detector. The change detector monitors the residuals for changes. The change detector computes the log-likelihood of departure from the current model and the most likely time of departure. Both of these statistics, and the model likelihoods computed by the estimators, are sent into a single decision procedure. This procedure uses all these process statistics and the graph to determine a collection of likely paths for the measurements through the graph using a search strategy. Since each of these paths terminates in a state, the procedure also computes an estimate of the current state of the robot. Therefore, this process reduces the problem of observing the contact state to the problem of observing the current contact model.

The coding of the task, and all of the modeling work lies in determing appropriate measurement models and how they should be related given the geometry and applied command. These measurement models are called *contact features* or just *features*. The rest of the procedure follows from the specified process statistics.

The rest of the thesis presents the components of the observer in detail and a demon-

[1]The likelihood of a measurement is the conditional probability or density for the measurement given the parameters of the statistical model. The log-likelihood is the logarithm of the likelihood.

stration using the maze is developed to illustrate all the components acting together. The first component is the collection of measurement models and their associated estimators. The first model is a time series model for a vibration measurement formed from the force measurements. The second measurement model is a model of constraint. We present the estimators, indicate performance, and indicate how they can be used for selecting the most likely model for a given batch of measurements.

Of course real data comes from a sequence of measurement models. It is therefore essential to be able to detect when a model has changed. This is the problem of segmentation or change detection. Lastly, the contact state observer is presented. The observer simultaneously segments and labels measurement sequences in order to determine an estimate of the robot's state.

The resulting complete observer can be applied to robot programming in order to: 1) mediate multiple step robot motions, 2) detect completion of a motion, 3) recover from unexpected contact events, and 4) detect model errors. The new techniques presented in this thesis are based on two insights: 1) additional, useful, manipulation sensing primitives exist in the force and position signals caused by contact other than just the raw force, and 2) sequential estimation and decision theory provides a powerful tool for detecting and isolating these primitives.

The core thesis contributions are:

1. Development of a useful model-based definition of contact features.

2. Development of a multi-outcome representation, the task feature graph, for relating features to steps in a task.

3. Development of efficient, robust, model-based, maximum likelihood (MLE) feature estimation, and segmentation techniques, for temporal and contact constraint features.

1.1 Thesis Approach

There are two unifying threads in the thesis which are used many times. The first thread comes from the ideas of statistical modeling and decision making. We model the basic problem of contact sensing in terms of determing the current measurement model for the contact force and position measurements. Because we start from a statistical representation, we are able to utilize the powerful tools that have been developed for stochastic measurement processes. Maximum likelihood estimation [Ljung, 1983] is used for extracting the feature parameters. Sequential hypothesis tests, originally developed by Wald [Wald, 1947], are used for on-line segmentation of the measurement processes into a sequence of measurement models.

The second important thread comes from dynamics in configuration space [Lozano-Pérez, 1983, Arnold, 1989]. *Configuration space* is the space of all possible generalized coordinates, or configurations, for the moving robot. Manipulation and mating are processes which are best modeled as motions in configuration space. The contact forces in manipulation are configuration space constraint reaction forces.

Although the contact feature sensing framework developed in this thesis can handle other information, this thesis focuses on the information available from measurements of force and position for a single end-effector. Position and force information is all the contact measurement information available in mating tasks. A good representation and perception of this information is essential to manipulation since mating is major component of active manipulation. These measurements provide information about the geometry of configuration space and the location of the robot in configuration space.

Therefore, three ideas come from the study of configuration space :

1. The process dynamics, and the applied command, provide a notion of connectivity in phase-space which is used to build the feature graphs.

2. Forces applied against contact constraints, represented in configuration space, produce the low frequency, quasi-static component of the measured contact forces. The form of these configuration space constraints must be understood in order to interpret the measured forces.

3. Motions along textured constraint surfaces, or changes in the forms of the constraint, produce distinct dynamic contact events. The source of these events is best represented by motions along configuration space surfaces, and discontinuous changes in the surface geometry.

1.2 Guide to Thesis

The thesis has been broken down into four broad sections. Chapters 2 and 3 provide background on the problem of manipulation programming and contact sensing. This area has been of major interest to robotics researchers for many years. Therefore, chapter 2 provides a historical review of the problem of sensor guided manipulation programming. The historical review provides a motivation for the idea of feature based programming. Feature based programming uses local controllers, which are selected by determining the active contact feature. The approach rests on ideas of feedback control and observer theory.

Chapter 3 discusses manipulation cues in general, before focusing on the cues available from internal force and position measurements. This subset of contact sensing has been called intrinsic contact sensing; and the relation of this thesis to that work is discussed here.

Chapters 4 and 5 form the bulk of the contact feature sensing theory. Chapter 4 reviews configuration space dynamics. Some basic concepts from differential geometry are presented in order to facilitate work in later chapters. This chapter essentially presents configuration space as a framework for understanding intrinsic contact information. It can be skimmed and referred to as needed.

Chapter 5 defines contact features in terms of the measurement models and then uses the definition to create several graph structures useful for sensing task progress. In particular, the feature graph is defined using forward projection. Chapter 6 then presents a contact feature observer. Maximum likelihood estimation and sequential hypothesis testing are used to develop the observer.

Chapters 7, 8, and 9 apply the general theory to develop particular results useful for rigid body manipulation. Chapter 7 considers the problem of temporal feature estimation and segmentation. This chapter shows how impacts, and changes in signal spectrum, can be detected. Chapter 8 develops a novel constraint estimation scheme which is able to incorporate all the available contact information for finite rotations in the plane. Extensions of the ideas to semi-rigid bodies are also discussed. Finally, chapter 9 uses the results of chapters 7 and 8, in order to demonstrate the general theory.

Chapter 10 concludes by summarizing the results of the thesis, and presenting ideas for future work.

Manipulation Programming

Chapter 2

A basic problem of autonomous manipulation and semi-autonomous telerobotics is to perform a basic control primitive: move object until some condition is satisfied. In fine manipulation, the primitives may be defined in terms of contact constraints and forces on the object. For example, we might like to instruct a Mars rover to probe for hard rocks using a stick-like probe. In assembly, we would like to instruct a robot to mate object A with object B. As a last example, a maintenance robot might be instructed to tighten a fastener until "it is tight". The facility with which humans perform all these tasks hides the complexity and detail involved in these simple instructions.

Each of these tasks involves the interaction of a grasped part with the environment. Therefore, the contact interaction forces cannot be directly measured. Instead, the form and state of this interaction must be inferred from sensors in the robot wrist or end-effector. In addition, often in robotic grasping the end-effector cannot directly measure grasp forces. Instead force and torque sensors are placed in the wrist or fingertips which must be used to infer the grasp force and thereby the current grasp state. Therefore, an understanding of perception which helps to track the progress of mating tasks has wide applicability in robotic manipulation.

Manipulation tasks like these are termed fine manipulation problems because the motions involved are usually small. A fine manipulation task is usually accomplished by a sequence of steps. Generally each step reduces the relative positioning error between the manipulated parts by incrementally adding contact constraints. Constraints are added incrementally because the approach has a much higher probability of succeeding than a strategy which attempts to go from no constraint to full constraint in one step. For example, one strategy for peg-in-hole assembly is to bring the the peg into contact with the top surface. Then a new motion is used to place the tip of the peg in the hole. Then the peg is approximately aligned with the hole. Finally, an insertion step is used to push the peg into the hole.

Each step is usually a compliant motion[1] and the sequence of motions is indexed by some termination predicate. The sequence of steps may depend upon the effects of the interaction. Since there can be multiple outcomes in any given interaction branching and context based decision making is always necessary.

Multi-step strategies also occur in grasping. As an example consider the experiment of grasping, lifting, and replacing an object of unknown mass given by [Howe *et al.*, 1990]. An idealized version of this task is shown in figure 2.1. In their paper, the sensors measured fingertip forces, accelerations, and relative slip while a two finger grasper was used to grasp, lift, and replace a single test object. Six distinct control steps, or phases, were determined for this task: pre-contact, loading, manipulation, unloading, post-contact, and slip. Each phase required a different control algorithm, and termination of each phase was signaled by different signal predicates.

The idea is to connect a set of continuous control algorithms into a discrete network. Transitions in the network are controlled by a decision process that uses the measurements and possibly the controls. If the decision process only sets thresholds on the current values of the measurements, the resulting controller is a mixture of continuous controllers embedded within a nondeterministic finite state machine. More general structures are also possible.

This idea of event driven manipulation has been evolving under different names at many research centers. Nevins et al [Nevins *et al.*, 1973] presents one of the earliest event driven frameworks for programming a robot for assembly. Brooks [Brooks, 1985, Brooks, 1987] argues for an implementation called the subsumption architecture. Howe and Cutkosky use event transitions, which they term contact events, to drive their grasping algorithm. Another work [McCarragher and Asada, 1993a] focused on the transitions to drive an assembly algorithm.

This mixed approach of continuous control within discrete domains or contexts is very appealing. It breaks the problem down into components which can then be solved in isolation. By appropriately combining solutions, more complex behaviors

[1]Compliance control or force control has a long history in robotics. [Whitney, 1985] provides a historical perspective of force control techniques which have since become standard and can be found in textbooks such as [Craig, 1989].

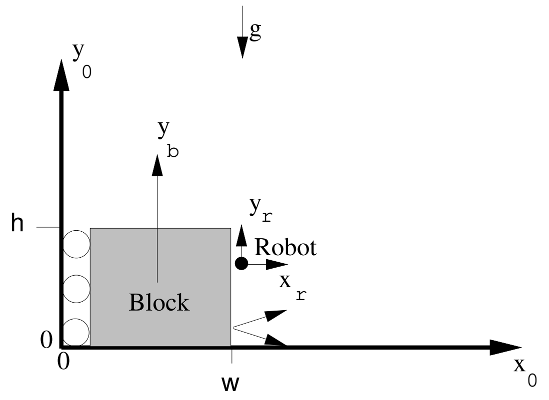


Figure 2.1: An idealized version of the lifting problem treated by Howe and Cutkosky. The rollers on the block are frictionless. The weight and contact friction coefficient of the block are in general unknown.

can be synthesized. However, the vast array of possible interactions between the robot and the part and problems that can arise in the interaction is currently the basic obstacle to this approach.

Although many single tasks have been programmed by hand for various robots, the innumerable number of potential problems makes direct programming of a large number of solutions impractical. Decomposing each task into components and selecting robot-dependent thresholds is a difficult problem for even one task. Instead, a method of programming which gives the robot competence in a large class of fine manipulation tasks must be found. Possible solutions might use more general representations, involve learning, or be based on some model-based planning. Uncertainty is what makes this problem difficult. There are several sources of uncertainty in manipulation:

1. There can be error in both the geometric and the topologic models of objects, especially if the model was extracted from vision or touch.

2. The location of objects in the environment and in the robot's gripper is not exactly known.

3. The parameters describing the physics of interaction, including the coefficient of friction, may be only partially known and may not even be directly modeled.

4. The control is subject to random disturbances.

5. The effect of the control on the motion of objects may be uncertain or unknown.

6. Measurements of the position and interaction forces for the moving objects suffer from measurement noise and unmodeled effects.

This thesis looked at two problems that arise within manipulation programming. First, how should the notion of a *contact event* be defined so that the discrete control networks can be related to task geometry and physics. Second, given an event definition how can the discrete state of the robot in the network be estimated or observered. An observation of the discrete state is necessary in order to select the appropriate control algorithm.

We will define *events* as statistically significant deviations from model predictions. Our approach is model-based. For every contact situation a statistical model of the measurement process is defined. We term these models *contact features* or just *features*. A model-based approach has several advantages. First, uncertainty is explicitly incorporated. The event detectors and observation process can be derived directly from the models. Second, models allow some parameterized generalization. Third, the representation is amenable to analysis and Monte Carlo simulation in order to determine performance.

Our programming approach is to create a network of measurement models. An observer is developed to track the state of the robot in this network. The network can be hand-programmed, possibly learned, or computed from models and first principles. Finally, although this is not developed here, it is possible that contexts can be defined as collections of nodes in this network and that actions and controllers can be created relative to these contexts. The complete approach provides a framework for programming robots to solve unstructured tasks.

Before developing our approach, some preliminaries and background are provided. This chapter provides a historical review of treatments of the manipulation programming problem. This provides some context for our approach. Chapter 3 discusses possible manipulation cues and past work on feature sensing. Finally, chapter 4 gives mathematical background on dynamics and configuration space. This material is used to justify the network representation, the feature models, and to develop tools for constraint estimation.

2.1 History of Fine Manipulation Programming

The challenge of automatic robot programming for fine manipulation has produced a rich history of research in robot design, force control, sensing, and robot planning. This section presents some of the more recent approaches to the problem of fine motion programming with uncertainty and shows how this thesis relates to that work.

2.1.1 Guarded Moves

Guarded moves [Bolles and Paul, 1973] is the earliest approach to fine motion programming. A guarded move is a motion which terminates when a force or position limit of a certain value is crossed. A branching control strategy can be written by combining primitive motions with guarded move termination tests. The combination of branching, passive compliance in the form of free uncontrolled joints, and early vision incorporated a range of sensing modalities and implicitly anticipated much future work in robotics. A similar approach along with the definition of guarded moves can be found in [Will and Grossman, 1975].

Bolles and Paul presented an assembly demonstration for a model T Ford water pump. The water pump base and top were to be assembled with 6 screws and a joining gasket. The assembly cell was strongly structured in order to control the consequences of uncertainty. Fixtures and alignment blocks were designed to control placement errors. Parts were painted white and the background was painted black for high contrast vision. Even with all this branching control strategies were still needed to deal with the remaining errors.

One step in the assembly, the insertion of an alignment pin, shows how guarded moves can be used to handle and test for errors. Two alignment pins, with a cone insertion end, were to be inserted into two of the screw holes in the pump base. Because of fixturing, positional errors in the plane are the only errors that can occur. Axis angular misalignment is prevented by the fixtures.

A spiral search strategy is used to compensate for the possible positional error. An analysis of the geometry shows that any one attempted insertion can result in three possibilities: 1) the pin can go in the hole, 2) the pin can miss the hole and land on the top of the base beside the hole, 3) the pin can miss the base. A two part guarded move strategy can be used to discriminate between these cases:

1. Move down sufficiently to guarantee an insertion will occur. If a large upward force is encountered, case 2 must have occurred so continue searching with a spiral of ever increasing radius. If the large force is not encountered go to step 2.

2. Try and seat the pin in the hole. If resistance (due to friction) is not felt the pin must have missed the base so continue the spiral search. If resistance is encountered the pin is seated.

The uses and limitations of guarded moves can be seen in this example. First, designing such simple tests for a more complex situation requires analyzing every possible contact possibility (and possibly sequence of possibilities) and determining a scalar test which can discriminate the cases. In this example there are only two degrees-of-freedom that can have errors (because of fixturing). The number of possible conditions increases rapidly with the dimension of the underlying space. As part of his thesis [Buckley, 1987] determined the number of force or position distinguishable contacts for a three dimensional square peg in a square hole. This study showed that there are 1714 possible contact pairings that could occur. Case by case analysis of the possible situations is impossible without computer aid. Futhermore, it is very unlikely that a single test using only a few force measurements will be sufficient to discriminate the cases.

Second, each test is a very simple threshold on the force. Force is a very noisy signal and situations can easily occur which will fool the test. For example, the pin could hit the base on the edge of the cone in the second step. The pin would have gone in part way in the first step, then collision with the cone would cause acceptance on the second step. A more complex test that probes the degree and form of constraint on the pin would not be confused by such an example.

Lastly, errors in axis alignment could cause problems with the decisions based on force thresholds and would cause problems with control. Axial misalignment (prevented in this problem through fixturing) can cause large forces. The remote center compliance (RCC) and compliance, or impedance, specification and control was developed to deal with small misalignments implicitly using appropriate control.

2.1.2 Compliance Specification

Bolles and Paul anticipate compliance control in their idea of a free joint which is aligned with the direction of possible errors. This idea was generalized in the work on generalized damper control by [Nevins *et al.*, 1973, Whitney, 1977, Whitney, 1982], stiffness control [Salisbury, 1980], impedance control [Hogan, 1985] and hybrid control [Craig and Raibert, 1986]. The relationship of compliance specifications to geometric constraint was formalized in [Mason, 1981]. A vast amount of additional literature has been presented on compliance and force control.

Compliances are specified relative to a constraint frame. Forces and torques are measured and motions are produced relative to this constraint frame. As a strategy, compliance control has the advantage that certain types of errors are accommodated automatically. For example, angular or translation misalignments can be accommodated in pin insertion while still controlling the displacement into the hole. Therefore it is not necessary to sense errors and switch the control law for a range of errors.

The complete manipulation will still require branching and a sensor based control strategy. The pin insertion task, described above, would still need to use a sequence of guarded moves to begin to put the peg into the hole. However once the robot is certain the peg is in the hole and beyond the jamming depth, a simple compliant insertion will suffice to complete the task.

2.1.3 Strategy Skeletons

The first idea for removing some of the burden from the programmer was to write a set of strategy skeletons [Lozano-Pérez, 1976, Taylor, 1976] which would be available as robot programming primitives. There would be a skeleton for maintaining grasp, a skeleton for inserting two parts, a skeleton for opening a door, et cetera. Each skeleton would have a set of free parameters that would be set when a specific problem was to be performed. Skeletons could be written by hand, and then a computer program would be used to determine the appropriate parameter values for a particular example of a task.

Taylor wrote such a program. The program used numerical evaluation of part locations and errors to propagate the effects of errors and uncertainties through the task model. The results were used to fill in the parameters in the skeleton. [Brooks, 1982] extended the approach to use symbolic propagation. By back-propagating the constraints from the goal, constraints on the skeleton parameters could be deduced.

The skeleton approach rested on the assumption that any one skeleton would provide an approach for many examples of a task such as peg-in-hole, close door ect. Each skeleton was intended to provide for a range of examples of a task. A few skeletons would then suffice to make the robot capable in a given task. The problem is that small changes in geometry can necessitate large changes in strategy [Lozano-Pérez *et al.*, 1984]. Therefore, a single task such as peg-in-hole insertion may require a many skeletons. Instead, a technique for directly incorporating geometry and a model of the errors is needed for task programming.

2.1.4 Fine Motion Planning

Preimage fine motion planning deals with error and the effects of geometry by planning over models of the error and geometry [Lozano-Pérez *et al.*, 1984]. In this formal approach, robot programming is treated as an automatic program synthesis problem given knowledge of geometry, physics, and sensing. The programmer writes a single program called a planner. Then the planner examines the problem specification in terms of the geometry, physics, and possible sensory operations and then computes a robot control program which is guaranteed to work. This program is then run on the robot. The robot program takes in sensory measurements and outputs appropriate controls in order to accomplish the task. Once this single program is written, the robot programming problem is solved for all tasks which the planner understands. This approach is important because it provides a formal problem statement and approach for fine manipulation programming.

This approach directly incorporates uncertainty. It showed how both contact constraints and sensing could be used to reduce uncertainty. This formulation of the control problem is very close to stochastic dynamic programming [Bertsekas, 1976] and shares the problem of dimensionality. One approach to solving the problem is to discretize the task state space and then consider sequences of motions and sensing operations in the discretized state space. Planning must be done over the possible sequences of sensing and controls or the knowledge space.

It will be easier to understand the approach, the complexity of the approach, and contrast the approach with other techniques, if we first loosely define some terms. This presentation will use a discrete representation of time. The *state* of the robot and all the objects in the environment is specified by a vector of variables $\mathbf{x}$. In quasi-static manipulation the state is just the position of the objects, in dynamics the velocities are also required. The set of all states is denoted by $\mathcal{X}$.

The path planning problem, without uncertainty, is the problem of constructing a trajectory, or path, through the state space which does not violate any constraints. One approach to solving this problem is backchaining from the goal, over single step motions.

The *goal* $\mathcal{G}$ is a subset of $\mathcal{X}$. The dynamics of the control law, without uncertainty, induces a *forward projection* operation $\mathbf{x}_{j+1} = \mathcal{F}(x_j, \mathbf{u}_j)$ which takes the current state $\mathbf{x}_j$ and maps it to the next state $\mathbf{x}_{j+1}$ given the control applied at time j, $\mathbf{u}_j$. The *control* $\mathbf{u}_j$ ranges over a set $\mathcal{U}(\mathbf{x}_j)$. For example, the control might be an increment in $\mathbf{x}$ that satisfies the constraints.

The *backprojection*, without uncertainty, can now be defined. Let $\mathcal{G}_j$ be the current goal set. The backprojection of $\mathcal{G}_j$ is the set of all points $\mathbf{x}$ in $\mathcal{X}$ such that there is a control in $\mathcal{U}(\mathbf{x})$ which when applied to $\mathbf{x}$ will result in an $\mathbf{x}_j$ in the goal set. Formally

$$\mathcal{B}(\mathcal{G}_j) = \{\mathbf{x} \in \mathcal{X} : \exists \mathbf{u} \in \mathcal{U}(\mathbf{x}) \,\text{and}\, \mathcal{F}(x, u) \in \mathcal{G}_j\}.$$

Backchaining is now defined by the recursion $\mathcal{G}_{j-1} = \mathcal{B}(\mathcal{G}_j)$ with $\mathcal{G}_0$ the initial goal set.

How complex is this? Let card denote the cardinality of a set. At every stage of the backchaining operation, the computer searches over the entire state space (in general) and the entire control space in order to determine if there is a control that will reach the current goal set. This step is essentially computing controls that connect any two states. If this connectivity is memorized, a k step path can be computed in order $O(k \,\mathrm{card}(\mathcal{U})\, \mathrm{card}(\mathcal{X})^2)$ [Erdmann, 1989]. In particular, the problem has polynomial complexity in the size of the state space. Of course, the size of the state space is typically exponential in the degrees-of-freedom, or dimension, of the state space.

Things become more complicated when there is uncertainty. But, there is a classic information reduction which places the problem with uncertainty into the path planning problem presented above [Bertsekas, 1976]. With uncertainty everything becomes set valued. The *information vector* at stage j is the set of all observed measurements and applied controls up to time j

$$\mathcal{I}_j = \{\mathbf{y}_1^j, \mathbf{u}_1^j\}$$

where the notation $\mathbf{y}_1^j$ represents the first through j^{th} measurement, and similarly for the control. This vector represents all the measurement and control information available to the computer up to time j. In addition, there is prior information.

The *interpretation set* is the set of all states that are consistent with the information vector and the prior information. There is a recursive mapping between the information vector and the interpretation set. Let $\mathcal{K}_0$ be a subset of $\mathcal{X}$. This set represents th prior information about the initial state. A function $\mathcal{M}(\mathbf{y})$, the *measurement interpretation function*, returns the set of all states that are consistent with the measurement $\mathbf{y}$. The forward projection operator now acts on sets. The operator takes an input set in $\mathcal{X}$ and returns the set of possible outcomes given the uncertainty in $\mathbf{u}(\mathbf{x})$. The recursion for the interpretation set is then defined by

$$\mathcal{K}_j = \mathcal{F}(\mathcal{K}_{j-1}, \mathbf{u}(\mathcal{K}_{j-1})) \bigcap \mathcal{M}(\mathbf{y}_j).$$

Now the backchaining solution to the planning problem proceeds over the interpretation sets instead of $\mathcal{X}$. In this model of the problem, the computational problem is order $O(k \operatorname{card}(\mathcal{U}) 2^{\operatorname{card}(\mathcal{X})^2})$. The difficulty is exponential in the cardinality of the state.

The output of the planner is a set of *subgoal* regions $\mathcal{G}_i$ and an associated action such that, given knowledge of the current subgoal, the robot can execute the sequence of actions and be guaranteed success. A subgoal is said to be recognizable if upon execution the sequence of actions and measurements which place the robot in $\mathcal{G}_i$ will generate an interpretation set which is contained in $\mathcal{G}_i$. A recognizable subgoal is called a preimage.

Heuristically, the difficulty of the full problem can be seen in the indexing of the interpretation sets. The interpretation sets run forward in time, whereas the planning solution must run backward in time. This means, in the worst case, in order to determine what to do on the next step, we have to know what we did for all previous steps and what we might measure at every step.

For even the simplest of realistic problems, the complexity of the procedure is too great. In order to make the problem tractable, the requirements, the assumptions, the basis for the state space, or the search algorithm must be changed. This has produced several areas of work: sensorless manipulation, probabilistic manipulation, planning with feedback, reactive behaviors, and landmark based planning. Finally after discussing these approaches we discuss a new approach, Local Control about a Nominal Plan (LCNP), [Narasimhan, 1994], to which this thesis directly relates.

2.1.5 Sensorless Manipulation

The constraints imposed by contact can be used to decrease uncertainty. For example, polygonal parts can be oriented by pushing on them with a stick. An early proponent of this approach was Mason in [Mason and Salisbury, 1985]. Only certain orientations will be stable, and the rest will be filtered out by the act of pushing. This is an example of *sensorless manipulation*. Sensorless manipulation relies on the task physics to accomplish the task without any sensing except for a clock. Although the direct approach to sensorless manipulation planning is still intractable, more powerful representations have made some problems tractable.

One technique for simplifying the problem, which is also important in sensor based manipulation, is to compute the effect of an action and show that it has only a finite number of outcomes. The problem is further simplified by showing that the continuous set of actions fall into a finite number of action classes based on which of the finite outcomes they allow. For example, a tray tilting planner was devised in [Erdmann and Mason, 1988] based on the observation that a polygonal object could only come to rest in a finite number of orientations when it fell against a wall. An action sequence can then be derived by considering the action to be a sequence of filtering operations each step of which removes some of the unwanted possible orientations.

This idea is very important for parts orienting and fixturing in manufacturing using vibratory bowl feeders or other similar systems. A good review of this work, and applications of using the constraints of shape to filter out and design the required function can be found in [Caine, 1993].

This approach can be placed in the preimage framework. Since the manipulations are sensorless, the information vector is just $\mathcal{I}_j = \{\mathbf{u}_1^j\}$ and the interpretation update is

$$\mathcal{K}_j = \mathcal{F}(\mathcal{K}_{j-1}, \mathbf{u}(\mathcal{K}_{j-1})).$$

Planning still proceeds over the power set of the state space. However, in this case since only a small number of states are stable, the dimensionality of the problem is tractable.

A close dual to the sensorless manipulation problem can be found in [Brock, 1993]. Brock considers the problem of determining the configuration of an object through a sequence of sensing operations. The goal is to decrease the interpretation set until it is small enough to fit within a given goal set.

Within the preimage framework, he considered the class of manipulation strategies where there is very little uncertainty in the robot configuration and control relative to the uncertainty in the location of the object in the environment. In this case, it is possible to plan a sequence of motions which are guaranteed to reduce the object location uncertainty with every step.

The state space is the combination of the robot's configuration and the object configuration. Since the error in locating the robot's configuration is very small, the analysis focuses purely on the interpretation set for the object. Finally, since the object is assumed to be fixed, the forward projection operation for the object is the identity operation. Therefore, in the preimage framework the interpretation set update after every measurement is

$$\mathcal{K}_j = \mathcal{K}_{j-1} \bigcap \mathcal{M}(\mathbf{y}_j)$$

where $\mathcal{M}(\mathbf{y}_j)$ are the possible configurations of the object consistent with the new measurement.

Every update step causes the interpretation set to either remain the same or decrease, since the update step is a sequence of intersection operations. The interpretation set can be made to strictly decrease by always moving the robot into the previous interpretation set. This will eventually shrink the interpretation set until it is sufficiently small to lie in the goal set expressed relative to the object configuration. A probabilistic viewpoint on this idea is given in [Hager, 1992].

2.1.6 Probabilistic or Randomized Strategies

Probabilistic or randomized strategies can help the planning and manipulation problem by: 1) removing the requirement that every step in the plan be guaranteed to be recognizable, and 2) preventing infinite control looping caused by model or control error [Erdmann, 1989].

At the simplest level, a randomized strategy is just a random walk on the state space of the problem. If the state space is closed, bounded, and free of trap states, eventually, although slowly, the random walk will come within epsilon of every point in the space, including the desired goal. By biasing the search in the desired direction, using a measure of progress, convergence times can be significantly decreased.

An everyday example of biased randomization is inserting a key in a tight lock. A constant force in approximately the right direction is applied to the key and then the key is jiggled about rapidly to randomly break constraints and make progress against the friction.

Biased randomization can be seen as an example of randomization with simple feedback. These are strategies that only consider the current sensor value in deciding the next action to execute. The progress measure can be used to guide the selection of the next action. The resulting set of motions can be analyzed in terms of the expected progress to the goal. If, on average, the feedback controller causes motion towards the goal, the goal will be achieved asymptotically with probability 1, and the expected time to completion will be bounded. However, it can be very difficult to show that on average the motion does result in progress.

Another advantage of randomization is that it tends to even out incorrect assumptions about the environment. If a fixed plan fails in a certain configuration because of modeling errors than it will always fail in that configuration. However, with randomization a plan may succeed at this configuration by chance.

2.1.7 Changing the State Space and Landmark Approaches

An approach to simplifying the problem, which has appeared in many forms, is to change the representation of the state space. This representation should have two properties: 1) the representation should reduce the infinite, continuous, state space to a finite representation, and 2) the sensors should be able to separate the elements of the representation at least probabilistically.

Primitive contact pairings between objects in the environment were defined as atoms in [Buckley, 1987]. This same representation was used in [McCarragher and Asada, 1993a]. In order to incorporate the notion of sensor separation, Buckley defined two atoms as being disambiguous if the sensors (force and position) could distinguish the two atoms with a single reading from the sensors. Information states can then be defined as collections of atoms which are not disambiguous. These collections were termed *distinguished* sets.

In the preimage framework, the state space is defined as the union of a disjoint collection of sets $\mathcal{X} = \bigcup_i \{\mathcal{A}_i\}$. The information space is then the power set of $\mathcal{A} = \{\mathcal{A}_i\}$. Forward projection, backward projection, and preimages can then be defined relative to $\mathcal{A}$.

Unfortunately, this notion of distinguished sets is too strong. In Cartesian configuration spaces, force cones do provide a notion of distinguished sets. However, in problems with rotations the distinguished sets can be very large collections of the atoms. [Robles, 1995] discusses this point; this thesis will revisit this problem in chapter 5 when features are formally defined.

A second difficulty is that atoms can be too large. Each atom represents some region of configuration space. In Buckley's approach the forward projection operation is applied to the entire atom to see if the next desired atom can be reached. Buckley shows that it is easy to construct examples where only a subset of the first atom can achieve the next goal. Both difficulties can be partially circumvented by limiting the state space by supplying a nominal path.

Another approach to creating a finite set on which to reason is landmark based planning as presented in [Lazanas and Latombe, 1992]. This paper shows that the original motion planning problem of finding a guaranteed plan despite uncertainty can be solved in polynomial time if state space is inhabited by regions called landmarks and the goal in one of the landmarks. Within each landmark the robot is able to navigate and know exactly what its state is. Between landmarks the robot moves with uncertain control and no sensing.

2.1.8 Behavior Approaches

A behavior based approach to robot task control evolved as an alternative to planning to try and circumvent the problems of computational complexity. Behaviors are a collection of feedback loops connecting sensor measurements to actions [Brooks, 1987, Brooks, 1991].

Behavior based approaches are important because they change the framework of the planning and design problem. Whereas before the planning problem was considered to be the determination of a set of command sequences, which might be motion primitives, for every sensor measurement sequence and action sequence, the behavior paradigm is to determine a set of feedback loop primitives that will be applied in different contexts. For example, in the peg-in-hole problem, the planner would look for a sequence of velocity commands to apply to a single admittance controller. The behavior approach might use a set of admittance controllers and surface following controllers to achieve the task. The idea of increasing the power of the admittance controller appears in [Schimmels and Peshkin, 1993]. The behavior approach attempts to exploit the error reducing property of feedback to make the general problem more tractable.

Although the approach is promising, there are two basic problems that remain to be solved, and which must be solved in a domain specific way. The first is the development of the feedback controllers. One approach is to develop feedback controllers which achieve nominal progress in the absence of sensor uncertainty. These can be developed through learning over simulated data, or directly from models of the task physics.

Second, once a set of controllers is developed, the domain of each controller must be determined. For feedback controllers, without information state, the domain is the region of state space over which the controller should be applied. In the subsumption architecture, the domain is arbitrated through behavior suppression. [Mataric, 1994] used reinforcement learning to determine the suppression rules given a set of real sensors. This set of rules attempts to approximate the optimal regions of applicability given the limitations of the sensors. [Erdmann, 1993] showed that backchaining planning, assuming perfect control, implicitly defines the regions of applicability for the controllers. He then discussed the problem of designing a sensor which can, with a single measurement, return the current region and an appropriate sensed value.

2.2 Feature Based Programming

This historical review suggests the following about a complete solution to the manipulation programming problem:

- global knowledge of the geometry is required;
- randomization should be used and expected;
- feedback is necessary;
- the feedback law should depend upon the local context; and
- the local context should come from a finite graph representation of the state space, where the representation is derived from the sensors capabilities.

An approach described in [Narasimhan, 1994] synthesizes these ideas into an approach which uses a set of indexed local controllers to control the motion along a nominal plan. He terms this idea Local Control about a Nominal Plan (LCNP). The idea is to plan a path through the state space which will accomplish the goal as if there were no uncertainty. This problem has polynomial complexity in the size of the state space. Then a set of feedback controllers is learned, from simulated examples, which keeps the robot on the desired path. As in classical feedback control, the controller computes a command to the actuators as if its sensor inputs were exact. This approach is sometimes called *certainty equivalent control*.

A controller is learned for each local context by inverting the experimentally learned input-output relationship between small motions of the robot and motions of the part. For the task of pushing, the local context is a discrete representation of the local configuration space surface. In operation, each sensor value returns the local context and an estimate of the robot's location. The local context is then used to determine the appropriate feedback controller, and the robot's location and the desired path are used to determine the next control action. The result is a diffusion process which on average makes progress to the goal.

The path can also be iteratively modified to increase the expected performance of the plan. Modification can be seen as a form of the expectation modification algorithm (EM). After the initial seed path is generated, assuming no uncertainty, the algorithm iteratively simulates possible outcomes and modifies the path in order to determine a more robust strategy.

A similar idea is presented in [Robles, 1995], although here the focus is assembly. In this thesis, an action map for states that might be visited given the nominal path is determined using backchain planning without uncertainty. Upon execution the controller iteratively computes the interpretation set and selects an action from the states in the current interpretation set. The interpretation set is represented as a list of bounding polyhedra in the configuration space. The select operator may or may not involve randomness.

When the decisions about the current contact state are made on the basis of local signal models, the approach can be termed feature based programming. A feature is a model of the measurement process that applies to a local patch of state space. By determining which model best explains the measurements, the robot controller can determine approximately where it is in state space. If the features select the controller, then determining the current feature is sufficient for determining the form of the local controller for each patch of state-space.

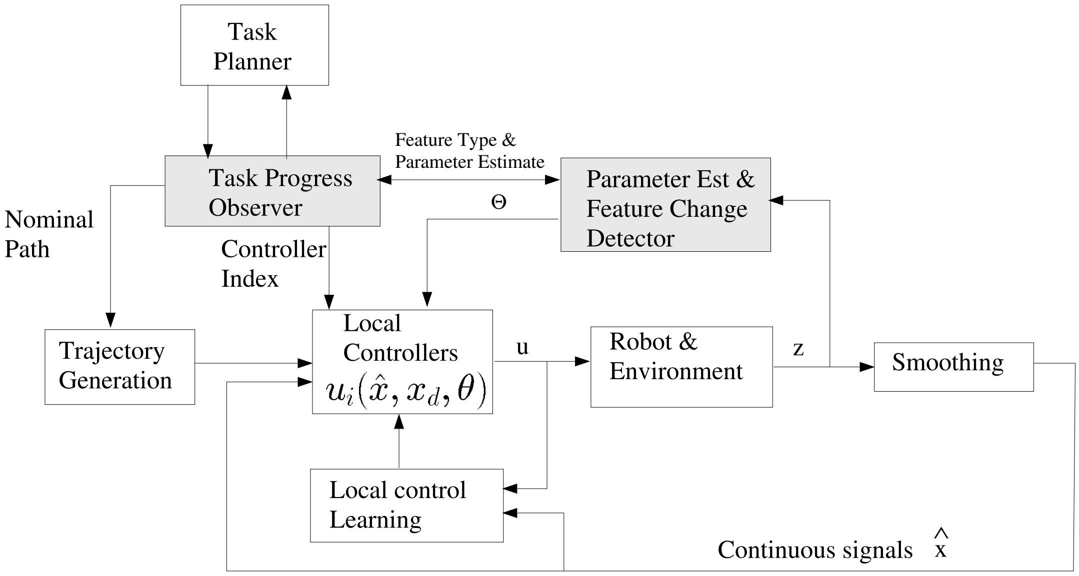


Figure 2.2: The components of the planning and control system for the feature approach to robot control. This thesis considers the shaded components.

There are three components to the feature approach: 1) the nominal plan, 2) the local controllers, and 3) the feature observer. As in the LCNP approach, the nominal plan is constructed assuming no uncertainty. The local controllers can be learned from real or simulated examples. This thesis focuses on the feature observer component of this approach. The observer is designed to determine the probabilities of each feature model. The probability distribution over the features is the context which can be used for selecting the local controller. The components of the planning and control system for the feature approach are illustrated in figure 2.2.

In this thesis, a statistical modeling approach is adopted for modeling the force and position features. This eliminates the need for setting thresholds for different problems. Instead the system automatically calibrates itself to the noise conditions and detects changes which are statistically relevant. Second, the statistical viewpoint allows the derivation of different levels of feature sensing within a single decision making framework. Lower levels can be used independently for simpler robot manipulation problems. Furthermore, these feature detectors isolate the raw force and velocity signal from the geometric part of the observer. Only changes in feature vectors need to be processed by the geometric part of the observer in order to determine the new contact state. This significantly reduces the computational load.

The problem of building an observer based on force measurements was also considered in [McCarragher and Asada, 1993a]. They used a Petri net to model the different contact conditions and possible transitions for known geometry. Transitions were detected using transient contact force features. The approach did not check the consistency of the forces with the current contact state. Models for the effect of Petri net transitions on the force were derived through qualitative reasoning on the dynamics. Using this technique, they were able to insert planar pegs into a planar hole.

Before focusing on the features used in this thesis, it is useful to consider the range of features that are practically observable in manipulation. The next section first considers a possible framework for the types of cues available to humans. Then, the subset of cues available to a robot which is only able to sense position, force, and torque on the contacting link is discussed. These are the only measurements available in the mating phase of manipulation. Therefore, an increased understanding of just these signals would significantly enhance a basic component of manipulation. Finally, previous work in this area of sensing called intrinsic tactile sensing is presented to relate the features used in the thesis to prior work.

Contact Manipulation Cues

Chapter 3

Manipulation sensing is a complex task requiring the interpretation of many different types of measurements. It is useful to consider the cues used in human manipulation and then to consider what subset of these cues is available to a robot which is equipped with only a force/torque and position sensor. These two sensors provide the measurement equivalent of tool based manipulation, with which humans have great perceptual facility. It is also the only source of contact measurement information available in mating tasks. A better representation and understanding of this information should make it possible to give robots the same level of perceptual capability.

This chapter provides a general overview of human manipulation cues, and then discusses the cues available in mating tasks. The processing of contact information without direct contact sensing is termed *intrinsic contact sensing*. The relationship of previous work in this area to this thesis is discussed. Finally, the hardware used in this thesis is presented.

3.1 Human Manipulation Cues

In general, touch tasks can be broadly divided into exploration tasks and manipulation tasks [Srinivasan, 1991]. Exploration tasks are sensor dominant, which means that they depend primarily on sensory inputs for successful completion. These tasks typically involve discrimination or identification of surface properties (such as shape and surface texture) and volumetric properties (such as mass and compliance) of an object. Manipulation tasks are motor dominant, depending primarily on voluntary motor activity to modify the environment, although sensory feedback is essential for successful performance. In a motor dominant task, sensory feedback provides cues about the state of the contact configuration both between the hand and the grasped object, and between the object and the rest of the world. Manipulation tasks can be further subdivided into precision or dexterous tasks and power tasks. Dexterous tasks are performed primarily with the fingertips, while power tasks are performed with the entire hand.

Exploratory tasks involve so called identification cues which can be passively or actively acquired. Passively acquired identification cues can be obtained by the act of holding the object in the hand. Three major cues in this category are:

- *Determining the local contact normal and curvature of the surface at each contact.* Determining the local contact curvature and labeling it as a point, edge, or planar contact is important for grasp acquisition and object identification. Different contact types have different grasp properties which affects stability [Salisbury, 1982]. In addition, the local contact normal and curvature provide a strong pruning heuristic rule for identifying objects and object pose [Grimson, 1990]. Local object curvature can be measured passively by examining the normal contact force distribution.
- *Determining the surface texture at each contact.* Surface texture affects grasp stability. Rough textures are generally easier to grasp then smooth textures. In addition, surface texture can also be used as a pruning heuristic in identifying objects and object pose. Texture cues are produced both from the spatial distribution of the contact force and the dynamic vibration effects produced at the skin during motion.
- *Determining the gross object shape.* By using the finger geometry and the location of the contact points in the hand, the shape and pose of known objects can be estimated [Siegel, 1991]. For example, an initial grasp might be a power grasp in order to get a lot of contact information about an object. Once the object and its shape is identified, a dexterous grasp might be used for manipulation.

With some local active exploration, the following properties relating the reaction forces to the applied force can also be determined:

- *The local stiffness at each contact.* The local surface stiffness can be estimated by applying varying normal forces and measuring the change in contact area. Softer surfaces will produce a larger contact area for a given applied force. The local stiffness of an object can also be estimated by tapping with a probe and looking at the initial component of the resulting impact signature.

- *The local frictional properties.* The local friction along with the local contact type strongly effects grasp stability. Friction can be measured by applying varying tangential forces and then detecting the onset of slip.

In addition to identification cues there are manipulation cues. Manipulation cues are interpretations of contact events that occur only during manipulation. The cues require relative motion between the hand and the object, or between a grasped object and the environment. Some of the basic cues are:

- *Detecting slip.* Detecting the onset of slip between an object and the hand is essential for grasp maintenance. Slip detection is used to determine the necessary grasp forces at each contact during all stages of manipulation.

- Determining object mass, center of mass, and moments of inertia.
  By manipulating a grasped object and measuring the resulting joint torques and net contact forces at the hands for different configurations, the mass, center of mass, and moments of inertia of the object can be computed. This information can be used for object identification and pose determination [Siegel, 1991], as well as for computing the necessary torques for throwing or manipulating the object.

- *Estimating directions and type of contact constraint.* Assembly is the process of bringing a moving part into a constrained relationship with a fixed part. In order to control the assembly process, the directions in which movement is constrained need to be estimated. Contact constraints are estimated using measurements of the reaction forces and allowed velocities as a function of position, and by measuring the direction of impact forces.

- *Detecting changes in contact constraints.* This is one of the most common cues during manipulation. The detents in switches, the termination in screws, the impacts from mating two parts are all examples. The onset of the change can be detected by looking for impact forces. The direction of the impact force provides some information about the new constraint.

3.2 Intrinsic Contact Cues

In mating or contact sensing through a tool the robot is only able to sense the generalized position and force on the contacting part or link.[1] This is only a small subset of the cues available in manipulation sensing.

The robot may or may not know the pose (relative to itself) and geometry of the contacted part. If the geometry and pose are known, the type of sensing is termed intrinsic contact sensing. For example, this is the case when a force sensor is attached with a known configuration and geometry to a robot. Knowledge of the geometry and pose of the part makes it possible to determine the position of the contact on the part under the assumption of a single point region [Salisbury, 1984]. With a convex sensor, and the assumption of a single point contact, the local contact normal and the object shape can be constructed through probing.

Without knowledge of the geometry and/or the pose, only the temporal and cross-correlation structure of the measurements is available to the robot. The robot needs additional prior knowledge in order to relate this abstract measurement information to more general manipulation cues. For example, if the robot knows that a temporal signal is the result of stroking a texture, then different textures can be sorted based on temporal effects. Similarly, the additional vibration induced by slip can be used as an indicator of slip, if the robot knows that slip is the only possible source of additional vibration. Techniques for both of these measurements are considered in chapter 7.

Constraint and changes in constraint can be directly measured by the robot. Techniques for estimating the constraints for certain classes of environments are considered in chapter 8. Determining the object mass, center of mass, and moments of inertia is a direct application of estimation techniques in robot calibration and adaptive control. Not all of these terms are directly observable, but certain combinations can be determined using recursive estimation techniques [Slotine and Li, 1987].

[1]The generalized position is the set of parameters need to specify the position and orientation of an object. This is also termed the object's configuration. The generalized force is the force or torque on each of the generalized directions.

3.3 Previous Work on Intrinsic Contact Sensing

During the last decade, considerable research has been performed on tactile sensing. [Howe and Cutkosky, 1991] provides a recent comprehensive review of current and past research. Most of this research has focused on designing surface array sensors and using these sensors for obtaining geometric information from static measurements. Some research has looked at the information that can be acquired by actively moving the contact sensor and monitoring both the sensor and joint locations. This is termed haptic sensing. Prior haptic research has primarily focused on actively tracing the contours of objects to determine geometry and critical features [Brock and Chiu, 1985, Stansfield, 1987]. This work assumes that each measurement is taken with the force sensor in a quasi-static state so that normal forces and contact locations can be computed. All of this work essentially treats sensing with a tactile array sensor as a primitive form of vision.

In contrast, only recently have investigations examined contact information that is characteristic of the physics of motion [Howe and Cutkosky, 1991]. Mechanical properties of objects like mass, friction, and damping can only be determined by actively probing and manipulating the object. Similarly, detecting the initial contact with an object, and slip between the sensors and environment require sensing the effects of relative motion.

A few studies have been done on this type of sensing. By monitoring the acoustic emission from a metal gripper, [Dornfeld and Handy, 1987] detected the onset of slip for some metallic workpieces. [Howe and Cutkosky, 1989] constructed an instrumented latex covered finger. Piezoelectric sensors are embedded in the latex cover and a miniature accelerometer is mounted on the inside surface of the cover. The piezoelectric sensors are very sensitive to strain rate. Because of the small mass of the cover, the accelerometers are sensitive to very small forces normal to the surface of the sensor. They found that the piezoelectric sensor was very sensitive to the changes in tangential strain associated with slip, and that the accelerometer was fairly sensitive to the vibrations normal to the sensor associated with slip.

Determining the contact location, given the sensor's geometry and the assumption of a single contact has received considerable attention since the original paper [Salisbury, 1984]. Additional extensions have appeared in [Bicchi, 1990, Bicchi *et al.*, 1990, Eberman and Salisbury, 1989, Bicchi *et al.*, 1989, Bicchi *et al.*, 1989, Gordon and Townsend, 1989, Kanekp and Tanie, 1992]. Most of these papers discuss how the contact location for a single point contact can be determined from joint torque or

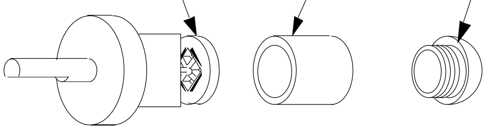


Figure 3.1: 6-axis fingertip force-torque sensor

internal wrench measurements and a model of the geometry. [Bicchi *et al.*, 1993] presents a general theory for this problem.

A six-axis fingertip force-torque sensor was used in [Bicchi *et al.*, 1989] to estimate the onset of slip (figure 3.1). This sensor has a Maltese-cross connecting the outer shell to the base. The cross is instrumented with 8 strain-gauge half-bridges. The shell has a lightly damped natural frequency of approximately 690 Hz when the base is fixed and the shell free. In his experiments, Bicchi first determined the coefficient of friction for the object to be grasped. Then, by monitoring the ratio of the tangential force to the normal force, he was able to determine when the contact state was approaching the slip condition determined earlier.

This thesis extends this work of extracting primitive features to include temporal features and constraint features. Furthermore, the feature primitives are placed into a single framework which can be used for manipulation. In an earlier paper [Eberman and Salisbury, 1993], we showed how intrinsic contact sensing could be used to detect changes in the spectral signal characteristics of the contact wrench. Greater detail on this approach is contained in [Eberman and Salisbury, 1994]. Chapter 7 covers some aspects of this work.

Although constraint is an important part of the assembly process, and is critical to force control, there appears to be little work in active sensing or processing of the information for the purposes of identifying the constraints on a part. [Simunovic, 1979] presents one of the first results in this area. This paper considered determining the relative position between the grasped part and the mating part using a sequence of position measurements assuming all measurements came from a single known contact configuration. He formulates an Extended Kalman Filter for the unknown positional bias between two parts. The filter takes in position measurements and recursively estimates the bias. His suggestions for future work anticipate part of the work in this thesis. To quote from his thesis:

> The implementation of the methods in this work requires further development in the following areas:
>
> - ...
> - develop the necessary checks and implementation to allow for parallel estimation and identification of alternative touching configurations;
> - ...

More recently [Delson, 1994] looked at programming a robot using human demonstration. As part of this work he developed some tests for segmenting the motion data into section that come from a single contact configuration. The tests were done either only on the force or on the velocity depending upon the contact condition. Because this is the focus of this thesis, this thesis significantly extends his work by explicitly modeling the process noise, making decisions on all of the data, making all the decisions on both the position and the force information, and creating a framework that can incorporate more than just constraint models.

3.4 Hardware

This thesis used a 6-axis force torque sensor, used by Bicchi, that is designed for the Salisbury Hand (figure 3.1) for all the experiments. The sensor is attached to a number of different experimental apparatus. The first device was a single glass epoxy link which was driven by a torque controlled brushless motor (figure 3.2). Under closed-loop torque control, this system is able to control the forces to 1 part in 500 with very little internal vibration. The motor's design and control is described in [Levin, 1990].

The second device mounted the fingertip sensor on the PHANToM. The PHANToM is a three degree-of-freedom force reflecting haptic interface. The design gives it a high force dynamic range (around 100:1) and low vibration levels during motion. In this thesis it was used as a large finger (figure 1.2).

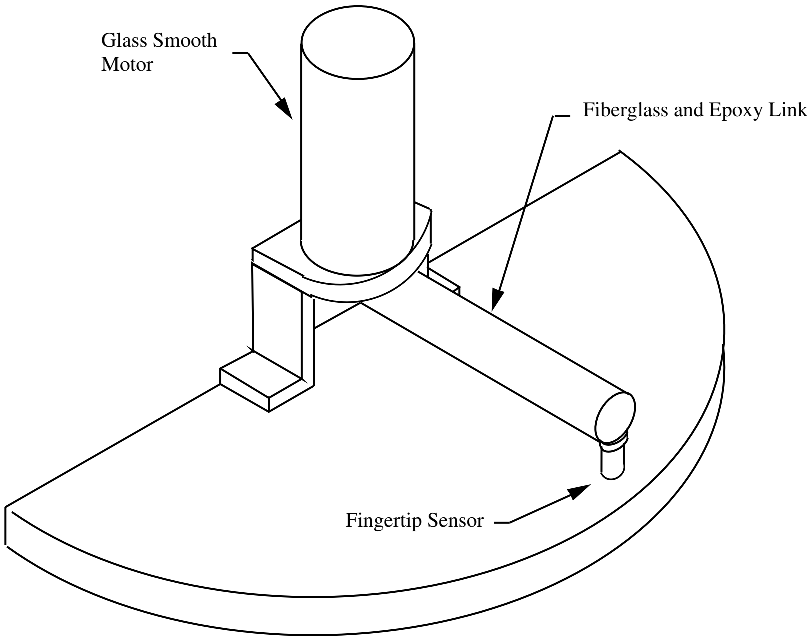


Figure 3.2: Single degree-of-freedom hardware

3.5 Conclusion

Force and torque information combined with velocity information provides the sensing equivalent of human tool manipulation. Although this is only a small subset of the information used during human manipulation, it is one of the most common and important forms of environmental interaction. Only recently has the tactile sensing community begun to work in this area. Previous work in this area has tended to focus on grasping issues.

The next chapter presents configuration space as necessary background material for the remaining thesis presentation.

Manipulation and Constraints

Chapter 4

This chapter discusses a representation for the motions of and forces on objects with contact constraint. The forces that the force sensor measures are caused by reaction forces from the constraints and by relative motion along the constraints. Additional background on this chapter's treatment of configuration space and deterministic dynamics can be found in books on advanced classical mechanics. A very readable treatment of the methods of mathematical classical mechanics including manifolds, tangent bundles, cotangent bundles, and forms from a physics viewpoint is given by [Arnold, 1989]. An introductory mathematics text on this material is [Munkres, 1991].

The structure of the dynamics and the sensor measurements in configuration space creates a natural framework in which to construct a feature observer. Motions in configuration space give rise to contact models, namely: 1) impacts, 2) constraints, and 3) velocity dependent characteristics of the noise. An observer can be designed to work with these models in order to estimate the current measurement model. The current model determines a set of configurations, corresponding to a piece of configuration space topology that the robot might currently occupy. These pieces of topology are connected in the configuration space and induce notions of connectivity between the measurement models. The next chapter formalizes this idea and discusses possible definitions for connectivity between the models.

This chapter provides background for the other chapters and can be skimmed and then referred to when reading the other chapters. The essential ideas to get from this chapter are:

1. The representation of motion in terms of configuration space.
2. That configuration space consists of the union of smooth surfaces.
3. That each surface has a collection of tangent surfaces, called the *tangent bundle*, and that each surface also has a collection of generalized normals called the *cotangent bundle*.
4. That contact forces are generated by contact with these surfaces.
5. That impacts occur because of transitions between these surfaces.

Finally, the end of this chapter provides a statistical treatment of dynamics and force sensing for a robot. This section is not central to the main thread of the thesis.

4.1 Configuration Space

Manipulation is the problem of changing the location and orientation, or *configuration*, of objects in the environment in order to bring objects into contact. Every contact on an object presents constraints on the allowed motions of the object. As the object moves the contacts and therefore the constraints on an object change discontinuously. *Configuration space* makes these kinematic constraints explicit.

The position of a body relative to a fixed origin is defined by a set of generalized coordinates $\mathbf{x}$. The set of all allowed coordinates is called the configuration space denoted by $\mathcal{C}$. For example, the configuration of a point in space can be specified by a vector giving the Cartesian coordinate values for the location of the point. The configuration of a n jointed serial linkage can be specified by its n joint coordinates. The configuration of a polygon in the plane can be specified by the position and orientation of its coordinate frame.

Configuration space has many important properties. First, obstacles in the environment can be mapped into constraint surfaces in the configuration space. Motion into these surfaces is prevented by a configuration space constraint force which we term a *constraint wrench*. The total contact wrench is the sum of the constraint wrenches and the wrenches tangent to the constraints. The sum all of all the contact wrenches is what is measured by the intrinsic contact sensor.[1]

[1]Wrench and twist typically refer to the generalized force and velocity of a three dimensional

When the accelerations of the body can be neglected all of the bodies dynamics can be represented in configuration space. The quasi-static assumption is that the net wrench on the body is zero. An analysis, given in this chapter, shows that this assumption is equivalent to a stationary statistical assumption for the second order dynamics. For the majority of robot interactions this assumption is enforced by using an appropriate control algorithm. Section 4.3 discusses this assumption in greater detail. With the quasi-static assumption, the generalized velocity of the body, or twist, is a function only of the desired motion and the configuration.

4.2 Kinematic Constraints

Since the position of the body is represented by a point in $\mathcal{C}$, constraints created by contact between objects are transformed to constraints on the motion of the point. The point in $\mathcal{C}$ which correspond to the overlap of objects are called *configuration space obstacles* denoted by $\mathcal{O}$. The complement of the obstacles space is the free space and is denoted by $\mathcal{F}$. This section discusses the characteristics of these kinematic constraints in general. The computation of the constraints for polygons in the plane and points in Cartesian space is deferred to chapter 8. As an example of the concepts, we discuss the configuration space for a moving polygon on a plane with other polygons as obstacles.

Most man-made objects can be modeled as the union of several generalized surfaces. For example, a circle has one surface: the outer edge. A triangle has six surfaces: three edges and three vertices. These are different surfaces because there is a discontinuity in the triangle at each corner. The dimension of a surface is the number of coordinates it takes to parametrically specify a location on the surface. In general, objects can be broken down into surfaces of equal dimension by separating the object along places of geometric discontinuity. Each of these surfaces is called a *manifold*. This separation can be extended to include changes in surface properties like texture and friction. In this section, each surface of an object is assumed to have uniform texture and friction.

The geometric form of the manifold depends upon some geometric parameters $g_i \in \Re^k$. For example, the location of a vertex is specified by its coordinates in the object frame. The location of an edge can be given by the outward normal of the line

body when these terms are represented as screws [McCarthy, 1990]. We are using these terms for any generalized force and velocity to avoid confusion with the force and velocity of a point mass.

supporting the edge and the distance to the edge. This last example shows that the boundaries of a manifold are not described directly by the geometric parameters. A boundary is defined by the intersection of two manifolds which is itself another manifold.

The structure of objects as collections of manifolds induces the same type of structure on the configuration space obstacles. A single contact between a surface on the moving object and a surface in the environment can be defined by a constraint equation

$$\mathbf{C}_{(f_m, f_e)} : \Re^{k_m} \times \Re^{k_e} \times \mathcal{C} \rightarrow \Re. \tag{4.1}$$

$\mathbf{C}_{(f_m, f_e)}$ is the constraint equation for contact between surface f_m on the moving object, described by k_m parameters, and surface f_e, described by k_e parameters, of the environment.

The constraint manifolds define the boundary of the configuration space obstacles. For many object types, each of these surfaces can come from only a finite class of primitives. For polygons each surface must be either a vertex or an edge. For polyhedra the surface must be either a vertex, an edge, or a face. Therefore, a constraint equation can be considered to be of a certain type

$$\mathbf{C}_{type} : \Re^{k_i} \times \Re^{k_j} \times \mathcal{C} \rightarrow \Re. \tag{4.2}$$

The set of $\mathbf{x}$ which satisfy $\mathbf{C}_{type}(\mathbf{g}_i, \mathbf{g}_j, \mathbf{x}) = 0$ defines a constraint manifold $\mathcal{M}_{i,j}$ for feature pair (i, j). Note that the free space is itself a manifold.

The configuration space manifolds for a point robot in $\Re^3$ are easy to visualize. The configuration space for a point robot is just $\Re^3$. The configuration space constraints are the surfaces that the point can contact in the original space. For example, if the point is touching a plane, then that plane creates a plane constraint manifold in the configuration space.

The configuration space of a polygon in a plane is the space of coordinate transformations $\Re^2 \times SO(2)$. $SO(2)$ is the two dimensional rotation group which is parameterized by a single value θ. Therefore, the configuration space is three dimensional. Therefore, transformations can be specified by the triple (x, y, θ) which makes visualization easier.

For polygons there are two basic constraint types. *Type A* is contact between an edge on the moving polygon and a vertex on the fixed polygon. *Type B* contact is contact between a vertex on the moving polygon and an edge of the fixed polygon (

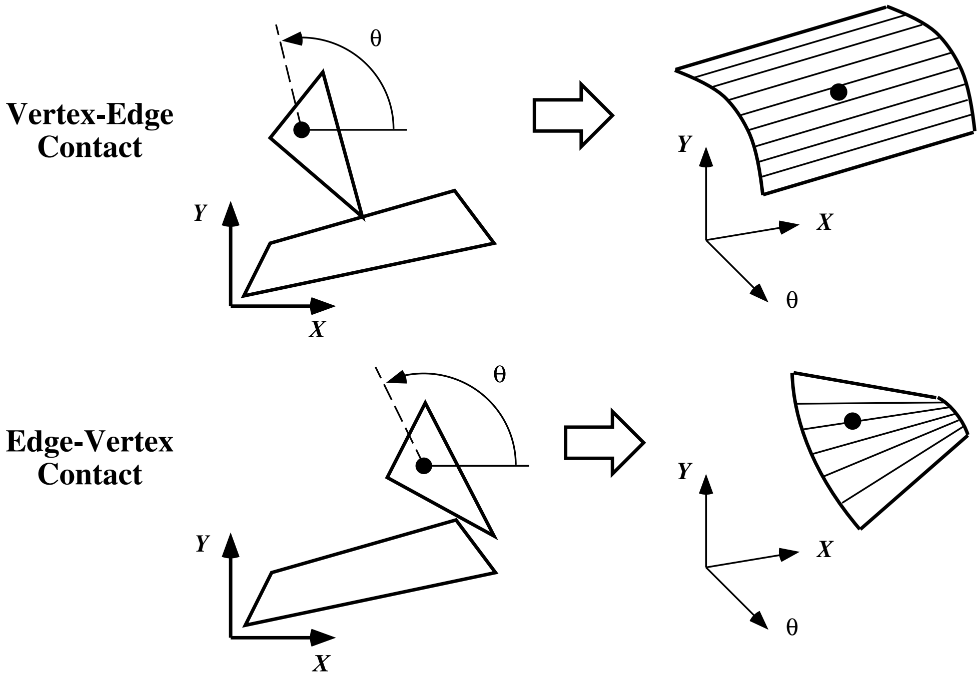


Figure 4.1: Type B (vertex-edge) and type A (edge-vertex) contact between two polygons. Figure from [Caine 1993].

[Lozano-Pérez, 1983]). Contact between two vertices is a limiting case of contacting two edges. Contact between two edges can also be treated as two contacts. Both cases will be considered in the next section.

Figure 4.1 illustrates the configuration space obstacles for type A and B contacts. Because polygons consist of straight lines, the configuration space obstacles are ruled surfaces. At any angle θ the configuration space obstacle is a straight line. The orientation of this line changes as the orientation of the moving object changes.

The curvature of the surface is a function of the signed distance of the moving origin from the contact. The curvature of the surface can be changed by moving the control point. The surface can be changed from convex to concave by moving the control point to the other side of the contacted edge.

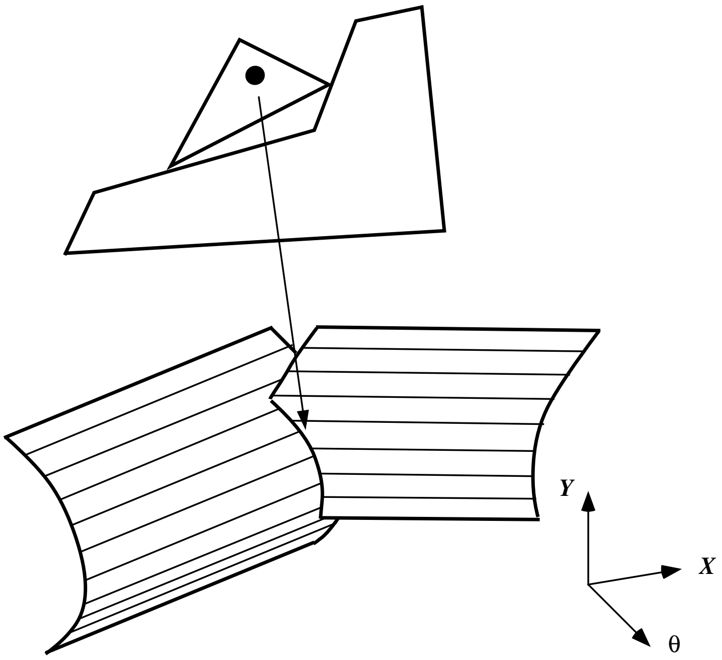


Figure 4.2: Two type B contacts creating a corner [Caine 1993].

4.2.1 Multiple Contacts

Multiple contacts are specified by requiring several constraint equations to hold simultaneously. With m constraint equations the complete constraint equation can be written as

$$\mathbf{C}_{group} : \Re^{k_{group}} \times \mathcal{C} \to \Re^m \tag{4.3}$$

where k_{group} is the vector of parameters specifying the contact geometry. The set of all $\mathbf{x}$ which satisfy $\mathbf{C}_{group}(\mathbf{g}_{group}, \mathbf{x}) = 0$ is again a manifold $\mathcal{M}_{group}$. This surface is the intersection of all the $\mathcal{M}_{i,j}$ for each pair (i, j) in the constraint group.

Just like with the underlying objects, the intersection of two configuration space manifolds creates a manifold of lower dimension. Two 2 dimensional surfaces intersect to create a one dimensional curve. Two curves intersect to create a point.

Figure 4.2 shows the configuration space obstacle for two vertices contacting a corner. The obstacle takes the form of a section of a helix. The obstacle is the intersection of the two constraints imposed by the contact between the two vertices and the two edges.

4.2.2 Connectivity of Configuration Space

Figure 4.3 shows the configuration space for a triangular object with a single quadrilateral obstacle. There are several important properties shown in this configuration obstacle. First, the complete configuration space consists of the union of many smooth manifolds. Each manifold surface is generated from one particular pairing of features on the moving object and features on the fixed object. The connectivity of these manifolds forms a natural graph of the topology of the configuration space.

There are two important types of connectivity for the configuration space topology. *Path connectivity* labels two manifolds as connected if there is a path connecting the manifolds which lies entirely in the two manifolds. *Phase flow* connectivity refines path connectivity be requiring that the path be an integral curve of the dynamics. That is the path must be physically realizable given the dynamics and any kinematic constraints.

Figure 4.4 shows a graph representation of the topology for the configuration space for the indicated objects . Each node in the graph has been labeled by the featuring pairings that are active at that graph node. This graph has been called the graph of assembly process states or the contact states in [Asada and Hirai, 1989]. For non-convex objects, such as peg-in-hole assembly problems, the connectivity is significantly more complicated than the graph shown. In either case, the graph shows the possible contact transitions and is invariant to changes in the reference point.

Phase flow connectivity represents the possible connections given the dynamics. Nonholonomic systems have differential constraints on the allowed velocities at any point in configuration space. A car can only move in the direction of the instant center formed by the wheels. In general, this is the important definition of topological connectivity. Contact conditions are connected if there is an integral curve that connects the two contacts.

Figure 4.3 also illustrates that the surface of any individual manifold is smooth or differentiable.[2] The surface is smooth since each manifold is defined by the set of allowed configurations for pairings between smoothly varying features on the moving and constraint objects. Therefore, the wrenches of constraint and the allowed motions on the surface do not change abruptly, and the second order dynamics of motion for the moving object is continuous for motions constrained to lie on a single

[2]Note that the straight lines in the figure are artificial and only serve to illustrate the curvature of the surface.

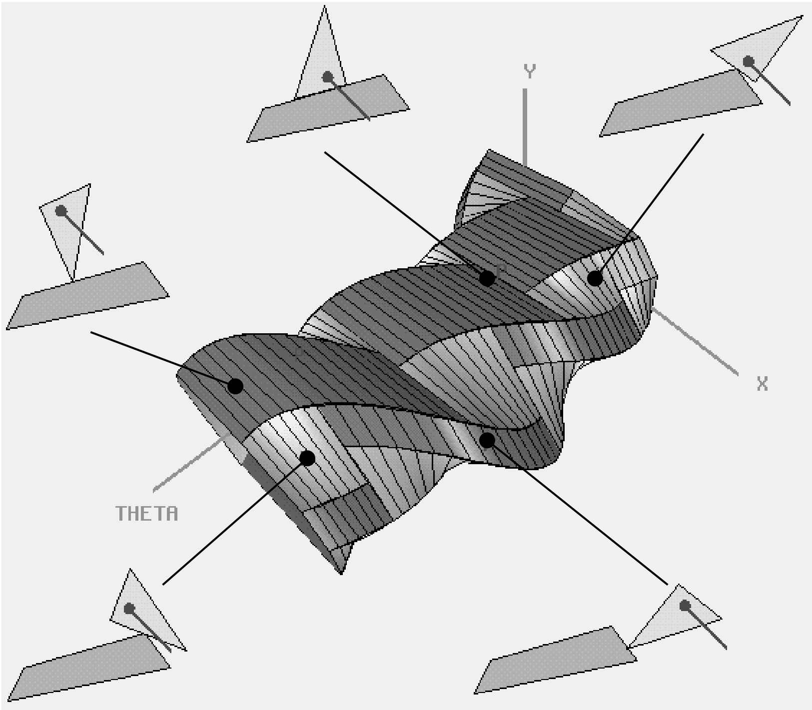


Figure 4.3: Triangular object with a single quadrilateral obstacle. The figure shows the configuration space with regions labeled with the feature pairings. Note how edge-edge contacts are formed by the intersection of the bounding vertex-edge contacts. (Figure from [Caine 1993]).

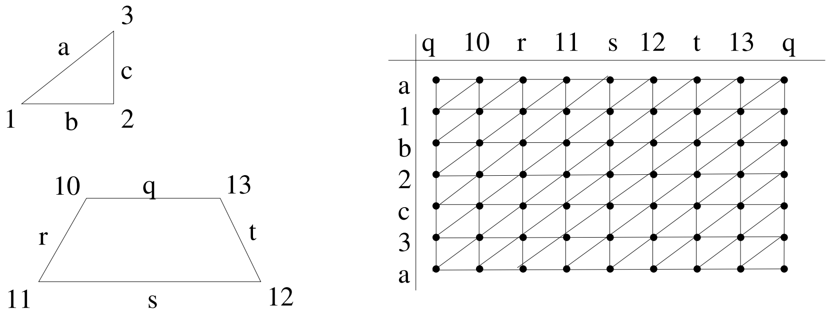


Figure 4.4: Path connectivity of the configuration space topology for the indicated convex objects. The configuration space is the (x, y, θ) coordinates of the triangle. Each dot in the matrix indicates a possible pairing between features on the triangle and features on the parallelogram. A segment indicates the pairings are connected. The other diagonal direction does not appear because those segments represent paths that cause objects to pass through each other.

manifold.

Therefore, the nodes of the topology graph represent connected, smooth regions of configuration space and edges represent discontinuous transitions between surfaces. As section 4.3 discusses, this structure implies the quasi-static assumption will be true on any one manifold, with dynamic effects restricted to the transitions given a smooth, stable controller.

4.2.3 Tangent and Cotangent Bundles

For every point on a constraint manifold $\mathcal{M}$, there are two associated vector spaces: the tangent space and the cotangent space. The tangent space is the vector space of all velocities which would results in motions that differentially remain on the constraint. The cotangent space is the dual vector space which is in the null space of the tangent space. The cotangent space is the space of all wrenches of constraint. The cotangent space is the space spanned by the derivative of the constraint equation at a point $\mathbf{x}$. Again, the tangent space lies in the null space of the cotangent space. In the force control literature, the inner product of vectors in the tangent space and covectors in the cotangent space is termed the *reciprocal product*. This inner product

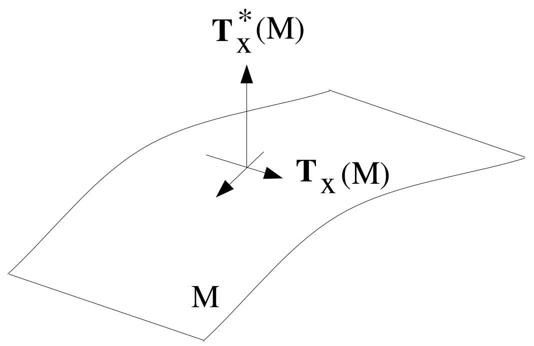


Figure 4.5: The tangent $\mathcal{T}_x(\mathcal{M})$ and cotangent $\mathcal{T}^*_{\mathbf{x}}(\mathcal{M})$ spaces at a point on a manifold.

is always zero since the tangent space is the null space of the cotangent space.

We denote the tangent space at a point $\mathbf{x}$ as $\mathcal{T}_{\mathbf{x}}(\mathcal{M})$ and the associated cotangent space as $\mathcal{T}^*_{\mathbf{x}}(\mathcal{M})$. The union of these spaces over all points on the manifold is called the tangent bundle and cotangent bundle.

$$\mathcal{T}(\mathcal{M}) = \bigcup_{\mathbf{x}\in\mathcal{M}} \mathcal{T}_{\mathbf{x}}(\mathcal{M}) \tag{4.4}$$

$$\mathcal{T}^*(\mathcal{M}) = \bigcup_{\mathbf{x}\in\mathcal{M}} \mathcal{T}^*_{\mathbf{x}}(\mathcal{M}) \tag{4.5}$$

Figure 4.5 illustrates these ideas. The configuration space together with its tangent bundle is called the *phase-space*.

When the constraint surfaces are produced through contact, and not through connection of a physical linkage, $\mathcal{T}^*_{\mathbf{x}}(\mathcal{M})$ is unisense because the constraint equations are properly inequalities. Constraint wrenches can only be produced to prevent the body from penetrating the obstacles. Constraint wrenches which prevent the body from leaving the obstacles are not possible. Our constraint estimation procedures will be interested in isolating only those times when the constraint is active. In this case, the constraint can be treated as an equality constraint.

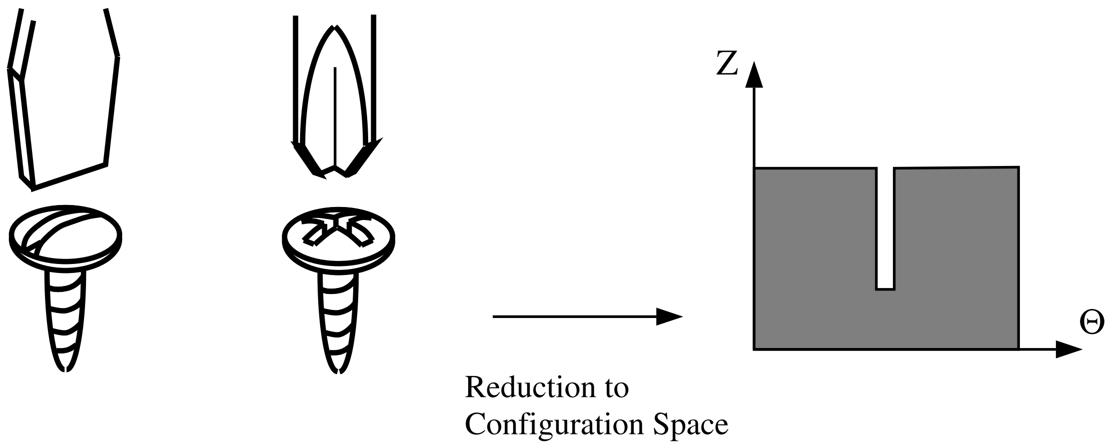


Figure 4.6: Configuration space manifold equivalence for a Phillips and flathead screw [Caine 1993].

4.2.4 Configuration Space Equivalence

While the geometric parameters, $\mathbf{g}_{group}$, of any constraint surface are sufficient to describe the surface, they may not be unique. Thus, the problem of determining equivalence between two different descriptions arises. Two descriptions $\tilde{\mathbf{g}}$ and $\mathbf{g}$ produce equivalent manifolds, or are equivalent, if they produce the same constraint surface.

For example, a Phillips screw and screw driver produce the same constraint surfaces as a flathead screw and screwdriver for (z, θ) motions (Figure 4.6). Several examples of this are given in [Caine, 1993]. Some mechanisms also have this property. Figure 4.7 shows a moving triangle contacting the environment with two type B contacts. These contacts, except for the limited range of motion, are equivalent to a mechanism with two sliders as shown. Each contacting vertex can be considered as attached to a slider aligned with the edge through a pivot. An equivalent mechanism can be constructed by moving the attachment points and slider directions. Both the triangle and the indicated mechanism generate the configuration space constraint curve shown.

Equivalence can be tested by examining the tangent bundles. Two descriptions $\tilde{\mathbf{g}}$ and $\mathbf{g}$ are equivalent if at every $\mathbf{x} \in \mathcal{M}$ the span of the cotangent and tangent spaces are the same. Clearly, two equivalent manifolds have the same tangent bundles because they are exactly the same surface. Two equivalent tangent bundles also determine

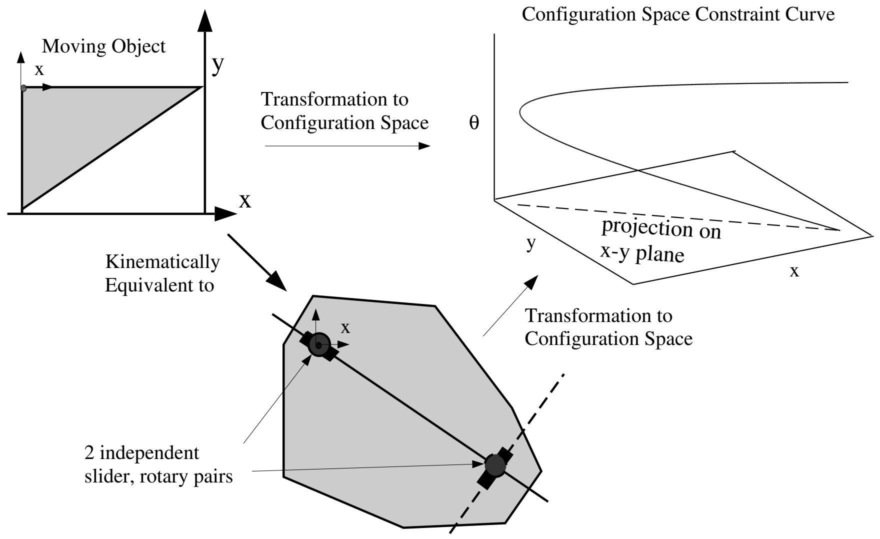


Figure 4.7: Configuration space manifold equivalence for two different slider assemblies.

the same surface because they have the same integral curves.

The tangent bundle notion of equivalence is sometimes much easier to test. Two tangent bundles are equivalent if the functions which generate a basis for the tangent space span the same space at every point. For two constraints in a three degree-of-freedom problem, the span consists of only a single vector. Equivalence can be checked by looking to see if the two mechanisms produce the same vector, up to a change in scale, by looking at the functional form for the vector. This example case of two type B contacts is shown to be equivalent in section 8.1.3.1.

4.3 Dynamics

One of the difficult and challenging properties of manipulation is that it takes place in a dynamic environment where the dynamics switch discontinuously as a function of the generalized coordinates. Although this class of problem arises often in practice, a formal control treatment has not yet been developed for analyzing stability or robustness.

In this section we discuss the physics and statistics of manipulation for a robot The analysis starts by formulating the second order, Hamiltonian, stochastic equations of motion for a system consisting of a robot, an sensor, an end-effector, and the environment for a single contact condition. The Hamiltonian formulation uses the momentum and position as basic state variables. The effect of assuming statistical stationarity on the state and measurement statistics is then determined. This analysis shows that the quasi-static assumption for manipulation is equivalent to the expected value of the stationary statistics of the second order dynamics. The analysis also shows how variation in the constraint equations creates a velocity dependent disturbance and how the other disturbances create errors in the wrench signal.

Then transitions between constraints are considered. These transitions correspond to edge transitions in the topology graph. Transitions almost always create impact forces because of either: 1) the discontinuous change in momentum required to satisfy the constraint, or 2) the discontinuous loss of constraint resulting in the release of stored elastic energy. Detecting and isolating these transitions is important in rigid body manipulation because they correspond to regions where the quasi-static assumption does not hold. Furthermore, the direction of momentum change provides information about which edge in the topology graph was the source of the transition.

4.3.1 Constant Contact Constraints

The equations of dynamics and the process statistics are most easily represented by considering the momentum and position form (Hamiltonian) of the dynamic equations. This approach results in two sets of first order differential equations instead of a single set of second order differential equations.

Figure 4.8 shows an example of contact between the end-effector (this could be an object held in a grasp) and the environment. The configuration of the robot and the end-effector is specified by $\mathbf{x}_r$ and $\mathbf{x}$ respectively. They are connected by the sensor which is represented as a generalized spring $\mathbf{K}$ and damper $\mathbf{B}$. They have inertia $\mathbf{H}_r$ and $\mathbf{H}$ respectively. The generalized momenta of the robot and end-effector are $\mathbf{p}_r = \mathbf{H}_r\dot{\mathbf{x}}_r$ and $\mathbf{p} = \mathbf{H}\dot{\mathbf{x}}$. The total momentum is $\mathbf{p}_t = \mathbf{p}_r + \mathbf{p}$. We will assume that the generalized inertia matrix is always invertible. This implies that that robot is always away from singular configurations.

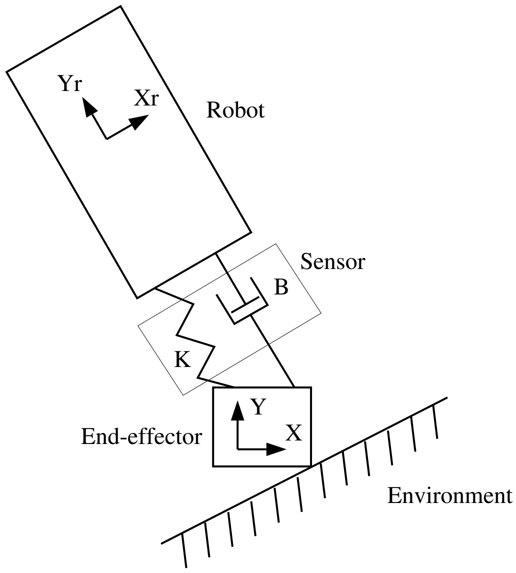


Figure 4.8: Model of the robot, the force/torque sensor, the environment and the contacting link.

The wrench measured by the sensor is

$$\mathbf{w}_m = \mathbf{K}(\mathbf{x} - \mathbf{x}_r) + \mathbf{B}(\dot{\mathbf{x}} - \dot{\mathbf{x}}_r) \tag{4.6}$$

since both the stiffness and damping loads are measured by the strain gauges in the sensor. With this notation, the dynamics of the robot and end-effector are given by the coupled equations

$$\begin{aligned} \dot{\mathbf{p}}_r + \mathbf{G}_r(\mathbf{x}_r) &= \mathbf{w}_r + \mathbf{w}_m && (4.7) \\ \dot{\mathbf{p}} + \mathbf{G}(\mathbf{x}) + (\mathbf{C}_{\mathbf{x}})\boldsymbol{\lambda} &= \mathbf{w}_d - \mathbf{w}_m && (4.8) \\ \text{subject to} \qquad \mathbf{C}(\mathbf{g}, \mathbf{x}) &= 0 && (4.9) \end{aligned}$$

where $\mathbf{G}_r$ is the gravity wrench on the robot, $\mathbf{G}$ is the gravity wrench on the end-effector, $\mathbf{w}_r$ and $\mathbf{w}_d$ are the applied robot wrench and a contact disturbance wrench, $\mathbf{C}_{\mathbf{x}}$ is the derivative of the constraint equation $\mathbf{C}(\mathbf{g}, \mathbf{x}) = 0$ with respect to $\mathbf{x}$, and $\boldsymbol{\lambda}$ is a Lagrange multiplier. The product $(\mathbf{C}_{\mathbf{x}})\boldsymbol{\lambda}$ is a constraint vector in the cotangent space of the contact constraint. The disturbance wrench lies entirely in the null space of the constraint wrench. To simplify the notation let $\mathbf{C}_{\mathbf{x}} = \mathbf{C}_{\mathbf{x}}$. The constraint equation can also be expressed as the inner product of the generalized velocity and the cotangent vector space.

$$\mathbf{C}_{\mathbf{x}}^T \dot{\mathbf{x}} = 0. \tag{4.10}$$

Adding the momentum equations gives the total momentum equation

$$\dot{\mathbf{p}}_t + \mathbf{G}_r + \mathbf{G} + \mathbf{C}_{\mathbf{x}}\boldsymbol{\lambda} = \mathbf{w}_r + \mathbf{w}_d. \tag{4.11}$$

Now assume that the robot controller takes the form of a generalized spring and damper with gravity compensation and applies its wrench with a zero mean random error

$$\mathbf{w}_r = \mathbf{K}_r(\mathbf{x}_d - \mathbf{x}_r) + \mathbf{B}_r(\mathbf{v}_d - \dot{\mathbf{x}}_r) + \mathbf{G}_r + \mathbf{G} + \mathbf{w}. \tag{4.12}$$

In order to implement this equation, we assume that $\mathbf{x}_r - \mathbf{x}$ is small, so that $\mathbf{G}$ can be implemented as $\mathbf{G}(\mathbf{x}_r)$ creating only a small error.

Changing the control equation into momentum variables and substituting the result into equation 4.11 yields the approximation for the total momentum

$$\begin{aligned} \dot{\mathbf{p}}_t + \mathbf{C}_{\mathbf{x}}\boldsymbol{\lambda} + \mathbf{B}_r\dot{\tilde{\mathbf{x}}}_t + \mathbf{K}_r\tilde{\mathbf{x}}_t &= \mathbf{w} + \mathbf{w}_d & (4.13)\\ \mathbf{p} - \mathbf{H}_t\dot{\tilde{\mathbf{x}}} &= \mathbf{H}_t\mathbf{v}_d & (4.14) \end{aligned}$$

where $\tilde{\mathbf{x}}_t = \mathbf{x}_r - \mathbf{x}_d$ is the tracking error and $\dot{\tilde{\mathbf{x}}}_t$ is the derivative of the tracking error.

Now, we rewrite the dynamics of the end-effector in error coordinates. Let $\tilde{\mathbf{x}} = \mathbf{x} - \mathbf{x}_r$ be the difference between the end-effector configuration and the robot's configuration and let the momentum of the difference be $\tilde{\mathbf{p}} = \mathbf{H}\dot{\tilde{\mathbf{x}}}$. Then equation 4.8, the momentum of the end-effector, can be expressed as

$$\dot{\tilde{\mathbf{p}}} + \mathbf{B}\dot{\tilde{\mathbf{x}}} + \mathbf{K}\tilde{\mathbf{x}} = -\mathbf{G}(\mathbf{x}) - \mathbf{C}_{\mathbf{x}}\boldsymbol{\lambda} - \frac{\mathrm{d}}{\mathrm{dt}}[\mathbf{H}\dot{\mathbf{x}}_r] + \mathbf{w}_d \tag{4.15}$$

since the wrench measurement is $\mathbf{w}_m = \mathbf{K}\tilde{\mathbf{x}} + \mathbf{B}\dot{\tilde{\mathbf{x}}}$. The statistics of the resulting pair of equations can now be treated by first solving for the statistics of the total momentum, including the effects of constraint, and then using the result as part of the forcing function in equation 4.15.

Before considering the statistics for constrained motion, it is illustrative to examine the statistics for free flight with a constant desired velocity. In free flight, the expected value of a first order expansion of 4.13 and 4.14 is

$$\begin{aligned} \frac{\mathrm{d}}{\mathrm{dt}}\mathrm{E}[\mathbf{p}_t] + \mathbf{B}_r\frac{\mathrm{d}}{\mathrm{dt}}\mathrm{E}[\tilde{\mathbf{x}}_t] + \mathbf{K}_r\mathrm{E}[\tilde{\mathbf{x}}_t] &= 0 & (4.16)\\ \mathrm{E}[\mathbf{p}_t] - \mathbf{H}_t\frac{\mathrm{d}}{\mathrm{dt}}\mathrm{E}[\tilde{\mathbf{x}}_t] &= \mathbf{H}_t\mathbf{v}_d. & (4.17) \end{aligned}$$

For a stationary solution, the expected values must be constant yielding

$$\mathrm{E}[\mathbf{p}_t] = \mathbf{H}_t\mathbf{v}_d \qquad \mathrm{E}[\tilde{\mathbf{x}}_t] = 0 \tag{4.18}$$

The covariance of the coupled system is best represented by rewriting the equation for free flight as

$$\begin{bmatrix} \mathbf{Id} & \mathbf{B}_r \\ 0 & \mathbf{H}_t \end{bmatrix} \dot{\boldsymbol{\zeta}} + \begin{bmatrix} 0 & \mathbf{K}_r \\ -\mathbf{Id} & 0 \end{bmatrix} \boldsymbol{\zeta} = \begin{bmatrix} 0 \\ -\mathbf{H}_t \end{bmatrix} \mathbf{v}_d + \begin{bmatrix} \mathbf{Id} \\ 0 \end{bmatrix} \mathbf{w} \tag{4.19}$$

$$\mathcal{A}_t \dot{\boldsymbol{\zeta}} + \mathbf{B}_t \boldsymbol{\zeta} = \mathbf{u} + \mathbf{S}_t \mathbf{w} \tag{4.20}$$

where the matrices are defined appropriately and $\boldsymbol{\zeta}^T = [\mathbf{p}_t^T \quad \tilde{\mathbf{x}}_t^T]$. With this notation, the steady-state covariance of $\boldsymbol{\zeta}$, $\mathbf{V}_{\boldsymbol{\zeta}}$ is the solution of the continuous time Lyapunov equation and satisfies

$$\mathbf{B}_t \mathbf{V}_{\boldsymbol{\zeta}} \mathcal{A}_t^T + \mathcal{A}_t \mathbf{V}_{\boldsymbol{\zeta}} \mathbf{B}_t^T = \mathbf{S}_t \mathbf{V}_{\mathbf{w}} \mathbf{S}_t^T \tag{4.21}$$

where $\mathbf{V}_{\mathbf{w}}$ is the covariance of the control error.

Similarly, the difference momentum equation for free flight is given by

$$\begin{bmatrix} \mathbf{Id} & 0 \\ 0 & \mathbf{H} \end{bmatrix} \dot{\boldsymbol{\nu}} + \begin{bmatrix} 0 & \mathbf{K} \\ -\mathbf{Id} & 0 \end{bmatrix} \boldsymbol{\nu} = \begin{bmatrix} -\mathbf{G}(\mathbf{x}) \\ 0 \end{bmatrix} - \begin{bmatrix} \mathbf{Id} \\ 0 \end{bmatrix} \frac{\mathrm{d}}{\mathrm{dt}} [\mathbf{H} \dot{\mathbf{x}}_r] \tag{4.22}$$

$$\mathcal{A} \dot{\boldsymbol{\nu}} + \mathbf{B} \boldsymbol{\nu} = \begin{bmatrix} -\mathbf{G}(\mathbf{x}) \\ 0 \end{bmatrix} - \mathbf{S} \frac{\mathrm{d}}{\mathrm{dt}} [\mathbf{H} \dot{\mathbf{x}}_r] \tag{4.23}$$

where $\boldsymbol{\nu}^T = [\, \tilde{\mathbf{p}} \quad \tilde{\mathbf{x}}\,]$. Taking expectations of a first order expansion gives the stationary solution

$$\mathbf{K}\mathrm{E}[\tilde{\mathbf{x}}] = -\mathbf{G}(\mathbf{x}) - \dot{\mathbf{H}} \mathbf{v}_d \qquad \mathrm{E}[\tilde{\mathbf{p}}] = 0 \tag{4.24}$$

since $\frac{\mathrm{d}}{\mathrm{dt}} \mathrm{E}[\mathbf{H} \mathbf{v}_r] = \frac{\mathrm{d}}{\mathrm{dt}} (\mathbf{H} v_d)$ and $\mathbf{v}_d$ is constant. This shows that the expected measurement is affected by both the gravity wrench and the changes in inertia for the moving frame. The driving term in 4.23 is $\frac{\mathrm{d}}{\mathrm{dt}}[\mathbf{H}\dot{\mathbf{x}}_r]$. Its covariance is

$$\mathbf{V}[(\frac{\mathrm{d}}{\mathrm{dt}}[\mathbf{H}\dot{\mathbf{x}}_r])(\frac{\mathrm{d}}{\mathrm{dt}}[\mathbf{H}\dot{\mathbf{x}}_r])^T] = \left[\frac{\mathrm{d}}{\mathrm{dt}}(\mathbf{H}\mathbf{H}_t^{-1})\right] \mathbf{F} \mathbf{V}_{\boldsymbol{\zeta}} \mathbf{F}^T \left[\frac{\mathrm{d}}{\mathrm{dt}}(\mathbf{H}\mathbf{H}_t^{-1})\right]^T \tag{4.25}$$

$$- \left[\frac{\mathrm{d}}{\mathrm{dt}}(\mathbf{H}\mathbf{H}_t^{-1})\right] \mathbf{F} \mathbf{V}_{\boldsymbol{\zeta}} \mathbf{B}_t^r \mathcal{A}_t^{-T} \mathbf{F}^T \left[(\mathbf{H}\mathbf{H}_t^{-1})\right]^T \tag{4.26}$$

$$- \left[(\mathbf{H}\mathbf{H}_t^{-1})\right] \mathbf{F} \mathcal{A}_t^{-1} \mathbf{B}_t \mathbf{V}_{\boldsymbol{\zeta}} \mathbf{F}^T \left[\frac{\mathrm{d}}{\mathrm{dt}}(\mathbf{H}\mathbf{H}_t^{-1})\right] \tag{4.27}$$

$$+ \left[(\mathbf{H}\mathbf{H}_t^{-1})\right] \mathbf{F} \mathcal{A}_t^{-1} \left[\mathbf{B}_t \mathbf{V}_{\boldsymbol{\zeta}} \mathbf{B}_t^T + \mathbf{S}_t \mathbf{V}_{\mathbf{w}} \mathbf{S}_t^T\right] \mathcal{A}_t^{-T} \left[(\mathbf{H}\mathbf{H}_t^{-1})\right]^T \tag{4.28}$$

where $\mathbf{F} = [\,\mathbf{Id} \quad 0\,]$ selects the momentum component of $\boldsymbol{\zeta}$. The control disturbance creates measurement variance both directly and through its effect on the variance of the robot velocity. This effect will be treated as constant and incorporated into a single modified variance $\mathbf{V}_{\mathbf{w}_w}$. This is justified for slow changes in the inertia tensor. Using this definition for the driving variance, the variance of $\boldsymbol{\nu}$ satisfies

$$\mathbf{B}\mathbf{V}_{\boldsymbol{\nu}}\mathcal{A}^T + \mathcal{A}\mathbf{V}_{\boldsymbol{\nu}}\mathbf{B}^T = \mathbf{S}\mathbf{V}_{\mathbf{w}_w}\mathbf{S}^T. \tag{4.29}$$

Finally let $\mathbf{D} = [\,\mathbf{B}\mathbf{H}^{-1} \quad \mathbf{K}\,]$, then $\mathbf{D}\boldsymbol{\nu} = \mathbf{w}_m$ and the variance of the measurement is $\mathbf{D}\mathbf{V}_{\boldsymbol{\nu}}\mathbf{D}^T = \mathbf{V}_{\mathbf{w}_m}$.

Now, consider the equations of constrained dynamics. For constrained motion, the average dynamics of the total robot looks like a point being pulled through a generalized spring and damper. The motion complies to the constraints because the average constraint wrench exactly balances the effect of the component of the command into the constraints. These wrenches deflect the connecting spring and add to the measured wrench. In addition, the expected value of the contact disturbance force also adds to the average measured wrench. Velocities which have a non-zero average will create an average bias disturbance through the effects of friction.

Besides the control error, the variance of the process depends on the variance of the contact disturbance and errors in the constraint model. These disturbances are caused by stick-slip during motion and textures and other small features. This second set of disturbances are velocity dependent. The equations make this precise.

The constraint dynamics equations consist of three coupled equations

$$\dot{\mathbf{p}}_t + \mathbf{C}_{\mathbf{x}}\boldsymbol{\lambda} + \mathbf{B}_r\dot{\tilde{\mathbf{x}}}_t + \mathbf{K}_r\tilde{\mathbf{x}}_t = \mathbf{w} + \mathbf{w}_d \tag{4.30}$$

$$\mathbf{p}_t - \mathbf{H}_t\dot{\tilde{\mathbf{x}}} = \mathbf{H}_t\mathbf{v}_d \tag{4.31}$$

$$\mathbf{C}_{\mathbf{x}}{}^T\dot{\tilde{\mathbf{x}}} = \mathbf{C}_{\mathbf{x}}{}^T\mathbf{v}_d. \tag{4.32}$$

The stationary solution requires $\mathbf{C}_{\mathbf{x}}{}^T\mathbf{v}_d = 0$. Then $\mathbf{p}_t = \mathbf{H}_t\mathbf{v}_d$ since the velocity is compatible with the constraints. Finally, errors in position are balanced by the average disturbance force and the constraints $\mathbf{K}_r\mathrm{E}[\tilde{\mathbf{x}}] + \mathbf{C}_{\mathbf{x}}\mathrm{E}[\boldsymbol{\lambda}] = \mathrm{E}[\mathbf{w}_d]$. A more illuminating assumption is that $\frac{\mathrm{d}}{\mathrm{dt}}\mathrm{E}[\mathbf{p}_t] = 0$, that is the system behaves quasi-statically on average.

The constraint dynamics equation 4.14 can be solved for $\dot{\mathbf{x}}_t$ and the result substituted in to the constraint equation 4.10 to get

$$(\mathbf{C}_{\mathbf{x}}{}^T\mathbf{B}_r^{-1}\mathbf{C}_{\mathbf{x}})\boldsymbol{\lambda} = \mathbf{C}_{\mathbf{x}}{}^T\mathbf{B}_r^{-1}(-\dot{\mathbf{p}}_t - \mathbf{K}_r\tilde{\mathbf{x}}_t + \mathbf{B}_r\mathbf{v}_d + \mathbf{w} + \mathbf{w}_d). \tag{4.33}$$

We have assumed that $\mathbf{B}_r$ is invertible. If we define the projection matrix

$$\mathbf{P} = \mathbf{B}_r^{-1}(\mathbf{Id} - \mathbf{C}_{\mathbf{x}}(\mathbf{C}_{\mathbf{x}}{}^T\mathbf{B}_r^{-1}\mathbf{C}_{\mathbf{x}})^{-1}\mathbf{C}_{\mathbf{x}}{}^T\mathbf{B}_r^{-1}) \tag{4.34}$$

then the error dynamics can be rewritten as

$$\mathbf{P}\dot{\mathbf{p}}_t + \mathbf{P}\mathbf{B}_r\dot{\tilde{\mathbf{x}}} + \mathbf{P}\mathbf{K}_r\tilde{\mathbf{x}} = \mathbf{P}(\mathbf{w} + \mathbf{w}_d). \tag{4.35}$$

Note that the projection matrix has the property $\mathbf{P}\mathbf{C}_\mathbf{x} = 0$. Therefore, the projection matrix takes covectors and projects them into the tangent space of the constraints.

Taking expectations and applying the assumption $\frac{\mathrm{d}}{\mathrm{dt}}\mathrm{E}[\mathbf{p}_t] = 0$, a first order differential equation results for the robot velocity

$$\frac{\mathrm{d}}{\mathrm{dt}}\mathrm{E}[\mathbf{x}] = \mathbf{P}(-\mathbf{K}\mathrm{E}[\tilde{\mathbf{x}}] + \mathbf{B}_r\mathbf{v}_d + \mathrm{E}[\mathbf{w}_d]) \tag{4.36}$$

Therefore, the average velocity is the result of projecting the control wrench $\mathbf{B}_r\mathbf{v}_d - \mathbf{K}\mathrm{E}[\tilde{\mathbf{x}}]$ plus the average disturbance wrench onto the tangent space through the damping matrix. The average momentum is the product of this velocity and the inertia tensor.

This also implies that the covariance of the momentum is not of full rank. This shows statistically that contact constraints reduce position uncertainty. The covariance of the momentum and tracking error is again a steady-state solution to a Lyapunov equation which can be derived using the projected form of the dynamics 4.35. More importantly, the constraint on velocity can be rewritten as a constraint on momentum yielding

$$\mathbf{C}_\mathbf{x}{}^T\mathbf{H}_t^{-1}\mathbf{V}_{\mathbf{p}_t}\mathbf{H}_t^{-T}\mathbf{C}_\mathbf{x} \quad = \quad 0 \tag{4.37}$$

as a constraint on the covariance of the momentum. Since $\mathbf{H}_t$ is full rank, this condition implies that $\mathbf{V}_{\mathbf{p}_t}$ has zero components in the directions of the constraints. The covariance is driven by both the control disturbance and random errors in the contact wrench.

The last derivation assumed that the constraint equations were known exactly. However, small surface imperfections or textures produce a small variability in the constraint equation. Therefore, to first order the constraint equation can be expanded to

$$(\mathrm{E}[\mathbf{C}_\mathbf{x}{}^T] + \delta\mathbf{C}_\mathbf{x}{}^T)\mathbf{H}_t^{-1}(\mathrm{E}[\mathbf{p}_t] + \delta\mathbf{p}_t) = 0. \tag{4.38}$$

By assumption $\mathrm{E}[\mathbf{C}_\mathbf{x}{}^T]\mathrm{E}[\mathbf{p}_t] = 0$, therefore,

$$\delta\mathbf{C}_\mathbf{x}{}^T\mathbf{H}_t^{-1}\mathrm{E}[\mathbf{p}_t] = -\mathrm{E}[\mathbf{C}_\mathbf{x}{}^T]\mathbf{H}_t^{-1}\delta\mathbf{p}_t \tag{4.39}$$

to first order. Therefore, the constraint on the covariance is modified to

$$\mathrm{E}[\mathbf{C_x}^T]\mathbf{H}_t^{-1}\mathbf{V}_{\mathbf{p}_t}\mathbf{H}_t^{-T}\mathrm{E}[\mathbf{C_x}] = \mathrm{E}[\delta\mathbf{C_x}^T\mathbf{H}_t^{-1}\mathrm{E}[\mathbf{p}_t]\mathrm{E}[\mathbf{p}_t]^T\mathbf{H}_t^{-T}\delta\mathbf{C_x}]. \tag{4.40}$$

Instead of having zero component in the direction of the constraints, the covariance now has a component in the direction of the average constraint which depends upon the square of the projection of the momentum into the error in the constraints. In actual practice, this velocity dependent error will produce measurement variance components in all directions. The magnitude of the variance will depend upon both the direction of travel and the square of the travel speed. Direction is important because many materials may have directional structure to their textures.

Now consider the difference momentum equation 4.15. The constraint modifies this equation to

$$\mathcal{A}\dot{\boldsymbol{\nu}} + \mathbf{B}\boldsymbol{\nu} = \begin{bmatrix} -\mathbf{G}(\mathrm{x}) \\ 0 \end{bmatrix} - \mathbf{S}\frac{\mathrm{d}}{\mathrm{dt}}[\mathbf{H}\dot{\mathbf{x}}_r] - \begin{bmatrix} \mathbf{C_x} \\ 0 \end{bmatrix}\boldsymbol{\lambda} \tag{4.41}$$

Therefore, taking expectations, gives the stationary solution

$$\begin{aligned} \mathrm{E}[\tilde{\mathbf{p}}] &= 0 && (4.42)\\ \mathbf{K}\mathrm{E}[\tilde{\mathrm{x}}] &= -\mathbf{G} + \dot{\mathbf{H}}\mathbf{v}_d - \mathrm{E}[\mathbf{C_x}\boldsymbol{\lambda}] && (4.43)\\ \mathrm{E}[\mathbf{w}_m] &= \mathbf{G} + \mathrm{E}[\mathbf{C_x}\boldsymbol{\lambda}] - \dot{\mathbf{H}}\mathbf{v}_d. && (4.44) \end{aligned}$$

The covariance of the measured force takes the same form as in free flight, but with the addition of a noise component that depends on the errors in the constraint.

In summary, the expected motion of the robot at steady-state on a single contact manifold can be described by mapping the applied wrenches onto the tangent space. Motion then occurs along the tangent space. The constraint wrenches necessary to achieve this motion cause a control error and create a possible large average measured wrench. The form of constraint wrench depends upon the type of the contact for systems with a finite number of types. Therefore, in order to predict the expected value of the measured wrench, the constraint type and the parameters of the constraint must be estimated. This is the topic of chapter 8. Second, the variance of the constraint wrench depends upon the contact disturbances, the robot control error, and on a velocity dependent term from the variability of the constraints through a Lyapunov equation. This velocity dependent term can depend both on the direction of the velocity and on the magnitude. Chapter 7 considers this velocity dependent term by examining the temporal characteristics of the noise from the force sensor. The dynamic effect is changes in constraint which is considered in the next section.

The equations can also be used for systems that are accelerating on a single manifold. In this case, the change in momentum will be measured by the sensor and the measurement covariance must be treated as time varying.

4.3.2 Changes in Constraint

There are only two types of changes in constraint: 1) changes from a manifold of lesser constraint (higher dimension) to one of more constraint (lower dimension), or 2) changes from lower dimension to higher dimensional surfaces. It is impossible to cross between two manifolds of the same dimension without first passing through either the larger surface they are embedded in or through their intersection.

An impact wrench will result from the first kind of change if the velocity causing the transition is not tangent to the new more constrained surface. Impacts result from transition velocities which violate the constraints of the new surface. For example, to avoid an impact wrench a point in $\Re^3$ which is going to contact a plane must have a crossing velocity which lies entirely in the plane. This sort of motion can be seen as a limiting condition on a smooth transition motion. This can occur when two constraint manifolds smoothly blend into each other.

The second type of transition may or may not result in an apparent impact wrench. The same ideal point will not experience an impact when it leaves the plane, because the higher dimensional space can accommodate the entrance velocity. However, any real robot has some built up potential spring energy in the sensor structure, the contacting linkage, and the robot joints. During contact, this energy is being contained by the constraint surface. If contact is lost abruptly, a contact loss impact occurs because of the sudden release of this energy. Abruptness is a function of the bandwidth and damping of the sensor and the speed of the transition.

For the first type of transition we need to distinguish between stable and unstable transitions. An example of an unstable transition is an object being pressed into a surface and then dragged along that surface. When the orientation of the surface discontinuously changes (i.e. we cross an edge), a sudden change in the orientation causes the object to lose contact with the surface and a loss impact results.

A transition is stable if the transition into the intersection between two manifolds remains in the intersection instead of continuing onto the next manifold. To formalize this let $\mathcal{M}_i$ and $\mathcal{M}_j$ be two constraints in the configuration space with non-empty intersection $\mathcal{M}_{i,j} = \mathcal{M}_i \bigcap \mathcal{M}_j$. Let their co-tangent bundles be defined so that the set of wrenches they can support is given by the positive convex combination over the basis vectors in the co-tangent space and the disturbance vectors. The set of wrenches supported on $\mathcal{M}_i$ at $\mathbf{x}$ is then $\mathrm{Sup}_i(\mathbf{x}) = \mathrm{Convex}(\mathcal{T}_i^*(x), \mathrm{Range}[\mathbf{w}_d](x))$. The range value models the friction component which also depends upon the applied wrench.

A state $\mathbf{x}$ and control $\mathbf{v}_d$ will be totally stable, resulting in no motion, if the applied control can be entirely supported by the constraint

$$\mathbf{K}_r\tilde{\mathbf{x}} - \mathbf{B}_r\mathbf{v}_d \in \mathrm{Sup}_i(\mathbf{x}). \tag{4.45}$$

A less strict form of stability is that the control plus the resulting constraints result in a motion in the tangent space of the intersection manifold $\mathcal{M}_{i,j}$. This definition is more useful because it states that motions on a manifold stay on a manifold. This will be the case if at state $\mathbf{x}$ and control $(\mathbf{x}_d, \mathbf{v}_d)$ there is a nonnegative solution for $\boldsymbol{\lambda}$ in

$$(\mathbf{C_x}^T\mathbf{B}_r^{-1}\mathbf{C_x})\boldsymbol{\lambda} = \mathbf{C_x}^T\mathbf{B}_r^{-1}(-\mathbf{K}_r\tilde{\mathbf{x}} + \mathbf{B}_r\mathbf{v}_d + \mathrm{E}[\mathbf{w}_d]). \tag{4.46}$$

Such a triple $(\mathbf{x}, \mathbf{x}_d, \mathbf{v}_d)$ will be called manifold stable.

In order for a transition from $\mathcal{M}_i$ to $\mathcal{M}_j$ to remain on the intersection, several conditions must hold. First, the control $\mathbf{v}_d$ and the transition state $\mathbf{x}$ must be manifold stable on $\mathcal{M}_{i,j}$. Second, the control and state must be only manifold stable on $\mathcal{M}_i$. If this condition does not hold, the robot will become stuck on $\mathcal{M}_i$ and will never transition. Lastly, the velocity that results from the control on $\mathcal{M}_i$ must have a component that causes motion toward the intersection.

In an ideal stable impact the momentum changes discontinuously to bring the new velocity into the new tangent space. Let the initial manifold be $\mathcal{M}_i$ and the new manifold be $\mathcal{M}_j$. Let $\mathbf{v}_{new}$ be the velocity after transition and the velocity before transition be $\mathbf{v}_{trans}$. $\mathbf{v}_{new}$ must lie in the tangent space of $\mathcal{M}_{i,j}$ so that $\mathbf{C}_{i,j}(\mathbf{x})^T\mathbf{v}_{new} = 0$. The change in momentum is caused by an impulse which lies entirely in the cotangent space of $\mathcal{M}_j$. The equation for the change in momentum is

$$\mathbf{H}_t(\mathbf{x})\mathbf{v}_{new} = \mathbf{H}_t(\mathbf{x})\mathbf{v}_{trans} + \mathbf{C}_j(\mathbf{x})\mathbf{k}. \tag{4.47}$$

Using this relationship, the constraint gives

$$\begin{aligned} \mathbf{k} &= -(\mathbf{C}_j^T\mathbf{H}_t\mathbf{C}_j)^{-1}\mathbf{C}_j^T\mathbf{H}_t^{-1}\mathbf{p}_{old} && (4.48)\\ \mathbf{w}_{impulse} &= \mathbf{C}_j\mathbf{k} && (4.49)\\ \mathbf{p}_{new} &= (\mathbf{Id} - \mathbf{C}_j(\mathbf{C}_j^T\mathbf{H}_t\mathbf{C}_j)^{-1}\mathbf{C}_j^T\mathbf{H}_t^{-1})\mathbf{p}_{old} && (4.50) \end{aligned}$$

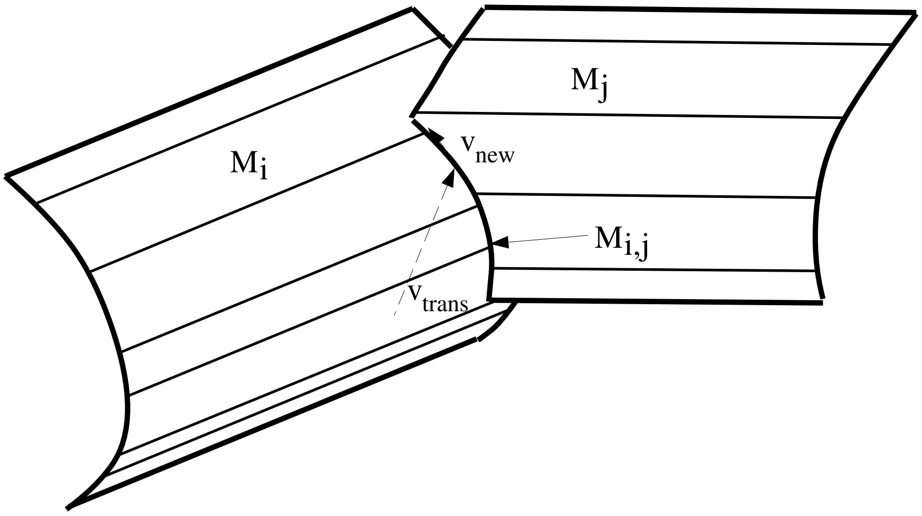


Figure 4.9: Impact transition from manifold $\mathcal{M}_i$ to manifold $\mathcal{M}_{i,j}$.

for the impulse force and new momentum. The impulse force is generated over a small window of time

$$\mathbf{w}_{impulse} = \int_t^{t+\delta t} \mathrm{E}[\mathbf{w}(t)]\mathrm{d}t\,. \tag{4.51}$$

The impact wrench adds into the measured wrench. The idealized model for E[$\mathbf{w}$] is a delta function at time t. However, in actuality the workpiece bounces several times, in a complex way, before settling onto the new constraint. Energy is lost on every bounce due to damping in the robot and at the interface. The integral can be computed from the data if the beginning and end of the impact can be identified. Chapter 7 discusses this identification problem. The change in momentum can be used to help determine the new constraint manifold.

For an unstable impact, there is a release in potential energy. The change in the average force stored in the spring is

$$\mathbf{K}(\mathrm{E}[\tilde{\mathbf{x}}]_{after} - \mathrm{E}[\tilde{\mathbf{x}}]_{before}) = \mathrm{E}[\mathbf{C}_{\mathbf{x}}\boldsymbol{\lambda}_{before}] - \mathrm{E}[\mathbf{C}_{\mathbf{x}}\boldsymbol{\lambda}_{after}]. \tag{4.52}$$

This difference in wrench adds an impulse wrench into the measured wrench. The impulse causes a transient response which slowly dies out. The direction of change can be found by identifying the peak wrench.

4.4 Conclusion

This chapter formulated the constraints and dynamics of rigid bodies in the configuration space of allowed motions. The configuration space was shown to have an intrinsic topology graph. Each node in the graph represented a region of constant dynamics. The dynamics were shown to depend critically on the type and geometric description of the constraint. It was shown that different geometric descriptions could result in the same constraint.

For constant velocity motions on a single configuration space manifold, the stationary solution to the expected value of the motion was shown to take the form of a point sliding on the manifold pulled by a spring and damper. The statistics of the motion and the measured wrench were shown to depend upon not only the control and contact disturbances, but also on the direction and magnitude of the motion.

Transitions between nodes in the graph were shown to correspond to transitory events. Transitions to manifolds with more constraint were shown to be either stable or unstable. Stable transitions almost always result in impacts. Unstable transitions result in loss impacts if there is stored elastic energy.

A Contact Feature Model

Chapter 5

The core contribution of this thesis is an approach to building an observer for manipulation. The goal of any *manipulation observer* is to estimate the pose and relative motion of salient objects in a timely fashion. Observers can be built using a complete C-space representation [Robles, 1995], or local threshold tests on the signals can be designed to trigger events and, in effect, observe the relevant information.

A C-space approach is powerful. It represents all of the relevant information and provides a convenient geometric framework for information fusion. On the other hand, current C-space representations become computationally prohibitive and memory intensive for high dimensional systems. The potentially high computational cost may make it difficult to produce timely results. Local event triggers have the opposite properties. They are quick to make decisions, but do not provide a global measurement fusion framework.

This chapter discusses a model based approach that blends the two representations. We use models of the measurement process to produce *contact feature* signal descriptions. Each of these models provides a statistical description of the measurement process applicable to a small patch of phase-space. A fast, robust, local test can then be derived from the model using statistical decision theory.

Global knowledge is incorporated by generating a graph of possible contact features. This graph can be either programmed for local tasks, computed from geometry and first principles, or learned for repetitive tasks. Essentially, a local test determines that there has been a change in the contact conditions and the graph determines the possible new models.

This approach can produce timely results and incorporate global context. In addition, since change tests are derived from models, the assumptions made in picking a test threshold are explicit. Regions of C-space are lumped together in the model. This has the advantage that during a motion signals arising from a particular model will tend to persist. This makes it easier to detect the model and track the motion of the robot. However, geometric information about length and shape is not represented and is not available to the observer.

A later chapter looks at features that are relevant to the mating and assembly phase of rigid body manipulation. The ideas can be extended to other phases of manipulation, but this was not experimentally explored during the course of this work. This chapter will discuss some possible ways of applying the ideas to other areas of manipulation such as grasp control.

This chapter defines a *contact feature* and discusses what is and what is not a feature. We then discuss how contact features can be connected to form a graph called *feature graph*. The following chapter then presents an observer designed to work with the feature graph to estimate the current state of the manipulation task. The remaining chapters in this thesis develop the estimation algorithms for the basic features that arise in environments composed of rigid bodies. Two basic features are developed: 1) the constraints on motion caused by contacts, and 2) the temporal effects caused by changes in constraint or velocity dependent temporal characteristics of contact forces.

5.1 Definition of a Contact Feature

A useful definition of *contact feature* is needed to make our model based approach precise. A contact feature is

- a statistical parameterized measurement model,
- which is applicable at every point in phase-space,
- and which partitions phase-space into a countable number of sets.

A statistical representation makes it possible to apply the powerful machinery of statistical decision and estimation theory to determining the current model. The second requirement makes it possible to compare signal measurement histories.

The last requirement is the most important. This requirement ensures that a discrete graph will actually result from a given model and rules out some common models. This requirement is a function of both the chosen measurement model and the underlying phase-space. Models that are features in one phase-space may not be features in another phase-space.

We need to be able to take common measurement contact models and make them parameterized , statistical models. The parameters of any model will be indicated by $\boldsymbol{\theta}$. Most common measurement models can be placed in this form.

For example, the measurement of the position of the robot $\mathbf{y}$ might be described by

$$\mathbf{y}(t) = \mathbf{x}(t) + \boldsymbol{\nu}(t) \tag{5.1}$$

where $\boldsymbol{\nu}(t)$ is an independent, identically distributed (i.i.d.) normal process with zero mean and variance $\mathbf{V}$:

$$\boldsymbol{\nu}(t) \sim \text{i.i.d. } \mathrm{N}(0, \mathbf{V}). \tag{5.2}$$

The parameters of the distribution for $\mathbf{y}(t)$ are $\mathbf{x}(t)$ and $\mathbf{V}$. [1]

Friction for a point robot on a plane can be put in this form. Relative motion of the robot breaks the model into two possible cases. For the fixed position case, a model for the measured force $\mathbf{w}_m(t) = (f_x(t), f_y(t))$ is that it has a independent, identically distributed, maximum entropy distribution over the friction cone specified by a fixed normal $\hat{\mathbf{n}}$ and coefficient of friction μ. For the moving case, a model for the measured force is the distribution

$$f_x(t) - \mu f_y(t)\,\mathrm{sgn}(\dot{x}) \quad = \quad \nu(t) \tag{5.3}$$

$$\nu(t) \sim \text{i.i.d. } \mathrm{N}(0, V). \tag{5.4}$$

Therefore, in the friction model case the first parameter of the distribution for $\mathbf{w}_m$ is an index giving the appropriate case. The rest of the parameters fix the required probability distribution.

Parameterized statistical models can also be formed from constraint equations between the measured/grasped part and other objects in the environment. This is a very important class of model. Constraint equations can always be written in the

[1] It will often be the case that only conditions on probability distribution are available. In that case, we will adopt the maximum entropy distribution as the underlying distribution [Cover and Thomas, 1991]. The maximum entropy distribution for a real random variable when the mean and variance are fixed is the normal distribution. The maximum entropy distribution for a real bounded random variable is the uniform distribution.

form $\mathbf{C}(\mathbf{x}, \mathbf{g}) = 0$ which requires that the robot's phase $\mathbf{x}$ satisfy the constraint parameterized by the geometric parameters $\mathbf{g}$. In general, constraints hold only for a range of configurations. Free space is the null constraint equation.

The constraint equation can be turned into a measurement equation by projecting the measured force onto the tangent space and projecting the measured velocity on to the cotangent space (at a configuration). In a perfect system without any noise, modeling error, or friction these projections would yield zero. In the real world, the projection will be nonzero and the result can be modeled as an independent noise process:

$$\mathbf{C}^*_{\mathbf{x}}(\mathbf{x})^T \, \mathbf{w}_m = \boldsymbol{\nu}_{\mathbf{w}} \tag{5.5}$$
$$\mathbf{C}_{\mathbf{x}}(\mathbf{x})^T \, \dot{\mathbf{x}} = \boldsymbol{\nu}_{\dot{\mathbf{x}}} \tag{5.6}$$

where $\mathbf{C}^*_{\mathbf{x}}(\mathbf{x})$ is a basis for the constraint cotangent space, $\mathbf{C}_{\mathbf{x}}(\mathbf{x})$ is a basis for the constraint tangent space, $\mathbf{w}_m$ is the measured force, $\dot{\mathbf{x}}$ is the measured velocity, and $\boldsymbol{\nu}_{\mathbf{w}}$ and $\boldsymbol{\nu}_{\dot{\mathbf{x}}}$ are independent measurement processes. We use zero mean normal distributions for the measurement error statistics. The statistics of the measurement processes may or may not depend on the velocity. In either case, if we have a model for how the statistics depend on the velocity, a model which is applicable everywhere can be generated by setting the first parameter to be the constraint model type and letting the remaining parameters specify the geometric parameters and the measurement statistics.

We now formalize the definition of contact feature. The measurements $\mathbf{y}(t)$ (the force, position, both, or something else entirely) are assumed to come from a model of the form

$$\mathbf{y}(t) \sim \text{i.i.d.}\, \mathbf{h}(\mathbf{x}(t), \dot{\mathbf{x}}(t), \boldsymbol{\theta}) \tag{5.7}$$

All of the previous models are of this type. Now for a pair $(\mathbf{x}, \dot{\mathbf{x}})$ in the phase-space, let $M(\boldsymbol{\theta})$ be an instance of model M which applies to this pair. The instance is specified by the value of $\boldsymbol{\theta}$. Models are defined so that only one instance applies to any one pair. The set of all points where a given model applies is the domain of definition of the model when specialized by fixing the parameter value.

The set of all possible domains, for all possible values of $\boldsymbol{\theta}$, forms a partition of the phase-space. A *stationary feature* will be any model which produces a *countable* partition of the phase-space. The domain associated with each instance of the parameter is defined as the *feature instance partition*, and the feature model specialized by $\boldsymbol{\theta}$ is a *feature instance*.

The fixed normal friction model for the measured force is a feature for a point robot in a countable polyhedral world. The first parameter, the model type, partitions the phase-space into $(\mathcal{C}, \dot{\mathrm{x}} = 0)$ and $(\mathcal{C}, \dot{\mathrm{x}} \neq 0)$. If $\dot{\mathrm{x}} = 0$, the rest of the parameters are the vectors describing the convex cone. If $\dot{\mathrm{x}} \neq 0$, the rest of the parameters are the parameters of the cone and the statistics of the model error. Any one cone applies to an entire face, edge, or vertex of the polyhedral obstacles. Since there are a countable number of such geometric features there are a countable number of partitions for $\dot{\mathrm{x}} = 0$. If the number of model error variances is also countable, or they can just be lumped into a single statistical model, then there are a countable number of partitions for $\dot{\mathrm{x}} \neq 0$. Therefore, cones are a feature for this phase-space.

However, the fixed angle friction cone model is not a feature for a polyhedral object in a polyhedral world. This is because the direction of the friction cone depends upon the orientation of the moving object. Therefore the domains depend upon the orientation, which yields an uncountable partition. This emphasizes that the feature definition depends both on the statistical model and the underlying phase-space and implicitly the configuration space.

Contact constraints models are features for polyhedral and curved environments. This is clear because there are only a few contact constraint types for contacts between polyhedral objects. Each of these types corresponds to one of the partitions. In fact it is clear that constraints are a feature for any rigid world consisting of piecewise smooth surfaces. Since this is a very broad array of objects, constraints are obviously a very important class of feature. The contact constraint models can be extended to incorporate friction and this in essence makes the friction cone normal depend upon the configuration.

Our definition of a feature can be contrasted with an alternative concept of *disambiguous* sets [Buckley, 1987]. We will discuss the complementary sets the *confuseable* sets and restrict our attention to partitions of the configuration space. The argument also applies to phase-space. Confuseable sets are defined by defining two points in configuration space as equivalent if they are both consistent with a given measurement. That is both points are in the same interpretation set for a given measurement. The extension is that two points are equivalent if it is possible to generate a measurement which will place them in the same interpretation set. Essentially, a pair of points in the configuration space $(\mathbf{x}_1, \mathbf{x}_2)$ is confuseable if it is possible to generate a measurement for which we cannot decide with certainty if the measurement came from $\mathbf{x}_1$ or $\mathbf{x}_2$. It remains to test if this is an equivalence relation.

Lets see how the confuseable set definitions works for fixed angle friction cones in a polygonal world. Clearly all the points on one edge are confuseable, because they all have the same friction cone. The points on two edges which form a vertex will not be confuseable if their relative angles is greater than the friction cone angle.

Now suppose the world has a circle that the point can contact. Select three points $\mathbf{x}_1, \mathbf{x}_2, \mathbf{x}_3$ on the circle so that the friction cones of $\mathbf{x}_1$ and $\mathbf{x}_2$ share some directions, and the friction cones of $\mathbf{x}_2$ and $\mathbf{x}_3$ also share some directions, but $\mathbf{x}_1$ and $\mathbf{x}_3$ do not. By definition $\mathbf{x}_1$ and $\mathbf{x}_2$ are confuseable and so are $\mathbf{x}_2$ and $\mathbf{x}_3$. Now in order for confuseability to be equivalence relation $\mathbf{x}_1$ must be confuseable with $\mathbf{x}_3$, but by construction it is not. Therefore confuseability is not an equivalence relation because the relation is not transitive, and therefore it cannot be used for constructing a graph. Clearly this problem also arises for polygonal objects in two dimensions. The model equivalence definition survives the test because every point is only equivalent to itself. Of course, this implies that fixed normal model of friction is not a feature in configuration spaces with curved surfaces.

The fact that sensor noise can destroy transitivity in the definition of confuseable sets was recognized in [Donald and Jennings, 1991]. Their work defined a configuration space perceptual equivalence sets in a manner similar to disambiguous sets.

The essential difficulty is that ambiguity or disambiguity is too strong a constraint to place on the problem. Disambiguity requires two sets to be uniquely separable with one measurement. Probabilistic measurement models provide only the possibility of drawing increasingly more likely conclusions with additional data but certainty is never guaranteed.

Features can be combined. Many possible features could exist for a given set of measurements and task. For example, a tactile array sensor could be used to compute the contact curvature, the contact stiffness, and possibly relative motion all of which are features. Each feature provides a different partition of the phase-space. The intersections of these partitions is again a countable partition, so a collection of features is again a feature. Combining features yields a finer partion of phase-space and in general this helps in both identifying the current state of the robot and in selecting the appropriate control action.

Nonstationary models are much more difficult to handle. Impacts will be the only nonstationary model needed for this thesis, because the only nonstationary events that will arise in this work are transitions between contact constraints. We will consider this type of model to be a *transition feature* because impacts occur only when transitioning from one stationary feature to another stationary feature.

5.2 Properties of Useful Features

The last section provided a definition of a feature. A constructive method of determining available features for a phase-space and dynamics is not yet available. However, we can comment generally on what makes a good feature.

The fundamental assumption behind the feature approach to programming is that a single continuous control law will exist for each feature instance. Features should have this property. The exact parameters of the control law may then depend on the other non-feature properties of the configuration space. However, the parameters are assumed to vary smoothly within the partition for a single feature instance. For example, the form of the hybrid force control law is determined by the form of the constraints which is a feature. The direction of constraint and free motion is then determined by the current contact location.

Although we can test this property given a control law and feature definition, we cannot use this property as a guide to feature selection because of the interplay between control strategy and the feature choice. Considering both together would place us back in the pre-image framework. A simpler test is to choose features that are adapted to the dynamics. As discussed in the last chapter, the dynamic equations of the complete system generally take on different discrete forms depending upon the contact conditions. For a single rigid body, the dynamics change whenever a contact constraint is made or broken. The different forms for the dynamic equations also create a partition of the phase-space. A good feature would produce a partition which is adapted to the partition created by the dynamics. That is, given a feature instance, the form of the dynamic equations should be uniquely determined.

The assumption behind this requirement is that a complete control algorithm can be written for accomplishing the task by analyzing the dynamics for the different possible dynamic equations. We assume that a local controller can be written for each dynamic partition which will locally keep the robot on a desired path. This approach was shown to work for pushing by [Narasimhan, 1994].

Finally, it should be relatively easy to determine, at least pointwise, feature membership for points in phase-space. It would be even better if it was possible to compute the feature domains.

5.3 Encoding Tasks with Features

In order to accomplish a task, the partitions induced by the features have to be organized for the given task. There are many possible organizations. We will discuss two possibilities: feature connectivity based on topologically connectivity, and feature connectivity based on forward projection. Both definitions yield a feature graph. The graphs are based on the underlying connectivity of configuration space. A graph is produced which can be used to infer information about the state of the manipulation task. Which organization is most useful for a given task depends upon the complexity of the problem at hand.

5.3.1 Connectivity from Topological Path Connectivity

As discussed in the last section, a collection of stationary features creates by definition a partition of the configuration space. A graph of this partition can be constructed by using the tangent space connectivity of the underlying configuration space. We identify a graph node with each element of the partition. Let the set of all nodes be $\mathcal{N}$. Every node $\mathcal{N}_j$ can be associated with a label, and a vector of parameters for that node, and its associated partition

$$\mathcal{N}_j = (j, \boldsymbol{\theta}_{k_j}, \mathcal{P}_j) \tag{5.8}$$

where j is the label, k_j is an index into the set of possible model instances, and $\mathcal{P}_j$ is the partition associated with the model instance. For any pair of nodes in $\mathcal{N}$ create an edge connecting the nodes if there exists a configuration in the first node and a configuration in the second node which are connected in the underlying configuration space. Let $\pi_{\mathbf{x}}$ be the natural projection from phase-space to configuration space, and let $\pi_{\dot{\mathbf{x}}}$ be the natural projection from phase-space to the tangent bundle. Then we have the following definition.

Definition 5.1 The edge $\mathcal{E}(i,j)$ exists if there exist $\mathbf{x}_0 \in \pi_{\mathbf{x}}(\mathcal{P}_i)$, $\mathbf{x}_1 \in \pi_{\mathbf{x}}(\mathcal{P}_j)$, and an integral curve $\sigma(t)$ such that:

1. $\sigma(0) = \mathbf{x}_0$, and $\sigma(1) = \mathbf{x}_1$,

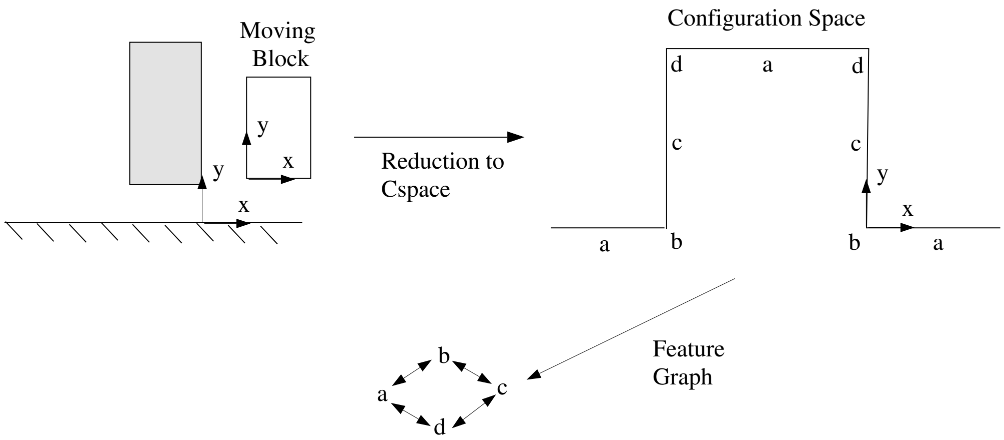


Figure 5.1: A feature graph based on path connectivity. The moving block is constrained to move rectilinearly in x and y. The force sensor measures the x and y contact force, so the contact feature model is determined by the contact normal. Each configuration space surface which shares the same contact model has been indicated with the same letter. This induces the indicated configuration space with four different contact constraint models. The partition for each model consists of disjoint domains in configuration space. The graph is induced by the connectivity of the domains.

2. $\sigma(t) \subset \pi_{\mathbf{x}}(\mathcal{P}_i) \bigcup \pi_{\mathbf{x}}(\mathcal{P}_j)$,

3. and $\dot{\sigma}(t) = \sum_{i=1}^{m} \alpha_i(t) B_i(\sigma(t))$, where $\{B_i(\mathbf{x})\}$ is a basis for the tangent space $\mathcal{T}_{\mathbf{x}}$.

Let the collection of all such edges be $\mathcal{E}$. The resulting graph $\mathcal{G} = (\mathcal{N}, \mathcal{E})$ is called the feature graph. This notion of feature connectivity produces a very coarse graph, because it lumps unconnected components of the phase-space together. However, it can be used to accomplish useful tasks, if we limit the size of the graph by using spatial locality.

Spatial locality limits the range of the phase-space under consideration to a particular volume. This volume is chosen because the system knows the robot is in the particular region from other measurements or prior knowledge.

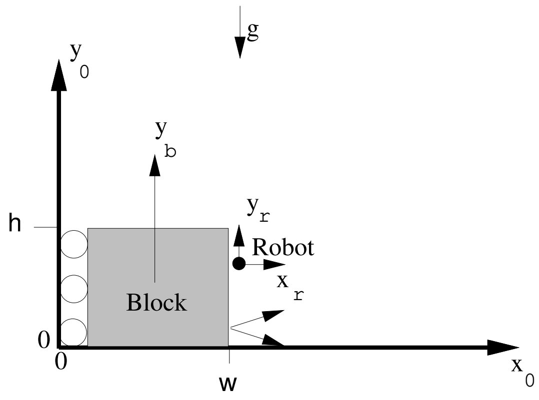


Figure 5.2: An idealized grasp maintenance problem. The rollers are only to indicate constrained sliding motion. We assume that the robot is touching the block for this tasks so that $x_r = w$. With this simplification, the configuration of the parts are specified by the height of the robot y_r and the height of the block y_b measured with respect to the global coordinates. The configuration space is therefore the set of pairs (y_r, y_b). The robot task is to lift and replace the block.

A finer local partition can be constructed by splitting each node in the graph into components which are topologically path connected in the phase-space. In general computing if two sets are topologically connected is difficult.

The nodes created by this splitting operation will no longer be uniquely labeled by the feature instance for the points. Instead any feature instance will correspond to a collection of nodes. In figure 5.1 the single node representing feature a would become three nodes. However, if we can assume that the robot remains in a local subset of the graph, we can prune the extra nodes. The resulting local feature graph can be used for controlling the robot in a small region of phase-space.

This notion of connectivity, along with locality, can be used as a basis for finite state based control. A grasp maintenance problem shown in figure 5.2. From locality the robot is known to be in contact with the block. Only lifting and slip need to be controlled. Therefore, the configuration space for this problem is the pair (y_r, y_b). The stable contact configurations which can occur during grasping are

$$\begin{aligned} \mathcal{C}_1 &= \{y_r \in y_b + [-h/2, h/2] \text{ and } y_b = h/2\} \\ \mathcal{C}_2 &= \{y_r \in y_b + [-h/2, h/2] \text{ and } y_b > h/2\} \end{aligned}$$

These two pieces of configuration space are clearly connected to each other.

The features of slip and contact force based on friction can be used to control the grasp. We assume the robot can sense vibration and force. The construction starts by defining the features for the vibration v and force $\mathbf{w}_m$ measurements, and then each feature is associated with its partition of the phase-space. The measured force is the reaction of the environment to the force applied by the robot. We consider only partitions of $\mathcal{C}_1$ and $\mathcal{C}_2$

Slip is indicated by the magnitude of the vibration. There will be a higher vibration level when the block is slipping then when it is not slipping. Vibration can be sensed in a number of ways. One way is to look at the magnitude of the energy in the derivative of the force signal. Another is to measure the magnitude of an acceleration signal, or the magnitude of stress rate sensor such as a piezoelectric film.

The measured vibration when there is no relative motion can then be modeled as a normally distributed variable given by a calibrated mean and variance. So under the no-slip condition, the vibration measurement has the model

$$v(t) \sim \text{i.i.d.}\, \mathrm{N}(\mu_0, V_{v1}),$$

where N is the normal probability distribution.

The statistics for when the sensor is slipping can either be calibrated, or an exponential model (given a variance) could be used for the difference between the measured signal and the no slip vibration mean. This model for the vibration signal under the slip condition is

$$(v(t) - \mu_0)/V_{v2} \sim \text{i.i.d.}\, \mathrm{Exp}$$

where Exp is the unit exponential distribution

$$\mathrm{Exp}(x) = e^{-x}$$

The measured contact force has three possible states: contact, lifted, and falling. When the robot is in contact, but has not yet lifted the block, the force must lie in the friction cone. The distribution of the force can be taken as the maximum entropy distribution over the cone. When the block is lifted, the y force must have a mean equal to the weight, and the x force must be sufficiently large to support the y force through friction. Finally, if the block is falling, the x and y forces must be smaller than the required lifting forces. The weight of the block is f_W a positive number, and $\mathbf{w}_m$ is the measured force.

$$
\begin{array}{lll}
\text{Force :} & & \\
F_1 & \mathbf{w}_m(t) \sim \text{i.i.d.}\,\text{ME}(|f_y| < \mu f_x, f_y > -f_W, f_x > 0) & \Rightarrow \text{Contact} \\
F_2 & f_y \sim \text{i.i.d.}\,\text{N}(-f_W, V_{f2}), f_x \sim \text{i.i.d.}\,\text{ME}(f_x > f_W/\mu) & \Rightarrow \text{Lifted} \\
F_3 & \mathbf{w}_m(t) \sim \text{i.i.d.}\,\text{ME}(|f_y| < \mu f_x, f_y > -f_W, 0 < f_x < f_W/\mu) & \Rightarrow \text{Falling}
\end{array}
$$

where ME is the maximum entropy distribution over the indicated set.

Each sensor partitions the phase-space into components. The partitions for each feature instance are:

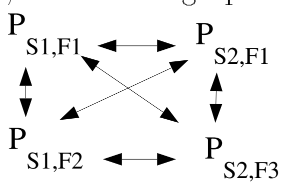
$$
\begin{aligned}
\mathcal{P}_{S_1} &= ((\mathcal{C}_1, (\dot{y}_r = \dot{y}_b)), (\mathcal{C}_2, (\dot{y}_r = \dot{y}_b))) \\
\mathcal{P}_{S_2} &= ((\mathcal{C}_1, (\dot{y}_r \neq \dot{y}_b)), (\mathcal{C}_2, (\dot{y}_r \neq \dot{y}_b))) \\
\mathcal{P}_{F_1} &= (\mathcal{C}_1, (\dot{y}_b \,\text{and}\, \dot{y}_r \text{unconstrained})) \\
\mathcal{P}_{F_2} &= (\mathcal{C}_2, (\dot{y}_r = \dot{y}_b)) \\
\mathcal{P}_{F_3} &= (\mathcal{C}_2, (\dot{y}_r \neq \dot{y}_b))
\end{aligned}
$$

Now consider the intersections of the different feature instance partitions.

$$
\begin{aligned}
\mathcal{P}_{S_1,F_1} &= (\mathcal{C}_1, (\dot{y}_r = \dot{y}_b)) \\
\mathcal{P}_{S_1,F_2} &= (\mathcal{C}_2, (\dot{y}_r = \dot{y}_b)) \\
\mathcal{P}_{S_1,F_3} &= (\mathcal{C}_1, (\dot{y}_r = \dot{y}_b)) \\
\mathcal{P}_{S_2,F_1} &= \phi \\
\mathcal{P}_{S_2,F_2} &= \phi \\
\mathcal{P}_{S_2,F_3} &= (\mathcal{C}_2, (\dot{y}_r \neq \dot{y}_b))
\end{aligned}
$$

where ϕ indicates the empty set.

Now apply definition 5.1 to connect the partitions. Since $\mathcal{C}_1$ is on the boundary of $\mathcal{C}_2$ and there exist velocities on both manifolds that will cause transition between the manifolds all the the features are connected to each other. The definitions only requires that points in configuration space be connected not that the velocities match in the two partitions. Therefore, the feature graph is



Since each feature in this graph is unique, observing the current feature instances is trivially equivalent to determine the state of the robot in the feature graph. This will be sufficient if the feature instance is enough to determine the appropriate control algorithm. This will be the case for essentially differentially local problems, such as this grasping example. In other words, if the control problem can be broken down into: 1) finding a finite set of local controllers, 2) selecting the appropriate controller based on a unique feature instance, and 3) the controllers will keep the robot within its local domain of definition, then this is a good representation. When the control law also depends upon the location of the feature instance in the configuration space, this representation is not sufficient.

The graph can be extended to incorporate transient features. Every transient feature corresponds to an edge in the initial task feature graph. Replace every such edge with an additional node labeled with the transient feature. If the transient feature is guaranteed to occur on transition from the initial to final feature node, connect the initial node to the transient node, and then the transient node to the final node. If it is not guaranteed to occur also connect an edge from the initial node to the final node.

As a very simple example, of transient features, consider the feature of the existence of contact. A simple contact model is

$$\mathbf{w}_m(t) \sim \text{i.i.d. } \mathrm{N}(0, \mathbf{V}_1) \quad \text{for forces in free space} \tag{5.9}$$

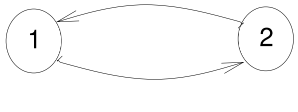
$$\mathbf{w}_m(t) \sim \text{i.i.d. } \mathrm{U}(\text{Measurable Forces}) \quad \text{for forces in contact} \tag{5.10}$$

where U is the uniform probability distribution over the given set.

The two equivalence classes for this model are the free space $\mathcal{F}$ and its associated tangent space $\mathcal{T}_{\mathcal{F}}$, and the configuration space obstacle surface $\mathcal{O}$ and its associated tangent space $\mathcal{T}_{\mathcal{O}}$. Therefore, the two equivalence classes in the partition of phase-space for this model are

$$\{(\mathcal{F}, \mathcal{T}_{\mathcal{F}}), (\mathcal{O}, \mathcal{T}_{\mathcal{O}})\}.$$

Labeling the first element 1 and the second as 2, gives the feature graph



If impact transient features also sometimes occur when going from free space to contact, the feature graph becomes

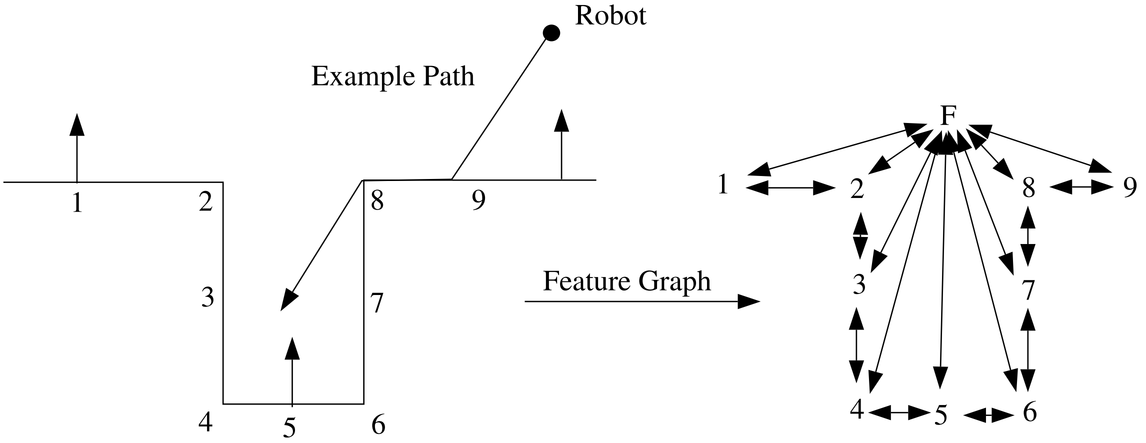
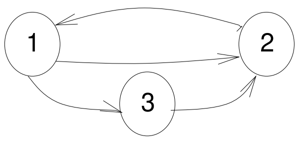


Figure 5.3: Example path and feature graph for a point robot and a two dimensional hole. Although the path is the correct path, an observer based on the feature graph shown at the right will be unable to recognize termination because the transition from 8 back into free space will reset the observation process. The second time the upward normal is observered, the observer will report it as possibly coming from 1, 5, or 9.



where 3 is the transient feature node. In this model, the impact node can only be visited when transitioning from free space to contact. In some cases, there may be sufficient stored energy in the sensor so that a release impact occurs when transitioning from contact back to free space.

5.3.2 Adding Actions to the Graph

The local approach of linking feature models based on phase-space connectivity can fail in a larger context. The approach can fail because the models become too connected, and therefore the graph provides insufficient constraint. This is illustrated in figure 5.3.

The starting configuration of the robot is above and to the right of the hole (feature F). A downward command is applied and the robot contacts the right of the hole (feature 9). The local feature graph indicates that three features exist which have the same normal: the side to the left of the hole (feature 9), the side to the right of the hole (feature 1), and the bottom of the hole (feature 5). The observer will return all three features and report them as equally probable.

Now the robot is commanded to move to the left. This causes the robot to cross the edge and transition back into free space (feature 8 followed by F). The transition returns the observer to its starting condition of free space (feature F), the additional information that was gathered by moving and losing contact with the edge is not represented in the local feature graph. Therefore, a decision algorithm cannot tell from this representation that the robot is actually closer to the hole and that the next time the vertical normal is observered it must be from feature 5. A richer representation is required for this type of task.

The example also suggests a solution. Predictions about the consequences of a given motion should be incorporated into the observer. Fortunately this can be done by making the feature graph dependent on the action. This is done formally by incorporating forward and back projection. It should be noted that computation of time projection of sets is a very hard task in general, and that the additional information being provided here by the action dependence is substantial [Canny, 1987].

In addition, the geometric parameters estimated for the constraint feature contain information about the pose of the contacted objects. The estimates of these parameters could be used to update an estimate of object configuration.

Forward projection is prediction or simulation of the possible outcomes of an action given the current estimate of the range of configurations for the robot and its environment. The dynamics, the current feedback local controller, and the configurations of all the objects define the equations of motion for the robot. These dynamics will hold until termination is signaled from a decision algorithm, which uses the probability estimates from the local feature observer. As long as a new trajectory and controller are not introduced, the motion of the robot can be represented with a single stochastic differential equation:

$$\mathrm{d}\mathbf{x} = n_i(\mathbf{x}(t), \mathbf{u}) + \mathrm{d}\mathbf{v}. \tag{5.11}$$

where n_i is the dynamics of the robot when in contact with points in feature node i, $\mathbf{u}$ is the control, and $\mathrm{d}\mathbf{v}$ is the stochastic control disturbance.

Forward projections are defined using equation 5.11 following [Erdmann, 1986]. The forward projection of a set $\mathcal{S}$ is the set of all points that could be reached by the dynamics given the control error. Formally

Definition 5.2 The *unbounded time forward projection* of a set $\mathcal{S}$ is

$$\mathcal{F}(\mathcal{S}) = \{\mathbf{x}' : \exists t > 0 \text{ and } \boldsymbol{\sigma}_{\mathbf{dv}(t)}(t) \text{ such that } \boldsymbol{\sigma}_{\mathbf{dv}(t)}(0) \in \mathcal{S} \text{ and } \boldsymbol{\sigma}_{\mathbf{dv}(t)}(t) = \mathbf{x}'\}$$

where $\boldsymbol{\sigma}_{\mathbf{dv}(t)}$ is a possible trajectory of the robot given a sequence of control disturbances.

Back projection is defined similarly. The *unbounded time weak back projection* of a set $\mathcal{S}$ is the set of all points that can reach $\mathcal{S}$.

Definition 5.3 The *unbounded time weak back projection* of a set $\mathcal{S}$ is

$$\mathcal{B}(\mathcal{S}) = \{\mathbf{x}' : \exists t > 0 \text{ and } \boldsymbol{\sigma}_{\mathbf{dv}(t)}(t) \text{ such that } \boldsymbol{\sigma}_{\mathbf{dv}(t)}(0) = \mathbf{x}' \text{ and } \boldsymbol{\sigma}_{\mathbf{dv}(t)}(t) \in \mathcal{S}\}$$

Now back and forward projection can be used to create a graph. The results depend upon the controller and the action selection procedure. We will discuss some approaches that can be used for limiting the scope of the computation based on the expected properties of the observer. We also suggest a point based method which only approximately, and probabilistically, computes the *feature forward projection.*

5.3.3 Equivalence under an Action

Back projection and forward projection can be used to refine the collection of feature equivalent sets. Let $\mathcal{P} = \{\mathcal{P}_i\}$ be a collection of feature partitions. For this section we will have to assume that the collection is finite. Finiteness is required because we will be considering a product set derived from the feature partitions. Each partition consists of points which have the same feature model and which are connected in the phase-space. To simplify notation here when referring to points in a partition $\mathcal{P}_i$, we mean the points in the configuration space associated with the partition.

Definition 5.4 Two points $\mathbf{x}_1$ and $\mathbf{x}_2$ are *equivalent under a single action* if

1. $\mathbf{x}_1$ and $\mathbf{x}_2$ are elements of the same partition $\mathcal{P}_i$.

2. Every partition $\mathcal{P}_j$ reachable from $\mathbf{x}_1$ is also reachable from $\mathbf{x}_2$.

3. Every partition $\mathcal{P}_j$ reachable from $\mathbf{x}_2$ is also reachable from $\mathbf{x}_1$.

The second and third requirements simply ensure that the same feature transitions can occur from both $\mathbf{x}_1$ and $\mathbf{x}_2$.

This definition can be related to weak back projections. For every partition $\mathcal{P}_i \in \mathcal{P}$, let $\mathcal{B}_i = \mathcal{B}(\mathcal{P}_i)$. That is $\mathcal{B}_i$ is the set of all points which could reach partition $\mathcal{P}_i$. Now let $\mathcal{I}$ be an index set over all possible indices and let $\bar{\mathcal{I}}$ be its complement in the indices. The index set partitions are defined as

$$\mathcal{P}_{\mathcal{I}} = (\bigcap_{i \in \mathcal{I}} \mathcal{B}_i) \backslash (\bigcup_{i \in \bar{\mathcal{I}}} \mathcal{B}_i),$$

i.e. a point is in a power set partition if and only if it can only reach the feature partitions in the index set. Therefore, any two points selected from the same power set partition can both reach exactly the same partitions and are therefore equivalent under a single action.

Figure 5.4 shows what this construction looks like for the classic cone model of uncertainty for generalized damper control [Lozano-Pérez *et al.*, 1984]. Generalized damper control only tracks the velocity command. The position is not used in the control law. Under this model of control, the path of the robot starting from any point will lie in a cone emanating from the point. Most of the power set partitions are empty. The first figure shows the original feature partitions and the cone of possible velocity commands. The figure to the right shows all the power set partitions that are not empty. The figure on the bottom shows how the power set partitions augment the original feature graph. The new nodes were connected using reachability.

The index set used for each power set partition are given below.

$$\begin{aligned}
\mathcal{F}_1 &= 1 \\
\mathcal{F}_2 &= 1,2,3,4 \\
\mathcal{F}_3 &= 1,2,3,4,5 \\
\mathcal{F}_4 &= 1,2,3,4,5,8,9 \\
\mathcal{F}_5 &= 3,4,5,8,9 \\
\mathcal{F}_6 &= 3,4,5 \\
\mathcal{F}_7 &= 3,4 \\
\mathcal{F}_8 &= 4,5
\end{aligned}$$

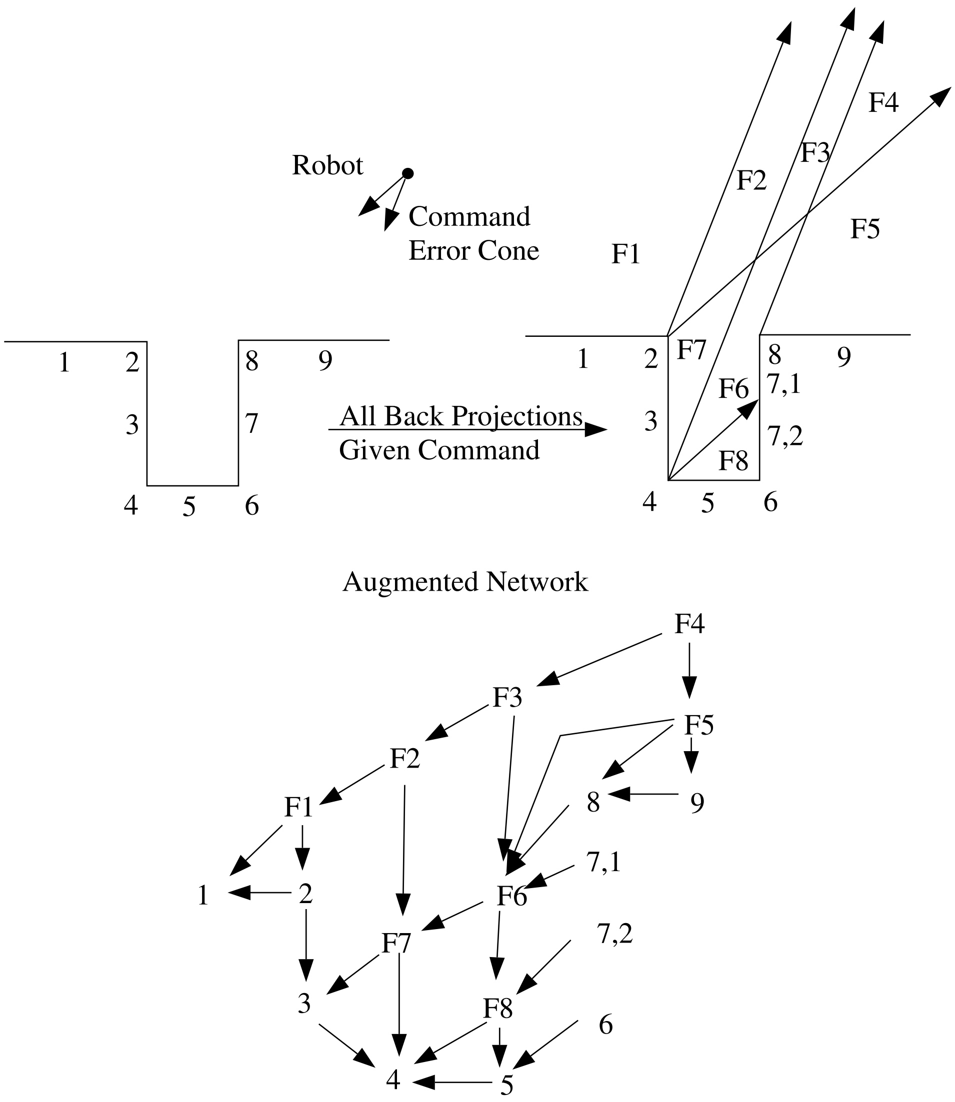


Figure 5.4: A planar assembly problem showing the back projections from each feature and the intersection of the back-projections and partitions. The control is generalized damper control with a cone model for the control error.

We have not connected partitions of free space that share only a vertex, because this transition has zero probability of occuring.

This new graph solves the original motivating problem illustrated in figure 5.3. Now when the object leaves feature 8, after sliding along 9, it will enter $\mathcal{F}_6$ instead of the entire free space. Therefore, the observer will know that the next contacted feature can only be 3, 4, 5 but not 9.

5.3.4 Forward Projection Graphs and LCNP

In LCNP (Local Control Nominal Path) [Narasimhan, 1994], the controller is assumed to keep the robot within an uncertainty ball about a nominal path using a stiffness type of control. The ball is at its maximum size in free space. Contact with a C-space surface projects the uncertainty ball onto the surface. The nature of the projection depends upon the model of friction. An uncertainty tube can be created by sweeping the maximum ball along the nominal path. This tube limits the range of feature possibilities and provides a control dependent feature connectivity.

Figure 5.5 shows an uncertainty ball and tube around a nominal path for a point in the plane. To construct a graph, begin with the collection of feature equivalence sets. Now keep only those points in each set which also lie in the uncertainty tube. Now use a compliance model for action to compute the action equivalent sets and their connectivity to compute the desired graph. Figure 5.6 shows the result.

A further approach is to remove the nominal path and consider the effect of an action everywhere in the space. The same definition can be used to produces a feature graph for every action. The collection of all such graphs will be a super-graph which is indexed by the selected action. We might use this super-graph to plan an appropriate action.

5.3.5 Graphs and Nonparameteric Interpretation Set Representations

For complex problems requiring a full knowledge representation, feature graphs might be computed on demand from the knowledge set and point-set simulation. Forward

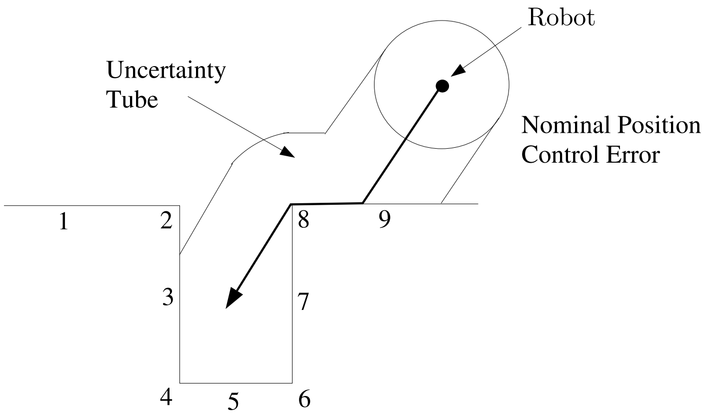


Figure 5.5: Nominal path and uncertainty tube for a LCNP peg-in-hole strategy for a point robot in a two dimensional configuration space.

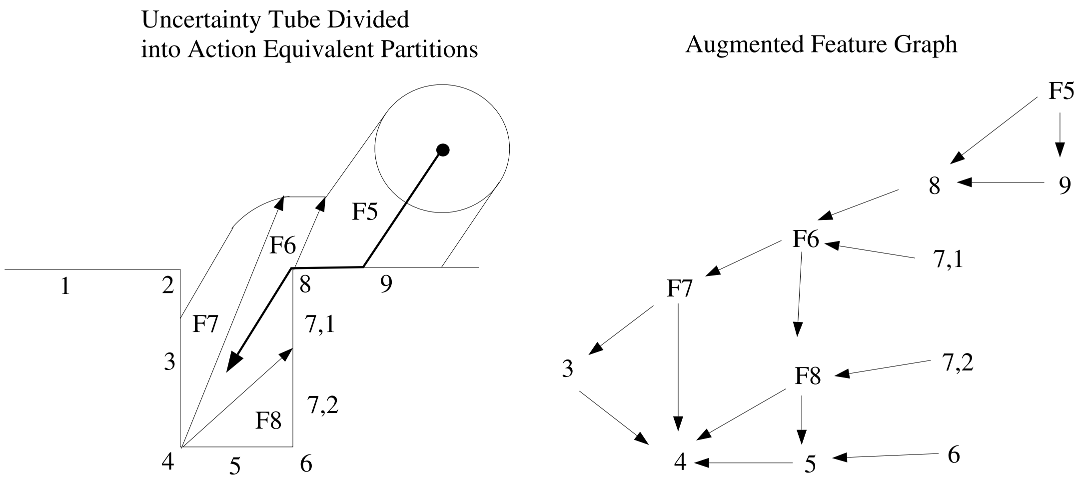


Figure 5.6: Action dependent graph for an LCNP insertion strategy.

projection of point sets in configuration space was addressed in the plane with rotation in [Caine, 1993]. These algorithms could be applied to computing the forward projection graph. We suggest, but have not implemented, the following Monte Carlo approach which does not require representing C-space explicitly.

Represent the possible configurations of every object using a non-parametric distribution. Each configuration is represented by a collection of n samples. Now randomly select a representative point from each of these distributions and do the forward simulation. Record which features come into contact and which feature models become active. Every simulation produce one possible forward projection trace and proceeds down one branch in the desired feature graph. Repeat the simulation until the features produced and the chance of hitting each feature becomes stably estimated.

The simulation is likely to become stable quickly. The most likely constraint features will be found quickly in the simulation. The very unlikely features will take a long time to detect via simulation, but these features are irrelevant because they are also unlikely to occur in practice. We now have the desired feature graph which can be used by the observer.

The same non-parametric distribution can also be used for incorporating information represented by the constraint parameter estimates. Every contact produces a constraint feature. The parameter estimates produced by the constraint estimator, after every new measurement, restrict the range of configuration for each object. The statistics of this restriction are also produced by the estimator in terms of the statistics of the relevant geometric parameters.

For example, an estimator for a type B contact will return a normal $\mathbf{n}$ and a contact point $\mathbf{p}$ estimate and a covariance matrix for both. These two estimates produce an estimate of the vertex location for the contacting object, and require one edge of the contacted object to have normal $\mathbf{n}$ and pass thru $\mathbf{p}$. This induces a distribution on the allowed configuration of each object considered independently.

The information from the measurements can now be fused with the non-parametric distribution for each object independently. There are two distributions, the prior $\mathrm{p}(t-1)$ and the new measurement distribution $\mathrm{p}_m(t)$. These can be fused using random sampling. For each distribution generate n random points. There are now $2n$ random points that represent possible locations of the object. Using this new non-parametric distribution, draw another n random samples. These n samples represent an estimate of $\mathrm{p}(t)$, the configuration distribution after measurement fusion.

The distributions for the objects are of course not at all independent. Many pairs of configurations for the two objects would produce a collision. This is explicitly represented in configuration space. Furthermore, this information could be incorporated if the non-parametric distribution was defined in configuration space. This is the approach in [Robles, 1995]. However for n three dimensional objects the configuration space representation has complexity exponential in $6n$. Because the approach outlined above ignores this information and fuses n independent distributions the complexity is only $6n$.

5.4 Conclusion

In this chapter the idea of contact feature was formalized as a model-based description of the measurements. The hard requirement of disambiguity of sets was loosened to a definition in which sets can only be probabilistically separated. This was done by defining features as models which produce finite (or at most countable) partitions of the phase-space. This is critical, because a discrete graph for observing progress is only defined for features which generate a countable partition.

Feature partitions were then collected into a local feature graph using phase-space connectivity as the equivalence relationship. Finally, the notion of a feature graph was extended to show how forward projection could be used to produce a more restricted from of feature connectivity and thereby incorporate information about the consequences of actions.

Now that we have defined features and put some structure on sets of features, how do we build an observer for the current state of the robot using this measurements and this structure? Our approach uses two subcomponents. The first component maps the raw signal into feature models and detects when the signal no longer comes from a given model. This is the change detection problem. The second component uses the graph and the measurement likelihoods computed from the first component to approximately compute the best measurement path and therefore the probability of each feature instance. These two components are discussed in the next chapter.

A Contact Feature Observer

Chapter 6

The central point of this thesis is to develop a technique for observing the robot's contact state. The observer uses knowledge of the c-space and force and position measurements to determine where the robot is in a given task.

The previous chapters discussed the central representational idea of this thesis, namely that the statistics of force and position measurements that arise in a given task can be represented as a graph of models called the feature graph. Figure 1.1 shows one example. This graph encodes the prior knowledge available from the geometry, the commanded motion, and the local measurement models. This chapter shows how to use this representation to derive a statistical decision algorithm that observers the contact state.

Figure 6.1 shows the basic components of the contact state observer. The forces and positions are input into a collection of candidate estimators. There is one estimator for each hypothesized contact model. The estimators output a measure of match between the underlying model and measurements called the log-likelihood.[1] The estimators also output a residual process, which conditioned on the model being true, is a white noise sequence. The residual processes gets fed into a change detector. The change detector monitors the residuals for changes. The change detector computes the log-likelihood of departure from the current model and the most likely time of departure. Both of these statistics, and the model likelihoods computed by the estimators, are sent into a single decision procedure. This procedure uses all these process statistics and the feature graph to determine a collection of likely paths for

[1]The likelihood of a measurement is the conditional probability or density for the measurement given the parameters of the statistical model. The log-likelihood is the logarithm of the likelihood.

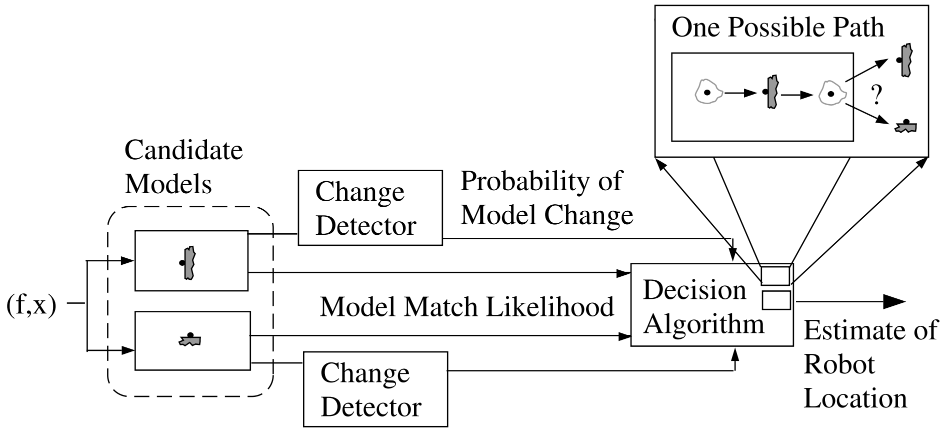


Figure 6.1: Contact state observer processing. The figure shows the basic architecture of the contact state observer processing force and position information measured by the robot as it explores the maze.

the measurements through the feature graph using a beam Viterbi search strategy. Since each of these paths terminates in a feature node, the procedure also computes an estimate of the current feature node of the robot. Therefore, this process reduces the problem of observing the contact state to the problem of observing the current contact model.

This chapter first presents the statistics of the decision problem. Then we specialize to the approach indicated in figure 6.1. Recursive estimators and their output statistics are discussed, and then change detection is presented. We conclude by showing how these components can be reassembled to produce the algorithm in figure 1.3.

6.1 Statistical Observer Formulation

The observer problem can be formulated in terms of assigning a feature node to each measurement given the constraints imposed by the graph. In general, there are m feature nodes $\mathcal{N}_i \quad i = 1, ..., m$. Each node provides a statistical description $\mathrm{p}_i(\mathbf{y}(k), ..., \mathbf{y}(l))$ of part of the measurement process which depends upon some parameters $\boldsymbol{\theta}$. There may or may not be a distribution provided for $\boldsymbol{\theta}$. As a series of sensor measurements $\mathbf{y}_0^n = \{\mathbf{y}(0), ..., \mathbf{y}(n)\}$ are taken, the problem is to generate an estimate $\hat{\mathcal{N}}_0^n = \{\hat{\mathcal{N}}(0), ..., \hat{\mathcal{N}}(n)\}$ for the sequence of nodes from which the measurements were produced. Initially, a distribution for the initial feature nodes is given $\pi_i(0) = \mathrm{P}(\hat{\mathcal{N}}(0) = \mathcal{N}_i)$. Let $\boldsymbol{\pi}(0)$ be a vector representing this collection of probabilities.

The parameters of each node fix the measurement distribution. To complete the statistical description, we must determine how these parameters are picked. We will use two approaches. Most of the time we are interested in labeling or matching the signal against a set of *a priori* models. In this case, it is useful to determine from a set of training data a probability distribution for the parameters for each feature instance. This approach assumes that every time the robot enters a given feature node, a value for the parameters is chosen from the parameter distribution. The value of the parameter then remains fixed until the robot leaves the given feature node.

We associate with every feature node $\mathcal{N}_i$:

- A parameterized Markov measurement process $\mathbf{y}(n) \sim \mathrm{p}(\mathbf{y}(n)|\mathbf{y}_{n-k}^{n-1}, \boldsymbol{\theta}_i)$ that depends upon at most k samples in the past.
- A parameterized distribution for $\boldsymbol{\theta}_i \sim \mathrm{p}(\boldsymbol{\theta}_i|\boldsymbol{\psi}_i)$. $\boldsymbol{\psi}_i$ is assumed known and fixed based on training data.

The statistics of the measurement signal up to time n is completely described by specifying the sequence of feature nodes which produced the measurements and the time at which each node began.

This model for the observation statistics is one additional level of stochastic abstraction than what is commonly used in Hidden Markov Models. It provides a very useful additional modeling tool. Experimentally, the feel of stroking a single texture tends to be produced by a consistent, constant parameter model for any short stroke. However, new motions across the texture, or extended motions, may be produced by different parameter values. All of these values can be collected into a single distribution for the parameters. Every time the robot enters a given feature, a value of the parameters is chosen from the parameter prior associated with the feature. This is then used to generated the observations until the robot leaves the feature.

Note that this creates the possibility of self-transitions. The robot can transition from a feature instance back into the same instance, because this models a change in the parameter values. Therefore, feature instances, when doing the processing, are specified by both the feature model and a starting time.

Let $a_{i,t} = (\mathcal{N}_i, t)$ be a feature instance which specifies the feature node and the

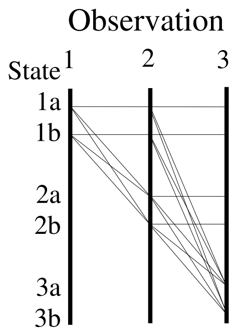


Figure 6.2: The computational lattice for the first three measurements for computing the feature node probability for a graph with two nodes.

start time for the instance. The lattice in figure 6.2 shows the possible paths for the first three measurements for a system with two possible feature nodes (a, b). The number to the left of the model designator indicates the starting time for each model. Horizontal lines are paths which stay in the same model without restarting, angled lines indicate self-transitions in the path. Self-transitions model the process switching from one set of measurement parameters to a different set within a single model. This graphically illustrates that at time n there are mn possible feature instances and approximately m^n possible paths. The set of all possible states at time n will be denoted $\mathcal{A}(n)$ and the set of all paths will be denoted $\mathcal{S}(n)$.

The probability of receiving the first n measurements given any sequence of feature instances $s \in \mathcal{S}(n)$ is

$$\mathrm{p}(\mathbf{y}_1^n|s) = \prod_{i=1}^{\mathrm{len}(s)} \mathrm{p}(\mathbf{y}_{t_i}^{t_{i+1}}|s(i)). \tag{6.1}$$

$s(i)$ is the i^{th} feature instance in the sequence. The sequence measurement distribution is computed from the single measurement distribution

$$\mathrm{p}(\mathbf{y}_{t_i}^{t_{i+1}}|s(i)) = \int_{\boldsymbol{\theta}} \prod_{j=t_i}^{t_{i+1}-1} \mathrm{p}(\mathbf{y}_j|\mathbf{y}_{j-k}^{j-1}, \boldsymbol{\theta})\mathrm{p}(\boldsymbol{\theta}|\mathcal{N}_i), \tag{6.2}$$

where we take $t_{i+1} = n$ if the feature instance $s(i)$ is the last one in the sequence. The probability of any feature path, given the measurements, can be computed from these terms by applying Bayes theorem

$$\mathrm{P}(s|\mathbf{y}_1^n) = \frac{\mathrm{p}(\mathbf{y}_1^n|s)\mathrm{P}(s)}{\mathrm{p}(\mathbf{y}_1^n)}. \tag{6.3}$$

The probability of a sequence of measurements is much easier to compute if we first whiten the measurements with a recursive estimator. An estimator should produce an *innovations* process $\boldsymbol{\nu}(n)$ which is a white process that summarizes the statistics of the measurement process. For linear predictive models, which we used for the temporal signals, the Kalman filter is the correct estimator. For the constraint models, we derive a recursive maximum log-likelihood estimator.

The feature observer must compute an estimate of the probability of being in each of the feature nodes at time n given the measurements. This probability is the sum of the path probabilities over all paths which reach the given node at time n

$$\mathrm{P}(\mathcal{N}(k)|\mathbf{y}_1^n) = \sum_{s\in\mathcal{S}(n):\mathrm{s}(\mathrm{len}(s))=\mathcal{N}(k)} \mathrm{P}(s|\mathbf{y}_1^n). \tag{6.4}$$

Evaluating this directly is computationally hopeless because there are an exponential number of paths. If transition probabilities between the elements of $\mathcal{A}(n)$ and $\mathcal{A}(n+1)$ are available, it is possible to form a recursive computation which uses order nm computations for the n^{th} step. Transition probabilities are required, because the computation requires a summation over all paths which reach a given feature instance. The transition probabilities are needed to make this summation meaningful. Unfortunately, these are not readily available in our problem.

Alternatively, the optimal path terminating in any element of $A(n)$ can also be determined with order nm computations using a form of the Viterbi algorithm, a form of dynamic programming, without transition probabilities. The standard Viterbi algorithm was developed for HMM models and does require transition probabilities. Let $\mathcal{S}_{\max}(n)$ be the set of nm paths which maximize the probability to each of the elements of $\mathcal{A}(n)$. The estimate of the feature probability

$$\mathrm{P}(\mathcal{N}(k)|\mathbf{y}_1^n) \approx \frac{\sum_{s\in\mathcal{S}_{\max}(n):\mathrm{s}(\mathrm{len}(s))=\mathcal{N}(k)} \mathrm{P}(s|\mathbf{y}_1^n)}{\mathrm{Norm}} \tag{6.5}$$

where Norm is a normalizing constant, can be computed from the available information. Furthermore, this approach can be further computationally bounded for real-time calculations by incorporating a change detection algorithm.

The change detection approach, will compute an estimate of the best path by testing each current best path for deviations from the current feature instance. When a change is detected, alternative paths starting from an estimated change time are expanded and compared to all the current best paths. Only the best path to each element of $\mathcal{A}(n)$ is kept. The computations can be further bounded by ranking the paths in $\mathcal{S}_{\max}(n)$ and keeping only enough paths to ensure keeping the overall best path with high probability.

The change detection approach relies on the empirical observation that changes in the feature instance are generally well separated in time. It further relies on the fact that the change test is a renewal process. Therefore, the estimate of the change time does not drift from the current estimate unless the test resets. Therefore, the observer will never expand more than one path for a single detected change even if the observer does not immediately decide to change its best path estimate.

The sections 6.3 and 6.3.1 discuss the estimator and change detector in general. Finally, a formal definition of the observer is given in 6.4.

6.2 Relationship to Other Detection Problems

Our approach is related to techniques that have been applied in speech processing, failure detection in dynamic systems, and modal change detection for structures.

In segment based approaches to computer perception of speech, the first step is to roughly segment the observered signal into phonemes. [Andre-Obrecht, 1988] looked at segmenting the speech signal using an autoregressive model for the measurements and three different change detection algorithms. The change detection algorithms are similar to the one discussed here.

Both our detector and the work of Andre-Obrecht is based on a number of papers by Basseville and Benveniste [Basseville and Benveniste, 1983, Basseville, 1986, Basseville *et al.*, 1986, Basseville *et al.*, 1987, Basseville, 1988, Benveniste *et al.*, 1987]. These works have been applied to segmentation of EEG, ECG, speech, and geophysical signals. The best reference is the collection of papers in [Basseville and Benveniste, 1986].

Their works are all related to the sequential likelihood ratio test, originally developed by [Wald, 1947], and the generalized likelihood ratio test (GLR). The GLR test was applied to changed failure detection for linear dynamic systems in [Willsky and Jones, 1976, Willsky, 1976, Chien and Adams, 1976, Tanaka and Muller, 1990] and to detection of incidents on freeways in [Willsky *et al.*, 1980]. Optimality of the GLR test has been investigated under many different conditions, one useful reference is [Zeitouni *et al.*, 1992]. An alternative test, which can be computed sequentially, is developed in [Hall, 1985].

Our path scoring algorithm is based on the Viterbi algorithm for Hidden Markov Models (HMM). The major difference between our representation and HMM, is that the HMM model assumes that the measurements are independent conditioned on the state in the Markov graph. Our model says that the measurements come from a measurement model, which can be whitened with an estimator, conditioned on the state in the graph. This allows for correlated measurements within each state in the graph. The best tutorial on the HMM model is in [Rabiner, 1989]. The HMM model was applied to contact perception in [Hannaford and Lee, 1991]. An approach similar to ours for dynamic systems is presented in [Tugnait and Haddad, 1979]. Segmental approaches to understanding speech [Goldenthal, 1994] also model the correlation in the speech signal.

6.3 Estimating the Current Feature Parameters

The first step in developing a feature observer, is to develop an estimator for the chosen feature model. The temporal models used in this thesis were linear predictive models, so the Kalman filter is the appropriate estimator. The constraint models use a combination of a maximum likelihood estimator for the parameters, and a Kalman filter prediction of the measurements. In either case, we assume that it is possible to develop an estimator which produces estimates via

$$\hat{\boldsymbol{\theta}}(n) = \hat{\boldsymbol{\theta}}(n-1) + \gamma \mathbf{H}(\hat{\boldsymbol{\theta}}(n-1), \mathbf{y}(n)) \tag{6.6}$$

and measurement innovations via

$$\boldsymbol{\nu}(n) = \mathbf{y}(n) - \mathrm{E}[\mathbf{y}(n)|\hat{\boldsymbol{\theta}}(n-1)]. \tag{6.7}$$

The generated innovations process will be assumed to be asymptotically white and normal. The covariance of the innovations are either provided or estimated on-line using

$$\mathbf{V}_{\boldsymbol{\nu}}(n) = \mathbf{V}_{\boldsymbol{\nu}}(n-1) + \frac{1}{n}(\boldsymbol{\nu}(n)\boldsymbol{\nu}(n)^T - \mathbf{V}_{\boldsymbol{\nu}}(n-1)). \tag{6.8}$$

The innovations process has the property that

$$\mathrm{p}(\mathbf{y}_1^n|a_1^l) = \mathrm{p}(\boldsymbol{\nu}_1^n|a_1^l). \tag{6.9}$$

Furthermore, because the innovations are white, the log of the probability of receiving the measurements (the log-likelihood) given the feature is

$$\log \mathrm{p}(\mathbf{y}_t^{t_{i+1}}|a_{f,t}) = \sum_{j=t}^{t_{i+1}} \log \mathrm{p}(\boldsymbol{\nu}(j)|f). \tag{6.10}$$

Thus the innovations reduce the problem to summing up the log probabilities of a sequence of measurements. Unfortunately if the variance of the innovations must be estimated, this computation cannot be done recursively. Instead, the current estimate of the variance must be used to compute the log likelihood of the current innovation. For changes which occur sufficiently separated in time, this approach will be sufficient.

6.3.1 Testing for Changes in the Model

The goal of the observer is to estimate the feature instance path given the graph. Dynamic programming can produce the optimal path up to the current time, but at the cost of a linearly increasing number of models and computations with each new measurement. In order to limit the computational cost, the observer orders the possible paths by probability. Then only enough paths to capture, with high probability, the best future path are tracked. To further limit the computation, the observer only branches and produces new path possibilities when a currently tracked path has most likely undergone a change to a new model. Detecting these changes is covered in this section.

The *change detector* takes the form of a sequential hypothesis test on the innovations process produced by the feature parameter estimation algorithm. The area of sequential hypothesis test for detecting jump changes in statistical processes has been an active area of research in statistics and signal processing since its initial development by Wald [Wald, 1947]. A mathematical review is given by Siegmund [Siegmund, 1985]. There have been a number of important results during the last decade [Willsky, 1976, Basseville, 1988, Benveniste *et al.*, 1987]. These methods are relevant to any signal processing task which can be modeled as a stochastic measurement process on an underlying system which undergoes discontinuous changes. The methods are particularly useful when accurate and rapid decisions about the time of change are required. This includes edge detection, continuous speech segmentation, and contact sensing.

In sequential hypothesis testing it is assumed that the time for the algorithm to detect a transition is short compared to the holding time before a second transition. Therefore it is assumed: 1) that transitions can be detected by considering only the data, and 2) only one transition from this hypothesis needs to be considered.

In order to apply the approach we need two hypotheses for the innovations process. The first hypothesis, the null hypothesis H_0, is that the current feature model is correct. Under this assumption the test statistic

$$\boldsymbol{\nu}(n)^T \hat{\mathbf{V}}(n)^{-1} \boldsymbol{\nu}(n) \tag{6.11}$$

is approximately $\chi^2(m)$ distributed for a vector measurement process of size m, because $\boldsymbol{\nu}(n)$ is asymptotically normal and $\hat{\mathbf{V}}(n)$ converges to the covariance of $\boldsymbol{\nu}$. If the process changes, both the mean and variance of $\boldsymbol{\nu}$ might change. For our purposes, the simple alternative hypothesis that the change in $\boldsymbol{\nu}$ is reflected in a change in magnitude of the covariance seems to suffice. More sensitive tests which involve estimating the direction of change can also be applied [Eberman and Salisbury, 1994]. Therefore, our alternative hypothesis H_1 is that $\boldsymbol{\nu}(n) \sim \mathrm{N}(0, q\hat{\mathbf{V}}(n))$, where q is the user selected change in variance.

Given these definitions we want to test the innovations process for a possible change at time r between 1 and n. We form the likelihood ratio between the hypothesis that the process was generated by H_0 from time 0 to time $r-1$ and then in H_1 from time r to n versus the hypothesis that the process was always in H_0.

Because the innovations process is white, the likelihood ratio is

$$\begin{aligned} L(0,1,r,\boldsymbol{\nu}_0^n) &= \frac{\mathrm{p}(\boldsymbol{\nu}_0^{r-1}|\mathrm{H}_0)\ \mathrm{p}(\boldsymbol{\nu}_r^n|\mathrm{H}_1)}{\mathrm{p}(\boldsymbol{\nu}_0^n|\mathrm{H}_0)} && (6.12)\\ &= \prod_{t=r}^{n} \frac{\mathrm{p}(\boldsymbol{\nu}(t)|\mathrm{H}_1)}{\mathrm{p}(\boldsymbol{\nu}(t)|\mathrm{H}_0)}. && (6.13)\end{aligned}$$

That is whiteness, or independence, implies that the likelihood is the product of simple terms.

To simplify the calculations let $\gamma_0(t) = \log(\mathrm{p}(\boldsymbol{\nu}(t)|\mathrm{H}_0))$, $\gamma_1(t) = \log(\mathrm{p}(\boldsymbol{\nu}(t)|\mathrm{H}_1))$, $\Xi_{01}(t) = \gamma_1(t) - \gamma_0(t)$, and $S_k^j(0,1) = \sum_{t=k}^{j} \Xi_{01}(t)$. Simplifying the expressions for the required change in covariance test shows that

$$\Xi_{01}(t) = 1/2(-m\log(q) + (1-1/q)\boldsymbol{\nu}(t)^T \hat{\mathbf{V}}(t)^{-1} \boldsymbol{\nu}(t)). \tag{6.14}$$

The decision function, DF, for a change from state 0 to state 1 is

$$DF(0,1,\boldsymbol{\nu}_0^n) = \max_{r\in[0,n]} \log L(0,1,r,\boldsymbol{\nu}_0^n) \tag{6.15}$$

which results in the binary rule

$$DF(0,1,\boldsymbol{\nu}_0^n) \underset{\mathrm{H}_0}{\overset{\mathrm{H}_1}{\gtrless}} T^2. \tag{6.16}$$

This rule says that H_1 will be chosen as the new state if $DF(0, 1, \boldsymbol{\nu}_0^n)$ becomes larger than T^2, otherwise H_0 will be maintained as the current hypothesis. T^2 is the decision threshold and is a design parameter that controls the essential trade-off between the detection delay and the false alarm rate. Increasing T^2 increases the detection delay and decreases the false alarm rate. For a process with multiple changes a large detection delay could lead to missing changes.

The decision function is equivalent to the Page-Hinkley (PH) cumulative sum stopping rule

$$DF(0, 1, \boldsymbol{\nu}_0^n) \quad = \quad S_0^n(0, 1) - \min_{0 \leq j \leq n} S_0^j(0, 1). \tag{6.17}$$

This test minimizes the time taken to reach decision H_1 over all tests that have the same false alarm rate [Siegmund, 1985]. Further, it is easily computed recursively by

$$DF(0, 1, \boldsymbol{\nu}_0^n) = \max(0, DF(0, 1, \boldsymbol{\nu}_0^{n-1}) + \Xi_{01}(n)). \tag{6.18}$$

This is essentially integration with reseting at 0.

Intuitively, the value of q is chosen such that as long as H_0 is correct the integration will on average drift negatively. Reseting the integration to zero then always restarts the process at zero. When a change occurs, the process integrates in the positive direction and eventually crosses the decision threshold. In some sense this test is thresholding a lowpass filtered signal with the lowest possible frequency filter.

There are two important characteristics of any hypothesis testing procedure: 1) the false alarm rate, 2) the delay to detection. The earliest time at which the decision function exceeds the threshold, given that the system is still in state 0, is the false alarm time $t_f = \inf(n : DF(0, 1, \boldsymbol{\nu}_0^n) > T^2)$ which has distribution $\mathrm{P}_{FA}(n)$. The probability of no alarm at time n is $\mathrm{P}_{NA}(n) = 1 - \mathrm{P}_{FA}(n)$. The asymptotic false alarm rate is defined to be $f = 1 - \lim_{n->\infty} \frac{\mathrm{P}_{NA}(n)}{\mathrm{P}_{NA}(n-1)}$. This reflects the rate at which false alarms will occur over the long-term. In contrast, the delay to detection is a transient performance measure. The delay to detection, given that a change to state 1 occurred at time 0, is $t_D = \inf(n : DF(0, 1, \boldsymbol{\nu}_0^n) > T^2 | \text{Initial state is} \mathrm{H}_1)$. The distribution of t_D is $\mathrm{P}_D(n)$ and its expected value is $\bar{t}_D = \sum_{t=0}^{\infty} t\mathrm{P}_D(t)$. Both statistics are controlled by T which is a design parameter. Increasing T decreases the false alarm rate and increases the time to detection. Determing both of these relationships requires solving a first passage problem. Closed form solutions to this type of problem are rare and difficult to derive. Approximations for some simple cases are discussed in [Eberman and Salisbury, 1994].

In particular, the Page-Hinkley (PH) test can be compared to the very popular method of filtering followed by thresholding. Both approaches take exactly the same amount of computation, but as the following figures show the PH test provides superior performance. A comparison was done for tests which have the same false alarm rates.

The asymptotic false alarm rate ${}^{PH}f$ and time to detection ${}^{PH}\bar{t}_D$ for the Page-Hinkley test can be approximated by applying Wald's identity and approximations [Siegmund, 1985]. The results are

$$\begin{aligned} {}^{PH}\bar{t}_F &\approx |e^{T^2} - T^2 - 1|/\beta_0 \\ {}^{PH}\bar{t}_D &\approx (e^{-T^2} + T^2 - 1)/\beta_1 \end{aligned}$$

where

$$\beta_i = \int \log\left[\frac{\mathrm{p}_1(\zeta)}{\mathrm{p}_0(\zeta)}\right] \mathrm{p}_i(\zeta) d\zeta.$$

Since the false alarms are the interarrival times of a Bernoulli process they are geometrically distributed. Therefore the asymptotic false alarm rate is

$${}^{PH}f = \frac{1}{{}^{PH}\bar{t}_F}.$$

For the change in mean between two Gaussian processes with the same standard deviations σ, β_i is

$$\beta_i = 1/2\left(\frac{\Delta\mu}{\sigma}\right)^2.$$

A plot of the trade-off between the time to detection, $\bar{t}_d$, and the time to false alarm,$\bar{t}_f$ is called the receiver operating characteristic (ROC). It is a function of the signal-to-noise ratio $s = \frac{\Delta\mu}{\sigma}$. Graph 6.3 shows the value of $\bar{t}_d$ and $\log_{10}\bar{t}_f$ parameterized by T for a fixed value of s. The ROC for this test is shown in figure 6.3 for $s = 0.5, 1.0, 1.5, 2.0$. Both the mean time to a false alarm and detection increase with increasing threshold. At a fixed false alarm time, an increase in the signal-to-noise ratio will decrease the time to detection.

The performance of the alternative test of lowpass filtering followed by thresholding can be bounded using the following asymptotic approximation derived by Hall [Hall, 1985]. The approximations are valid in the limit of an increasing threshold and short sampling time. Consider a filter realized by a stable, linear, time invariant vector process x

$$x(k+1) = Ax(k) + w(k+1) + \Delta\mu u_{-1}(k-r)$$

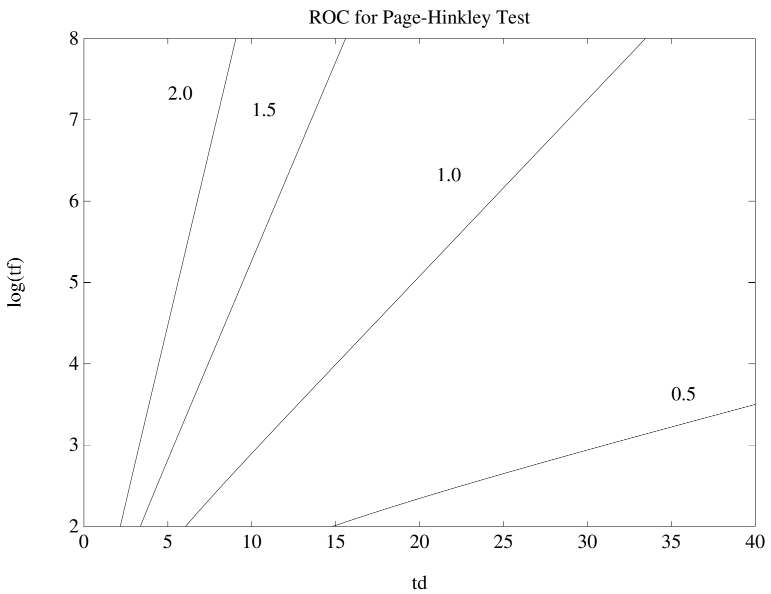


Figure 6.3: Receiver operating characteristic (ROC) of Page-Hinkley test between two Gaussian distributions with different means and the same variance as a function of the signal to noise ratio $s = \frac{\Delta\mu}{\sigma}$. The $\log_{10}(\bar{t}_f)$ is shown as a function of the mean time to detection $\bar{t}_d$ for s =0.5, 1.0, 1.5, and 2.0.

driven by a white, zero-mean, Gaussian noise $w(k)$ with noise intensity Q. A change of size $\Delta\mu$ is applied by the unit step u_{-1} at time r. The covariance of x is a steady-state solution to the discrete Lyapunov equation and satisfies $S = ASA^T + Q$. The decision function is $DF(k) = x^T(k)S^{-1}x(k)$. In principle it is possible to determine $\mathrm{P}_{FA}(k)$ by propagating the density for $x(k)$, $\mathrm{p}(x,k)$, forward in time and then integrating over the decision region. The propagation equation is

$$\mathrm{p}(x, k+1) = \int_D \mathrm{p}_w(x - A\zeta)p(\zeta, k)d\zeta$$

where $D = \{x : DF(k) \leq T^2\}$. Then $\mathrm{P}_{FA}(k)$ is given by

$$\mathrm{P}_{FA}(k) = 1 - \int_D \mathrm{p}(u, k)du.$$

Unfortunately there are no closed form solutions to this problem. However by treating the discrete system as a sampling of a continuous system, an approximation valid for large k can be determined. Using this approximation, the steady state false alarm rate f is found to be asymptotically bounded by

$$f \leq 1 - \exp\left(\frac{\ln(\det(A))T^p}{\Gamma(p/2+1)}\exp^{-T^2/2}(1 - p/T^2)\right)$$

where p is the dimension of x. In the case of a first-order lag filter $x(k+1) = ax(k) + w(k)$, the bound is

$$f_0 \leq 1 - \exp\left(\sqrt{\pi/2}\ln(a)T\exp^{-T^2/2}(1 - 1/T^2)\right).$$

This is the bound for $x^2/S > T$. The PH test is equivalent to $X/S^{1/2} > T$ which has a false-alarm rate bounded by $f_0/2$.

To approximate $\mathrm{P}_D(k)$ note that $DF(k)$ is a noncentral chi-squared random variable with p degrees of freedom and noncentrality parameter $\delta^2(k) = \bar{x}^T(k)S^{-1}\bar{x}(k)$ [Anderson, 1984]. The process mean $\bar{x}$ satisfies

$$\bar{x}(k+1) = A\bar{x}(k) + \Delta\mu$$

with initial condition $\bar{x}(0) = 0$ for a change in mean of $\Delta\mu$, where we have assumed for simplicity $r = 0$. If the cumulative noncentral chi-square distribution of DF at value T^2 is denoted by $F(T^2, \delta^2, p)$, then $\mathrm{P}_D(k)$ is bounded by

$$\mathrm{P}_D(k) \geq 1 - F(T^2, \delta^2, p)$$

which can be computed numerically or approximated.

For a scalar, first-order lag-filter, the ROC can be computed as a function of the signal-to-noise ratio s as in the PH test. In this case, the values of $\bar{t}_d$ and $\log_{10} \bar{t}_f$ are parameterized by a. The optimal threshold for the test is $\frac{\Delta\mu^2}{4S}$ where $S = \frac{(1-a)}{(1+a)}\sigma^2$. This gives a threshold of $T^2 = \left(\frac{s}{2}\right)^2 \frac{(1-a)}{(1+a)}$. With the one-sided test, an approximation for $\mathrm{P}_D(k)$ is simply the probability of drawing a value greater that $\Delta\mu/2$ from a Gaussian random sample which has mean $\bar{x}(k)$ and variance S, given that the test has not already terminated. The probability of terminating at time k given that the test has not already terminated is

$$F(k) = 1 - \mathrm{erf}\left(\frac{s}{2}\sqrt{\frac{1-a}{1+a}}\right).$$

The probability of terminating at time k is then given by the recursion

$$\begin{aligned} P_D(0) &= F(0) \\ P_D(k) &= F(k)(1 - P_D(k-1)). \end{aligned}$$

This gives an underestimate of the termination time. An overestimate is given by the rise time for $\bar{x}(k)$ to $\Delta\mu/2$. Figure 6.5 shows the logarithm of $\bar{t}_f$ as a function of $\bar{t}_d$ for a signal-to-noise ratio of $s = 0.5$, 1.0, 1.5, and 2 computed using these two approximations. The curve for $s = 0.5$ has been cut short, because the approximation is not valid for small $\bar{t}_d$.

An examination of both figures shows that the performance is better for the Page-Hinkley stopping rule for all signal-to-noise ratios greater than 0.5. With a signal-to-noise ratio of 0.5 the figures indicate that filtering performs better. This is most likely do to the approximations used in computing these curves. The ROC curve for the filtering approach is only an upper bound and the true performance is probably lower. The ROC curve for the Page-Hinkley test is also computed from an approximation. According to the theory the Page-Hinkley test will always perform better at all signal-to-noise ratios.

Figure 6.6 indicates that the lowpass filter approach has a longer delay to detection compared to the PH test when they have the same false alarm rate. The test shown in figure 6.4 will signal an alarm on average every 6×10^6 samples and the change will be detected after 28 samples. To get equivalent performance from the lowpass filter, a must equal 0.98. With this value, the estimate of $\bar{t}_D$ is 29.5 and the rise time is 34.5. These results demonstrate that the PH test gives an improvement in performance without an increase in computational cost. In addition, an estimate of the change time is possible by storing a single number.

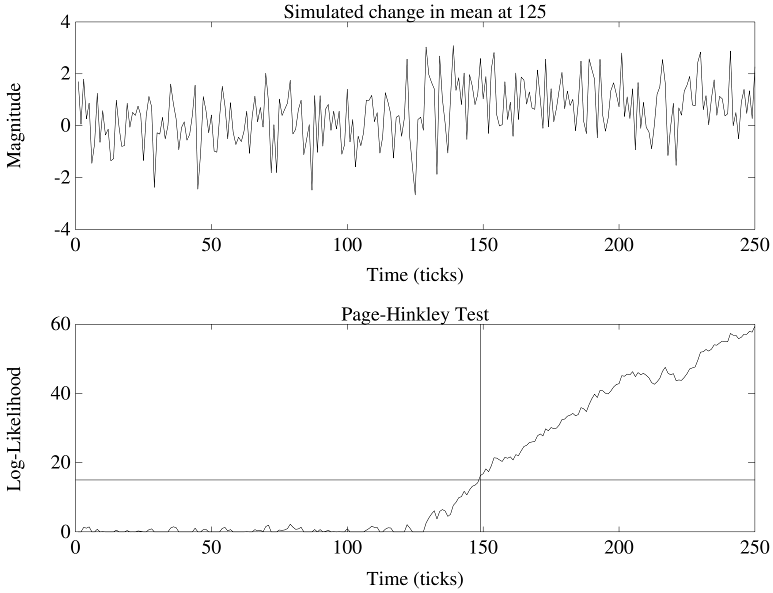


Figure 6.4: Behavior of the Page-Hinkley stopping rule to a simulated change in mean at tick 126 for a Gaussian process. Signal has standard deviation of 1 before and after the change, and mean of 1.0 after the change. Change is detected with a threshold of 15 at tick 149. The estimate of the time of change is the last time the test equals zero which is at tick 128.

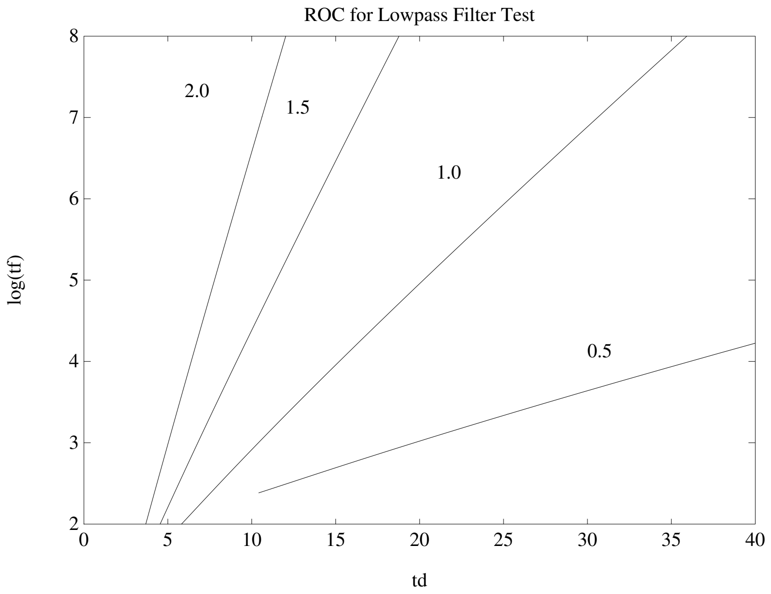


Figure 6.5: Receiver operating characteristic (ROC) of first order lag filter test with threshold between two Gaussian distributions with different means and the same variance as a function of the signal to noise ratio $s = \frac{\Delta\mu}{\sigma}$. The $\log_{10}(\bar{t}_f)$ is shown as a function of the mean time to detection $\bar{t}_d$.

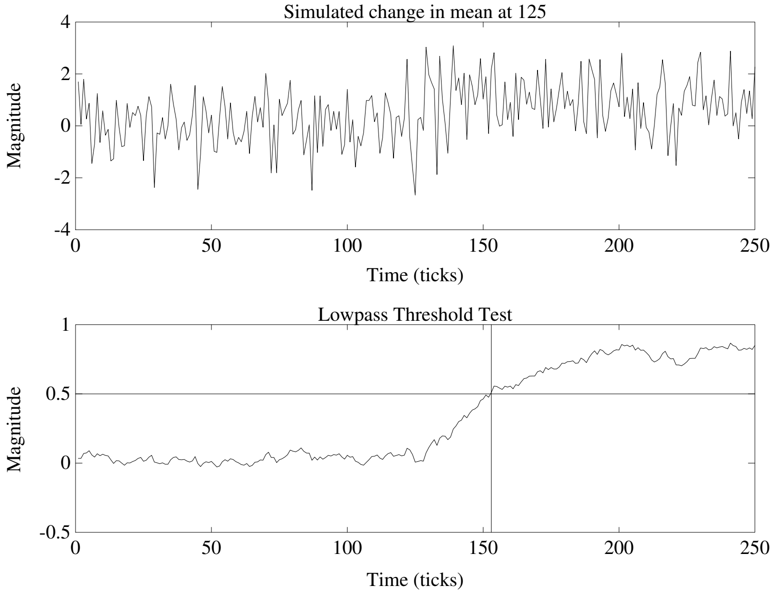


Figure 6.6: Lowpass filter of $x(n+1) = 0.98x(n) + 0.02y(n+1)$ on the same signal as figure 6-3. The threshold is 0.50. The change is detected at 153. This is a slower response then the response for the PH test. Further, an estimate of the change time is not computed.

6.4 Computing the Feature Probabilities

This section summarizes the local feature observer based on all the preceding background. The observer is given a local feature graph $\mathcal{G}$ consisting of a collection of nodes $\mathcal{N}$ and edges $\mathcal{E}$. An initial probability distribution over the graph for the current contact state $\boldsymbol{\pi}(0)$, and a probability pruning threshold $\mathcal{P}_{\text{prune}}$ are also provided. Finally, a change detection threshold T^2 and a change accumulation magnitude q is provided.

A Markov model is associated with every graph node n for the measurement parameterized by a set of known, fixed parameters $\boldsymbol{\psi}_n$. The values of these parameters were determined by estimating their values using previously segmented training data.

In order to initialize the observations process, the observer:

1. Creates a buffer to store past measurements.
2. Sorts the nodes by their initial probabilities from largest to smallest.
3. Selects nodes from this sorted list until the sum of the probabilities of all the nodes is greater than $\mathcal{P}_{\text{prune}}$.
4. Creates for each selected node, an observer for the measurement model and a change detector for the residual process. The threshold for the detector is set to T^2 and the size of the alternative hypothesis is set to q.
5. Initializes the path log-likelihood of each model to the logarithm of the model's prior probability.
6. Creates a buffer for each measurement model to store past values of the path log-likelihood for that model.

As each new measurement arrives, the observer:

1. Stores the value of the measurement in the measurement buffer.
2. Updates all the current models and change detectors using the new measurement.
3. Accumulates the log-likelihood of each residual in order to track the path log-likelihood.

4. Stores the path log-likelihoods for each model in each model's path log-likelihood buffer.
5. Records the maximum time in the past, over all current models, at which a change could have occurred.

If a change in any model is detected, then for every model in which a change is detected, the observer:

1. Creates a new estimator for every possible model to which a transition can occur given the edges $\mathcal{E}$ in the feature graph, and for which the same model and starting time are not already being tracked.
2. If an estimator with the same starting time and model is being computed, the initial path log-likelihood of the estimator is compared to the new spawning model. The current estimator is set to have the higher initial path log-likelihood. This implements the maximization step of the path tracking algorithm.
3. If the new model and starting time are not being tracked, the path log-likelihood of the new model is initialized to the path log-likelihood of the spawning model at the change time, and the measurements from the change time to the current time are incorporated into the new model.
4. Resets the change test statistic for the spawning model.

The observer now has a new collection of distinct paths. These paths are sorted on their path log-likelihoods. The log-likelihoods are then normalized and turned into probabilities. Enough of the new paths are then kept so that probability mass $\mathcal{P}_{\text{prune}}$ is retained. Finally, the probabilities are renormalized.

If a change is not detected in any current path, the observer shortens all the current buffers to their minimum possible lengths. For the measurement buffer, this is the minimum change time over all models. For the path log-likelihood buffers, it is the minimum change time for that path. The observer then computes the relative probability of each path by renormalizing the log-likelihoods.

Finally, for both change and no change, the probability of each feature is computed by summing path probabilities over paths that are currently in that feature. The resulting observer estimates the probability of being in each node in the feature graph by computing estimates of the best path for all the measurements. Computations are minimized by incorporating a set of change detectors on each active model. The detectors monitor the innovations produced by the parameter estimators and signal a change, or an event, when a cumulative sum stopping rule crosses a threshold.

6.5 Conclusion

This chapter presented a sequential decision approach to observing manipulation state. The manipulation state was encoded by a feature graph and the observer tracked the feature state by determining the most likely sequence of measurement models given by the features.

Change detection was used to control the expansion of the path likelihoods. It is critical in manipulation tasks that changes in the contact conditions be rapidly and robustly detected. Change detection theory provides a general tool for designed algorithms for this task. The probabilities of the paths were then related to the probabilities of each feature node. Since each feature node corresponds to a region of phase-space, the system can determine the probability that the robot occupies a region of phase-space. Thus, the robot can track its motion in a task.

Time Series Models

Chapter 7

This chapter looks at temporal models which capture properties of the force (or strains) signal considered purely as a time series. The force signal is strongly affected by textural properties, frictional stick-slip, and contact transients. These effects appear in the time series and affect the high frequency range of the force sensor. Time series models of the derivative of the force are useful models for these effects.

For example, figure 7.1 shows a spectrogram of an impact event. The impact results in an increase in energy at all frequencies locally around the event, and a persistent residual vibration at the sensor's natural frequency. It is important to isolate both the beginning and end of impact events in manipulation sensing.

Depending on the stiffness of the contacting materials, the beginning is usually easy to sense. For soft materials, the peak force is easy to sense but the beginning may actually be hard to sense. For hard materials, the impact rise time is so short that the beginning and peak are essentially the same. Sensing both the beginning and the peak is nice, because the rise-time is a good indication of the stiffness of the impacted material.

The end of impacts is always difficult to determine because the sensor and the robot vibrate after each contact. This vibration slowly dies away and an impact event is over when the vibration has sufficiently decayed. The end of the vibration needs to be sensed, so that a constraint estimator can be started for low frequency forces. The constraint estimator cannot be run while the sensor is in an impact event, and we would like it to run as soon as possible so that the robot can use the estimated constraints for force control. Uniform textures also produce temporal patterns that

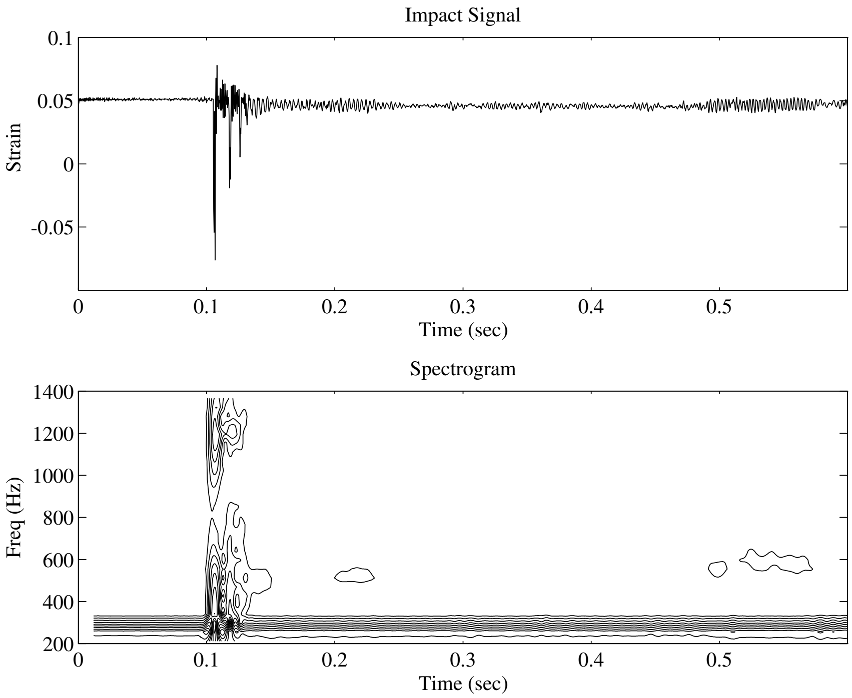


Figure 7.1: Spectrogram of an impact event. The upper figure shows a single strain signal with an impact event. The bottom figure shows a contour plot of the energy in frequencies from 200-1350 Hz as a function of time. The signal was sampled at 2700 Hz. Sixty-four points windowed with a Hamming window were used for each fast Fourier transform (FFT). The FFT was computed for every new data point. Note the broad frequency band that occurs at an impact and the short time scale of this event.

can be useful for identifying the contact location and contact type. We used time series models to model steady vibration levels, impacts and uniform textures. The raw strain signal and the derivative of the strain signal are used in the modeling.

Two basic problems in temporal modeling were examined during the course of this research: signal labeling and signal description. Signal labeling is matching the measurements against a set of predefined models. The predefined models are created off-line from hand segmented training data. Signal description is regression against a set of model classes so as to best describe the signal over the classes. This chapter discusses labeling. Signal segmentation and description is discussed in previous papers [Eberman and Salisbury, 1993, Eberman and Salisbury, 1994]. A simple change detection based segmentation procedure was used to produce the training samples for labeling.

Both problems require estimators to compute the likelihood of a model. In labeling, estimators incorporate prior knowledge of parameters in computing model likelihoods. We used a square-root implementation of the Kalman filter for labeling because it is fast and numerically robust. The standard implementation of the square-root filter was slightly modified for some of the models to produce an orthogonal regressor square-root filter. Appendix A.2, on filtering and estimation, discusses the detailed implementation of the filters.

The experiments discussed in this chapter show:

- That autoregressive models of the strain signal produce segmentation boundaries that make sense from the physics of the motion.

- That, at least for this sensor, most of the information for labeling textures is in the vibration energy.

- That the logarithm of the energy in the time derivative of strain is a good measure of vibration. Impact events can be isolated from the signal in this form at lower frequencies then in the raw strain signal.

- Context in the form of the correct model sequence, for a given measurement history, significantly helps in recognition.

7.1 Signal Measurements and Models

For temporal modeling the raw strain signals and generalized forces measured by a force/torque sensor provide a basis for modeling. We used the strain signals for temporal modeling because they are directly available and equivalent, up to linear transformations, to the generalized forces.

The strains $\mathbf{y}(n)$ were sampled at 1000 Hz, and the first order difference $\mathbf{y}_d(n) = \mathbf{y}(n) - \mathbf{y}(n-1)$ was computed. A high sampling rate is required to accurately capture the rapid vibration of short time events like impacts. The raw signals were then lowpass filtered with a second order filter at 125 Hz and then subsampled at 4:1 to produce a 250 Hz signal, $\mathbf{y}_l$. Filtering in this way empirically isolated the strain signal that was due to contact from the vibration signal. The contact induced strains are all low frequency in nature because they are caused by the motions of the robot. We filter and decimate in this ratio to maintain the flat, or white, spectrum of the measurement error. The change detection algorithm does not work as well with colored noise.

In addition to this signal, the logarithm of the energy in the derivative of the strain signal was computed. This high energy signal is formed by the filter

$$\mathbf{y}_h(n) = \log(|\mathbf{y}_d(n)|).$$

The logarithm signal was then lowpass filtered by averaging blocks of 8 points together and then sub-sampling at 4:1 again producing a 250 Hz signal with a flat noise spectrum. The averaging reduces noise in the signal. The effect of the logarithmic transformation and filtering is to estimate the envelope of the energy signal.

This can be seen by writing $\mathbf{y}_d$ as the product of two terms

$$\mathbf{y}_d(n) = q(n)\nu(n)$$

where $q(n)$ is a slowly varying envelope and ν is a white noise process. Expanding $\log(\mathbf{y}_d^2(n))$ yields

$$\log(\mathbf{y}_d^2(n)) = \log(q(n)^2) + \log(\nu(n)^2).$$

An analysis of the spectrum of these two processes shows that $\log(q(n)^2)$ has a low frequency spectrum and $\log(\nu(n)^2)$ is broad spectrum. Therefore, lowpass filtering removes some of the noise and emphasizes the envelope. This approach is a form of homomorphic filtering [Oppenheim and Flanagan, 1989].

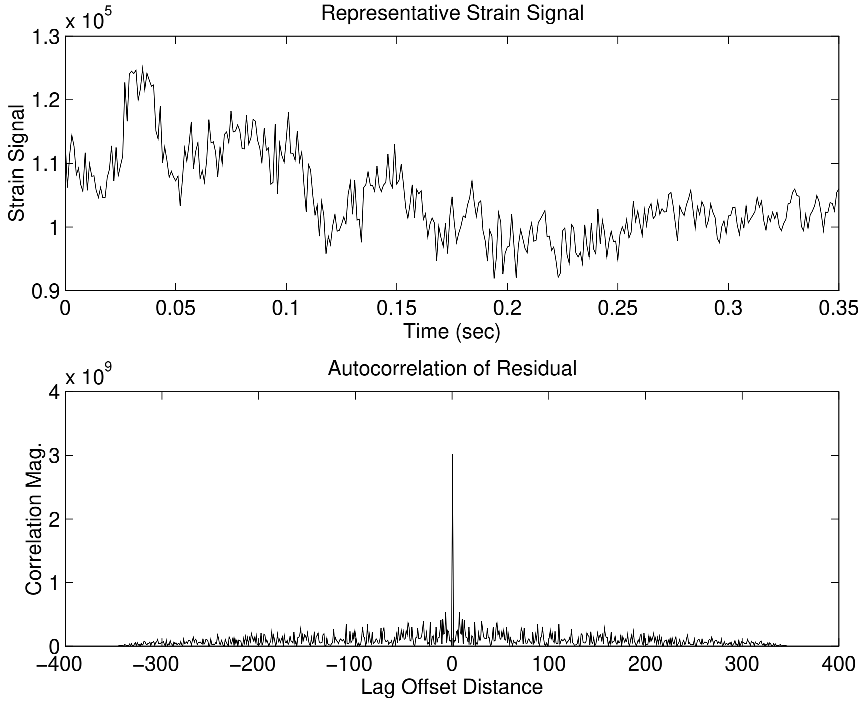


Figure 7.2: Fitting a stationary texture signal with an autoregressive process. The top figure shows one of the measured strain signals, and the bottom figure shows the autocorrelation of the residual after estimating four AR parameters. The spiked form of the autocorrelation shows that the AR model produces fairly white residuals and is a good fit to the measurements.

We used linear predictor coding (LPC) for both measurement processes. Each signal was treated as independent. Autoregressive (AR) models with and without a mean were used for models of textures and steady motions for both the raw strain signals and the high energy signal. The AR model is

$$\mathbf{y}(n) = \sum_{i}^{p} a_i \mathbf{y}(n-i) + \mu + \nu(n)$$

where $\nu(n)$ is a white, Gaussian processes, $\{a_i\}$ are the autoregressive parameters, and μ is the forcing mean. Figure 7.2 shows an example AR fit to a stationary strain signal. The feature vectors were formed from the linear predictor coefficients generated by this model and the parameter order $\{\mu, a_1, a_2, ...\}$.

The dominant characteristic of impacts is the sharp rise of the event and then exponential decay of the vibration envelope. If the impact is caused by gaining a contact, the strain signal will also show a change in mean. Intermittent impacts will not show a change in mean. The effects of the change in mean are eliminated by looking for impacts in the difference signal $\mathbf{y}_d$.

Figure 7.3 shows the difference formed from one of the strain signals from an impact and the high energy signal computed from this strain in dotted lines. An envelope produced by lowpass filtering the log signal is also shown in solid in both traces. The plots clearly show that the effect of filtering is to produce an estimate of the envelope. This envelope estimate can then be interpreted at the lower processing rate of 250 Hz. The lower logarithm plot also shows that a step rise in energy followed by a linear decay provided a good model of the signal in the logarithmic domain. The appropriate model is

$$\mathbf{y}(n) = \mu + an + \nu(n)$$

for a signal beginning at time 0. Again the feature vectors were formed from the linear predictor coefficients generated by this model and the parameter order $\{\mu, a\}$.

7.2 Labeling Stationary Models

Autoregressive models provide a good model for stationary strain and force signals. The feature parameters are the linear predictor coefficients generated by the model and the variance of the driving noise. In order to test the labeling performance of this feature, a simple two texture experiment was performed.

The PHANToM was used to stroke the sample in the x direction under open-loop force control. The complete motion command took 5 seconds. Data was recorded for almost 7 seconds. A downward force of 200mN was applied with zero stiffness and damping in the vertical direction. Position control was used in the x direction. The fingertip position was measured through the PHANToM using encoders on the motors.

The first sample (sample A) consisted of a single two inch section of aluminum. One inch of the aluminum was sanded crosswise with 80 grit sandpaper. The second half was polished. The second sample (sample B) was a polished piece of aluminum inlayed into a rough cross-cut piece of wood. Eight trials were performed for each sample. A typical set of strain measurements and the x position for the first sample is shown in figure 7.5.

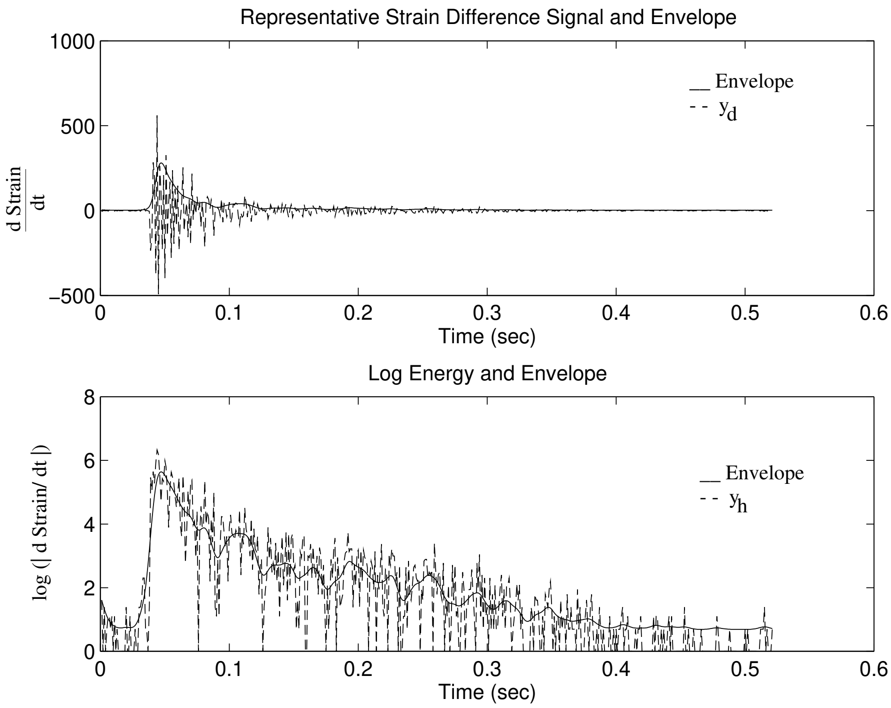


Figure 7.3: Characteristics of an impact processes. The primary characteristic is the sharp rise in strain followed by vibration within an exponentially decaying envelope. The upper figure shows the difference signal and an exponential envelope fit from the logarithmic trace. The lower figure shows the logarithm of the absolute value of the difference signal and an envelope created by lowpass filtering.

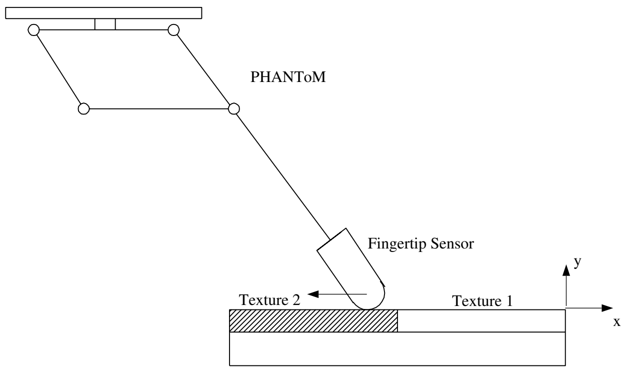


Figure 7.4: Two-texture labeling experiment. Each sample consisted of two textures placed side by side. Two different samples were used in the experiments. The surface was stroked with the sensor under force control.

There are 7 distinct phases to the motions which appear in the plot. The first phase is caused by the sensor resting on the plate. The next impact-like burst is caused by the start of the motion command. The sensor moves slightly forward and then is halted by friction. The third segment is the slow build up of strain as the position controller ramps up the torque commands to overcome contact friction. The fourth segment is motion over the rough surface. The fifth segment is motion over the smooth surface. The sixth segment, the short fast drop in strain, is caused by the termination of the motion command. Upon termination the system switches to a command which freezes the robot in its current position, thus the strains decrease. The last section is is from the sensor again resting against the surface of the aluminum.

7.2.1 Data Segmentation

In order to test the performance of autoregressive models in labeling example signals, segmented components were created using a change detector based segmentation. After automatically segmenting the data, the pieces that corresponded to steady motion over the textures were selected. These were collected together for modeling training and testing. For completeness, this subsection summarizes the segmentation procedure and empirically shows the effect of the decision threshold on the number of segments.

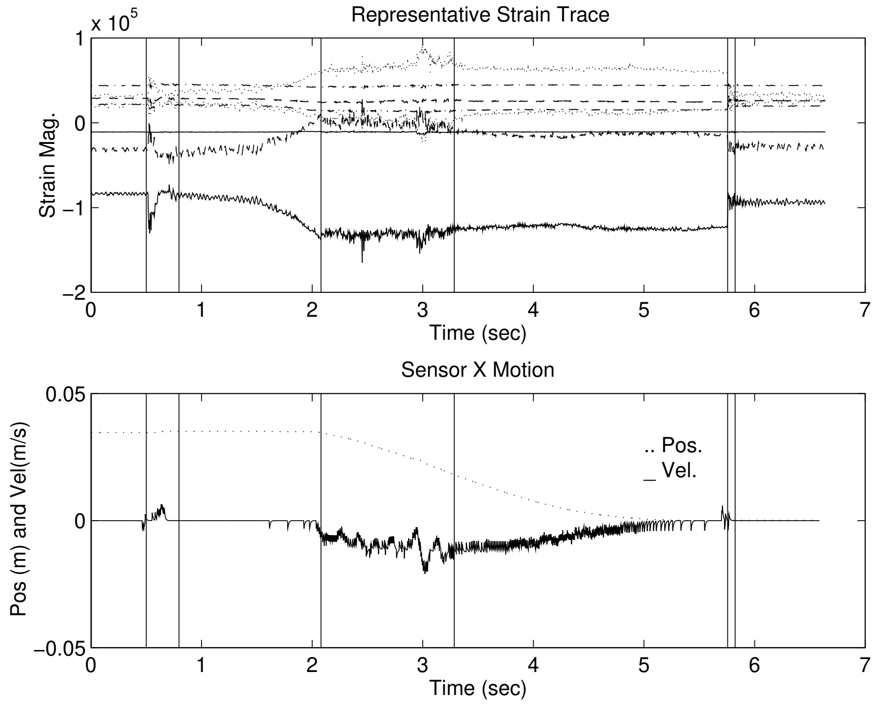


Figure 7.5: Typical result from the stroking experiment. The upper figure shows the preprocessed strain signals versus time. The lower figure shows the global x coordinate and velocity versus time. The signal division lines were generated by the segmentation procedure with a decision threshold of 100.

The segmentation approach uses the Page-Hinkley cumulative sum test (see chapter 5) to detect changes in the residuals produced by a parameter estimator. Figure 7.6 shows the complete signal processing architecture for segmenting the strains or forces using this approach. The estimator is discussed in detail in appendix A.2. The Page-Hinkley test is presented in section 6.3.1.

After preprocessing, each of the strain signals is run through a square-root parameter estimator. Two autoregressive terms are estimated by the filter. Each of the residuals produced by the estimators is then sent to a variance estimation scheme. The variance estimator starts on the ninth measurement. The sum of the residuals squared divided by their variance estimate is then sent to the Page-Hinkley change detector. This statistic is chi-squared with eight freedoms. The change detector looks for changes that correspond to a change in covariance of size 2.5. This level essentially corresponds to a change that is approximately $\sqrt{1.5}$ standard deviations away from the estimate of the variance. Experimentally this level was a good balance between sensitivity to changes and false alarms. Both increases and decreases in covariance magnitude are tested. The change likelihood is then accumulated using the Page-Hinkley rule, and if the change statistic crosses a decision threshold a change is indicated. After detecting a change, both the square-root estimator and the variance estimators are restarted.

The performance of the segmentation procedure can be gauged by comparing the segmentation points at different decision thresholds. Figure 7.7 shows the effect of the decision threshold on the number and position of the detected changes for one of the experiments with the first sample. Lowering the threshold increases the number of boundaries. However, the location of previous boundaries remains relatively constant. The number and location of the boundaries is heuristically reasonable for most threshold levels. At very low values of the threshold the test detects very small changes in the signal. Because the larger boundaries are maintained, these could possibly be removed by subsequent post-processing. Finally, note that the boundary caused by the change in surface texture appears at all threshold levels.

Since the test specimen has two distinct surface textures, the boundaries produced with $\lambda = 200$ were used for segmentation. Each experiment was segmented at this threshold and two training batches were formed for each test sample.

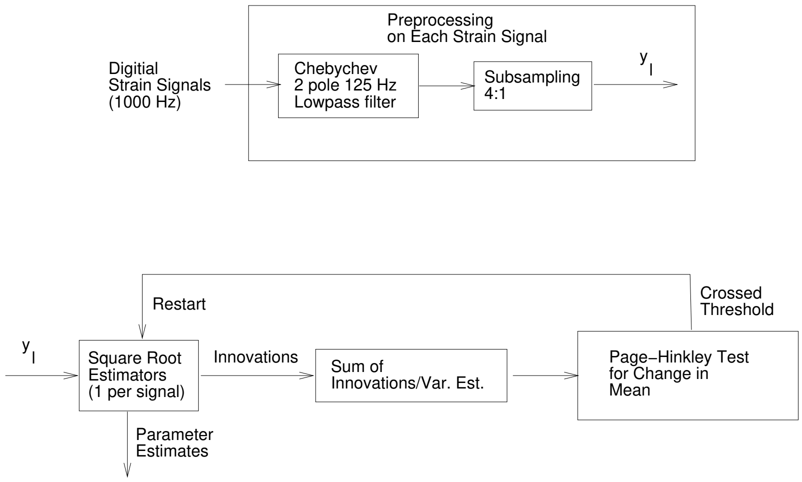


Figure 7.6: Signal processing architecture for on-line segmentation of strains and forces using an auto-regressive model. The strain or force signal is preprocessed using a low pass filter and decimation. Then a square-root parameter estimator is applied to the signal to produce the reflection coefficients. The estimator residuals are used in the Page-Hinkley cumulative sum test. If this test crosses the decision threshold, the estimator is restarted using the next measurement.

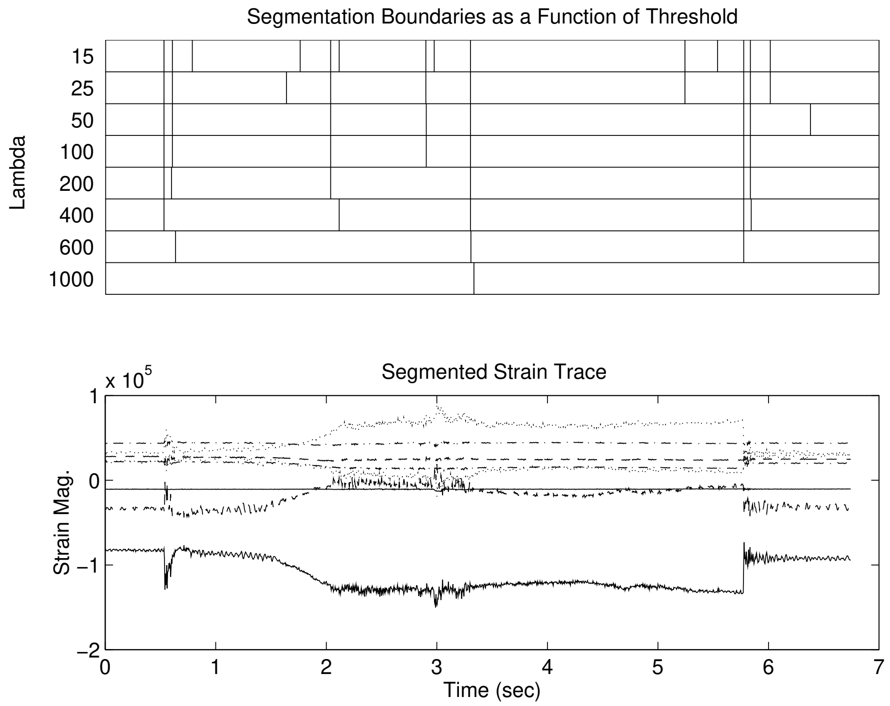


Figure 7.7: Effect of the decision threshold on the number and location of the segmentation boundaries for autoregressive segmentation. The bottom figure shows the preprocessed strains signals from one experiment using sample A. The top figure shows the placement of the boundaries for different levels of the decision threshold.

7.2.2 Batch Performance

A leave-one-out procedure was used to gauge the batch performance of the autoregressive models. In this procedure one example is left out of the example set. The models are then trained on the remaining examples. The left out example is then tested against the models. It is marked either correctly or incorrectly by the labeling process. Each sample is then left out in turn. The total performance is computed from the performance over all samples.

For sample A, the 8 training segments were subdivided into sections of length 100. This produced 67 blocks of data where each block consisted of the 8 strain measurements. For every individual strain signal a second order autoregressive model was estimated and the two LPC coefficients and the driving variance was recorded. This produced a 24 dimensional feature space consisting of 8 independent groups of three features.

The decision procedure is a binary hypotheses test between two cases. The autoregressive measurement model for each strain and each case is of the form

$$\mathbf{Y} = [\mathbf{Y}(-1), \mathbf{Y}(-2)]\boldsymbol{\theta} + \boldsymbol{\nu} \tag{7.1}$$

$$\mathbf{Y} = \mathcal{A}\boldsymbol{\theta} + \boldsymbol{\nu} \tag{7.2}$$

where $\mathbf{Y}$ is the complete measurement vector for a single strain, $\mathbf{Y}(-i)$ is the i^{th} lagged version of the vector, $\boldsymbol{\theta}$ are the AR coefficients and $\boldsymbol{\nu}$ is the noise term. The LPC measurement model is formed by decomposing $\mathcal{A}$ into $\mathbf{QR}$ using the QR decomposition. After decomposition the model is

$$\mathbf{Y} = \mathbf{QS}^{1/2}\mathbf{k} + \boldsymbol{\nu} \tag{7.3}$$

where $\mathbf{S}$ is the diagonal part of $\mathbf{R}$ and $\mathbf{k}$ is the vector of LPC coefficients.

Both $\boldsymbol{\nu}$ and $\mathbf{k}$ were treated as normal random variables. For each block of data, the mean and covariance of $\mathbf{k}$ was estimated from the examples with the test block removed. The variance of the driving noise, $\mathbf{V}\boldsymbol{\nu}_i$, was also estimated. Since $\mathbf{Y}$ is the sum of two independent normal random variables, its distribution is also normal. The mean and covariance of $\mathbf{Y}$ are

$$\mathrm{E}[\mathbf{Y}] = \mathbf{QS}^{1/2}\mathrm{E}[\mathbf{k}] \tag{7.4}$$

$$\mathbf{V}[\mathbf{Y}] = \mathbf{QS}^{1/2}\mathbf{V}[\mathbf{k}](\mathbf{QS}^{1/2})^T + \mathbf{V}\boldsymbol{\nu}. \tag{7.5}$$

This distribution was used in a likelihood based procedure which resulted in 100% correct marking of the examples.

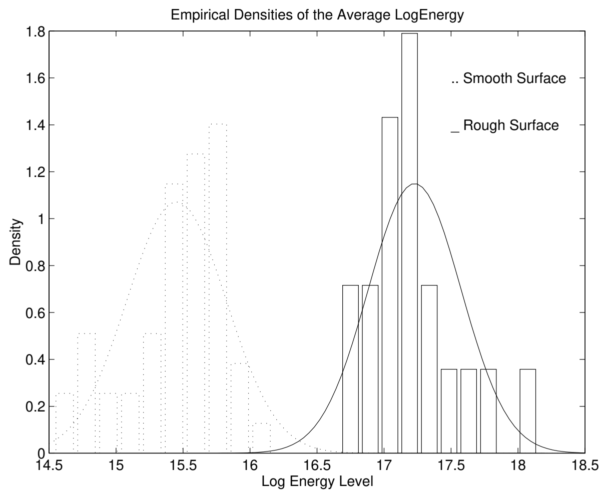


Figure 7.8: Histogram of the logarithm of the energy in the time derivative of the strains. The relative separation of the empirical densities is confirmed by the normal fit to the densities.

The relative contribution to this success of the LPC coefficients versus the magnitude of the noise is not clear. It is possible to achieve similar levels of success by testing just the magnitude of the vibration energy. Figure 7.8 shows the empirical and estimated densities of the logarithm of the energy in the time difference of the strains. This is the signal $\mathbf{y}_h$. Based on the normal fit to the histogram an error probability of $5\times\ 10^{-4}$ would be expected.

A similar series of computations was performed for experiment B (the wood and aluminum block). In this experiment, the 8 training segments were divided into blocks based on the 6 wrench measurements. Each individual wrench signal was used to generate LPC feature vectors. Each signal was again treated as independent and the equivalent binary hypothesis test was performed. In this case, the results are summarized by the confusion matrix below.

		Chosen Hypothesis	
		Wood	Al.
Correct	Wood	0.8125	0.1875
Hypothesis	Al.	0.0323	0.9677

Again much of the discrimination information is contained in the vibration magnitude. An error probability of 0.225 can be predicted from a normal fit to the average energy in $\mathbf{y}_h$. This is very close to the measured rate of 0.2198.

Based on these experiments we conclude that, at least for this type of discrimination the AR models provide only marginally more information than the information available from the log-energy. Therefore, if we are only interested in discrimination the log-energy signal is sufficient. However, if other processing will be done on the strains, perhaps in conjunction with position measurements, the AR models provide a fast, unstructured, method of segmentation. In our experiments the boundaries found by this technique corresponded to interesting physical events so the segments produced should be useful for additional processing. Furthermore, segmentation on the strain rather than the log-energy should produce better results because there are more degrees-of-freedom to fit in the estimates.

7.3 Recognition using High Frequency Models

The high frequency vibration signal provided by $\mathbf{y}_h$ clearly is an important signal for understanding manipulation. The last section showed that in two forced choice experiments quite good discrimination could be achieved purely on the mean value of this signal. This section extends this work by examining segmentation and labeling for the high frequency signal. Stationary models were represented by LPC coefficients and the driving noise variance. In addition, nonstationary impact events are represented by linear decay models.

Real-time recognition involves simultaneous segmentation and labeling of the data, against prior models, as the data is received. Recognition of high frequency events is important for isolating the beginning and end of impacts and as a secondary source of information about textures and motions. As an experimental demonstration, we looked at recognition using $\mathbf{y}_h$ against four hypotheses:

- H_0: The PHANToM is stationary and not touching anything.
- H_1: The PHANToM is moving and not touching anything.
- H_2: The sensor is being pressed against a surface, but is not moving along the surface.
- H_3: The sensor has been hit and signal represents the falling edge of an impact.

These hypotheses cover events generated by guarded moves from free space.

Fifteen training runs were performed to calibrate the four models. The first five moved the robot from $(0, 0, 0)$ to $(0, 0.7, 0.7)$ in 0.5 seconds. The acceleration of the desired trajectory is a sinusoid, so the trajectory is smooth. The sensor hit and came to rest against a block at the end of its motion. The second group of five moved the robot $(0, 0, 0)$ to $(0, 0.7, 0.7)$ in 0.5 seconds with no impact at the end. The last group moved the robot from (0,0,0) to (1,0,0) in 2.0 seconds with an impact and contact at the end.

The test set consisted of 12 runs in blocks of three. The first block moved the robot from $(0, 0, 0)$ to $(0.0, 0.7, 0.7)$ in 0.5, 1.0, and 2.0 seconds respectively with an impact and contact at the end. The second block performed the same motion without an impact and contact at the end. The third block moved from (0,0,0) to (1,0,0) in 0.5, 1.0, and 2.0 seconds respectively with an impact and contact at the end. The last block executed the same motion without an impact or contact.

The data from the training runs was hand segmented, and collected into examples for the four hypotheses. The data from the three stationary hypotheses were additionally segmented into units of 100 measurements. For each stationary example, the LPC coefficients generated by a mean and a single AR coefficient was determined. In addition, the driving noise variance was estimated. Figure 7.9 shows a two dimensional scatter plot for the two LPC coefficients over the three stationary models. An approximate estimate of the confusion matrix can be determined empirically from a normal fit to this scatter plot. The result is given below. The actual confusions are likely to be worse because the data does not appear to match the normal assumption well, but we still expect labeling performance between 80 and 90 %.

		Chosen Hypothesis		
		H0	H1	H2
Correct	H0	0.92	0.0	0.08
Hypothesis	H1	0.00	0.97	0.03
	H2	0.13	0.03	0.84

7.3.1 Recognition of High Frequency Models

As both a preliminary experiment and a useful tool, we examined recognition of high frequency events. The first experiment allowed transition between any pair of models.

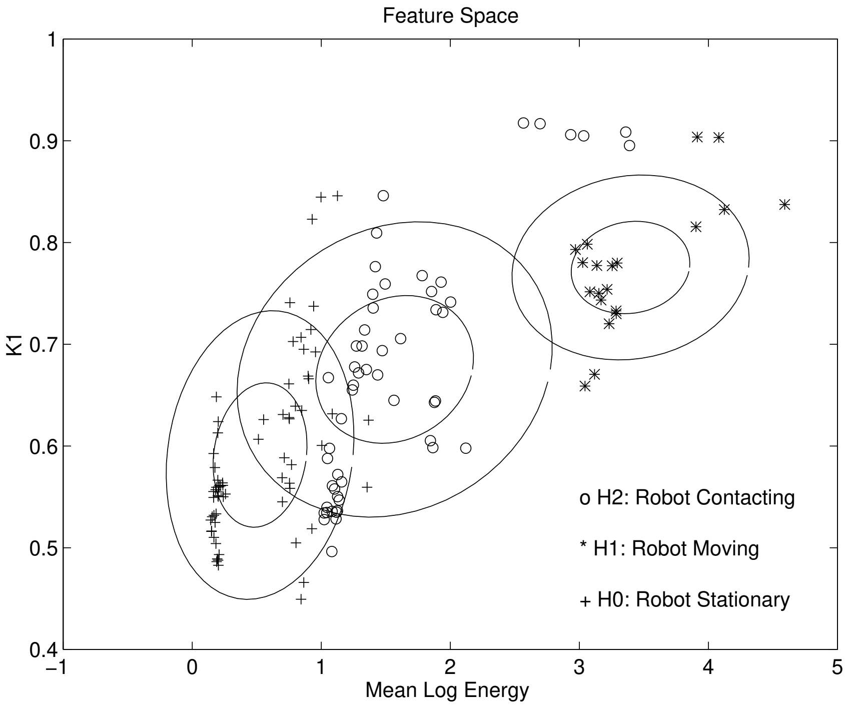


Figure 7.9: Scatter plot of the feature vectors for the three stationary high frequency models. The contours show the 1 and 2 sigma boundaries.

The context provided by the applied action and geometry was not provided. The second experiment used the applied action and geometry to constrain the allowed transitions between models. Since recognition is both segmentation and labeling, performance is reported in terms of the number of measurements that are misslabeled. Misslabeling can occur because transitions were missed or detected late, or a segment was just misslabeled. Because there are two sources of error, expected performance is lower than the estimated labeling performance.

The labeling program uses the path-based formulation discussed in chapter 5 to estimate the current model. After estimating the two LPC coefficients and driving noise variance for each segment, the mean and square-root of the covariance of LPC coefficients was stored for each model. The mean of the variance was also stored for each model. These terms were then used in the orthogonal square-root Kalman filter to generate the test residuals. The change detector used a change magnitude of 2.0 and a decision threshold of 8.0. A change of magnitude 2.0 means one standard deviation changes will cause the detector to accumulate the change statistic. A threshold of 8.0 produced an empirically good trade-off between false alarms and missed detections.

The program charts an estimate of the probability of each model, given an input feature graph and models. It also displays the raw data and statistics about the current paths and the change history. The user interface is shown in figure 7.10. The program can be run both off-line and in real-time (on a 68040) using 250 Hz data generated by the signal processing.

7.3.2 Context Free Recognition

The first experiment allowed transitions between any two models. In addition, the initial probability of each model was equal. Under these conditions the training data had the following point level confusions.

		Chosen Hypothesis			
		H0	H1	H2	H3
Correct Hypothesis	H0	6175	119	1508	0
	H1	19	2859	159	0
	H2	654	621	4279	208
	H3	0	7	12	981

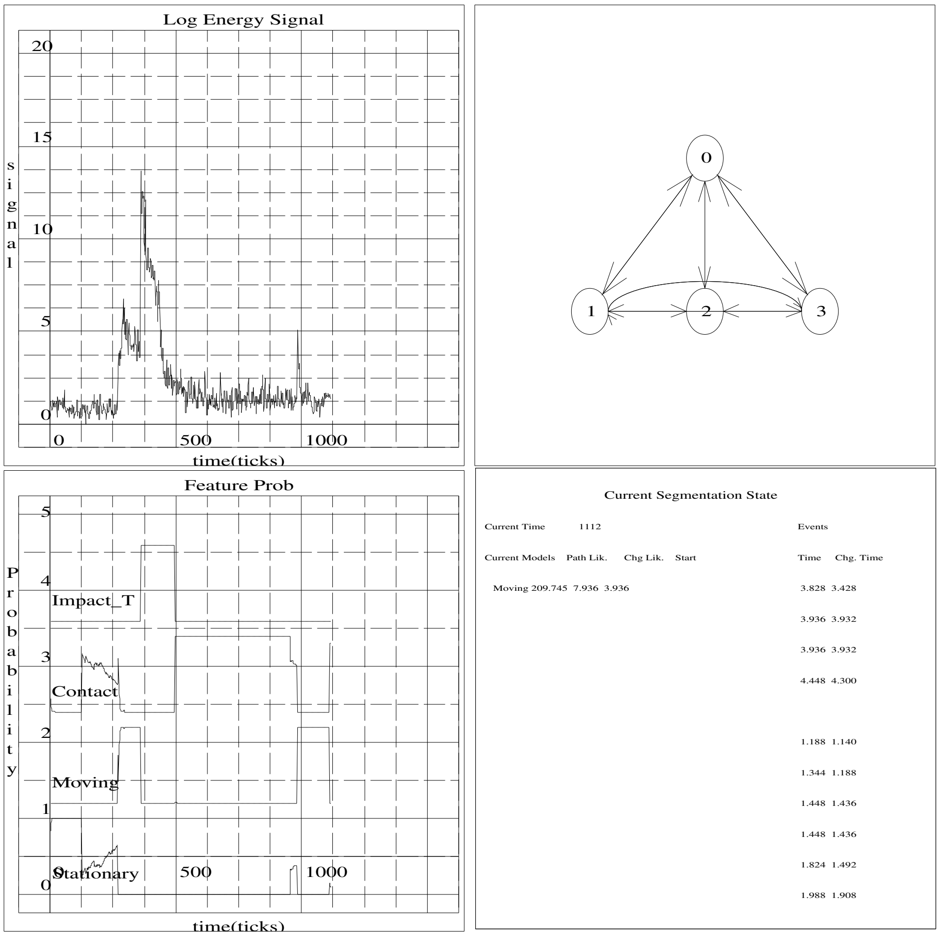


Figure 7.10: Display of the recognition program for high frequency events. The upper left figure shows the measured data. The upper right figure shows the feature graph. During execution, the current probability of each node is marked with color (not shown). The lower left shows the probability of each feature model as a function of time. The lower right shows the current models, their path log-likelihood and their change log-likelihood.

The systems marked 81% of all points correctly. Most of the errors are the expected confusions. The change detector increases the number of point confusions because models which do not quite match can take a long time to trigger a change.

Most of the required transitions are marked very closely. The system does insert extra transitions. The number of marks with a given delay over the complete training set was:

Delay	0	1	2	3	> 3
	28	8	1	3	15

Performance is lower with the test data. Over all the system marked 66% of the points correctly. The confusion matrix was:

		Chosen Hypothesis			
		H0	H1	H2	H3
Correct Hypothesis	H0	4769	86	3302	0
	H1	35	2904	222	0
	H2	794	1014	2774	101
	H3	0	49	12	405

For both the training and the test data, the major confusion is between contact and stationary. Since these two models are being separated purely on extra vibration sensed when the robot is touching an object, this is not surprising. Clearly sensing the wrench will eliminate this problem and greatly improve the scores. Moving is a sufficiently higher level of vibration that it is never confused with the stationary vibration level. Higher vibration levels which might be confusing with the moving level will result when moving along a surface. However, these can be distinguished from free space motions by the contact force.

The decision delays were similar to the training set:

Delay	0	1	2	3	> 3
	22	1	2	0	14

7.3.3 Recognition with Context

The performance of the system on the expected task can be greatly improved by providing context from the executed action and the geometry. The initial context is provided by the initial probability distribution over the models. The allowed transitions in the network encoded context of the expected sequence of models given the actions and geometry. This constrains the models to which the system can transition.

The drawback is that unexpected events will not be detected by the system. Depending on the network, these events could even cause the system to permanently deviate from the correct model. This trade-off exists any time an additional constraint is placed on the problem. The additional constraints help the path search so the change threshold parameter can be reduced and more of the paths can be searched. This will help prevent the permanent deviation problem. Also as in similar problems, a global reset can be used to restart the process when the measurements are very far from the current model.

The test set was used in a second experiment. In this case, each data set used the appropriately constrained network. In addition, the initial state was set correctly. This increased the performance to 95% correct marking of the points from 66%. Almost all the errors occured on the seventh test run when the impact was not detected and the system remained in the moving state instead of transition to the impact and then the contact state. The confusion matrix was:

		Chosen Hypothesis			
		H0	H1	H2	H3
Correct Hypothesis	H0	7981	176	0	0
	H1	18	3142	0	1
	H2	0	636	3946	101
	H3	0	47	2	405

7.4 Conclusion

This chapter showed how the theoretical ideas in chapter 5 could be applied to recognizing contact state from temporal models of the force and strain signals.

Autoregressive models were experimentally shown to be a good approach for automatically segmenting the signal into similar components. The technique used a simple local change detection test that is relatively simple to develop and apply. The decision boundaries were very close to where hand marked boundaries would be placed. In addition the boundaries remain relatively constant as the decision threshold is changed.

The AR models were of only marginal use in labeling textures. The AR coefficients do not encode substantially more information than what is available in the vibration energy, because there is not a lot of spectral content in the strain signal produced by stroking textures. However, they are still essential for producing the white residuals necessary for signal segmentation.

Because the high frequency information is so important in recognizing transient effects and textures, an experimental recognizer was developed using only the high frequency data. This recognizer was able to estimate the correct model, significantly better than chance, without context. With context, the system was quite accurate. The only substantial error was made when an impact was not detected leaving the system stuck in an earlier state.

These simple experiments demonstrated the utility of the approach. Constraint recognition has to be incorporated before a more complete task observer can be constructed. Chapter 8 investigates this problem. Finally chapter 9 presents some complete experiments that use both constraints and high frequency features in the task observer.

Constraint Estimation

Chapter 8

Constraint is fundamental to manipulation. A robot must command motions or forces which will not violate the constraints in order to avoid generating substantial, perhaps damaging, contact forces. From the representational and estimation viewpoint of this thesis, constraint is fundamental because the majority of the measured wrenches are generated by quasi-static interactions with constraints.

The field of robot force control has studied automatic compliance to constraints for many years. This large body of work has focused on two key problems: 1) representation of the constraints, and 2) writing control laws which automatically comply to the known constraints. Assuming the constraints are known can cause problems in grasping or more general manipulation. Often the geometry and pose of objects is only approximately known. For example, the geometry may have been extracted from vision. The resulting model could have substantial errors in the geometry. Grasped objects and parts that are to be mated often have large uncertainties in pose.

The problem of estimating the constraints has received very little attention. A few papers have examined contour tracking in the plane when the geometry and location of the end-effector is known [Fedele *et al.*, 1993]. A constraint estimation procedure where either the wrench of twist measurements was uses as a basis was formulated in [Bruyninckx and De Schutter, 1992], but we are unaware of any experimental results.

This chapter solves the problem of estimating and identifying the constraints themselves when both the geometry and the contact type is unknown for Cartesian spaces with linear constraints and contacts between planar polygons. The estimator returns estimates of the geometric parameters which specify the contact constraints, and a set of measurement residuals. For multiple contact problems, uniqueness issues arise in recovering the contact geometry.

The estimation problem is shown to result in a coupled generalized eigenvalue-like problem for which we give solution procedures. The resulting approach is robust to friction and random noise. For Cartesian spaces the approach can be modified and the effect of friction can be accounted for. Extending frictional modeling to rotational spaces is still a challenge. Extension of the approach to 6DOF is discussed in the conclusion. Finally the residuals from the estimator are used to determine the likelihood of different contact models. This was then applied to model-based labeling, segmentation, and recognition using the framework discussed in chapter 5.

8.1 Constraint Estimation and Residuals

Constraint estimation uses the measured wrenches and twists to best estimate the geometric parameters of the constraint equation. Alternatively the measured wrenches and positions can be used to estimate the geometric parameters. For Cartesian spaces with linear constraints, both approaches result in the same equations. Since the position estimates are less noisy this is the preferred approach. In spaces with rotations, the extension is more difficult and has not yet been solved.

This section provides an outline of constraint estimation and its relationship to constraint labeling, segmentation, and recognition. The issue of changes in scale which preserve reciprocity is also discussed. The next sections present detailed estimator solutions for particular constraint cases. This chapter uses the discussion of configuration space and constraints given in chapter 4.

As discussed in chapter 4, configuration space constraint equations are of the form

$$\mathbf{C}(\mathbf{x}, \mathbf{g}) = 0 \tag{8.1}$$

where $\mathbf{x}$ is the configuration of the robot and $\mathbf{g}$ is a vector of geometric parameters. The constraint wrench $\mathbf{w_C}$ must lie in the cotangent bundle of the constraint manifold, and the velocity of the robot $\dot{\mathbf{x}}$ must lie in the tangent bundle of the manifold. The fundamental property of constraints is that they cannot do work on an object. Therefore the reciprocal product of the $\mathbf{w_C}$ and $\dot{\mathbf{x}}$ must be zero. In Einstein sum notation[1]

$$\mathbf{w}_{\mathbf{C}i}\ \dot{\mathbf{x}}_i = 0. \tag{8.2}$$

The cotangent space is spanned by the partial derivative of $\mathbf{C}$ with respect to $\mathbf{x}$, $\mathbf{C_x}$. Every constraint wrench is formed from a linear combination of the vectors in this basis.

$$\mathbf{w}_{\mathbf{C}i} = \mathbf{C_x}_{ik}\ \boldsymbol{\lambda}_k \tag{8.3}$$

The tangent space is spanned by $\mathbf{C}^*_{\mathbf{x}}$ and every velocity is formed from a linear combination of the vector in this basis.

$$\dot{\mathbf{x}}_i = \mathbf{C_x}^*_{ij}\ \boldsymbol{\gamma}_j \tag{8.4}$$

With these definitions the reciprocity condition 8.2 yields

$$\mathbf{w}_{\mathbf{C}i}\ \dot{\mathbf{x}}_i = (\mathbf{C_x}_{ik}\ \boldsymbol{\lambda}_k)\ (\mathbf{C}^*_{\mathbf{x}ij}\ \boldsymbol{\gamma}_j) = 0 \tag{8.5}$$

for every $\boldsymbol{\lambda}$ and $\boldsymbol{\gamma}$. Therefore

$$\mathbf{C_x}_{ik}\ \mathbf{C}^*_{\mathbf{x}ij} = 0 \tag{8.6}$$

for every k and j and the bases are themselves reciprocal.

Now suppose that a change of scale is applied to each wrench and velocity term. The change of scale is expressed as

$$\mathbf{w}'_{\mathbf{C}i} = \boldsymbol{\alpha}_i\ \mathbf{w}_{\mathbf{C}i} \qquad \dot{\mathbf{x}}'_i = \boldsymbol{\beta}_i\ \dot{\mathbf{x}}_i \tag{8.7}$$

The change of scale must preserve reciprocity.

$$\mathbf{w}'_{\mathbf{C}i}\ \dot{\mathbf{x}}'_i = 0 \tag{8.8}$$

Expanding this condition and again using the fact that it must hold for all $\boldsymbol{\lambda}$ and $\boldsymbol{\nu}$ yields the condition

$$\boldsymbol{\alpha}_i\ \boldsymbol{\beta}_i\ \mathbf{C_x}_{ik}\ \mathbf{C}^*_{\mathbf{x}ij} = 0 \tag{8.9}$$

for all k and j. This condition holds if and only if $\boldsymbol{\alpha}_i\ \boldsymbol{\beta}_i = d$ is a constant for every i. A change of scale of this form conserves the measured power.

Given these preliminaries, consider the estimation problem. The first step is to change the scale of measured wrench and velocities so that the terms are unit-less. This can be done by dividing the measurements by the measurement standard deviations for each unit. Care must be taken in mixed units to obey the above reciprocity

[1] In Einstein sum notation, sums are assumed over repeated indices.

condition. In the new units, a unit-less tangent and cotangent basis can be used to produce the conditions

$$\mathbf{C}^*_{\mathbf{x}i,j}\ \mathbf{w}_{m\,i} = \boldsymbol{\nu}_{\mathbf{w}\,j} \tag{8.10}$$

$$\mathbf{C}_{\mathbf{x}i,j}\ \dot{\mathbf{x}}_{m_i} = \boldsymbol{\nu}_{\dot{\mathbf{x}}\,j} \tag{8.11}$$

on the measured wrench and velocity. These two equations are the reciprocity conditions for the measured wrench and velocity. Under ideal conditions, the indicated reciprocal product would be zero. We make the statistical assumption that the product actually results in a normal, zero mean, random error. The covariance of these two errors depends upon the contact conditions.

All of the statistical problems for constraint can now be stated. Estimation is the problem of computing, from a vector of measurements of the wrench and velocity, estimates of $\mathbf{g}$, $\mathbf{V}_{\boldsymbol{\nu}_\mathbf{w}}$, and $\mathbf{V}_{\boldsymbol{\nu}_{\dot{\mathbf{x}}}}$. Segmentation is estimation followed by a change detection test on the residuals $\boldsymbol{\nu}_\mathbf{w}$ and $\boldsymbol{\nu}_{\dot{\mathbf{x}}}$. For labeling and recognition a hypothesis test is formed over the possible values of $\mathbf{g}$, $\mathbf{V}_{\boldsymbol{\nu}_\mathbf{w}}$, and $\mathbf{V}_{\boldsymbol{\nu}_{\dot{\mathbf{x}}}}$ given all the hypotheses. Labeling uses prior parameter values and a vector of measurements to form a likelihood hypothesis test. Recognition is simultaneous labeling and segmentation.

The next two subsections discuss particular solutions to the estimation problem. The first subsection looks at the configuration space $\Re^n$ with linear constraints. The solution in this space is relatively simple, and can be used both in its own right and as an input to a more general smoothing algorithm. The second subsection looks at the configuration space of planar polygons. This space is interesting because the configuration space constraints are curved. Despite this, it is still possible to develop an approach which uses all the data in estimating the constraints by using a curvature model of the constraint surfaces.

8.1.1 Fast Solution for Linear Constraint in $\Re^k$

Estimation for a point in Cartesian space provides both a motivation for the more complex polygon solution, and a technique for doing local estimation followed by smoothing in a more general setting. We first consider contacts that have an associated stiffness, and then consider the problem of rigid constraints. Since the problem is linear, the estimator can use the position measurements instead of the velocity measurements, which is more robust. Local smoothing would require the velocity measurements.

A free point in $\Re^k$ has k free motions. Every contact with a hyper-plane in this space removes one of these free motions and introduces a relationship between the motions and wrenches. For a rigid constraint the relationship is that arbitrary forces against the constraint are possible and no motion is possible. Therefore, the rank of the allowed motions plus the allowed forces is always k.

One way to write these constraints is to represent the contact as a stiffness relationship between the true configuration and wrench

$$\mathbf{K}(\mathbf{x} - \mathbf{x}_0) = \mathbf{w} \tag{8.12}$$

where $\mathbf{x}_0$ is the zero point of the stiffness. However, this representation becomes poorly conditioned as the constraints become rigid, because elements of $\mathbf{K}$ go to infinity. The compliance representation will also break down when there is no constraint. The difficulty is that one of the two coordinates $\mathbf{x}$ or $\mathbf{w}$ has been chosen as basic and the other one has been related to this basic coordinate through the constraint. The conditioning problem can be eliminated by directly representing the constraint in the joint measurement space. In this way both coordinates are treated equally. The joint measurement representation of the relationship is

$$\mathbf{P}^T(\begin{bmatrix} \mathbf{x} \\ \mathbf{w} \end{bmatrix} - \mathbf{p}_0) = 0 \tag{8.13}$$

where $\mathbf{P}^T$ is a $(k \times 2k)$ relationship projection matrix, and $\mathbf{p}_0$ is the zero point of the relationship. The stiffness form or the compliance form of the relationship can be recovered through column operations on $\mathbf{P}^T$.

The scaled measurements will not exactly obey equation 8.13, instead the projection of each measurement $\mathbf{y}_t$ will leave a residual $\boldsymbol{\nu}_t$.

$$\mathbf{P}^T(\mathbf{y}_t - \mathbf{p}_0) = \boldsymbol{\nu}_t \tag{8.14}$$

The estimation problem is to determine the projection which minimizes the size of this residual over the measurement vector. The estimation problem can be formulated as a constrained maximum likelihood problem.

The log-likelihood of the measurement residuals is

$$l = -\frac{1}{2}(n \log \det \mathbf{V}_{\boldsymbol{\nu}} + \sum_{t=1}^{n} \boldsymbol{\nu}_t^T \mathbf{V}_{\boldsymbol{\nu}}^{-1} \boldsymbol{\nu}_t) \tag{8.15}$$

with the constraint

$$\mathbf{P}^T\mathbf{P} = \mathbf{Id}. \tag{8.16}$$

Differentiating 8.15 with respect to $\mathbf{V}_{\boldsymbol{\nu}}$ and setting the result to zero yields

$$\mathbf{V}_{\boldsymbol{\nu}} = \frac{1}{n}\sum_{t=1}^{n} \boldsymbol{\nu}_t \boldsymbol{\nu}_t^T. \tag{8.17}$$

After much further algebra, the log-likelihood can be shown to be

$$l = \frac{-n}{2}(\log\det(\mathbf{V}_{\boldsymbol{\nu}}) + k) \tag{8.18}$$

at this optimal value for $\mathbf{V}_{\boldsymbol{\nu}}$. Therefore, the optimality problem is to minimize $\log\det(\mathbf{V}_{\boldsymbol{\nu}})$ subject to 8.16.

Setting $\mathbf{p}_0$ to the mean of $\mathbf{y}_t$ minimizes the criteria with respect to $\mathbf{p}_0$ independently of $\mathbf{P}$. For a calibrated and zeroed force and velocity sensor the mean can be assumed to be zero. The mean can then be used to define an error variable $\tilde{\mathbf{y}}_t = \mathbf{y}_t - \mathbf{p}_0$. The average covariance matrix $\mathcal{I}$ is the sum of the outer products of the error variables.

$$\mathcal{I} = \frac{1}{n}\sum_{t=1}^{n} \tilde{\mathbf{y}}_t \tilde{\mathbf{y}}_t^T \tag{8.19}$$

From equation 8.14 and 8.15, the variance of the innovations is a projected version of $\mathcal{I}$.

$$\mathbf{V}_{\boldsymbol{\nu}} = \mathbf{P}^T \mathcal{I} \mathbf{P} \tag{8.20}$$

It remains to determine the optimal value of $\mathbf{P}$ subject to the constraints.

One solution is that $\mathbf{P}$ is the matrix formed by the k eigenvectors associated with the k smallest eigenvalues of $\mathcal{I}$. Each of these vectors $\mathbf{p}_i$ maps the values in $\Re^{2k}$ onto a single dimension. This can be seen geometrically from the meaning of the eigenvalues of $\mathcal{I}$. The eigenvectors form an orthogonal basis of $\Re^{2k}$ aligned with the axes of the data. The projection operator projects the data onto a k dimensional subspace, and the determinant measures the volume of this subspace. The smallest volume will be produced when the projection takes the slice of the data that is aligned with the smallest directions in the data.

The complete solution set consists of all projections which are solutions of the form $\mathbf{PR}$ that is rotations of $\mathbf{P}$. This is proved by noting that

$$\det(\mathbf{P}^T \mathcal{I} \mathbf{P}) = \det(\mathbf{R}^T \mathbf{P}^T \mathcal{I} \mathbf{P} \mathbf{R})$$

since $\det(\mathbf{R}) = 1$.

8.1.1.1 Rigid Linear Constraints in $\Re^k$

Rigid constraints introduce an additional reciprocity constraint on the log-likelihood criterion (equation 8.18). This constraint enforces the condition that the vectors in the tangent space (at a point) of a constraint must be reciprocal to the vectors in the cotangent space. Another view is that the estimator is seeking two vector bases $\mathbf{P}_{\mathcal{T}} = \{\mathbf{p}_j\}$ and $\mathbf{P}^*_{\mathcal{T}} = \{\mathbf{p}^*_k\}$ at any configuration $\mathbf{x}$ which are reciprocal. These two bases can be collected into a single orthonormal, or rotation, matrix $\bar{\mathbf{P}} = [\mathbf{P}_{\mathcal{T}}, \mathbf{P}^*_{\mathcal{T}}]$. For Cartesian spaces, with linear constraints, all configurations on the same manifold surface have the same bases. The measurements of wrench must be reciprocal to the tangent basis (up to friction), and the measurement of change in position must be reciprocal to the cotangent basis.

The constraint projection matrix $\mathbf{P}$ can be written in the form

$$\mathbf{P} = \begin{bmatrix} \mathbf{P}_{\mathcal{T}} & 0 \\ 0 & \mathbf{P}^*_{\mathcal{T}} \end{bmatrix}. \tag{8.21}$$

Therefore the log-likelihood criterion and the constraint on the projection matrix can be written as

$$\begin{aligned} l &= -\frac{n}{2} \max_{\mathbf{P}} (\log \det \hat{\mathbf{V}} + k) \quad \text{subject to} && (8.22) \\ \hat{\mathbf{V}} &= \mathbf{P}^T \mathcal{I} \mathbf{P} \text{ and} && (8.23) \\ \bar{\mathbf{P}}^T \bar{\mathbf{P}} &= \mathbf{Id} && (8.24) \end{aligned}$$

where the last equation is the reciprocity constraint.

The reciprocity constraint makes minimizing this criterion difficult, and we had to solve it numerically. An alternative criterion based on the trace is easier to manipulate and produced good estimates. Unfortunately, the statistical meaning of the criterion is unclear. Therefore it was not clear how to use the criterion in recognition where different paths consisting of segments of different lengths must be compared. We discuss the trace procedure first, and then return to the original log-likelihood criterion.

The original log-likelihood score is equal to the sum of the logarithms of the eigenvalues of the error covariance. An alternative is to minimize the sum of the eigenvalues instead of their logarithms. Since the sum of the eigenvalues is equal to the trace of a matrix, this criterion is

$$l = -\frac{1}{2} \sum_i \mathbf{p}_i^T \mathcal{I}_i \mathbf{p}_i \tag{8.25}$$

$$\text{subject to} \bar{\mathbf{P}}^T \bar{\mathbf{P}} = \mathbf{Id} \tag{8.26}$$

where $\mathbf{p}_i$ is one vector in $\bar{\mathbf{P}}$ and $\mathcal{I}_i$ is its associated covariance matrix.

Taking the derivative of equation 8.25, after adding in the constraints, with respect to each column of $\bar{\mathbf{P}}$ results in the necessary conditions

$$\mathcal{I}_i \mathbf{p}_i + \bar{\mathbf{P}} \boldsymbol{\sigma}_i = 0 \tag{8.27}$$

where $\boldsymbol{\sigma}_i$ is the i^{th} column of the constraint multipliers. Note that $\boldsymbol{\sigma}$ is a symmetric matrix.

This coupled set of eigenvalue like problems can be solved to produce two bases which are reciprocal using Newton's method. A starting value was generated by solving an eigenvalue problem for each $\mathcal{I}_i$. This produces an initial guess for $\bar{\mathbf{P}}$ which is not orthonormal. Newton's method then rapidly converges on orthonormal vectors.

Two approaches to numerically solve the log-likelihood formulation of the problem numerically were explored. The first approach was to introduce the six constraints on the projection into the log-likelihood as a vector of Lagrange multipliers. Fifteen nonlinear equations result from the first derivative necessary conditions. The equations can be simplified by introducing some auxiliary variables, increasing the number of equations and variables to twenty. The Hessian matrix for these twenty variables was then derived. An initial guess at the solution was again produced by solving an eigenvalue problem for each $\mathcal{I}_i$. The difficulty is that this initial guess produces a singular Hessian matrix. Although this was not a problem in the trace criterion (an addition of a regularizer solved it), it prevented the log-likelihood formulation from making any progress toward a solution.

The next approach was to note that the solution sought is a rotation matrix. An initial guess at a rotation matrix was generated by orthogonalizing the matrix formed by the eigenvectors. The eigenvectors were sorted based on the magnitude of their associated eigenvalue from largest to smallest. Then the Gram-Schmidt orthonormalization procedure was applied to produce an orthogonal set. Then the eigenvectors were returned to their original order to produce an initial estimate $\mathbf{R}_{\text{initial}}$. The numerical procedure then sought a rotation matrix $\tilde{\mathbf{R}}$, represented by three Euler angles, which when applied to $\mathbf{R}_{\text{initial}}$ produce the optimal basis

$$\bar{\mathbf{P}} = \mathbf{R}_{\text{initial}} \tilde{\mathbf{R}}. \tag{8.28}$$

Powell's method of minimization [Press *et al.*, 1992], which does not require derivatives, was applied because the symbolic derivatives were much too complex. Since the initial estimate is usually close to the optimal estimate, Powell's method converges in a few iterations.

8.1.1.2 Applying Constraint Estimation in $\Re^k$

Both stiffness-based and constraint-based estimation were applied to test data. The PHANToM with the fingertip sensor was moved by hand to produce four groups of four experimental runs. The first group is motion in free space, the second group is motion along a single flat plane made of wood, the third group is motion along a line formed by the plane of wood and an aluminum block, the last group is a fixed contact in a corner formed by the wood and two aluminum blocks. Sixteen hundred samples of the low-passed force and position data were sampled at 250 Hz.[2]

To investigate the effect of sample size on measurement accuracy, each group of four experimental runs was lumped into a single block of data. The measurement accuracy for the position and force measurements were then used to normalize the units of each measurement. The effects of friction where then removed from every sample using the following filter

$$\tilde{\mathbf{w}}_i = \left(\mathbf{Id} - \frac{\dot{\mathbf{x}}_i \dot{\mathbf{x}}_i^T}{\dot{\mathbf{x}}_i^T \dot{\mathbf{x}}_i}\right) \mathbf{w}_{mi}. \tag{8.29}$$

This filter computes the power being expended by the motion, and normalizes it to get a frictional force. The velocity then gives the direction of the frictional force. This filter improves performance for the stiffness approach.

Samples were then randomly drawn from the block of measurements to create an example for regression. Ten examples were drawn for each sample size. The stiffness regression was applied to the data and the estimated frame was returned in $\hat{\mathbf{P}}$. This was compared to an assumed true $\mathbf{P}$ by computing the singular value decomposition of $\hat{\mathbf{P}}^T\mathbf{P} = \mathbf{U\Sigma V}$. If the two projection matrices were equivalent, the singular values would all be one. Since the singular values are the direction cosines of the misalignment, any deviation from one can be mapped to a positive angular error. We assumed a log-normal distribution for the deviation of the minimum singular value from 1. Table 8.1 shows the mean and standard deviation (times $\sqrt{n}$) of this log-normal distribution for each contact case, and the equivalent average angular error. Figure 8.1 graphically shows the effect of sample size on the estimated error.

[2]Chapter 7 describes the signal conditioning on the force signal

Number of Constraints	0	1	2	3
Mean of Log-Normal μ	-10.9	-5.09	-4.00	-10.0
Std of Log-Normal $\sigma\sqrt{n}$	1.3	11.0	23.2	2.2
Equivalent Mean Angular Err. (deg)	0.34	6.34	11.0	0.53

Table 8.1: Error statistics for stiffness based estimation in $\Re^3$ using position and force measurements and friction compensation.

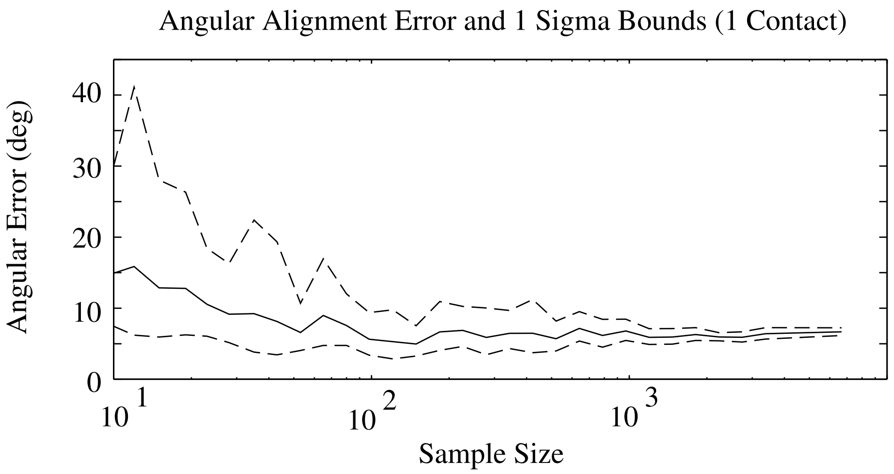


Figure 8.1: Angular alignment accuracy of the stiffness estimation procedure with 1 contact as a function of the number of samples. Measurements of position and force compensated for friction were used in the estimation.

Number of Constraints	0	1	2	3
Mean of Log-Normal μ	-10.6	-.228	-1.78	-10.0
Std of Log-Normal $\sigma\sqrt{n}$	1.54	6.37	5.89	1.9
Equivalent Mean Angular Err. (deg)	0.40	78.2	33.7	0.55

Table 8.2: Error statistics for stiffness based estimation in $\Re^3$ using position and force measurements without friction compensation.

The data shows, as expected, that both the full contact and no contact case produce very accurate results with little deviation. The contact cases have some bias and both have significant variance. A plot of the position data shows that the wood surface was not flat. For some data sets an $11°$ slope can be observered. How much of this is kinematic error in the PHANToM and how much is actual alignment error would require more careful measurements. It is very possible that the kinematic chain from the block, to the table, to the PHANToM, and finally to the fingertip has $1° - 2°$ of angular error.

Secondly, the data from the experiment with two contacts is not perfectly straight. There is enough kinematic error in the robot to produce a 1 mm deviation from a straight-line fit to the x and y position data over a maximum of 80 mm. In addition, the significantly greater standard deviation in the measurements is probably due to friction and vibration.

For comparison the same experiments were run without first preprocessing the data to account for friction. Table 8.2 shows the results. As the table indicates, stiffness based estimation without friction compensation is terrible. Friction completely contaminates the measurements and for one contact produced an estimate that is almost 90 degrees opposed to the correct normal.

The constraint based estimator was also tested. The same sampling procedure was used to generate examples for the constraint based estimator. The results are significantly improved by adding the information on the number of constraints. Table 8.3 summarizes the results. Figure 8.2 shows the performance of this estimator as a function of sample size. Even the results without friction compensation, and estimation based on velocities are quite accurate. Tables 8.4 and 8.5 show the results for estimation without friction compensation, and for velocity based estimation with friction compensation.

Number of Constraints	0	1	2	3
Mean of Log-Normal μ	NA	-8.5	-10.2	NA
Std of Log-Normal $\sigma\sqrt{n}$	NA	7.3	0.38	NA
Equivalent Mean Angular Err. (deg)	0.0	1.2	0.5	0.0

Table 8.3: Error statistics for constraint based estimation in $\Re^3$ using position and force measurements with friction compensation.

Number of Constraints	0	1	2	3
Mean of Log-Normal μ	NA	-8.19	-10.2	NA
Std of Log-Normal $\sigma\sqrt{n}$	NA	7.2	0.36	NA
Equivalent Mean Angular Err. (deg)	0.0	1.34	0.5	0.0

Table 8.4: Error statistics for constraint based estimation in $\Re^3$ using position and force measurements without friction compensation.

Number of Constraints	0	1	2	3
Mean of Log-Normal μ	NA	-10.5	-9.9	NA
Std of Log-Normal $\sigma\sqrt{n}$	NA	10.6	8.7	NA
Equivalent Mean Angular Err. (deg)	0.0	0.43	0.59	0.0

Table 8.5: Error statistics for constraint based estimation in $\Re^3$ using velocity and force measurements with friction compensation.

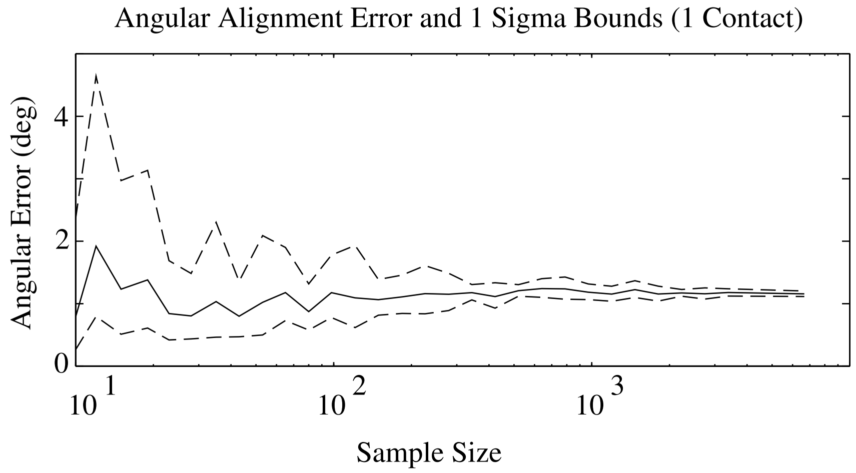


Figure 8.2: Angular alignment accuracy of the constraint estimation procedure with 1 contact as a function of the number of samples. Measurements where positions and forces compensated for friction. Note the significant improvement in performance over the stiffness approach.

The completely free and fixed case are fit without any error (indicated by NA), since there is nothing for the estimator to estimate. The partial constraint cases are also significantly improved.

If accurate estimates are required for assembly or other mating tasks, it may be worth the additional computational cost to use the constraint based approach. The eigenvalue computations are easier in the constraint approach since two $k \times k$ systems have to be solved, versus one $2k \times 2k$ system. However there is additional computational cost in the numerical iterations. In the test cases, this computation generally converged in two to three steps from the initial estimate, but each step requires searching for a minimum in three different directions.

8.1.2 Estimation for Planar Polygons

The configuration space for planar polygons is $\mathcal{C} = \Re^2 \times SO(2)$. Each configuration can be represented by a homogeneous transform $\mathbf{T}$. The rotation component $\mathbf{R}$ of the transform can be identified with a single angle θ, therefore the configuration space coordinate $\mathbf{x}$ is equal to the triple (r_x, r_y, θ). The position component (r_x, r_y)

will be represented by the vector $\mathbf{r}$. With these conventions, the planar rotation is given by

$$\mathbf{R}(\theta) = \begin{bmatrix} \cos(\theta) & -\sin(\theta) \\ \sin(\theta) & \cos(\theta) \end{bmatrix} = \begin{bmatrix} \mathrm{c} & -\mathrm{s} \\ \mathrm{s} & \mathrm{c} \end{bmatrix}, \tag{8.30}$$

where c and s are abbreviations for cos and sin. This representation for the configuration space will make some of the derivations easier.

For polygons any contact constraint consists of contacts between one or more points and lines. Let $\bar{\mathbf{p}}$ be the homogeneous coordinates of a point in space. Let $\bar{\mathbf{n}} = [\,\mathbf{n}, -d\,]$ be the homogeneous coordinates of a hyper-plane (line in 2D) with $\mathbf{n}$ a unit vector, and d the distance to the plane measured along the normal. With this notation the basic constraint that a point lie on the plane is

$$^{0}\bar{\mathbf{n}}^{T}\; {}^{0}\bar{\mathbf{p}} = 0. \tag{8.31}$$

This general constraint becomes a function of the configuration of the polygon. There are only two basic types of contact that can occur with polygons. Type B contact is contact between a vertex on the moving polygon and an edge on the fixed obstacles. Type A contact is contact between an edge on the moving polygon and a vertex on the fixed obstacles. Type B constraints are most easily expressed in the fixed frame, and by symmetry type A constraints are most easily expressed in the moving frame. However, for uniformity, all the constraints will be given in the fixed frame.

For type B contacts the general constraint becomes

$$^{0}\bar{\mathbf{n}}^{T}\; {}^{0}\mathbf{T}_{m}\; {}^{m}\bar{\mathbf{p}} = 0. \tag{8.32}$$

In this equation ${}^{m}\bar{\mathbf{p}}$ is the coordinates of the contact vertex expressed relative to the moving frame, ${}^{0}\mathbf{T}_{m}$ is the transform taking the local coordinates to the global coordinates, and ${}^{0}\bar{\mathbf{n}}$ is in the fixed frame. This equation can be expanded to

$$^{0}\mathbf{n}^{T}\; ({}^{0}\mathbf{R}_{m}\; \mathbf{p} + \mathbf{r}) - d = 0. \tag{8.33}$$

For type A contact the constraint is

$$^{m}\bar{\mathbf{n}}^{T}\; {}^{0}\mathbf{T}_{m}^{-1}\; {}^{0}\bar{\mathbf{p}} = 0. \tag{8.34}$$

In this equation ${}^{0}\bar{\mathbf{p}}$ is the coordinates of the contact vertex expressed relative to the fixed frame, ${}^{m}\bar{\mathbf{n}}$ is in the moving frame, and ${}^{0}\mathbf{T}_{m}^{-1}$ is the transform taking global coordinates to local coordinates. It is the inverse of ${}^{0}\mathbf{T}_{m}$. This equation expands to

$$^{m}\mathbf{n}^{T}\; {}^{0}\mathbf{R}_{m}^{T}\; (\mathbf{p} - \mathbf{r}) - d = 0. \tag{8.35}$$

The difference between the two constraint equations is whether ${}^0\mathbf{R}_m$ is applied or not applied to $\mathbf{r}$.

Both constraint equations involve a normal. To simplify the following notation we will drop the frame superscript. The choice of frame should be clear from the contact type.

8.1.2.1 Constraint Tangents and Cotangents

This section derives the constraint tangents and cotangents for both type A and type B contacts. It is shown that all the tangent bundles can be written as a projection of the product of a 4x4 curvature matrix with a constant 4 vector. This unifying view of the problem allows us to derive a single general form for the constraint estimator in the next section. In all of these equations $\mathbf{R} = {}^0\mathbf{R}_m$ to simplify the notation.

8.1.2.2 Type B

Taking the derivative of equation 8.33 with respect to $\mathbf{x}$ gives the cotangent vector

$$\mathbf{Y}_B(\mathbf{x}) = \begin{bmatrix} \mathbf{n} \\ \mathbf{n}^T\mathbf{X}\mathbf{R}\mathbf{p} \end{bmatrix} \tag{8.36}$$

where $\mathbf{X} = \begin{bmatrix} 0 & -1 \\ 1 & 0 \end{bmatrix}$ and is the cross product matrix. $\mathbf{X}$ has the properties

- $\mathbf{y}^T\mathbf{X}\mathbf{y} = 0$ for any vector $\mathbf{y}$
- $\mathbf{X}^T = -\mathbf{X}$
- and $\mathbf{X}\mathbf{R} = \mathbf{R}\mathbf{X}$, i.e. $\mathbf{X}$ commutes with rotations.

Expanding out the term $\mathbf{n}^T\mathbf{X}\mathbf{R}\mathbf{p}$ results in the expression

$$\begin{aligned} \mathbf{n}^T\mathbf{X}\mathbf{R}\mathbf{p} &= \mathbf{n}^T\mathbf{X}\mathbf{p}\cos(\theta) - \mathbf{n}^T\mathbf{p}\sin(\theta) & (8.37) \\ &= [1 \quad 0]\,\mathbf{R}\mathbf{a}, & (8.38) \end{aligned}$$

where $a_1 = \mathbf{n}^T\mathbf{X}\mathbf{p}$ and $a_2 = \mathbf{n}^T\mathbf{p}$. This means that $\mathbf{Y}_B(\mathbf{x})$ can be rewritten as

$$\mathbf{Y}_B(\mathbf{x}) = \mathbf{P}\begin{bmatrix}\mathbf{Id} & 0 \\ 0 & \mathbf{R}\end{bmatrix}\begin{bmatrix}\mathbf{n} \\ \mathbf{a}\end{bmatrix} = \mathbf{P}\ \mathbf{C}_{Y_B}\ \mathbf{Y}_B, \tag{8.39}$$

where $\mathbf{Id}$ is the identity matrix and $\mathbf{P}$ is a projection matrix which keeps the first three components. $\mathbf{Y}_B$ will refer to the constant four vector and $\mathbf{Y}_B(\mathbf{x})$ will refer to the function. The matrix $\mathbf{C}_{Y_B}$ encodes the curvature of the cotangent vectors and plays a critical role in the estimation problem.

The tangent vectors $\mathbf{J}_{B_1}$ and $\mathbf{J}_{B_2}$ are reciprocal to $\mathbf{Y}_B$

$$\mathbf{J}_{B_1}(\mathbf{x}) = \begin{bmatrix}\mathbf{X}\mathbf{n} \\ 0\end{bmatrix} \qquad \mathbf{J}_{B_2}(\mathbf{x}) = \begin{bmatrix}\mathbf{R}\mathbf{X}^T\mathbf{p} \\ 1\end{bmatrix}, \tag{8.40}$$

and these can also be written as products of the projection matrix and curvature matrices as

$$\mathbf{J}_{B_1}(\mathbf{x}) \quad = \quad \mathbf{P}\begin{bmatrix}\mathbf{Id} & 0 \\ 0 & \mathbf{Id}\end{bmatrix}\begin{bmatrix}\mathbf{X}\mathbf{n} \\ 0\end{bmatrix} = \mathbf{P}\ \mathbf{C}_{J_{B1}}\ \mathbf{J}_{B_1} \tag{8.41}$$

$$\mathbf{J}_{B_2}(\mathbf{x}) \quad = \quad \mathbf{P}\begin{bmatrix}\mathbf{R} & 0 \\ 0 & \mathbf{Id}\end{bmatrix}\begin{bmatrix}\mathbf{X}^T\mathbf{p} \\ 1 \\ 0\end{bmatrix} = \mathbf{P}\ \mathbf{C}_{J_{B2}}\ \mathbf{J}_{B_2}. \tag{8.42}$$

8.1.2.3 Type A

Taking the derivative of equation 8.35 with respect to $\mathbf{x}$ gives the cotangent vector

$$\mathbf{Y}_A(\mathbf{x}) = \begin{bmatrix}-\mathbf{R}\mathbf{n} \\ \mathbf{n}^T\mathbf{X}^T\mathbf{R}^T(\mathbf{p}-\mathbf{r})\end{bmatrix} \tag{8.43}$$

Expanding out the term $\mathbf{n}^T\mathbf{X}^T\mathbf{R}^T\mathbf{p}$ results in the expression

$$\mathbf{n}^T\mathbf{X}^T\mathbf{R}^T\mathbf{p} = \mathbf{n}^T\mathbf{X}^T\mathbf{p}\cos(\theta) - \mathbf{n}^T\mathbf{p}\sin(\theta). \tag{8.44}$$

Using the previous definitions of a_1 and a_2 shows that $\mathbf{Y}_A(\mathbf{x})$ can be rewritten as

$$\mathbf{Y}_A(\mathbf{x}) = \mathbf{P}\begin{bmatrix}\mathbf{R} & 0 \\ \begin{bmatrix}\mathbf{r}^T\mathbf{X}\mathbf{R} \\ 0\end{bmatrix} & \mathbf{R}\end{bmatrix}\begin{bmatrix}-\mathbf{n} \\ \mathbf{a}\end{bmatrix} = \mathbf{P}\ \mathbf{C}_{Y_A}\ \mathbf{Y}_A. \tag{8.45}$$

The tangent vectors $\mathbf{J}_{A_1}$ and $\mathbf{J}_{A_2}$ are orthogonal to $\mathbf{Y}_A$

$$\mathbf{J}_{A_1}(\mathbf{x}) = \begin{bmatrix}\mathbf{R}\mathbf{X}^T\mathbf{n} \\ 0\end{bmatrix} \qquad \mathbf{J}_{A_2}(\mathbf{x}) = \begin{bmatrix}-\mathbf{X}(\mathbf{p}-\mathbf{r}) \\ 1\end{bmatrix}, \tag{8.46}$$

and these can also be written as products of the projection matrix and curvature matrices as

$$\mathbf{J}_{A_1}(\mathbf{x}) = \mathbf{P}\begin{bmatrix}\mathbf{R} & 0\\ 0 & \mathbf{Id}\end{bmatrix}\begin{bmatrix}\mathbf{X}^T\mathbf{n}\\ 0\end{bmatrix} = \mathbf{P}\ \mathbf{C}_{J_{A1}}\ \mathbf{J}_{A_1} \tag{8.47}$$

$$\mathbf{J}_{A_2}(\mathbf{x}) = \mathbf{P}\begin{bmatrix}\mathbf{Id} & [\mathbf{Xr}\ \ 0]\\ 0 & \mathbf{Id}\end{bmatrix}\begin{bmatrix}-\mathbf{Xp}\\ 1\\ 0\end{bmatrix} = \mathbf{P}\ \mathbf{C}_{J_{A2}}\ \mathbf{J}_{A_2}. \tag{8.48}$$

8.1.3 General Case

The dimension of the tangent and cotangent vector space, at a point, is always three for planar rigid bodies. That implies that for a single constraint the cotangent vector can be computed from the cross product of the two tangent vectors up to sign. With two constraints there are two cotangent vectors and only one tangent vector. Therefore, the tangent vector can be computed as the cross product of the two cotangent vectors.

Therefore, we can always choose two vectors $(\mathbf{U_1}, \mathbf{U_2})$ as the base vectors for the appropriate bundle. Let $\mathbf{C}_1$ and $\mathbf{C}_2$ be their respective curvature matrices. Then the third vector $\mathbf{V}(x)$ can be generated at any configuration by

$$\mathbf{V}(\mathbf{x}) = \mathbf{U_1}(\mathbf{x}) \times \mathbf{U_2}(\mathbf{x}) = \bar{\mathbf{C}}(\mathbf{C}_1(\mathbf{x}), \mathbf{C}_2(\mathbf{x}))\ \mathbf{U_1} \otimes \mathbf{U_2} \tag{8.49}$$

where $\otimes$ is the tensor product of $\mathbf{U_1}$ and $\mathbf{U_2}$ arranged into an appropriate vector, and $\bar{\mathbf{C}}$ is the appropriate rearrangement of the two curvature matrices to give the required result. For any definition of arrangement of $\otimes$ and any $3 \times n$ matrices $\mathbf{X}$ and $\mathbf{Y}$

$$\bar{\mathbf{C}}(\mathbf{X}, \mathbf{Y}) = \begin{bmatrix}\mathbf{X}_2 \otimes \mathbf{Y}_3 - \mathbf{X}_3 \otimes \mathbf{Y}_2\\ \mathbf{X}_3 \otimes \mathbf{Y}_1 - \mathbf{X}_1 \otimes \mathbf{Y}_3\\ \mathbf{X}_1 \otimes \mathbf{Y}_2 - \mathbf{X}_2 \otimes \mathbf{Y}_1\end{bmatrix}$$

where $\mathbf{X}_i$ denotes the i^{th} row of $\mathbf{X}$. As a further notational convenience let $\mathbf{U_1} \otimes \mathbf{U_2} = T_e(\mathbf{U_2})\,\mathbf{U_1} = T'_e(\mathbf{U_1})\,\mathbf{U_2}$, where T_e and T'_e are appropriate arrangements of the terms. The rest of this development will use these general definitions.

8.1.3.1 Equivalent Systems of Two Contacts

Before presenting the solution for the contact estimation problem, this section discusses the question of uniqueness. For two or more contacts the geometric parameters which specify the constraint manifold are no longer unique. There are three separate cases: 1) two contacts along a line, 2) two same type contacts not in a line, and 3) two different type contacts not in a line.

The solutions are not unique because for any multiple contact situation there exists an infinite set of mechanisms which produce the same configuration space curve. This is very clear for contact along a line. Contact along a line creates a straight line constraint in $\mathcal{C}$. The same constraint is created by any pair of collinear sliders, with the same angle, attached anywhere on the body. Therefore, all such sliders are equivalent.

The problem of equivalent mechanisms is not a special problem of line-line contacts. Figure 8.3 shows two type B contacts on a body, the associated configuration space, and one equivalent mechanism. The motion of $\mathbf{x}$ is a straight line in Cartesian space and a helix in $\mathcal{C}$. Kinematically, the contact set acts as a pair of pivots attached to two fixed sliders. For any choice of direction for one of the sliders there exists a unique second direction and two attachment points on the body which will produce exactly the same configuration space surface.

To prove this, let $\mathbf{Y}_n^T = [\, n_1 \quad n_2 \quad a_1 \quad a_2 \,]$ and $\mathbf{Y}_m^T = [\, m_1 \quad m_2 \quad b_1 \quad b_2 \,]$ be the two bases for the two contact wrenches. Computing the twist freedom $\mathbf{J}$ from the cross product $\mathbf{J}(\mathbf{x}) = \mathbf{Y}_n(\mathbf{x}) \times \mathbf{Y}_m(\mathbf{x})$ yields

$$\begin{aligned}
\mathbf{J}_{(B,B)}(\mathbf{x}) &= \begin{bmatrix} (n_2 b_1 - m_2 a_1) & (m_2 a_2 - n_2 b_2) & 0 \\ (m_1 a_1 - n_1 b_1) & (n_1 b_2 - m_1 a_2) & 0 \\ 0 & 0 & (n_1 m_2 - n_2 m_1) \end{bmatrix} \begin{bmatrix} \mathrm{c} \\ \mathrm{s} \\ 1 \end{bmatrix} && (8.50) \\
&= \mathbf{J}_{\mathbf{T}(B,B)} \mathbf{f}_{(B,B)}(\mathbf{x}). && (8.51)
\end{aligned}$$

Therefore equivalence of two mechanisms is therefore the same as equivalence of the induced matrix $\mathbf{J}_{\mathbf{T}(B,B)}$. We term this matrix the tangent space tableau.

For any $\mathbf{n}$, $\mathbf{J}_{\mathbf{T}(B,B)}(3,3) = z_5$ determines the value of a corresponding $\mathbf{m}$. Then an equivalent matrix can be created by solving the equation

$$\begin{bmatrix} -m_2 & 0 & n_2 & 0 \\ 0 & m_2 & 0 & -n_2 \\ m_1 & 0 & -n_1 & 0 \\ 0 & -m_1 & 0 & n_1 \end{bmatrix} \begin{bmatrix} a_1 \\ a_2 \\ b_1 \\ b_2 \end{bmatrix} = \begin{bmatrix} \mathbf{J}_{\mathbf{T}(B,B)}(1,1) \\ \mathbf{J}_{\mathbf{T}(B,B)}(1,2) \\ \mathbf{J}_{\mathbf{T}(B,B)}(2,1) \\ \mathbf{J}_{\mathbf{T}(B,B)}(2,2) \end{bmatrix}. \qquad (8.52)$$

The determinant of this matrix is $(m_1 n_2 - m_2 n_1)^2 = (z_5)^2$. Therefore, the solution is unique as long as $\mathbf{m} \neq \mathbf{n}$. That is the constraint is not line-line contact. This shows

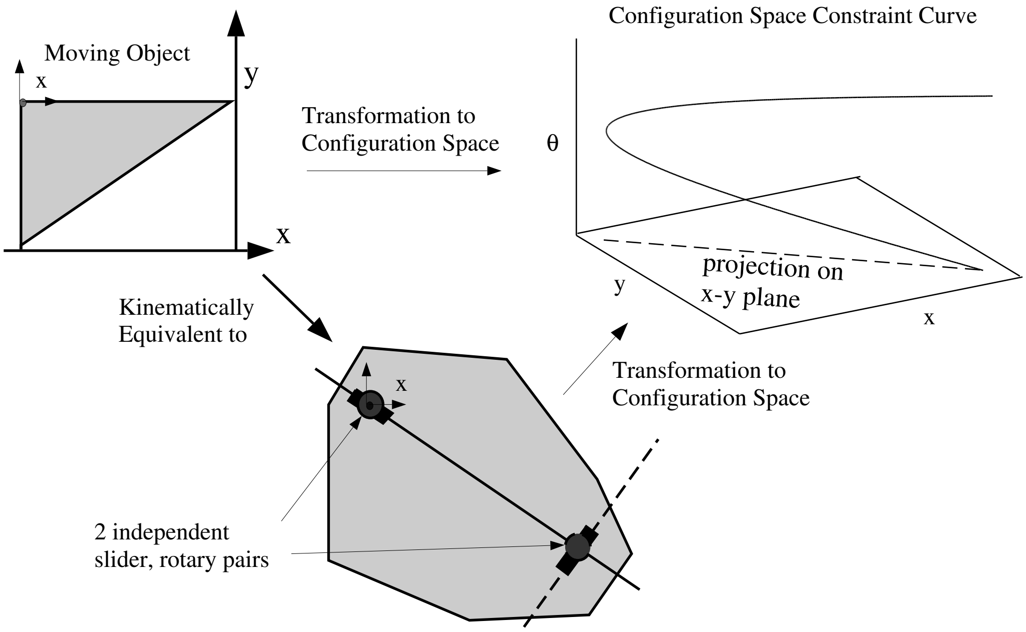


Figure 8.3: The configuration space for two non-collinear type B contacts and an equivalent mechanism. The triangular part is contacting two walls. Motions that obey these two constraints cause the control frame **x** to move in a spiral in configuration space. The same spiral can be produced with the indicated mechanism.

that type (B,B) contact is unique only up to the initial choice of one of the contact normals. Furthermore, it shows that for linear constraint the contact locations are arbitrary.

A similar result holds for the other contact pairings. For type (A, A) the tangent space tableau and vector are

$$\begin{bmatrix} \frac{(-\mathbf{a}^T\Delta\mathbf{m}+\mathbf{b}^T\Delta\mathbf{n})}{2} & \frac{(\mathbf{a}^T\mathbf{Z}\mathbf{m}-\mathbf{b}^T\mathbf{Z}\mathbf{n})}{2} & 0 & -\mathbf{m}^T\mathbf{X}\mathbf{n} & \frac{(\mathbf{a}^T\mathbf{X}\mathbf{m}-\mathbf{b}^T\mathbf{X}\mathbf{n})}{2} \\ -\frac{(\mathbf{a}^T\mathbf{Z}\mathbf{m}-\mathbf{b}^T\mathbf{Z}\mathbf{n})}{2} & \frac{(-\mathbf{a}^T\Delta\mathbf{m}+\mathbf{b}^T\Delta\mathbf{n})}{2} & \mathbf{m}^T\mathbf{X}\mathbf{n} & 0 & \frac{(\mathbf{a}^T\mathbf{m}-\mathbf{b}^T\mathbf{n})}{2} \\ 0 & 0 & 0 & 0 & \mathbf{m}^T\mathbf{X}\mathbf{n} \end{bmatrix} \quad (8.53)$$

$$\mathbf{J}_{(A,A)}(\mathbf{x}) = \mathbf{J}_{\mathbf{T}(A,A)} \begin{bmatrix} \cos(2\theta) & \sin(2\theta) & r_1 & r_2 & 1 \end{bmatrix}^T \quad (8.54)$$

where $Z = \begin{bmatrix} -1 & 0 \\ 0 & 1 \end{bmatrix}$, $\Delta = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}$, and $(\mathbf{a}, \mathbf{b}, \mathbf{n}, \mathbf{m})$ are the appropriate 2 vectors.

Finally, for type (A, B) the result is

$$\mathbf{J}_{\mathbf{T}(A,B)}^T = \begin{bmatrix} \frac{-(\mathbf{a}^T\Delta\mathbf{m})}{2} & \frac{-(\mathbf{a}^T\mathbf{Z}\mathbf{m})}{2} & 0 \\ \frac{(\mathbf{a}^T\mathbf{Z}\mathbf{m})}{2} & \frac{-(\mathbf{a}^T\Delta\mathbf{m})}{2} & 0 \\ -m_2n_2 & m_2n_1 & 0 \\ m_1n_2 & -m_1n_1 & 0 \\ -m_1n_2 & m_1n_1 & 0 \\ m_-2n_2 & m_2n_1 & 0 \\ b_1n_2 & -b_1n_1 & \mathbf{m}^T\mathbf{X}\mathbf{n} \\ -b_2n_2 & b_2n_1 & \mathbf{m}^T\mathbf{n} \\ \frac{(\mathbf{a}^T\mathbf{X}\mathbf{m})}{2} & \frac{(\mathbf{a}^T\mathbf{m})}{2} & 0 \end{bmatrix} \quad (8.55)$$

$$\mathbf{J}_{(A,B)}(\mathbf{x}) = \mathbf{J}_{\mathbf{T}(A,B)} \begin{bmatrix} \cos(2\theta) \\ \sin(2\theta) \\ r_1\cos(\theta) \\ r_2\cos(\theta) \\ r_1\sin(\theta) \\ r_2\sin(\theta) \\ \cos \\ \sin \\ 1 \end{bmatrix} \quad (8.56)$$

8.1.3.2 The Planar Constraint Estimator

All of the planar constraint problems can be stated using the appropriate basis vector pair $(\mathbf{U_1}, \mathbf{U_2})$ and their associated curvature matrices. The third vector $\mathbf{V}$ can be formed from the cross-product of these two vectors. We consider only the trace criterion 8.25 for the planar polygon problem. In this problem, we have to define three matrices:

$$\mathcal{I}_u = \frac{1}{n}\sum_i \bar{\mathbf{C}}(\mathbf{x}_i)^T\mathbf{u}(\mathbf{i})\mathbf{u}(\mathbf{i})^T\bar{\mathbf{C}}(\mathbf{x}_i) \tag{8.57}$$

$$\mathcal{I}_{v_1} = \frac{1}{n}\sum_i \mathbf{C}_1(\mathbf{x}_i)^T\mathbf{v}(\mathbf{i})\mathbf{v}(\mathbf{i})^T\mathbf{C}_1(\mathbf{x}_i) \tag{8.58}$$

$$\mathcal{I}_{v_2} = \frac{1}{n}\sum_i \mathbf{C}_2(\mathbf{x}_i)^T\mathbf{v}(\mathbf{i})\mathbf{v}(\mathbf{i})^T\mathbf{C}_2(\mathbf{x}_i). \tag{8.59}$$

The first matrix can be used to compute the projection of the measurements onto $\mathbf{V}$, but $\mathbf{V}$ is being determined by the cross-product of $\mathbf{U_1}$ and $\mathbf{U_2}$. The second two matrices can be used to compute the projections of the $\mathbf{v}_i$ measurements onto $\mathbf{U_1}$ and $\mathbf{U_2}$. Using these definitions the criterion to be maximized can be written as

$$l = \mathbf{V}^T\mathcal{I}_u\mathbf{V} + \mathbf{U_1}^T\mathcal{I}_{v_1}\mathbf{U_1} + \mathbf{U_2}^T\mathcal{I}_{v_2}\mathbf{U_2} \tag{8.60}$$

Finally, define

$$\bar{\mathbf{I}} = \begin{bmatrix} \mathbf{Id} & 0 \\ 0 & 0 \end{bmatrix}$$

and

$$\underline{\mathbf{I}} = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}.$$

The constraints can now be added to create the final optimization problem

$$\begin{aligned} l^* &= \mathbf{V}^T\mathcal{I}_u\mathbf{V} + \mathbf{U_1}^T\mathcal{I}_{v_1}\mathbf{U_1} + \mathbf{U_2}^T\mathcal{I}_{v_2}\mathbf{U_2} \qquad (8.61) \\ &+\frac{1}{2}\eta_1(\mathbf{U_1}^T\mathbf{C}_1\mathbf{U_1} - 1) + \frac{1}{2}\eta_2(\mathbf{U_2}^T\mathbf{C}_2\mathbf{U_2} - 1) - \boldsymbol{\gamma}_1^T\mathbf{D}_1\mathbf{U_1} - \boldsymbol{\gamma}_2^T\mathbf{D}_2\mathbf{U_2}. \end{aligned}$$

The $\mathbf{C}_i$ are one of the two constraint matrices $\bar{\mathbf{I}}$ or $\underline{\mathbf{I}}$ depending upon the constraint type. The $\mathbf{D}_i$ are additional constraint conditions that arise in the single contact case because in this case $\mathbf{U_1}$ and $\mathbf{U_2}$ have constrained forms, and $\boldsymbol{\gamma}_i$ are their associated Lagrange multipliers.

Differentiating and using the different expressions for $\otimes$ yields the two necessary conditions

$$T_e(\mathbf{U_2})^T\,\mathcal{I}_u\,T_e(\mathbf{U_2})\mathbf{U_1} + \mathcal{I}_{v_1}\,\mathbf{U_1} + \mathbf{D}_1^T\,\gamma_1 = \eta_1\mathbf{C}_1\,\mathbf{U_1} \tag{8.62}$$

$$T'_e(\mathbf{U_1})^T\,\mathcal{I}_u\,T'_e(\mathbf{U_1})\mathbf{U_2} + \mathcal{I}_{v_2}\,\mathbf{U_2} + \mathbf{D}_2^T\,\gamma_2 = \eta_2\mathbf{C}_2\,\mathbf{U_2}. \tag{8.63}$$

The result is two coupled generalized eigenvalue like problems which must be solved simultaneously. The solution procedure depends upon the particular contact type.

8.1.3.3 One Contact

In the one type B contact case $\mathbf{U_1} = \mathbf{J}_{B_1}$, $\mathbf{U_2} = \mathbf{J}_{B_2}$, $C_1 = \bar{\mathbf{I}}$ and $C_2 = \underline{\mathbf{I}}$. The solution procedure consists of iteratively solving a single eigenvalue problem and a linear equation. Let

$$\mathcal{I} = T_e(\mathbf{U_2})^T \mathcal{I}_u T_e(\mathbf{U_2}) + \mathcal{I}_{v_1} = \begin{bmatrix} \mathcal{I}_{11} & \mathcal{I}_{12} \\ \mathcal{I}_{12}^T & \mathcal{I}_{22} \end{bmatrix} \tag{8.64}$$

Then the eigenvalue problem $\mathcal{I}_{11}\acute{\mathbf{n}} = \eta_1 \acute{\mathbf{n}}$ results from equation 8.62 where $\mathcal{I}_{11}$ is the indicated 2×2 submatrix. After solving this problem, let

$$\mathcal{I} = T_e'(\mathbf{U_1})^T \mathcal{I}_u T_e'(\mathbf{U_1}) + \mathcal{I}_{v_2} = \begin{bmatrix} \mathcal{I}_{11} & \mathcal{I}_{12} & \mathcal{I}_{13} \\ \mathcal{I}_{12}^T & \mathcal{I}_{22} & \mathcal{I}_{23} \\ \mathcal{I}_{13}^T & \mathcal{I}_{23}^T & \mathcal{I}_{23} \end{bmatrix}. \tag{8.65}$$

Then the linear equation $\mathcal{I}_{11}\acute{\mathbf{p}} = \mathcal{I}_{12}$ results from equation 8.63 where $\mathcal{I}_{11}$ is the indicated 2×2 matrix and $\mathcal{I}_{12}$ is the indicated 2×1 matrix. The solution for $\acute{\mathbf{n}}$ and $\acute{\mathbf{p}}$ can be determined by iterating these two solution procedures. Because the cost metric is strictly positive definite the solution will be unique. Furthermore, the procedure has a quadratic convergence rate.

After solving for $\acute{\mathbf{n}}$ and $\acute{\mathbf{p}}$ the final solution is given by

$$\acute{\mathbf{n}} = \mathbf{X}\mathbf{n} \qquad \acute{\mathbf{p}} = \mathbf{X}^T\mathbf{p}. \tag{8.66}$$

One type A contact results in the same equation for $\acute{\mathbf{p}}$ and $\acute{\mathbf{n}}$, however the final solution is given by

$$\acute{\mathbf{n}} = \mathbf{X}^T\mathbf{n} \qquad \acute{\mathbf{p}} = \mathbf{X}^T\mathbf{p}. \tag{8.67}$$

8.1.3.4 Solution Procedure for Two Contacts

Constraint equivalence appears in the estimation solution as an indeterminacy in the basic equations (8.62,8.63). Expansion of

$$T_e(\mathbf{U_2})^T \mathcal{I}_u T_e(\mathbf{U_2}) + \mathcal{I}_{v_1}$$

for the line-line contact case shows that for every $\mathbf{U_2}$ the resulting value is only rank 1. The minimum is obtained by selecting the eigenvector which has nonzero normal component and the smallest eigenvalue, and the eigenvector which has a nonzero value in the third component.

For two contacts of the same type the two estimation equations are redundant. Therefore two eigenvectors are sought for one of the equations. For the non-collinear case, the two solution eigenvectors are the two vectors with nonzero normal component.

Finally, with two contacts of different type both equations must be simultaneously satisfied. High convergence rates are obtained by iterating between the two equations using the solution for the smallest eigenvalue at every step.

8.1.3.5 Simulated Estimator Performance

To confirm that the estimator algorithm works correctly, the wrenches and motions from different contacts were simulated. The system generated an example desired trajectory of 100 points using a random motion bounded to stay within a (-0.3, 0.3) radian range. The translational random walk was always of unit length 1. A constraint solver is then used to produce the output motion by minimizing the energy in a stiffness matrix subject to the constraints. The constraint forces and the motion are recorded and then read back into the estimator simulation.

The true force and velocity measurements were then corrupted using a normal noise model. The average variances of the translational and rotational velocities, and of the force and torques are computed. The noise magnitude is then set to the product of the average variance with a signal-to-noise ratio. For point contacts (type A and B) the torque error was set to zero, because no torque can be applied about the point contact. For each configuration, a basis which completes the tangent space and a basis which completes the cotangent space is chosen. For type B contacts the velocity noise basis is the normal to the contact line and the force noise basis is the direction along the contact line. A normal noise was then generated, multiplied by the appropriate basis vector, and added to the true value to get the measurement value. Because the algorithm relies on the relative magnitude of the forces and not the absolute values, noise does not have to be added in the direction of the constraint to test the system.

The estimator for the appropriate model was then run on the simulated data. Lengths of (1, 2, 8, 16, 32, 64) and signal-to-noise ratios of (0.707, 0.1, 0.01, 0.001, 0.0001) were used. Figure 8.4 shows the average simulated accuracy of the estimator for a type B contact. The results for type A experiments were similar. For this type of noise the angular error is unbiased and the variance decreases as the signal-to-noise ratio increases. The errors depend on the product of the length and the signal-to-noise

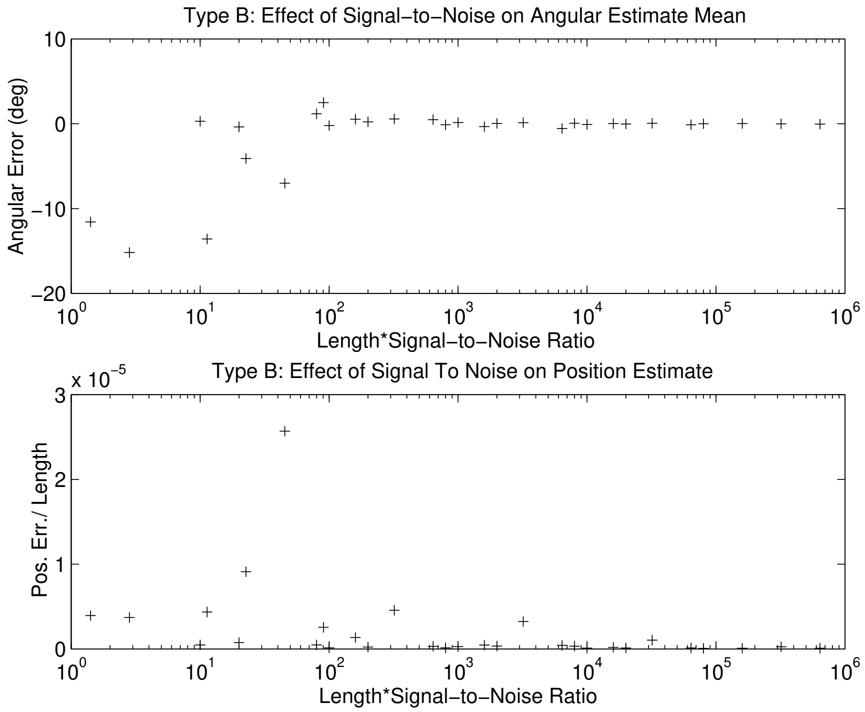


Figure 8.4: Angular alignment accuracy and relative position error for the polygonal constraint estimation procedure for a type B contact as a function of the product of the contact vector length times the signal-to-noise-ratio.

ratio. A better signal-to-noise ratio or a longer length decreases both the relative position error and the angular error. For products on the order of 100 both estimates are quite accurate. A value between 10 and 100 can be reasonable expected. The simulation suggests that the technique should work well in practice.

A type BB contact test was also performed. In this test, two perpendicular walls were used as the constraints. The walls were equally distant from the origin with distances of (1, 2, 8, 16, 32, 64). The same signal-to-ratios were investigated. Figure 8.5 shows the average simulated accuracy of the estimator for a type BB contact. The position accuracy was computed from the contact tableau for this contact type. The first two elements of each of the first two rows was used as a position vector and this was compared to the true tableau. The angular error was computed by finding the maximum angular error between the direction cosines of these vectors. Note that this measure of angular error is guaranteed to produce positive errors. As the figure shows, type BB contacts are more difficult to estimate. It takes a product of

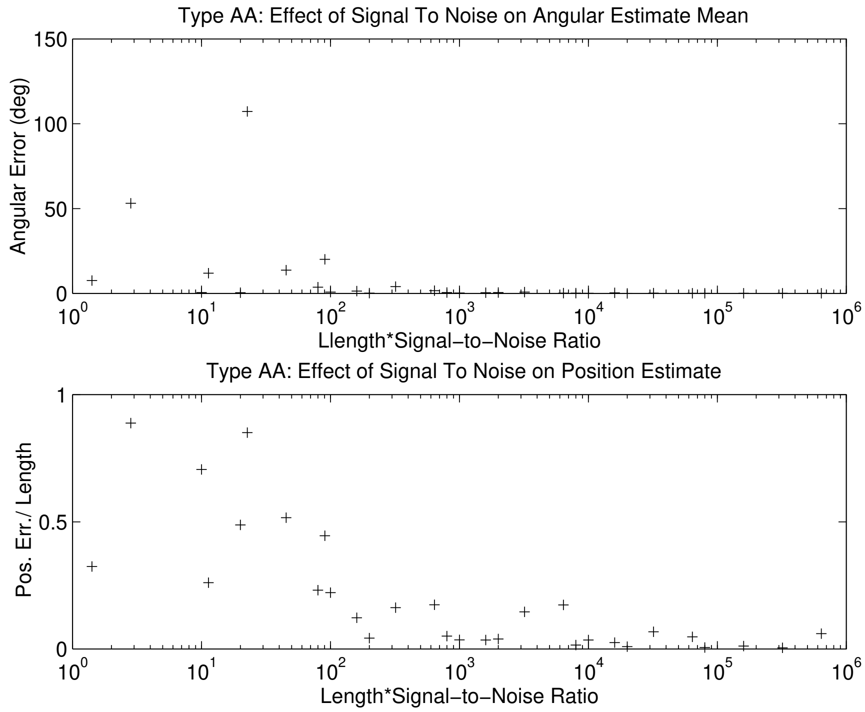


Figure 8.5: Angular alignment accuracy and relative position error for the polygonal constraint estimation procedure for a type BB contact as a function of the product of the signal-to-ratio and the contact vector length.

length and signal-to-noise ratio of 1000 before accurate measurements of geometry are produced.

8.1.4 Constraint Extensions

This section looked at two particular constraint estimation problems. Algorithms based on minimizing the measurement error covariance were derived and experimentally tested. Both algorithms produced good estimates of the constraint directions and the geometric parameters. However, the derivations are limited in generality. There are two extensions that would be useful for general manipulation: 1) Extensions to polyhedral interactions, and 2) general stiffness/constraint representations.

The cross-product approach used in the planar derivation will not work for polyhedral interactions. In addition, the representation of configuration breaks down. The first problem can be potential solved by treating both the tangent and cotangent spaces equally and then enforcing the reciprocity condition through a Lagrange multiplier. The second problem might be solved by changing to a representation based on the dual-quaternion [McCarthy, 1990]. This approach was used for linear constraints with the trace criterion. Since quaterions represent the rotation implicitly, the problem of representational singularity that occurs in an angle based representation does not occur. Furthermore, McCarthy's exposition shows how the constraints of kinematic mechanisms can be uniformly represented in terms of vector quadratic equations. This representation might simplify the derivation of results equivalent to the polygonal contact results. Constraint representation based on quaternions is also used in [Canny, 1987].

A more general representation is needed for bodies with complex geometries or compliant surfaces. The stiffness estimation procedure combined with a smoothing algorithm may provide such an approach. Stiffness estimates could be generated locally from change in position and wrench measurements for a whole range of locations and operating conditions. This second order tensor field could then be smoothed using a regularizer based smoothing technique. This might provide a general representation for what is felt, at low frequencies, when two objects interact. One speculation is that this could be used for representing the additional constraint/stiffness that is felt as two objects engage in an assembly. A recursive state estimator could then use this field and the wrench and position measurements to estimate the configuration of an interaction.

8.2 Determining the Degree of Constraint

There are two basic questions we can ask in constraint labeling: 1) can the degree or form of the constraint be recognized from the measurements, and 2) to what degree can constraints with different geometric parameters within the same form be differentiated. This section looks at the first question, the next section formulates and tests the statistics for the second question.

The constraint based estimator can be used directly to determine the degree of constraint because it breaks all possible projections down into constraint classes. The class can be determined by comparing the projected trace log-likelihood scores for the best projection, returned by the estimator, for each class.

To test the approach four hundred blocks of 25 points, each of which takes 0.1 seconds to measure, were pulled at random from the measured data for each constraint class. The estimator was then run on each sample block, and the model with the highest log-likelihood score was selected as the correct model. The approach yielded the following confusion matrix:

		Chosen Hypothesis Number of Constraints			
		0	1	2	3
Correct Hypothesis	0	1.00	0.00	0.00	0.00
	1	0.00	0.97	0.03	0.00
	2	0.00	0.00	0.97	0.03
	3	0.00	0.00	0.00	1.00

Another approach to determining the number of constraints is to use the eigenvectors from the stiffness regression. The degree to which these eigenvectors align with the ideal motion or wrench projection directions, computed from the singular values, can be used as a measure of constraint. The test statistic is the sum of the sample alignment probability statistic for each selected triple of vectors weighted by the probability that the selected triple is the correct triple. This approach has the advantage that it uses the relatively fast computation of the eigenvectors for the comparison, followed by many 3×3 singular value computation.

One alignment statistic is the sum of the singular values of the upper 3×3 matrix from $\mathbf{P}$, and the sum of the singular value for the lower 3×3 matrix from $\mathbf{P}$. The singular values are the direction cosines between the two bases. The mean and covariance of this alignment statistic can be computed from the contact examples. The test is then

$$\mathrm{p}(\tilde{\mathbf{x}}_1^n|\mathcal{H}) = \sum_{\mathbf{i}} \mathrm{p}(\mathbf{s}(\mathbf{P}(\mathbf{i}))|\mathcal{H})\mathrm{P}(\mathbf{i}) \tag{8.68}$$

where $\mathbf{i}$ is a three element index, $\mathbf{s}$ is the singular values computed from the columns selected with the index, and $\mathrm{P}(\mathbf{i})$ is the probability that the given index contains the eigenvectors associated with the smallest eigenvalues.

The index probability, and the associated summation, is what makes this approach difficult. For 3 eigenvectors of size 6 there are $\binom{6}{3} = 20$ possible indices. The index probability can be computed from the asymptotic normal statistics of the eigenvalues given in [Anderson, 1984]. The index probability is the probability that the selected eigenvalues fall in the given order given the uncertainty in their estimates. If the data is sufficiently rich such that only one of these probabilities is relevant then the test is very fast. However if the data is just noise, which may have been measured prior to exploration, all of the terms will be necessary.

A simplified version of the test was tested by generating four hundred blocks of 25 points as in the constraint test. These blocks were assumed to be sufficiently rich, so that only the three eigenvectors associated with the three smallest estimated eigenvalue had to be compared. Because the data was rich, this gave very good results.

		Chosen Hypothesis Number of Constraints			
		0	1	2	3
Correct Hypothesis	0	1.00	0.00	0.00	0.00
	1	0.00	1.00	0.00	0.00
	2	0.00	0.005	0.995	0.00
	3	0.00	0.00	0.00	1.00

However, when the same approach was applied to noise, the test selects one of the hypotheses at random but assigns a very high relative likelihood to its guess. This is wrong, and can only be fixed by including the summation over all indices in the test.

Given this additional complexity, and uniformity of the maximum likelihood framework for the constraint based approach, I feel that the constraint approach is a better test.

8.3 Constraint Direction Labeling

The recognition power of constraints is significantly enhanced by testing the direction of the constraints. This section adds two additional log-likelihood terms onto the constraint degree-of-freedom penalty to account for prior information about the constraint frame. With this enhanced model contacts with the same number of constraints but different constraint frames can be distinguished. This is demonstrated with experiments.

Recall that the log-likelihood criterion, given the measurements of position and force, for the number of constraints is

$$\begin{aligned} l &= -\frac{n}{2}\max_{\mathbf{P}}(\log\det\hat{\mathbf{V}} + k) \quad \text{subject to} \\ \hat{\mathbf{V}} &= \mathbf{P}^T\mathcal{I}\mathbf{P} \text{ and} \\ \bar{\mathbf{P}}^T\bar{\mathbf{P}} &= \mathbf{Id}(\text{equations 8.22 to 8.24}). \end{aligned}$$

where $\mathcal{I} = \frac{1}{n}\sum_{t=1}^{n}\tilde{\mathbf{y}}_t\tilde{\mathbf{y}}_t^T$ (equation 8.19).

Prior information on the nominal constraint frame can be added to this criterion by adding a penalty for the deviation in alignment of the estimated frame from a nominal frame. The alignment deviation can be measured, as in the estimation results section, by the singular values of the product of the estimated projection matrix and the nominal projection matrix

$$\alpha = 3 - \sum_{i=1}^{3}\text{svd}(\hat{\mathbf{P}}^T\mathbf{P}_N)_i \tag{8.69}$$

where $\hat{\mathbf{P}}$ is the estimated 6×3 projection matrix, $\mathbf{P}_N$ is the nominal projection matrix, and svd_i is the i^{th} singular value. The sum is bounded between 0 and 3 and measures the alignment of the frames.

An exponential distribution provides a convenient prior distribution for α. The exponential distribution is defined over positive numbers, is maximal at 0, and decreases exponentially. With this prior distribution, the combined log-likelihood criterion is

$$l = -\frac{n}{2}\max_{\mathbf{P}}(\log\det\hat{\mathbf{V}} + k) + \log(a) - a\alpha \quad \text{subject to} \tag{8.70}$$

$$\hat{\mathbf{V}} = \mathbf{P}^T\mathcal{I}\mathbf{P} \text{ and} \tag{8.71}$$

$$\bar{\mathbf{P}}^T\bar{\mathbf{P}} = \mathbf{Id}. \tag{8.72}$$

where a is the parameter of the exponential distribution. A convenient transformation from the prior standard deviation of the expected angular alignment error to the parameter of the exponential distribution is

$$a = 2/3(1/1 - \cos(\Delta\theta)).$$

This additional criterion makes it possible to distinguish contact frames that are quite closely aligned. Figure 8.6 shows the experimentally measured probability of error as a function of the angular difference between two candidate frames. The true

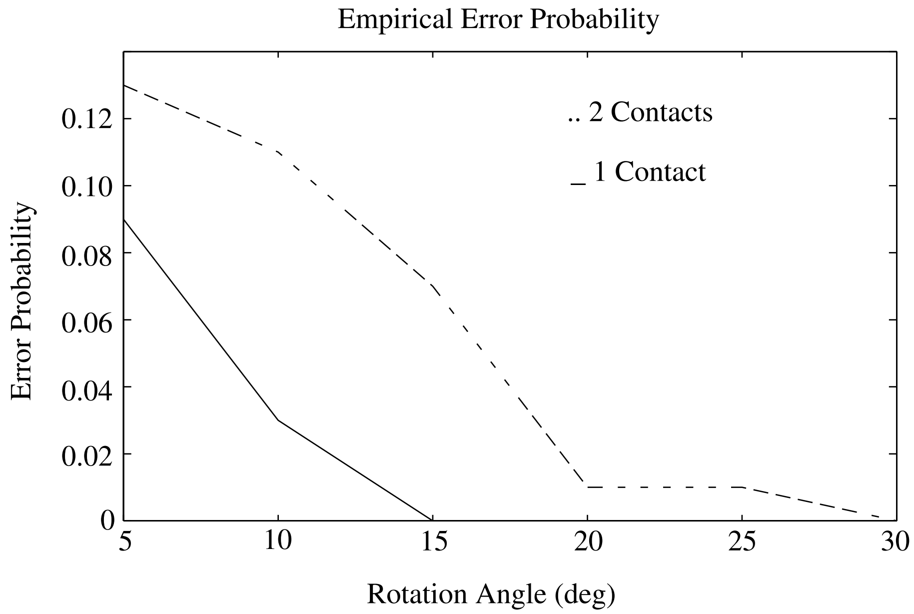


Figure 8.6: Experimentally measured probability of error as a function of the angular difference between two frame hypotheses. There is more error with two contacts than one contact. Over 98 % correct selection is demonstrated for differences in frame direction of 20 degrees.

frame was rotated by the indicated amounts about the x axis to create the alternative frame. Fifteen degrees was taken as the expected standard deviation of the alignment error for both frames. Twenty-five measurements were used for all decisions. One hundred samples were used in the estimation procedure for both models. The figure shows that even with only a 5 degree difference in the frames, the system is still able to select the correct frame over 90 % of the time.

This approach has one problem, in that it is unable to distinguish between two frames that span the same vector space, but have opposite signs in the cotangent basis. That is, the system cannot distinguish between contacts on the top of a plane and on the bottom of the plane. This problem is a consequence of the decision, discussed in chapter 4 to consider contacts as bidirectional constraints. Clearly these two types of contacts are easy to distinguish, and in fact can be distinguished purely from the applied action. However, fixing this problem within the feature framework requires some careful thought.

The first approach, that might be considered, is to simply test if the measured force lies in the positive span of the hypothesized force basis. But this is just the fixed angle friction cone model for forces, and as was already pointed out this model is not a feature in spaces with curvature.

A possibly better approach is to explicitly account for the unidirectionality of the constraint bases in the log-likelihood criterion. It may be possible to rework the estimation procedure in terms of a basis representation consisting of a collection of unidirectional basis vectors for the tangent and cotangent spaces. Each contact would shift one of the unidirectional vectors from the tangent space into the cotangent space. In this way the log-likelihood criterion would be able to distinguish between different contact signs. The singular value criterion could be similarly modified to measure the alignment of the unidirectional bases.

8.4 Conclusion

This chapter discussed the basic problem of constraint estimation and labeling. Constraints and changes in constraint are fundamental properties of manipulation. The ability to estimate the constraint directions and detect deviation from the current estimate are an essential component of any contact observer.

This chapter presented a log-likelihood based approach to constraint estimation. By considering projections of the forces onto the tangent bundle and velocities (or deviations in position) onto the cotangent bundle as normal noise, we were able to derive a log-likelihood criterion. This criterion was scale independent and considered sequences of measurements. Furthermore, since the criterion worked with the joint measurement information in the velocities and forces, the criterion was independent of the number of constraints.

The same basic approach was used for both Cartesian spaces and polygonal object spaces. The major difference between the two was in the form of the equations specifying the cotangent and tangent bundles. For Cartesian spaces, the tangent and cotangent bundles are constant as a function of position. However, in polygonal object spaces, the tangent space depends on the configuration and the notion of multidimensional curvature must be introduced to the estimation. For polygonal objects, the curvature depends only on the configuration and not on the basis vectors specifying the tangent and cotangent bundles. The problem could thus be separated, and again all of the information in a sequence of measurements could be exactly combined to estimate the basis of the bundles.

This basis is not unique in both the Cartesian space problem and the polygonal object contact problem. In Cartesian space, the basis in unique only up to a rotation that preserves the space spanned by the basis. In the polygonal space the problem is similar. Two bases were shown to be equivalent if they produced the same contact tableau.

Both estimators were implemented and tested. The Cartesian estimator was tested on real data, and the polygonal estimator on simulated data. In both cases, the estimators were extremely robust to noise, and the Cartesian estimator was able to reject the effect of friction. This shows the power of using a sequence of measurements. With multiple measurements and a little exploration, the frictional force tends to average out.

The Cartesian estimator was shown to lead directly to a log-likelihood procedure for labeling the number of degrees-of-freedom in a contact situation. In addition, with a small modification it was also able to determine the most likely direction of the constraints given two candidate models.

The procedures work because a little exploration gives a rich amount of data on which to base a decision. The main issue in performance, and any estimation problem, is sufficient richness of the data. Taking many measurements at one location or force level really does not improve performance. The important thing is to move about locally or apply forces in a variety directions. Then the model-based estimators developed here can incorporate all these measurements to produce good estimates and decisions.

Overall, the constraint estimation approach to understanding contact force seems to give good results. The next chapter brings constraint feature models together with temporal features to perceive tasks.

Tasks

Chapter 9

This chapter applies the ideas of the previous chapters to two tasks: tracking a guarded move, and following a maze. These tasks pull together all the work in the previous chapters. Constraint, high frequency temporal models, and velocity are used in distinguishing contact cases. In addition, the graphs are action dependent.

These experiments illustrate the utility of the method and bring out some future issues. The main utility is that programming becomes a problem of connecting up measurement models instead of accurately predicting measurement forces. This is much easier to do by hand and is potentially much easier to do with a computer. In addition, the models make it possible to base decisions on multiple measurements which is much more robust than a single measurement threshold. Furthermore, the number of measurements needed to support a decision is dynamically determined by the measurement probabilities.

The experiments also point out some issues that were not investigated in this thesis, but which are important to intelligent, autonomous, manipulation. First, actions which are designed to achieve goals do not necessarily gather sufficient information for making decisions. This problem arose in one guarded move task, when the commanded force did not explore the possible alternatives. In this case, the decision algorithm was forced to make decisions against noise and, as expected, returned arbitrary results. Therefore when designing an action selection procedure the trade-off between information gathering and goal achievement must be accounted for. Fortunately, the feature graph provides a starting point for this analysis.

Second, the system is robust to noise which fits within the context of the graph.

However, if there are unmodeled surfaces or things which can cause unmodeled events there is a tendency for the decision processes to proceed down the feature graph. The procedure can be made more robust by considering longer and more measurement paths. In addition, it may be possible to detect this error by again applying change detection theory to detect a mismatch between all the best current path traces and the measurements.

Lastly, there is the issue of feature graph complexity. Although only a planar task, the configuration-space maze has 63 distinct configuration space surfaces, and 148 feature models. Higher dimensional problems can be expected to have exponentially more models. There are three issues involved with this complexity. First, all the observer cares about is the graph branching factor. If this is small, the observer will still be able to track the state of the manipulation task. This should grow slowly with dimension. Second, most of the feature nodes in the graph will never be visited. For each range of starting configuration, each action will cause the robot to visit only a few of the feature partitions. If the graph for this set of starting configurations and action could be computed on-line, then the graph could be expanded as needed which would greatly reduce storage requirements. Lastly, even if the graph is computed on-line there is a need to be able to produce the graph from simulations to decrease the programming burden.

The first set of experiments looked at guarded move strategies. Guarded moves that contact one or many surfaces are a common component of a multi-step manipulation strategy. Section 9.1 will show how the measurements experienced in the typical guarded move can be encoded. It will also show how the system is easily able to distinguish the correct contact surface even when presented with closely related alternatives if there is sufficient information. First, a single step guarded move is given as an initial illustration of the technique. Then, multiple guarded moves and guarded moves which contact a sequence of surfaces are presented as a direct extension.

Section 9.2 applies the approach to tracking the position of the robot in a maze. The maze is an H shape with a single extra tail and is meant to simulate the constraints created by a gear-shift. The commands in this example are again forms of guarded moves. This example illustrates how the technique scales to actions which cause long sequences of features to be visited. It also illustrates how a collection of actions can be encoded as a collection of graphs in order to navigate the robot through desired task states.

In all of the experiments the action was applied to the robot for a fixed length of time and the measurements were recorded. The measurements where then analyzed by the host computer off-line and displayed. The constraint model estimation algorithms are numerically intensive and do not yet run in real-time. Also off-line computations were much easier to debug. With some code restructuring and optimization the program could be made to run in real-time on current microprocessors.

9.1 Tracking a Guarded Move

Three guarded move experiments were performed. The first, example simply moved the robot from free space into contact with a vertical surface. A small exploratory motion was added onto the basic motion, and this was used to show that the system could correctly determine the contacted constraints when presented with multiple possibilities.

The second is a single command that moved the robot from free space to contact with a vertical surface, to contact with a wall, and then finally into a corner-all with a single command. This example illustrates the decision algorithm's capabilities in tracking a long sequence of events. The final guarded move accomplished the same sequence of three contacts, but performed the task with three separate commands. This example illustrates changing the graph connectivity based on the commanded action.

9.1.1 Guarded Move Baseline: Move until Contact

As a baseline case the system was tested on a single step guarded move. The PHANToM was commanded to move the fingertip downward from free space onto a horizontal surface. This is the simplest possible guarded move and is just the classic move until contact. All the measurements were collected with the fingertip sensor on the PHANToM (figure 1.2). Motions were generated under automatic control. For this task, the low level controller was a position controller. A small sinusoidal exploratory motion, in all three directions, of amplitude 2 mm was added onto the downward command. The measurement models used in each experiment will be discussed and a single letter abbreviation will be introduced for the model. Feature models then consist of triples of these letters one letter - or each base feature model: high frequency, constraint, and velocity. The resulting graph is provided and labeled using these letter triples.

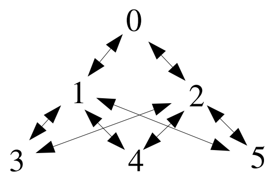
The high frequency signal was separated into two models. The first model was for the trailing edge of impact signals (I), the second model was a general, statistical, stationary signal model. The moving case treated in chapter 7 was used as a baseline, but the expected standard deviation of the mean was increased to 2.5 based on observation of the measurements when performing this task. This allowed this single model to capture all of the stationary signal conditions in the high frequency measurements. This model will be referred to as the stationary (S) model.

The Cartesian constraint models were used with the prior penalty discussed in section 8.3. Four constraint models were needed for this experiment. Free (0) is the constraint model for free space. Vertical constraint (Z) is the constraint model for contact with the surface. Two alternative constraints - a two constraint situation, one in x and one in z (XZ), and a single constraint in (Y) were also provided.

A feature model was also created for the velocity signal. The first model was a vector normal distribution with zero mean and variance corresponding to the measurement noise. This model is for zero velocity (F). The second model was a uniform distribution on all the velocities over the interval [-0.01,0.01] (M). The combination of these two models provided a test for deciding if the robot was moving or not moving.

The decision parameter for rejecting a path was set to 20.0. Any path whose likelihood was 20.0 worse than the current best path was eliminated. Since a likelihood of 20.0 corresponds to a probability ratio of $\exp(-20)$ this was felt to be reasonable. This value was small enough that only a few (less than 5) paths were maintained. A difference of 5.0 would be more aggressive and give faster performance.

The feature graph for this task is



which was defined in a file and then read into the program using a simple parser. The features corresponded to

Feature	High Freq. Model	Const. Model	Vel. Model
0	S	0	F
1	I	0	M
2	S	0	M
3	S	Z	F
4	S	XZ	F
5	S	YZ	F.

The expected measurement sequence was $0 \rightarrow 1 \rightarrow 2 \rightarrow 1 \rightarrow 3$. The robot starts out fixed in free space. The initial acceleration to start the motion generated an impact. The trailing edge of the impact occurs while moving, therefore the impact is considered to be in a moving velocity state. The robot is then smoothly moving and contacts the surface. This causes an impact (back to feature 1) which then transitions to the final constraint model (3). Features 4 and 5 were provided to test the system in order to see if the decision algorithm would become confused. The system marked 10 trial runs correctly. Three runs contacted wood, three contacted aluminum, and four contacted hard plastic.

Figure 9.1 shows the motion and the processing for one of the trial runs. For this example, most of the changes occur at impact events. An impact event is even detected during the contact with the surface. This is probably caused by a surface burr. After detecting the event, the system Initially selects feature 4 as more probable but quickly returns to the correct feature, number 3. Note that a simpler feature graph of $0 \rightarrow 1 \rightarrow 2 \rightarrow 1 \rightarrow 3$ would probably also have worked, and would have made the segmentation faster.

To give the system a more difficult test, the alternative feature (4) was modified to bring it into closer alignment with the correct choice (3). As can be expected from the constraint experiments, the system performed correctly with only a 5 degree difference in the frames. When the difference was reduced to 1 degree the system still performed correctly, but now it took more measurements (300) to make the correct decision. This is the advantage of a model driven approach. Closely related models will require lots of data to make decisions and this will be reflected in the feature probabilities.

However, it should be emphasized that the system performs well in this experiment because of the exploratory motion added onto the basic motion. Without exploration the system would not be provided with sufficient data to make a decision. Although Z would definitely be put in as a constraint direction, the system would not be able to determine if the other directions were free or constrained. If provide with these

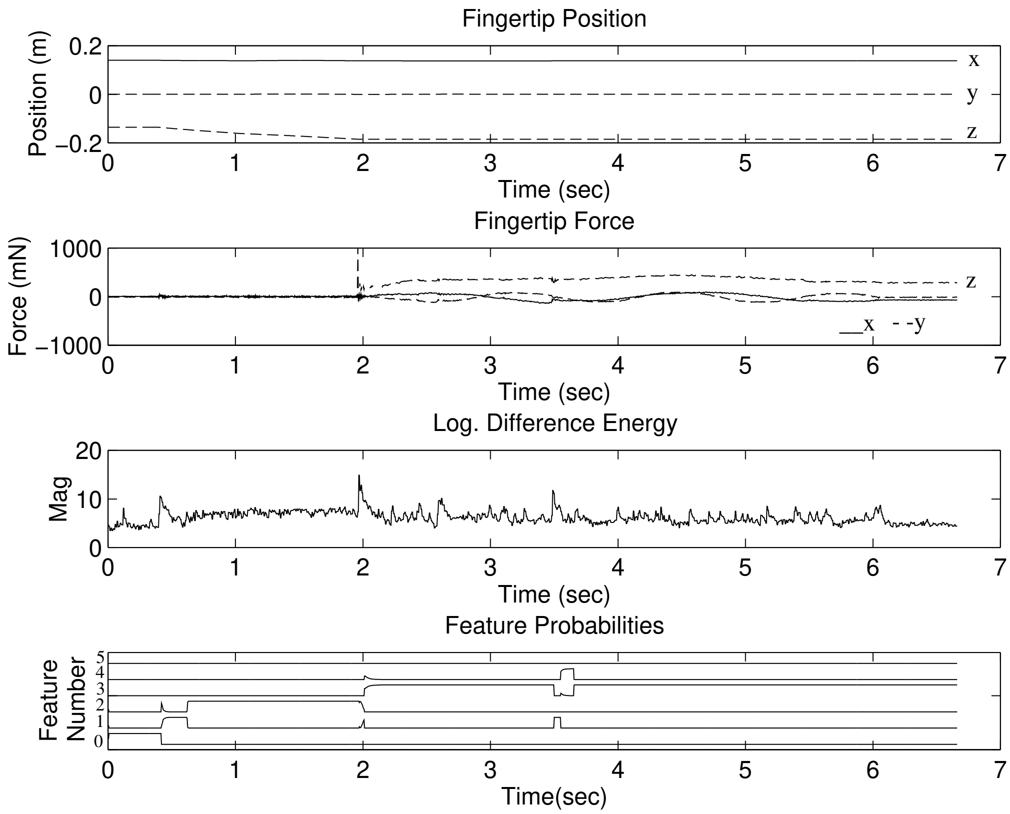


Figure 9.1: Example simple guarded move trace. The top plot shows the position of the fingertip. The x, y, and z plots are labeled in the figure. The next plot shows the force on the fingertip with the same line-type convention. The next plot shows the logarithm of the strain difference energy. The bottom plot shows the probability traces for each feature.

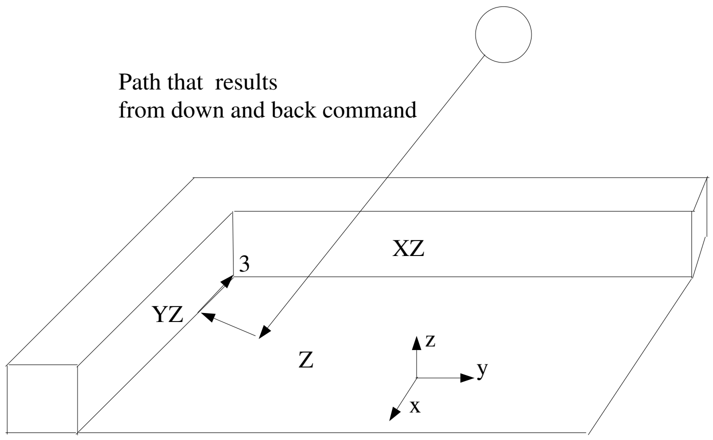


Figure 9.2: The fingertip on the PHANToM , represented by the sphere, is commanded to move down and back, with a single command, using position control. The constraint surfaces are labeled as (Z), (YZ), (XZ) and 3. The command takes the fingertip from 3 to Z to YZ to 3.

alternatives as possible decisions, it will make arbitrary decisions against what is essentially noise. An experiment discussed in the next section illustrates this issue.

9.1.2 Guarded Moves with Multiple Transitions

The second experiment was a single guarded move that was designed to contact three surfaces in succession. The PHANToM is presented with a flat surface on which there are two walls forming a corner (See figure 9.2). There are five possible constraint conditions abbreviated as: 1) 0 for no constraints, 2) Z for a single vertical constraint, 3) XZ for a x and a z constraint, 4) YZ for a y and a z constraint, and 5) 3 for all possible constraints. The desired motion took the system from $0 \to Z \to YZ \to 3$.

Four models were used for the high frequency signal. Stationary (S) for the signal when there was no movement, an impact trailing edge model (I), a moving in free space model (MF) and a moving in contact model (MC). The parameters of these models were chosen based on the measurements. The constraint model for each of the impact signals was chosen as the constraint surface to which the impact transitions.

Because of the low stiffness of the position controller, the fingertip underwent stick-

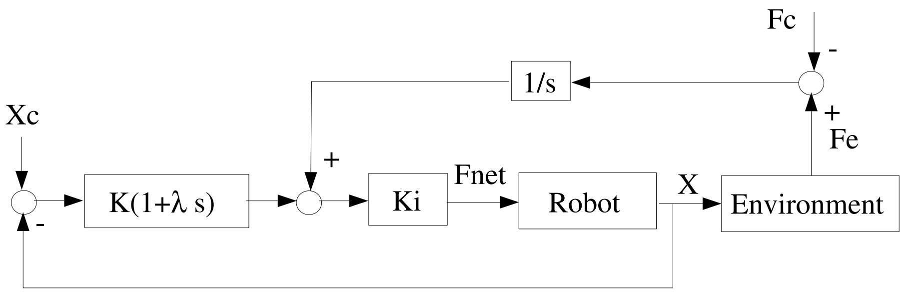
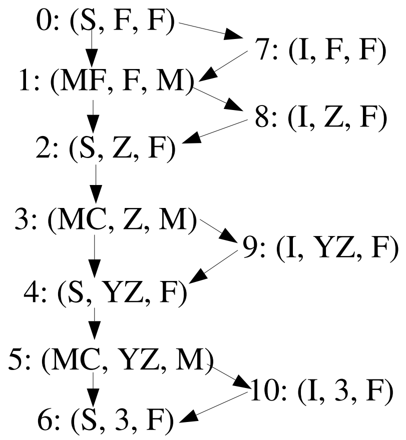


Figure 9.3: Integral endpoint force controller with position feedback.

slip as it moved across the flat surface. This had to be explicitly accounted for in the feature model. Sticking was modeled by allowing transition from (MC, Z, M) to a (S, 3, F). Then slipping was modeled by adding a connection from (S, 3, F) back to (MC,Z,M) possibly through an impact mode, thus the stick-slip that occurs as the fingertip moves along the surface is modeled a multiple transitions between these two .

Although this works reasonably well, a better approach was to change the underlying controller. An integral based endpoint force controller was added in parallel to the position controller (Figure 9.3). With this controller, the desired force input can be used as the basic command. In free space, this command becomes a constant velocity command. In contact, the controller will maintain a constant force against the constraints if the constraints support the desired force. A constant velocity will result from the forces that are not supported by the constraints. The controller acts like a generalized damper with a very narrow velocity error cone. All the remaining experiments given in this section and in the maze were performed with this controller.

With this controller, the desired action becomes the command (100,-100,200) where the command is a vector desired force to achieve in mN. The feature graph for this task and action is



Given the starting condition, the expected feature sequence is $0 \rightarrow 1 \rightarrow 8 \rightarrow 2 \rightarrow 3 \rightarrow 9 \rightarrow 4 \rightarrow 5 \rightarrow 10 \rightarrow 6$. The impacts in the sequence could be missed which is accounted for in the feature graph. This sequence, with or without the impacts, was reliably found by the system. The errors that did occur happened when an unmodeled event occured, such as catching on the wood, which occasionally forced the system down the feature graph prematurely. The system can be made more robust by maintaining more paths.

The same experiment was also done with the addition of a feature for the (XZ) constraint presented by the second wall. In this case the decision process is being asked to determine if the second guarded move terminated on the XZ wall or the YZ wall. Under integral force control the decision processor would often transition to the XZ constraint model while still moving along the flat Z constraint surface.

The problem occured because the force controller made the system move directly in the negative y direction. No motion or force information was gathered in the x direction. The decision processors was therefore forced to decide between Z constraint and XZ constraint without any significant measurements. Any small burr or just the grain in the wood surface making up the Z constraint would trigger a transition and often the XZ constraint direction would be chosen.

This problem could be detected by a more detailed examination of the covariance matrix used in constraint estimation. When this problem arises, the covariance matrix will have only two significant eigenvalues. The remaining four will be small in size. Two represent the true free directions, in the joint force and position space, and the four represent the lack of sufficient measurements.

In addition, this is not a problem intrinsic to the decision process. Rather, it is a higher level problem associated with action selection. The action selection processes must take into account not only the command required for reaching a given feature node but also all the features to which the system could next transition. If the features to which the system could transition can be disambiguated with the base action no exploration is necessary. However if the feature graph has multiple possible new features, sufficient exploration has to be added to the action. The need to explore can potentially be determined from the current markings on the feature graph and the transitions which can occur given the base action.

However, more work is needed in control to make exploration actually work. The integral force controller was used for controlling the contact forces because it gives measurements that are easier to interpret than the position controller, because it does not under go stick-slip. Furthermore, it is much easier to command a force when the position of objects is not well known. However, exploration using this controller is difficult. Any exploratory position command will be rejected by the controller. If the position exploratory command attempts to move the robot in a direction in which the robot is trying to control to zero force no motion will occur. If the force exploratory command is added, the robot will undergo a random walk on the surface, due to friction, and it will be impossible to predict the expected feature transitions. It appears that additional work in force control is needed to develop a hybrid control technique that combines integral force control and impedance position control is needed.

9.1.3 Action Dependant Graphs

Transition from free space to the corner can be made more robust using a sequence of three guarded moves. The decision process can also be made very robust by making the transitions in the feature graph depend upon the desired action. The resulting action dependent feature graph is shown in figure 9.5. The action dependent arcs provide a significant amount of information. If the current command is 0, vibration or tapping impacts will not trigger transitions to feature (S, Z, F) because this feature is not connected. Furthermore, the dependent arcs bound the number of transitions that can occur for each guarded move. Since each feature node, within the context of a single guarded move, is easily disambiguated, the decision process did not make any errors in any of the 20 trials.

One of the big advantages of the feature approach is that it is firing a sequence of

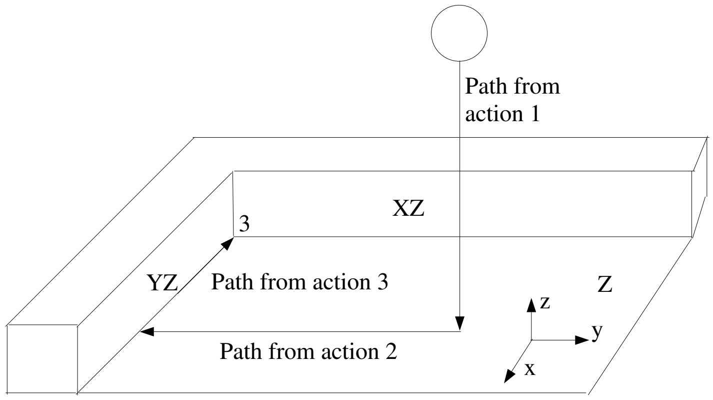


Figure 9.4: The PHANToM fingertip, represented by the sphere, follows a path created by three separate actions.

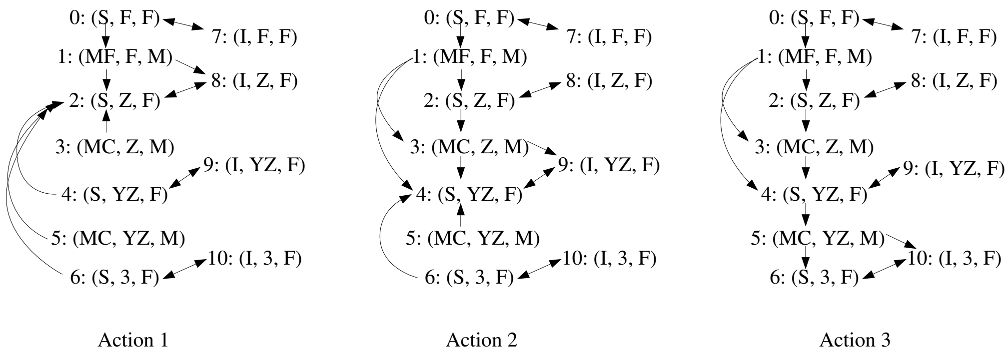


Figure 9.5: Feature graph for three step guarded move using a force controller. Each graph is labeled by the actions which make it active. The actions change the edges of the graph.

model estimators. The constraint estimators return estimates of the contact configuration and thus the object configuration. A complete robot control system could exploit this information to make the guarded move strategy more robust to angular misalignments.

An analysis of sticking shows that the these three actions will place the fingertip in the corner for angles between $-10°$ and $45°$. The angle is measured about the z axis from the nominal alignment of the YZ constraint wall with x. For angles less than $-10°$ the fingertip will stick to the YZ wall. For angles greater that $45°$ the fingertip will hit the XZ wall from the YZ wall. Both outcomes could be added as possible features.

For angles less than $-10°$ active exploration would determine that the system is in a single constraint situation and would return an estimate of the wall angle. The commanded action could then be adjusted and the system would again slide into the corner. In the other case, exploration would determine that the system is on the wall and not in the corner, and again the command could be adjusted to slide the fingertip into the corner.

This sort of approach introduces higher level feedback into the system. Actions depend upon the feature parameter and model estimates, and the feature graph depends upon the actions. Doing this intelligently without introducing instabilities remains as a major research challenge.

9.2 Following a Maze

Finally, a configuration space maze was used as an example of medium complexity to show how the system can be used to encode up a task with many features and several actions. In this task an H shaped maze with an additional tail was milled out of Lexan. The shape and the resulting configuration space are shown in figure 9.6.

The circular arcs in the configuration space are formed by contact of the fingertip with the corners of the maze. These should be modeled as circular constraint features. However, they were approximated as two linear constraint models because of the current limitations of the constraint estimator implementation.

The free space also has to be divided into components. As discussed in section 5.3.3

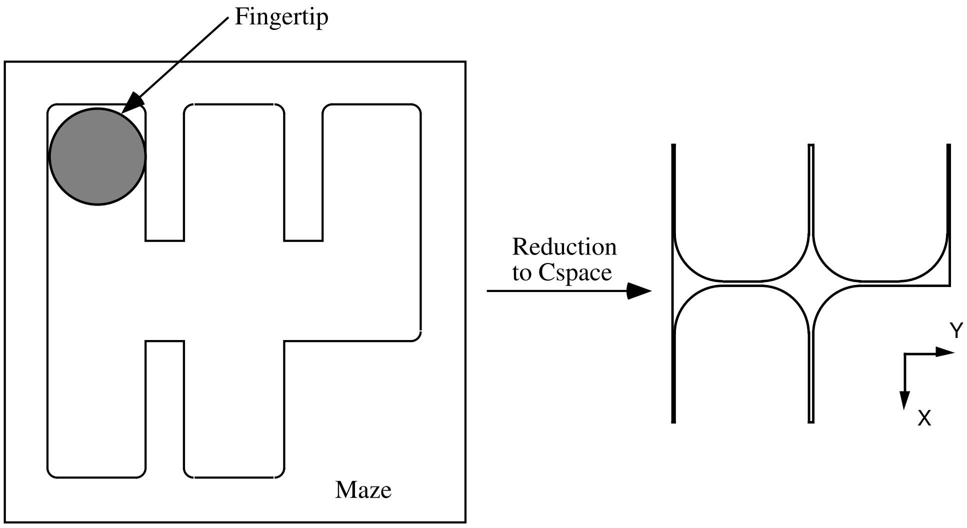


Figure 9.6: Gear-shift maze and its configuration space. The maze was milled out of Lexan to a depth less than the radius of the fingertip endtip sphere. The circle formed by slicing the sphere at the given depth is the moving object in this planar task. Together the two objects form the indicated configuration space.

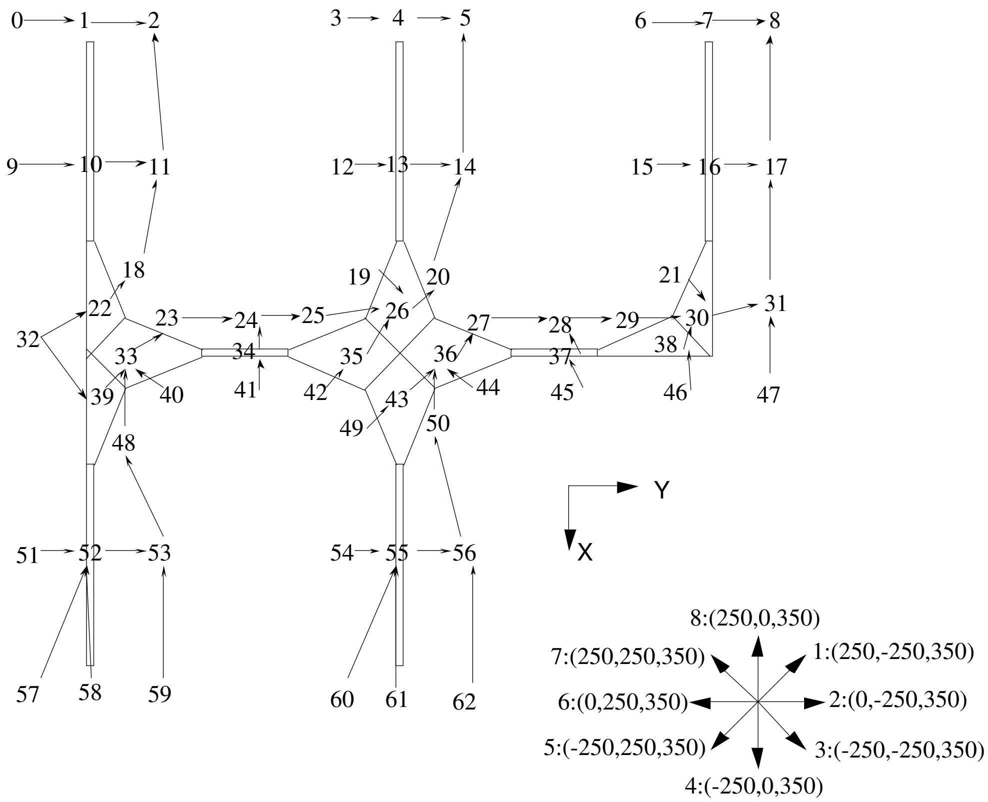


Figure 9.7: Simplified configuration space partitions. There are 62 basic configuration space partitions in this task after simplifying the back-projections and the curved surface. In addition, the eight basic force commands are shown in the lower right. The connectivity between the configuration space partitions are shown for action 1.

this should be done by considering the back-projections of each feature under the desired actions. Switching from one action to another entails first instantiating a second feature graph consisting of both new edges and nodes, and then setting the initial probabilities of all the new nodes. These probabilities are computed by covering the partitions associated with the nodes with non-zero probability in the original graph with partitions from the new graph. For this task, this was a bit too complex. Instead, the back-projections were computed and then heuristically unified into a fixed set of partitions. Different actions can then be encoded by just changing the connectivity of the fixed set of nodes. Figure 9.7 shows the 62 configuration space partitions that result, and the eight basic commands.

Each action induces transitions between the configuration space partitions. Figure 9.7 also shows the transition for action 1. The other actions produce different transition graphs. As expected, the figure shows that action 1 separates the features into three different regions. Action 1 will take the robot from partition 57 through a series of partitions to partition 5. It is also possible that the robot will become stuck at partition 23 or partition 20.

A simulation of the robot as a generalized damper with friction could be used to compute the transition graph. Multiple Monte Carlo simulation could be run to generate the possible transitions and even assign transition probabilities.

The feature graph is derived from the transition graph for the partitions. Figure 9.8 shows the feature graph for the statistically stationary feature models and the impact models. Each statistically stationary feature is labeled with a constraint model and either (f) for fixed or (m) for moving. Impact models have the constraint and velocity model of the stationary model they terminate in, and the models have not been put on the label.

To determine the stationary models, each of the configuration space partitions was simply doubled. One feature is for being in the configuration partition and not moving, and one feature is for being in the partition and moving. Impacts were introduced when the motion from one partition to a second partition would involve an abrupt change in constraint. Thus there is an impact model from 108 to 109, and from 90 to 80.

Our experiments focused on commanding action 1 from node 57. The most complex sequence of events occurs for this action-node combination. This sequence of events also appears in moving from 60 to 8, and from 5 and 8 under action 5. In this event sequence it is important to pick out feature 90 and 91 because these signal that the system has entered the "neutral" position of the H. If the system desired to move from feature 57 to feature 62 it would change action at one of these features.

Most of the connections between the stationary models come from the connectivity of the partitions. However, to make the system robust it was necessary to add additional connections. Certain features are very clear in the force and position signal traces because they are either distinctly different from the previous features or they come from persistent contacts. For action 1, starting from 57, feature 108, 109, 90, 81, 71, 116, and 5 are all very distinct. To make the system robust, we added extra connections from distinct features to the next distinct feature. For example, the transition from 108 to 90, under the basic connectivity from the action, requires

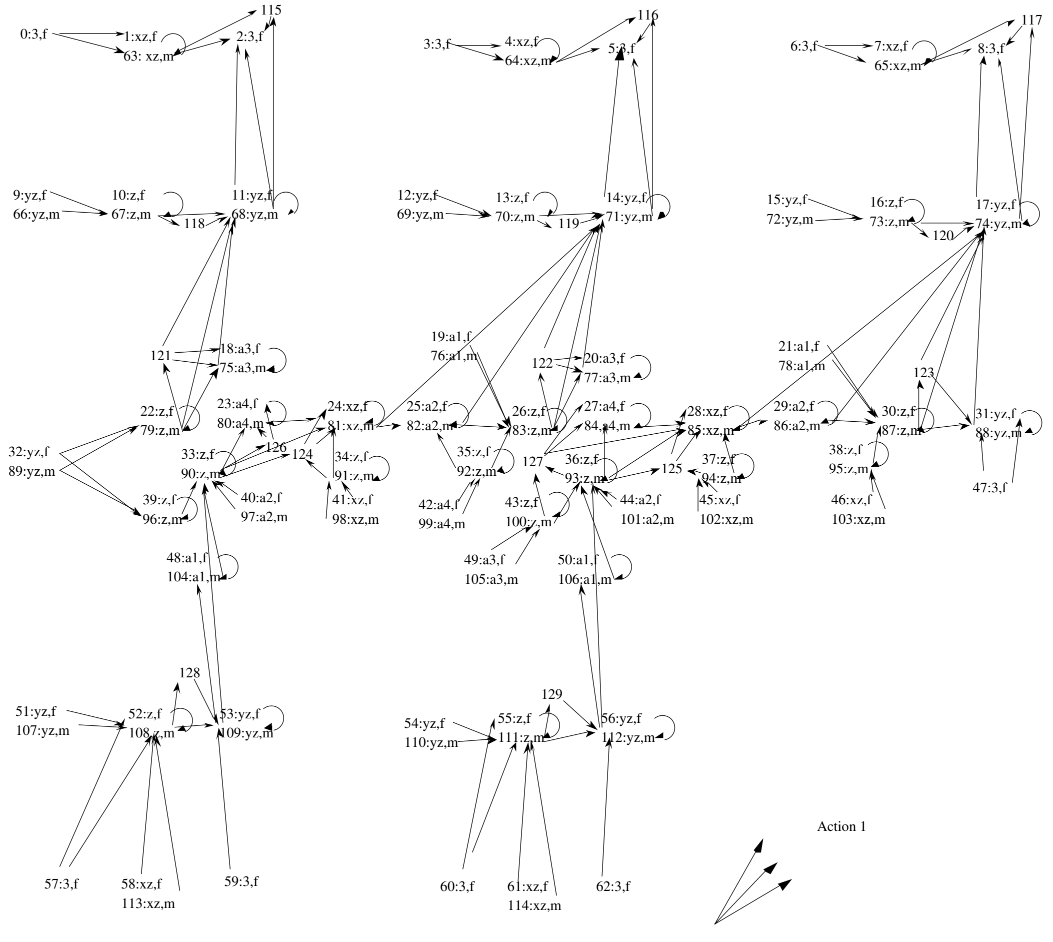


Figure 9.8: Feature graph for action 1.

passing through 104 which is not distinct. Errors can occur if 104 must be detected before 90 can be hypothesized. By adding a connection from 108 and 109 to 90 the system is made more robust. Similarly, 77, 82 and 83 are nominally required nodes from 81 to 71. These can also be missed. By adding a connection from 81 to 71 the system is made more robust.

In addition, the nominal model is that the initial motion from 57 transitions directly to 108. However, unloading the force from partition 5 in the corner with command 1 appears as a transition from 57 to 52. Adding these unloading transitions also improves performance.

This information can really only be gleaned either from a really good simulation or from actual measurements. In the HMM approach to both speech and force processing [Hannaford and Lee, 1991] the transitions are learned from repeated trials using the EM algorithm. A similar approach should be applicable here and would significantly enhance robustness for repeated tasks.

With these additional connections, 20 experimental trials from 57 under action 1 were correctly marked. The return from 5 to 57 under action 5 was also correctly marked. A typical feature trace is shown below. Each column corresponds to the order list of feature models considered possible.

57	52	108	108	109	109	109	90	81	81	71	116	5
		52	109	128	104			124	82			
			128									

9.3 Conclusion

This chapter discussed experiments with a system that observes all of the features discussed in the previous chapters. The observer uses constraint models, high frequency temporal models, and a simple model of velocity. Triples of these models are the basic feature model. Graphs of these feature models are formed from the connectivity of the underlying configuration space under a given action. This graph encodes the sequence of measurement models that can be expected given an action, which can then be used to observe where the robot is in configuration space for a given task. Different actions give rise to different graphs.

The possible measurement models are determined by the range of constraints that can arise in a given task. The connectivity depends upon both the connectivity imposed by the action, and on what will actually be observered during the task. Features which are visited for only a short time or which are ambiguous can be missed, and extra connectivity needs to be added to deal with this possibility. This extra connectivity is best determined by observing actual examples.

We showed that the feature observer was able to use the information encoded in the feature models and feature graph to track the progress of several tasks. This capability can form the basis for manipulation programming.

Selection of actions given the feature probabilities assigned by the feature observer is the other major component of a complete, intelligent, manipulation program. This action selection procedure must take into account both the basic action needed to accomplish the desired goal and the possibility that exploration may be required to determine the current feature state. Given a basic action, the feature graph shows where exploration is required, and the constraint covariance matrix indicates the directions in which there is insufficient information.

Several other important extensions to this work are indicated by the difficulty of creating the feature graph. Monte-Carlo simulation of the robot and environment would seem to be an important first step in automatically computing the feature graph. Learning using repeated trials and EM would also significantly enhance robustness and eliminate the need for hand programming both the initial feature model parameters and the connections.

The demonstrations discussed in this chapter show that contact features can be automatically detected and used in manipulation based programming. This is a step toward building a completely autonomous and automatically programmed system that incorporates contact information.

Conclusions

Chapter 10

This thesis presented a feature-based approach to observing the state of a manipulation task. Contact features were defined as predictive, statistical, measurement models which produced a countable partition of the task phase-space. This definition was contrasted with the idea of confuseable sets.

The statistical approach has two implications. First, the statistical definition implies that the robot can only observe its current state probabilistically. There are no guaranteed states, only high probability states. Second, the definition makes it possible to use the machinery of stochastic parameter estimation and change detection theory. Since this is a well formed body of knowledge, powerful tools are available for solving a contact task observer problem formulated in this way.

These tools were used to develop a statistical decision model which tracked measurement paths given prior knowledge encoded in the expected model parameter values and feature graph. The approach can be seen as a system which runs a collection of matched Kalman filters on the measurements. The Kalman filters take the form of model parameter estimators. The innovations produced by the filters are monitored by generic cumulative sum change detectors. These change detectors quickly and robustly locate deviations in the innovations process from whiteness. When a change is detected, the system determines which new Kalman filters to run by examining the feature graph. A filter is started for every node in the graph to which the system could have transitioned given the starting node. The complete observer estimates the most likely paths for the measurements through the feature graph. This was related to the probability of being in any feature in the feature graph, which solved the basic feature observer problem.

Feature estimators were then developed for two basic features: temporal sequences and constraint relationships. The constraint estimator combined the velocity and force information in a uniform manner and treated the estimation problem in the joint velocity and force space. This unified the development of the estimator and made it possible to develop a single detection algorithm for the number of degrees-of-freedom and all directions constraint. In addition, by including the curvature of the configuration space we were able to formulate an estimator for planar polygons which could recursively incorporate all of the measurement information. This estimator determines the geometric parameters describing the contact without requiring knowledge of the relative configuration between the robot and the environment. The relative configuration can be computed from the current robot configuration and the estimated geometric parameters up to equivalence in the parameters.

Finally, a demonstration of the complete theory was implemented for a sphere in Cartesian space. This system used the force measurements from a 6 axis force-torque sensor mounted on a smooth, force controllable robot (the PHANToM). The PHANToM is able to position the sensor in Cartesian space. Several experiments were performed with this system. These experiments showed that the system was able to track the sequences of features encoded by the feature graph. A number of lessons were learned from the implementation, and these motivate the discussion of future work.

10.1 Future Work

The work in this thesis laid the groundwork for a contact feature approach to manipulation programming. A number of significant steps were suggested by the implementation and experiments.

10.1.1 Computational Construction of Feature Graphs

The feature graphs become quite large quite quickly. To make the construction of these graphs practical, a program should be developed which can quickly compute the expected graph for a nominal plan and action. This leads to a two level system. The high-level system computes nominal trajectories and the expected feature contacts and transitions that could occur. The feature graph is then down-loaded to a faster process. Some of the features are labeled as terminal features where the system would like to consider a new action. This process monitors the measurements and looks for transitions from the current feature and determines when a terminal feature has been reached. Creating this high-level system is still a difficult computational problem, but because features encode large regions of phase-space the problem should be easier then reasoning based on point sets.

10.1.2 Task Control and Exploration

As has long been recognized, force control is needed to control the robot when there is very large uncertainty in the task geometry. Integral force control seems to provide the best explicit force control. However, with friction, this appears incompatible with exploration of a contacted surface. It may be necessary to use switching hybrid control laws. The difficulty is that the switching is determined by the feature observer and the information gathered by the observer depends upon the control law. This means the problem is a switching control feedback problem and the issue of stability becomes paramount.

It is important to determine when exploration is needed and what form it should take. The feature observer and graph provide two pieces of information that could be used in the analysis. First, the information matrix used in the constraint estimation procedure indicates when there is insufficient information in the measurements to make a robust decision. Second, the distribution over the feature graph and the possible transitions from the current set of possible nodes indicates first when the system does not know where it currently is, and second when the next transition could be confuseable given the current action.

10.1.3 Feature Graph Learning

Lastly, although computational construction of the feature graphs is important to solving new problems it may not be sufficient. Accurate simulation of the outcome of an action is difficult and for repetitive tasks the task itself is a better simulation. In addition, it is difficult to predict which transitions the observer may miss and therefore where extra graph connections need to be added.

Given an initial guess at the appropriate features, maximum likelihood in the form of the Expectation Modification (EM) algorithm could be used to refine the graph over multiple trials. In the implemented experiments, the experimenter essentially acted as this algorithm adding edges to make the system more robust.

10.2 Conclusion

The feature approach to manipulation sensing developed in this thesis is a step toward a method of robot programming that uses models of contact instead of just looking for force thresholds. This is a significant departure from the original paradigm for guarded moves. Models are much easier to write down for the task than the forces, because they are intrinsic to the task and are not a function of the forces applied by the robot. Furthermore, models make it possible to filter and make decisions over sequences of measurements. This approach is much more robust than making decisions over single measurements, and additional information about the contact conditions is automatically extracted by the estimation procedures. The approach requires further development to make it faster and to add actions. Development of a tight connection between the observer and actions will provided a new way to do manipulation programming.

Mathematical Background

Appendix A

A.1 Homogeneous Coordinates and Constraints

This section reviews homogeneous coordinates and transforms.

Let $\mathbf{p} \in \Re^n$ be the coordinates of some point. A transformation is a map $\mathbf{T} : \Re^n \rightarrow \Re^n$ which preserves distance. Any such transformation can be written as a rotation plus a translation. The transformation $\mathbf{T}$ takes $\mathbf{p}$ to $\mathbf{p}$ as

$$\mathbf{T}(\mathbf{p}) = \mathbf{P} = \mathbf{R}\mathbf{p} + \mathbf{r}. \tag{A.1}$$

Transformations are not linear maps on $\Re^n$ because $\mathbf{T}(\mathbf{p}_1 + \mathbf{p}_2) \neq \mathbf{T}(\mathbf{p}_1) + \mathbf{T}(\mathbf{p}_2)$. Transformations can be made linear by embedding $\Re^n$ in $\Re^{n+1}$ via $\mathbf{p} \rightarrow \begin{bmatrix} \mathbf{p} \\ 1 \end{bmatrix} = \bar{\mathbf{p}}$. In homogeneous coordinates, a transformation is given by

$$\mathbf{T} = \begin{bmatrix} \mathbf{R} & \mathbf{r} \\ 0 & 1 \end{bmatrix} \tag{A.2}$$

and $\bar{\mathbf{P}} = \mathbf{T}\bar{\mathbf{p}}$. The inverse of a transform $\mathbf{T}$ is

$$\mathbf{T}^{-1} = \begin{bmatrix} \mathbf{R}^T & -\mathbf{R}^T\mathbf{r} \\ 0 & 1 \end{bmatrix} \tag{A.3}$$

since $\mathbf{R}$ is an orthonormal matrix.

A rigid body $\mathcal{B}$ consists of a collection of points. Their locations are defined relative to a local coordinate system. Their coordinates relative to any other coordinate system is obtained by applying the same transform $\mathbf{T}$ to every point in the body. Therefore, the configuration space of a rigid body is the space of transforms. In mathematical notation $\mathcal{C} = \Re^n \times SO(n)$. The tangent bundle and cotangent bundle are defined on this configuration space.

In $\Re^n$ the dual object to a point is a hyper-plane of dimension $n-1$. This notion carries over to homogeneous coordinates. The homogeneous coordinates of a hyper-plane are $\bar{\mathbf{n}} = \begin{bmatrix} \mathbf{n} \\ -d \end{bmatrix}$.

A.2 Estimation and Filtering

The first step in building an observer is to develop algorithms for estimating the feature parameters given the measurements and to determine the likelihood of the measurement sequence. When the parameters are free (no prior distribution) the maximum likelihood estimator (MLE) provides the necessary computations. In matching, a prior distribution is provided, and this information must be folded in with the information from the measurements. The Kalman filter provides the necessary computations. In many cases, the feature parameters will not have a normal prior distribution on the natural regression directions because they are correlated in a nonlinear manner. However it may be possible to orthogonalize the regressors and produce orthogonal parameters which do have approximately normal distributions. We therefore, complete this section by showing how the regressors can be orthogonalized within the Kalman filter.

A.2.1 MLE Estimation of Feature Parameters

We will consider only linear, Gaussian, predictive measurement models with additive noise. The general form of this type of model is

$$\mathbf{y}(t) = \boldsymbol{\psi}(t)\boldsymbol{\theta} + \boldsymbol{\nu}(t) \tag{A.4}$$

where $\boldsymbol{\psi}^T(t)$ is a matrix of regression coefficients, $\boldsymbol{\theta}$ is the parameter vector, and $\boldsymbol{\nu}$ is a white Gaussian noise process. $\mathbf{y}$ is a vector of size m.[1]

When we have no prior information about the distribution of $\boldsymbol{\theta}$, the maximum likelihood estimate (*MLE*) provides the best basis for estimating and detecting changes in the parameters. For the linear predictor model, the MLE estimate of the parameters is the least-squares estimate. The estimate can be written in matrix form by collecting the measurements into a vector as $\mathbf{Y}(n) = \mathbf{y}_1^n$, and collecting each column of $\boldsymbol{\psi}$ into a vector as $\mathbf{s_i}(n) = \boldsymbol{\psi}_i(t)_{t=1}^n$. Then the least-squares estimate takes the form

$$
\begin{aligned}
\mathbf{I}_{ij}(n) &= \left\langle \mathbf{s_i}(n) | \mathbf{s_j}(n) \right\rangle && \text{(A.5)} \\
\mathbf{I}(n) &= \sum_{t=1}^{n} \boldsymbol{\psi}^T(t)\boldsymbol{\psi}(t) && \text{(A.6)} \\
\mathbf{X}_i(n) &= \langle \mathbf{s_i}(n) | \mathbf{Y}(n) \rangle && \text{(A.7)} \\
\mathbf{X}(n) &= \sum_{t=1}^{n} \boldsymbol{\psi}^T(t) y(t) && \text{(A.8)} \\
\hat{\boldsymbol{\theta}}(n) &= \mathbf{I}^{-1}(n)\mathbf{X}(n), && \text{(A.9)}
\end{aligned}
$$

where $\langle | \rangle$ denotes inner product. $\mathbf{I}$ is called the empirical information matrix and $\mathbf{X}$ is called the empirical information vector. $\hat{\boldsymbol{\theta}}(n)$ is the parameter estimate at time n.

The parameter estimates depend upon the model order because in general the vectors $\{\mathbf{s_i}(n)\}$ are not orthogonal. An orthogonal set of parameters, sometimes termed the reflection coefficients [Makhoul, 1975], can be formed by first orthogonalizing the vectors $\{\mathbf{s_i}(n)\}$. The orthogonal parameters can then be used to estimate $\boldsymbol{\theta}$ for any desired order.

When the computation is performed on-line only $\mathbf{I}(n)$ and $\mathbf{X}(n)$ are actually kept in memory. Therefore, the reflection coefficients need to be computed from these matrices. If the vectors $\{\mathbf{s_i}(n)\}$ were actually available, a Gram-Schmidt procedure could be applied to the vectors to generate an orthogonal basis. This decomposition represents the matrix $\mathbf{S} = [\mathbf{s_i}]$ as $\mathbf{S} = \mathbf{Q}\mathbf{R}^T$ where $\mathbf{Q}$ is the orthonormal basis, and $\mathbf{R}^T$ is the correlation structure of $\mathbf{S}$. The time dependence has been dropped for clarity. Then, the unnormalized reflection coefficients, $\tilde{\mathbf{k}}$, satisfy $\mathbf{k} = \mathbf{Q}^T\mathbf{Y}$.

Now note that $\mathbf{I} = \mathbf{S}^T\mathbf{S} = \mathbf{R}\mathbf{R}^T$, since $\mathbf{Q}$ is orthonormal, and $\mathbf{X} = \mathbf{S}^T\mathbf{Y} = \mathbf{R}\mathbf{Q}^T\mathbf{Y}$. Therefore the unnormalized reflection coefficients are generated by the first stage

[1] Much of this material can be found in advanced texts on signal processing and estimation [Anderson and Moore, 1979].

of Gaussian elimination. Gaussian elimination factors $\mathbf{I}$ into $\mathbf{R}\mathbf{R}^T$. Therefore, the reflection coefficients are the solution to $\mathbf{R}\tilde{\mathbf{k}} = \mathbf{X}$. This solution is just the result of the first stage of Gaussian elimination. The second step, back substitution, solves $\mathbf{R}^T\hat{\boldsymbol{\theta}} = \tilde{\mathbf{k}}$.

Now the unnormalized reflection coefficients can be used to reconstruct the solution for any model order. Given any order model m, let the first m coefficients be $\tilde{\mathbf{k}}_m$ and the corresponding sub-matrix of $\mathbf{R}^T$ be $\tilde{\mathbf{R}}_m^T$. Then the original model coefficients for an order m model can be determined from $\tilde{\mathbf{k}}_m = \tilde{\mathbf{R}}_m^T\hat{\boldsymbol{\theta}}_m$.

A square-root algorithm provides an efficient, numerically stable procedure for computing the coefficients. This procedure works directly with $\mathbf{R}$. When a new measurement $\mathbf{y}(n)$ is taken the following update is performed:

1. Update the regression matrix $\boldsymbol{\psi}(n)$ based on the model equation.

2. Update $\mathbf{R}$ by finding a Householder transform $\mathbf{T}$ such that

$$[\,\mathbf{R}(n) \quad 0\,] = [\,\mathbf{R}(n-1) \quad \boldsymbol{\psi}^T(n)\,]\,\mathbf{T}. \tag{A.10}$$

   Since $\mathbf{T}$ is orthonormal, this equation is factored

$$\mathbf{R}(n)\mathbf{R}^T(n) = \mathbf{R}(n-1)\mathbf{R}^T(n-1) + \boldsymbol{\psi}^T\boldsymbol{\psi} \tag{A.11}$$

   which is the update for the information matrix.

3. Solve

$$\mathbf{R}(n)\,[\,\tilde{\mathbf{k}}(n) \quad \bar{\boldsymbol{\psi}}(n)\,] = [\,\mathbf{X}(n) \quad \boldsymbol{\psi}(n)\,]\,. \tag{A.12}$$

4. If desired, solve for the model parameters of length m.

$$\hat{\boldsymbol{\theta}}_m \leftarrow \mathbf{R}^{-T}\tilde{\mathbf{k}}_m \tag{A.13}$$

5. Compute the residuals in the orthogonal frame.

$$\begin{aligned} \nu_j(n) &= \nu_{j-1}(n) - \bar{\boldsymbol{\psi}}_j(n)\tilde{k}_j(n) & \text{(A.14)}\\ &\quad \text{with } \nu_0(n) = y(n). & \text{(A.15)} \end{aligned}$$

6. Finally, compute an estimate of the log-likelihood[2] of the residual using the estimated variance

$$l(\mathbf{y}(n)|\mathbf{y}_1^{n-1}) = l(\nu(n)) = -\frac{1}{2}(\log\det\hat{\mathbf{V}}_\nu(n) + \nu(n)^T\hat{\mathbf{V}}_\nu(n)^{-1}\nu(n))^3 \quad \text{(A.16)}$$

   The measurement sequence log-likelihood is estimated by summing these terms.

7. Compute an estimate of the variance of the residuals via:

$$\hat{\mathbf{V}}_\nu(n)] = \hat{\mathbf{V}}_\nu(n-1) + \frac{1}{n}(\nu(n)\nu(n)^T - \hat{\mathbf{V}}_\nu(n-1)) \quad \text{(A.17)}$$

   which could also be done with a square-root update algorithm.

A.2.2 Estimating Parameters with Prior Values

When a prior distribution is provided for the feature parameters, this information must be incorporated into the estimation procedure. The Kalman filter incorporates the information and produces a white innovations process that can be used for testing for changes from the model. The square-root implementation of the Kalman filter provides a robust, efficient, implementation of the filter. Our review of the algorithm follows [Willsky, 1990]. The development of the algorithm shares many similarities with the previous estimation algorithm.

The state and measurement equations can be written in matrix form as

$$\begin{bmatrix} \mathbf{y}(n) \\ \boldsymbol{\theta}(n+1) \end{bmatrix} = \begin{bmatrix} \boldsymbol{\psi}(n) & \mathbf{V}_{\boldsymbol{\nu}}^{1/2} \\ \mathbf{Id} & 0 \end{bmatrix} \begin{bmatrix} \boldsymbol{\theta}(n) \\ \boldsymbol{\mu}(n) \end{bmatrix} \quad \text{(A.18)}$$

where $\boldsymbol{\mu}(n)$ is a white, normal, process with identity covariance. Given, $\hat{\boldsymbol{\theta}}(n|n-1)$ and $\mathbf{V}_{\boldsymbol{\theta}}^{1/2}(n|n-1)$ the measurement prediction and parameter update step are

$$\begin{bmatrix} \hat{\mathbf{y}}(n|n-1) \\ \hat{\boldsymbol{\theta}}(n+1|n-1) \end{bmatrix} = \begin{bmatrix} \boldsymbol{\psi}(n) \\ \mathbf{Id} \end{bmatrix} \hat{\boldsymbol{\theta}}(n|n-1). \quad \text{(A.19)}$$

[2]The log-likelihood is the log of the conditional probability of receiving the measurements given the model.

[3]In this case it is actually possible to compute the measurement sequence log-likelihood exactly. If $E_0 = \langle \mathbf{Y}(n)|\mathbf{Y}(n)\rangle$ is the total energy in the signal, the exact value of the log-likelihood is $-n/2(\log((E_0 - \tilde{\mathbf{k}}(n)^T\tilde{\mathbf{k}}(n))/n) + m)$.

These two equations can be combined to compute the innovations and the error in parameter estimate

$$\begin{bmatrix} \boldsymbol{\nu}(n) \\ \tilde{\boldsymbol{\theta}} \end{bmatrix} = \begin{bmatrix} \Psi(n)\mathbf{V}_{\boldsymbol{\theta}}^{1/2}(n|n-1) & \mathbf{V}_{\boldsymbol{\nu}}^{1/2} \\ \mathbf{V}_{\boldsymbol{\theta}}^{1/2}(n|n-1) & 0 \end{bmatrix} \boldsymbol{\eta}(n) \tag{A.20}$$

where

$$\boldsymbol{\eta}(n) = \begin{bmatrix} \mathbf{V}_{\boldsymbol{\theta}}^{-1/2}\tilde{\boldsymbol{\theta}}(n|n-1) \\ \boldsymbol{\mu}(n) \end{bmatrix}. \tag{A.21}$$

$\boldsymbol{\eta}$ has identity covariance.

Now we apply a Householder transform $\mathbf{T}$ such that

$$\begin{bmatrix} \Psi(n)\mathbf{V}_{\boldsymbol{\theta}}^{1/2}(n|n-1) & \mathbf{V}_{\boldsymbol{\nu}}^{1/2} \\ \mathbf{V}_{\boldsymbol{\theta}}^{1/2}(n|n-1) & 0 \end{bmatrix} \mathbf{T} = \begin{bmatrix} F & 0 \\ B & D \end{bmatrix}. \tag{A.22}$$

Then

$$\begin{bmatrix} \boldsymbol{\nu}(n) \\ \tilde{\boldsymbol{\theta}} \end{bmatrix} = \begin{bmatrix} F & 0 \\ B & D \end{bmatrix} \boldsymbol{\zeta} \tag{A.23}$$

$$\boldsymbol{\zeta} = \begin{bmatrix} \boldsymbol{\alpha} \\ \boldsymbol{\beta} \end{bmatrix} = \boldsymbol{T}^T \boldsymbol{\eta}. \tag{A.24}$$

Using these new definitions yields

$$\mathrm{E}[\boldsymbol{\nu}(n)\boldsymbol{\nu}(n)^T] = FF^T \rightarrow F = \mathbf{V}_{\boldsymbol{\nu}}^{1/2}(n|n) \tag{A.25}$$

$$\mathrm{E}[\tilde{\boldsymbol{\theta}}(n+1|n-1)\boldsymbol{\nu}(n)] = BF^T. \tag{A.26}$$

Therefore the new parameter estimate is given by

$$\hat{\boldsymbol{\theta}}(n+1|n) = \hat{\boldsymbol{\theta}}(n+1|n-1) + \mathrm{E}[\tilde{\boldsymbol{\theta}}(n+1|n-1)\boldsymbol{\nu}(n)^T] \; \mathrm{E}[\boldsymbol{\nu}(n)\boldsymbol{\nu}(n)^T]^{-1}\boldsymbol{\nu}(n) \tag{A.27}$$

$$= \hat{\boldsymbol{\theta}}(n+1|n-1) + B\boldsymbol{\alpha}. \tag{A.28}$$

Finally, the covariance of the parameters is also produced since

$$\tilde{\boldsymbol{\theta}}(n+1|n) = \boldsymbol{\theta} - \hat{\boldsymbol{\theta}}(n+1|n) \tag{A.29}$$

$$= D\boldsymbol{\beta} \tag{A.30}$$

$$\rightarrow \mathbf{V}_{\boldsymbol{\theta}}^{1/2}(n+1|n) = D. \tag{A.31}$$

The algorithm is summarized below.

1. Given $\hat{\boldsymbol{\theta}}(n|n-1)$, $\mathbf{V}_{\boldsymbol{\theta}}^{1/2}(n|n-1)$, $\mathbf{V}_{\boldsymbol{\nu}}^{1/2}(n|n-1)$, $\boldsymbol{\psi}(n)$ form

$$\mathbf{A} = \begin{bmatrix} \boldsymbol{\psi}(n)\mathbf{V}_{\boldsymbol{\theta}}^{1/2}(n|n-1) & \mathbf{V}_{\boldsymbol{\nu}}^{1/2} \\ \mathbf{V}_{\boldsymbol{\theta}}^{1/2}(n|n-1) & 0 \end{bmatrix}. \tag{A.32}$$

   The initial values of the estimates and the covariance matrices are provided by estimating the covariance and parameter values over a set of training examples.

2. Apply a Householder reduction $\mathbf{T}$ to reduce $\mathbf{A}$

$$\mathbf{AT} = \begin{bmatrix} \mathbf{V}_{\boldsymbol{\nu}}^{1/2}(n|n) & 0 \\ \mathbf{K}(n) & \mathbf{V}_{\boldsymbol{\theta}}^{1/2}(n+1|n) \end{bmatrix}. \tag{A.33}$$

3. Compute the innovation

$$\boldsymbol{\nu}(n) = \mathbf{y}(n) - \boldsymbol{\Psi}(n)\hat{\boldsymbol{\theta}}(n|n-1). \tag{A.34}$$

4. Solve for the normalized innovations $\boldsymbol{\alpha}$

$$\mathbf{V}_{\boldsymbol{\nu}}^{1/2}(n|n)\boldsymbol{\alpha}(n) = \boldsymbol{\nu}(n). \tag{A.35}$$

5. Update the parameter estimates

$$\hat{\boldsymbol{\theta}}(n+1|n) = \hat{\boldsymbol{\theta}}(n|n-1) + \mathbf{K}(n)\boldsymbol{\alpha}(n). \tag{A.36}$$

6. Compute a statistic for change detection

$$l_c(n) = -\frac{1}{2}\boldsymbol{\alpha}(n)^T\boldsymbol{\alpha}(n). \tag{A.37}$$

7. Update the estimate of the square-root of the variance using another Householder reduction

$$\begin{bmatrix} \mathbf{V}_{\boldsymbol{\nu}}^{1/2}(n+1|n) & 0 \end{bmatrix} = \begin{bmatrix} (1-1/n)^{1/2}\mathbf{V}_{\boldsymbol{\nu}}^{1/2}(n|n-1) & \sqrt{(1/n)\boldsymbol{\nu}(n)} \end{bmatrix} \mathbf{T}_2. \tag{A.38}$$

8. Compute an estimate of the log likelihood of the innovations

$$l(n) = -\frac{1}{2}\log\det(\mathbf{V}_{\boldsymbol{\nu}}^{1/2}(n+1|n)) + l_c(n) \tag{A.39}$$

A.2.3 Estimating Orthogonal Parameters with Priors

In order for the Kalman filter to work properly, the initial value of the parameter estimates and the covariance has to be at least approximately normal. As was discussed in the thesis, the training estimates used for the temporal models were very far from normally distributed. However, if an orthogonal set of parameters, generated by orthogonalizing the regression vectors, was used the orthogonal parameters were approximately normally distributed.

In training, the values of the feature parameters were estimated using the maximum likelihood square-root algorithm to produce orthogonal parameters. During labeling, the Kalman filter needs a small modification.

Orthogonal parameter estimates will be computed by the square-root Kalman filter, if orthogonal regressors are provided to the algorithm. If $\mathbf{R}(n)$ represents the current correlation structure, then the orthogonal regressors are produced by solving

$$\tilde{\boldsymbol{\psi}}(n)\mathbf{R}(n) = \boldsymbol{\psi}(n). \tag{A.40}$$

The correlation structure is maintained using yet another Householder reduction $\mathbf{T}$

$$\begin{aligned} [\,\tilde{\mathbf{R}}(n)^T \quad 0\,] &= [\,\mathbf{R}(n-1) \quad \boldsymbol{\psi}(n)^T\,]\,\mathbf{T} && \text{(A.41)} \\ \mathbf{D}^{-1/2}(n)\tilde{\mathbf{R}}(n) &= \mathbf{R}(n) && \text{(A.42)} \end{aligned}$$

where $\mathbf{D}^{-1/2}$ is the matrix formed from the diagonal elements of $\tilde{\mathbf{R}}$.

Using either of these two algorithms, signals are mapped into estimates of the feature parameters and an estimate of the log-likelihood that the given feature is the correct model for the measurements.

Bibliography

[Anderson and Moore, 1979] B. D. O. Anderson and J. B. Moore. *Optimal Filtering.* Information and System Sciences Series. Prentice-Hall, 1979.

[Anderson, 1984] T. W. Anderson. *An Introduction to Multivariate Statistical Analysis.* Probability and Mathematical Statistics. John Wiley & Sons, 2nd edition, 1984.

[Andre-Obrecht, 1988] R. Andre-Obrecht. A new statistical approach for the automatic segmentation of continuous speech signals. *IEEE Transactions on Acoustics, Speech, and Signal Processing*, 36(1):29–40, January 1988.

[Arnold, 1989] V. I. Arnold. *Mathematical Methods of Classical Mechanics.* Springer-Verlag, 1989.

[Asada and Hirai, 1989] H. Asada and S. Hirai. Towards a symbolic-level feedback: Recognition of assembly process states. In *Robotics Research: The Fifth International Symposium*, Tokyo, 1989.

[Basseville and Benveniste, 1983] M. Basseville and A. Benveniste. Sequential detection of abrupt changes in spectral characteristics of digitial systems. *IEEE Transactions on Information Theory*, 29(5):709–724, September 1983.

[Basseville and Benveniste, 1986] M. Basseville and A. Benveniste, editors. *Detection of Abrupt Changes in Signals and Dynamical Systems.* Springer-Verlag, 1986.

[Basseville *et al.*, 1986] M. Basseville, A. Benveniste, and G. V. Moustakides. Detection and diagnosis of abrupt changes in model characteristics of nonstationary digitial systems. *IEEE Transactions on Information Theory*, 32(3):412–417, May 1986.

[Basseville *et al.*, 1987] M. Basseville, A. Benveniste, G. V. Moustakides, and A. Rougée. Detection and diagnosis of changes in the eigenstructure of non-stationary multivariable systems. *Automatica*, 23(4):479–489, 1987.

[Basseville, 1986] M. Basseville. On-line detection of jumps in mean. In M. Basseville and A. Benveniste, editors, *Detection of Abrupt Changes in Signals and Dynamical Systems*. Springer-Verlag, 1986.

[Basseville, 1988] M. Basseville. Detecting changes in signals and systems - a survey. *Automatica*, 24(3):309–326, 1988.

[Benveniste *et al.*, 1987] A. Benveniste, M. Basseville, and G. V. Moustakides. The asymptotic local approach to change detection and model validation. *IEEE Transactions on Automatic Control*, 32(7):583–592, July 1987.

[Bertsekas, 1976] D. P. Bertsekas. *Dynamic Programming and Stochastic Control*, volume 125 of *Mathematics in Science and Engineering*. Academic Press, 1976.

[Bicchi *et al.*, 1989] A. Bicchi, J. K. Salisbury, and P. Dario. Augmentation of grasp robustness using intrinsic tactile sensing. In *Proc. IEEE International Conference on Robotics and Automation*, pages 302–307, 1989.

[Bicchi *et al.*, 1990] A. Bicchi, J. K. Salisbury, and D. L. Brock. Contact sensing from force measurements. AI Memo 1262, MIT Artificial Intelligence Laboratory, Cambridge, MA, October 1990.

[Bicchi *et al.*, 1993] A. Bicchi, J. K. Salisbury, and D. L. Brock. Contact sensing from force measurements. *International Journal of Robotics Research*, 1993.

[Bicchi, 1990] A. Bicchi. Intrinsic contact sensing for soft fingers. In *Proc. IEEE International Conference on Robotics and Automation*, pages 968–973, Cincinnati, Ohio, May 1990.

[Bolles and Paul, 1973] R. Bolles and R. Paul. The use of sensory feedback in a programmable assembly system. Memo 220, Stanford Artificial Intelligence Laboratory, 1973.

[Brock and Chiu, 1985] D. L. Brock and S. Chiu. Environment perception of an articulated robot hand using contact sensors. In *Proceedings of the ASME Winter Annual Meeting*, Miami, FL, 1985.

[Brock, 1993] D. L. Brock. *A Sensor Based Strategy for Automatic Robotic Grasping.* PhD thesis, MIT, Department of Mechanical Engineering, 1993.

[Brooks, 1982] R. A. Brooks. Symbolic error analysis and robot planning. *International Journal of Robotics Research*, 1(4), 1982.

[Brooks, 1985] R. A. Brooks. A robust layered control system for a mobile robot. Memo 864, MIT Artificial Intelligence Laboratory, 1985.

[Brooks, 1987] R. A. Brooks. Planning is just a way of avoiding figuring out what to do next. Technical report, MIT Artificial Intelligence Laboratory, 1987.

[Brooks, 1991] R. A. Brooks. Intelligence without reason. A.I. Memo 1293, MIT Artificial Intelligence Laboratory, 1991.

[Bruyninckx and De Schutter, 1992] H. Bruyninckx and J. De Schutter. A systematic derivation of on-line motion constraint identification equations for model-based compliant motions. In *Proc. IEEE International Conference on Robotics and Automation*, pages 1513–1518, Nice, France, 1992.

[Buckley, 1987] S. J. Buckley. *Planning and Teaching Compliant Motion Strategies.* PhD thesis, MIT, Department of Electrical Engineering and Computer Science, January 1987.

[Caine, 1993] M. E. Caine. *The Design of Shape from Motion.* PhD thesis, MIT, Department of Mechanical Engineering, 1993.

[Canny, 1987] J. F. Canny. *The Complexity of Robot Motion Planning.* PhD thesis, MIT, Department of Electrical Engineering and Computer Science, Cambridge, MA, 1987.

[Chien and Adams, 1976] T.-T. Chien and M. B. Adams. A sequential failure detection technique and its application. *IEEE Transactions on Automatic Control*, 21(5):750–757, Oct 1976.

[Cover and Thomas, 1991] T. M. Cover and J. A. Thomas. *Elements of Information Theory.* John Wiley & Sons, 1991.

[Craig and Raibert, 1986] J. J. Craig and M. H. Raibert. A systematic method hybrid position/force control of a manipulator. *International Journal of Robotics Research*, 1986.

[Craig, 1989] J. J. Craig. *Introduction to Robotics: Mechanics & Control.* Addison-Wesley Series in Electrical and Computer Engineering: Control Engineering. Addison Wesley, 2 edition, 1989.

[Delson, 1994] N. J. Delson. *Robot Programming by Human Demonstration.* PhD thesis, MIT, Department of Mechanical Engineering, 1994.

[Donald and Jennings, 1991] B. Donald and J. Jennings. Sensor interpretation and task-directed planning using perceptual equivalence classes. In *Proc. IEEE International Conference on Robotics and Automation*, pages 190–197, Sacramento, CA, April 1991.

[Dornfeld and Handy, 1987] D. Dornfeld and C. Handy. Slip detection using acoustic emission signal analysis. In *Proc. IEEE International Conference on Robotics and Automation*, pages 1868–1875, 1987.

[Eberman and Salisbury, 1989] B. S. Eberman and J. K. Salisbury. Determination of manipulator contact information from joint torque measurements. In V. Hayward and O. Khatib, editors, *Experimental Robotics I, First International Symposium*, Montréal, Canada, June 1989. Published in: Lecture Notes in Control and Information Sciences, Springer-Verlag, 1990.

[Eberman and Salisbury, 1993] B. Eberman and J. K. Salisbury. Segmentation and interpretation of temporal contact signals. In *Experimental Robotics III, Third International Symposium*, Kyoto, Japan, October 1993.

[Eberman and Salisbury, 1994] B. Eberman and J. K. Salisbury. Application of change detection to dynamic contact sensing. *International Journal of Robotics Research*, 13(5), Fall 1994.

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

[Erdmann, 1986] M. A. Erdmann. Using backprojections for fine motion planning with uncertainty. *International Journal of Robotics Research*, 5(1):19–45, Spring 1986.

[Erdmann, 1989] M. A. Erdmann. *On Probabilistic Strategies for Robot Tasks.* PhD thesis, MIT, Department of Electrical Engineering and Computer Science, 1989.

[Erdmann, 1993] M. Erdmann. Action subservient sensing and design. In *Proc. IEEE International Conference on Robotics and Automation*, pages Vol 2: 592–598, 1993.

[Fedele *et al.*, 1993] A. Fedele, A. Fioretti, C. Manes, and G. Ulivi. On-line processing of position and force measures for contour identification and robot control.

In *Proc. IEEE International Conference on Robotics and Automation*, pages Vol 1 369–374, 1993.

[Goldenthal, 1994] W. Goldenthal. *Statistical Trajectory Models for Phonetic Recognition.* PhD thesis, MIT, Department of Aeronautics and Astronautics, 1994.

[Gordon and Townsend, 1989] S. J. Gordon and W. T. Townsend. Integration of tactile-force and joint-torque information in a whole-arm manipulator. In *Proc. IEEE International Conference on Robotics and Automation*, Scottsdale, AZ, 1989.

[Grimson, 1990] W. E. L. Grimson. *Object Recognition by Computer: The Role of Geometric Constraints.* Artificial Intelligence. The MIT Press, 1990.

[Hager, 1992] G. D. Hager. Constraint solving methods and sensor-based decision making. In *Proc. IEEE International Conference on Robotics and Automation*, pages 1662–1667, 1992.

[Hall, 1985] S. R. Hall. *A Failure Detection Algorithm for Linear Dynamic Systems.* PhD thesis, MIT, Department of Aeronautics and Astronautics, 1985.

[Hannaford and Lee, 1991] B. Hannaford and P. Lee. Hidden Markov Model analysis of force/torque information in telemanipulation. *International Journal of Robotics Research*, 10(5):528–539, 1991.

[Hogan, 1985] N. Hogan. Impedance control: An approach to manipulation: Part I-theory. *ASME Journal of Dynamic Systems Measurement and Control*, 107(1), 1985.

[Howe and Cutkosky, 1989] R. D. Howe and M. R. Cutkosky. Sensing skin acceleration for slip and texture perception. In *Proc. IEEE International Conference on Robotics and Automation*, pages 145–150, 1989.

[Howe and Cutkosky, 1991] R. D. Howe and M. R. Cutkosky. Touch sensing for robotic manipulation and recognition. In O. Khatib, J. Craig, and T. Lozano-Pérez, editors, *Robotics Review 2.* MIT Press, 1991.

[Howe *et al.*, 1990] R. D. Howe, N. Popp, P. Akella, I. Kao, and M. R. Cutkosky. Grasping, manipulation, and control with tactile sensing. In *Proc. IEEE International Conference on Robotics and Automation*, pages 1258–1263, Cincinnati, Ohio, May 1990.

[Kanekp and Tanie, 1992] M. Kanekp and K. Tanie. Self-posture changeability (spc) for 3-d link system. In *Proc. IEEE International Conference on Robotics and Automation*, pages 1635–1640, 1992.

[Lazanas and Latombe, 1992] A. Lazanas and J.-C. Latombe. Landmark-based robot navigation. Technical Report STAN-CS-92-1428, Department of Computer Science Stanford University, May 1992.

[Levin, 1990] M. D. Levin. Design and control of a closed-loop brushless torque actuator. Technical report, MIT Artificial Intelligence Laboratory, AI-TR 1244, 1990.

[Ljung, 1983] L. Ljung. *Theory and Practice of Recursive Identification*. MIT Press, 1983.

[Lozano-Pérez *et al.*, 1984] T. Lozano-Pérez, M. T. Mason, and R. H. Taylor. Automatic synthesis of fine-motion strategies for robots. *International Journal of Robotics Research*, 3(1):3–24, Spring 1984.

[Lozano-Pérez, 1976] T. Lozano-Pérez. The design of a mechanical assembly systems. Technical Report 397, MIT Artificial Intelligence Laboratory, 1976.

[Lozano-Pérez, 1983] T. Lozano-Pérez. Spatial planning: A configuration space approach. *IEEE Transactions on Computers*, C–32(2):108–120, feb 1983.

[Makhoul, 1975] J. Makhoul. Linear prediction: A tutorial review. *Proceedings of the IEEE*, 63(4):561–580, 1975.

[Mason and Salisbury, 1985] M. T. Mason and J. K. Salisbury. *Robot Hands and the Mechanics of Manipulation*. MIT Press, Cambridge, Massachusetts, 1985.

[Mason, 1981] M. T. Mason. Compliance and force control for computer controlled manipulators. *IEEE Transactions on Systems, Man, and Cybernetics*, 11(6):418–432, 1981.

[Mataric, 1994] M. J. Mataric. *Interaction and Intelligent Behavior*. PhD thesis, MIT, Department of Electrical Engineering and Computer Science, May 1994.

[McCarragher and Asada, 1993a] B. J. McCarragher and H. Asada. A discrete event approach to the control of robotic assembly tasks. In *Proc. IEEE International Conference on Robotics and Automation*, pages 331–336, 1993.

[McCarragher and Asada, 1993b] B. J. McCarragher and H. Asada. Qualitative template matching using process models for state transition recognition of robotic assembly. *ASME Journal of Dynamic Systems Measurement and Control*, 115:261–269, 1993.

[McCarthy, 1990] J. M. McCarthy. *An Introduction to Theoretical Kinematics*. MIT Press, 1990.

[Munkres, 1991] J. R. Munkres. *Analysis on Manifolds.* Addison-Wesley, 1991.

[Narasimhan, 1994] S. Narasimhan. *Task-level Strategies for Robots.* PhD thesis, MIT, Department of Electrical Engineering and Computer Science, 1994.

[Nevins *et al.*, 1973] J. L. Nevins, D. E. Whitney, and S. N. Simunovic. System architecture for assembly machines – report on advanced automation. Technical Report NSF-RA-T-73-101, NSF, November 1973.

[Oppenheim and Flanagan, 1989] A. V. Oppenheim and R. W. Flanagan. *Discrete-time Signal Processing.* Prentice-Hall, 1989.

[Press *et al.*, 1992] W. H. Press, S. A. Teukolsky, W. T. Vetterling, and B. P. Flannery. *Numerical Recipes in C.* Cambridge University Press, 2 edition, 1992.

[Rabiner, 1989] L. R. Rabiner. A tutorial on hidden markov models and selected applications in speech recognition. *Proceedings of the IEEE*, 77(2):257–286, February 1989.

[Robles, 1995] J. L. Robles. *Contact Interpretation in Randomized Strategies for Robotic Assembly.* PhD thesis, MIT, Department of Electrical Engineering and Computer Science, 1995.

[Salisbury, 1980] J. K. Salisbury. Active stiffness control of a manipulator in cartesian coordinates. In *19th IEEE Conference on Decision and Control*, Albuquerque, NM, December 1980.

[Salisbury, 1982] J. K. Salisbury. *Kinematic and Force Analysis of Articulated Hands.* PhD thesis, Stanford University Mechanical Engineering and Computer Science Dept., June 1982.

[Salisbury, 1984] J. K. Salisbury. Interpretation of contact geometries from force measurements. In M. Brady and R. Paul, editors, *Robotics Research: The First International Symposium*, Bretton Woods, 1984. MIT Press, Cambridge, MA.

[Schimmels and Peshkin, 1993] J. M. Schimmels and M. A. Peshkin. The space of admittance control laws that guarantees force-assembly with friction. In *Proc. IEEE International Conference on Robotics and Automation*, pages Vol 3. p 443–448, 1993.

[Siegel, 1991] D. M. Siegel. *Pose Determination of a Grasped Object Using Limited Sensing.* PhD thesis, MIT, Department of Electrical Engineering and Computer Science, 1991.

[Siegmund, 1985] D. Siegmund. *Sequential Analysis, Tests and Confidence Intervals.* Springer-Verlag, 1985.

[Simunovic, 1979] S. N. Simunovic. *An Information Approach to Parts Mating.* PhD thesis, MIT, Department of Mechanical Engineering, April 1979.

[Slotine and Li, 1987] J. E. Slotine and W. Li. On the adaptive control of robot manipulators. *International Journal of Robotics Research*, 6(3), 1987.

[Srinivasan, 1991] M. A. Srinivasan. Background on human haptic system. Appendix to VETT Proposal, 1991.

[Stansfield, 1987] S. A. Stansfield. Visually-aided tactile exploration. In *Proc. IEEE International Conference on Robotics and Automation*, pages 1487–1492, 1987.

[Tanaka and Muller, 1990] S. Tanaka and P. C. Muller. Fault detection in linear discrete dynamic systems by a pattern recognition of a generalized-likelihood-ratio. *ASME Journal of Dynamic Systems Measurement and Control*, 112(2):276–282, 1990.

[Taylor, 1976] R. H. Taylor. The synthesis of manipulator control programs from task-level sepecifications. AIM 282, Stanford Artificial Intelligence Laboratory, 1976.

[Tugnait and Haddad, 1979] J. K. Tugnait and A. H. Haddad. A detection-estimation scheme for state estimation in switching environments. *Automatica*, 15:477–481, 1979.

[Wald, 1947] A. Wald. *Sequential Analysis.* John Wiley & Sons, 1947.

[Whitney, 1977] D. E. Whitney. Force feedback control of manipulator fine motions. *ASME Journal of Dynamic Systems Measurement and Control*, pages 91–97, June 1977.

[Whitney, 1982] D. E. Whitney. Quasi-static assembly of compliantly supported rigid parts. *ASME Journal of Dynamic Systems Measurement and Control*, pages 65–104, March 1982.

[Whitney, 1985] D. E. Whitney. Historical perspectives and state of the art in robot force control. In *Proc. IEEE International Conference on Robotics and Automation*, pages 262–268, 1985.

[Will and Grossman, 1975] P. M. Will and D. D. Grossman. An experimental system for computer controlled mechanical assembly. *IEEE Transactions on Computers*, 24(9):879–888, 1975.

[Willsky and Jones, 1976] A. S. Willsky and H. L. Jones. A generalized likelihood ratio approach to the detection and estimation of jumps in linear systems. *IEEE Transactions on Automatic Control*, 21(1):108–112, February 1976.

[Willsky *et al.*, 1980] A. S. Willsky, E. Y. Chow, S. B. Gershwin, C. S. Greene, P. K. Houpt, and A. L. Kurkjian. Dynamic model-based techniques for the detection of incidents on freeways. *IEEE Transactions on Automatic Control*, 25(3):347–360, June 1980.

[Willsky, 1976] A. S. Willsky. A survey of design methods for failure detection in dynamic systems. *Automatica*, 12:601–611, 1976.

[Willsky, 1990] A. S. Willsky. Recursive estimation. Department of Electrical Engineering and Computer Science, MIT, 1990.

[Zeitouni *et al.*, 1992] O. Zeitouni, J. Ziv, and N. Merhav. When is the generalized likelihood ratio test optimal? *IEEE Transactions on Information Theory*, 38(5):1597–1602, sep 1992.