Geometric and Dynamic Sensing:
Observation of Pose and Motion through Contact

Yan-Bin Jia
December 1997
CMU-RI-TR-97-50

The Robotics Institute
Carnegie Mellon University
Pittsburgh, PA 15213

Submitted in partial fulfillment of the requirements for the degree of Doctor of Philosophy.

Thesis Committee:
Michael A. Erdmann, Chair
Matthew T. Mason
Bruce R. Donald, Dartmouth College
Katsushi Ikeuchi, University of Tokyo

Copyright © 1997 Yan-Bin Jia

Support for this research was provided in part by Carnegie Mellon University, and in part by the National Science Foundation through the following grants: NSF Research Initiation Award IRI-9010686, NSF Presidential Young Investigator award IRI-9157643, and NSF Grants IRI-9213993 and IRI-9503648. The views and conclusions expressed in this document are those of the author, and do not necessarily reflect the views of Carnegie Mellon University, NSF, or the U.S. Government.
To my parents
Abstract

I investigate geometric and mechanical sensing strategies for objects of known shapes. Examples include industrial parts and everyday desktop items. Based on nonlinear control theory, I show that local observability from contact holds in typical manipulation tasks, and I present an approach for estimating the pose and motion of a manipulated object from a small amount of tactile data.

The first part of the thesis describes two geometric strategies, namely inscription and point sampling, for computing the pose of a polygonal part; both can generalize to other shapes in two and three dimensions. These strategies use simple geometric constraints to either immobilize the object or to distinguish its real pose from a finite number of apparent poses. Computational complexity issues are examined. Simulation results support their use in real applications.

The second and main part of the thesis introduces a sensing strategy called pose and motion from contact. I look at two representative tasks: (1) a finger pushing an object in the plane; and (2) a three-dimensional smooth object rolling on a translating horizontal plane. I demonstrate that essential task information is often hidden in mechanical interactions, and show how this information can be properly revealed.

The thesis proves that the nonlinear dynamical system that governs pushing in the first task is locally observable. Hence a sensing strategy can be realized as an observer of the system. I have developed two nonlinear observers. The first one determines its “gain” from the solution of a Lyapunov-like equation. The second one solves for the initial (motionless) pose of the object from as few as three intermediate contact points. Both observers have been simulated and a contact sensor has been implemented using strain gauges.

Through cotangent space decomposition in the second task, I derive a sufficient condition on local observability for the pose and motion of the rolling object from its path in the plane. This condition depends only on the differential geometry of contact and on the object’s angular inertia matrix. It is satisfied by all but some degenerate shapes such as a sphere.
Acknowledgments

I am very fortunate to have had Mike Erdmann as my thesis advisor. Mike was a tremendous source of advice, encouragement, and appreciation over the years. He understood the importance of fostering my independent research ability and yet made sure I had a big picture in mind and some sound strategies to carry the work through. A number of ideas reported in this thesis stemmed from numerous discussions with Mike. The thesis writing also benefited greatly from his feedback detailed enough to include even corrections of my English. As an intellectual he set a complete example of integrity, creativity, modesty, thoroughness, and hard work; as a friend he carried enthusiasm, openness, help, and a rich sense of humor.

I am also indebted to Matt Mason for his many insights and for helping stretch my view on robotic manipulation from computer science to mechanics and control. Matt was always able to interpret things intuitively and to offer constructive feedback by making connections to seemingly unrelated topics that turned out to be worthy of looking at. His pointer to nonlinear observers resulted in a change of the course of the thesis — the application of nonlinear observability theory greatly enriched its second part.

It was my great pleasure to have spent the summer of 1996 working closely with Bruce Donald, who showed me that many real problems, including some in robotics, could be tackled algorithmically. Lots of thanks to Bruce for his appreciation and support through the years, and to Katsu Ikeuchi for serving on my thesis committee and for providing valuable comments on the thesis draft.

It was very fortunate of me to have had the chance to meet many leading robotics researchers outside CMU and to have benefited from the interactions with them. I would especially like to thank Ken Goldberg, Seth Hutchinson, Ken Salisbury, Jeff Trinkle, Oussama Khatib, Bud Mishra, John Canny, and Dan Koditschek for their feedback and encouragement. Thanks also go to Jean-Daniel Boissonnat, Mark Overmars, Anil Rao, Kathleen Romanik, and the anonymous reviewers who provided valuable comments on my conference and journal papers which formed part of this thesis.

My life as a graduate student would have never been better without the remarkable working environment provided by the Robotics Institute and School of Computer Science at Carnegie Mellon. Thanks to Steve Shafer for overseeing my courses in the first one and a half years and for offering useful research tips. Thanks to Catherine Copetas, Marce Zaragoza, and Phyllis Pomerantz for always being eager to help out. My work also benefited from the generous help offered by a number of CMU research faculty and staff, especially Alan Guisewite, Mel Siegel, and Ralph Hollis.

Matt Mason and Mike Erdmann were responsible for making the Manipulation Lab such a stimulating and fun place to work. One of many nice things about having an Mlab membership was to go out and socialize with other members every two or three weeks, having some beer, drinks, ice cream, and tons of interesting chat. I would like to thank Kevin Lynch, Srinivas Akella, Tammy Abell, Nina Zumel, Wes Huang, Garth Zeglin, and Mark Moll for being wonderful labmates, and for being patient audience and sharp critics.
at my research/practice talks. Thanks to Kevin for introducing me to his related work on active sensing; to Wes for realizing the connection between my work and observer design; and to Garth for giving generous help with the experiments. Thanks also to our affiliated lab member Al Rizzi for his insights and helpful discussions.

Over the years I have had some wonderful officemates from computer science and robotics. Dennis Grinberg was always willing and generous to offer help when I needed. James Landay’s humor and whining only made our office a place many people liked to frequent. Lars Birkedal, Peter Berkelman, and Steve Chen were fun to be with and also good reminders of speeding up my graduation. Steve was always eager to share with me his broad sports and movie knowledge.

Thanks also go to Harry Shum, Yalin Xiong, Somesh Jha, Kan Deng, and others for sharing the joys of life as graduate students at CMU.

My life would have never been the same had I not been admitted to the Special Class for Gifted Young at the University of Science and Technology of China in 1984. In college I was very fortunate to be associated with some of the smartest talents I have ever met and to develop everlasting friendships with them. My view of life and style of learning, shaped mostly back then under the influence of these individuals, are still exerting major impact on me today. Jian Zhang convinced me of scholarly excellence and mathematical perfection in college, setting a solid standard that I always look up to and try to emulate. Charles Qin Huang was an early inspiration of pursuing graduate study in the states and is always generous with his help and insights. Xintian Lin provided those entertaining biweekly phone conversations for almost seven years, keeping my morale high, my eyesight open, while my engine relaxed. Thanks to Tiejun Wang, Xiaochun Luo, Wenshuo Li, and Yi Du for helping me focus on the Ph.D. work. Great thanks to Charles Huang, Chao Yuan, Jianzhong Shen, and many others for spending tremendous amounts of time and effort on helping me recover from an almost fatal car accident on December 21, 1993. I would have not bounced back this well so rapidly without your genuine and unselfish love and support.

I also thank my former supervisor Professor C. S. Tang in the Institute of Software, Chinese Academic of Sciences for his influence, encouragement, and philosophical insights that I could draw upon from time to time.

Finally, I thank my parents and sister for your love, understanding, patience, and support through the years. Dad started my interests in elementary mathematics and physics, taught me the meaning of integrity, and was instrumental in every step I took before graduate school. Mom was always passionate but also headstrong, having overcome a great deal of difficulty and pressure to take care of me after the aforementioned car accident. Her love and care contributed so significantly to my recovery. I love you all very much. I always shall.
# Contents

## 1 Introduction

1.1 Geometry and Mechanics in Sensing ........................................... 1  
  1.1.1 Geometric Sensing .................................................. 2  
  1.1.2 Mechanics-based Observation of Pose and Motion .................. 4  

1.2 Thesis Overview ............................................................... 5  
  1.2.1 Inscription ......................................................... 6  
  1.2.2 Point Sampling ..................................................... 6  
  1.2.3 Pose and Motion from Contact .................................... 7  
  1.2.4 Observer Simulations and Sensor Implementation ............... 7  
  1.2.5 Local Observability of Rolling ................................. 8  

1.3 Related Research ............................................................... 8  
  1.3.1 Sensor Design and Geometric Sensing ............................. 8  
  1.3.2 Parts Orienting ................................................... 9  
  1.3.3 Force and Tactile Sensing ....................................... 10  
  1.3.4 Model-based Recognition ....................................... 10  
  1.3.5 Computational Geometry and Complexity Theory ............... 11  
  1.3.6 Nonlinear Observability Theory ................................ 12  
  1.3.7 Contact Kinematics ............................................... 13  
  1.3.8 Rigid Body Dynamics ............................................ 14  
  1.3.9 Quasi-Static Pushing ........................................... 15  
  1.3.10 Dextrous Manipulation ........................................ 16  

## I Geometric Sensing

## 2 Polygon Inscription

2.1 Multi-Cone Inscription ..................................................... 20  
  2.1.1 A Triangle Sliding in a Cone ................................... 20  
  2.1.2 One-Cone Inscription ........................................... 22  
  2.1.3 Upper Bounds .................................................. 25  
  2.1.4 An Algorithm for Inscription .................................. 27  

2.2 Simulations ................................................................. 27  

2.3 Summary ................................................................. 32
# The Complexity of Point Sampling

3.1 The Formal Problem .................................................. 37
3.2 Discriminating Set .................................................. 39
3.3 Region Basis .......................................................... 41
3.4 Approximation ........................................................ 43
3.5 More on Discriminating Set ........................................... 47
3.6 Simulations ............................................................. 50
3.7 Summary ............................................................... 56

# Observing Pose and Motion through Contact

4 Mathematical Basics .................................................... 59
4.1 Elements of Differential Geometry .................................. 59
4.1.1 Curves .............................................................. 59
4.1.2 Surfaces .......................................................... 60
4.1.3 Normal and Gaussian Curvatures .................................. 60
4.1.4 Curves in a Surface ............................................... 61
4.1.5 Patch Computations ................................................ 62
4.1.6 Manifolds, Cotangent Bundles, Codistributions .................. 62
4.1.7 Lie Derivatives .................................................... 63
4.2 Nonlinear Control ..................................................... 64
4.2.1 Observability of a Nonlinear System ............................ 64
4.2.2 Nonlinear Observers ............................................... 66

5 Pose and Motion from Contact ......................................... 71
5.1 Motion of Contact ..................................................... 72
5.1.1 Degenerate Case ................................................... 75
5.1.2 Integral of Support Friction ....................................... 76
5.1.3 Initial Motion ...................................................... 78
5.1.4 Contact Breaking .................................................. 80
5.2 Local Observability .................................................... 80
5.2.1 The Disk-Polygon System ........................................ 80
5.3 Pose Observers ......................................................... 84
5.3.1 A Gauthier-Hammouri-Othman Observer ......................... 85
5.3.2 The Initial Pose Observer ........................................ 86
5.4 Contact Friction ....................................................... 88
5.4.1 Rolling ............................................................ 88
5.4.2 Sliding ............................................................ 90
5.5 Summary ............................................................... 91
6 Observer Simulations and Sensor Implementation
   6.1 On the GHO Observer ........................................... 94
   6.2 On the IP Observer ............................................. 100
   6.3 Preliminary Experiments ......................................... 101
   6.4 Sensor Implementation ......................................... 103
   6.5 Ambiguities of Sensing ......................................... 104

7 Rolling on a Moving Plane — Mechanics and Local Observability 107
   7.1 The Rolling Motion ............................................. 108
       7.1.1 Kinematics of Rolling ..................................... 109
       7.1.2 Dynamics of Rolling ....................................... 113
   7.2 Local Observability of Rolling ................................. 115
       7.2.1 Angular Velocity ........................................... 116
       7.2.2 Contact on the Object ..................................... 117
       7.2.3 Rotation about the Contact Normal ....................... 120
       7.2.4 Sufficient Condition for Local Observability .......... 123
   7.3 Summary ......................................................... 123

8 Conclusions 125
   8.1 Issues on Geometric Sensing ................................. 127
   8.2 Mechanics-based Observation ................................ 128
       8.2.1 Remarks on Local Observability ....................... 129
       8.2.2 Observer-related Issues .................................. 130

A A Worst Case of Inscription 133

B k-Independent Sets 137

C Integral $\Gamma$ of Support Friction (Disk-Polygon) 139
   C.1 Integrals $I_1$ and $I_2$ .................................... 139
   C.2 Partial Derivatives of $\Gamma$ ................................. 141
   C.3 Partial Derivatives of $I_1$ and $I_2$ on I.R.C. .......... 143

D Observer for Rolling (Disk-Polygon) 145

E Infeasibility of Linearization by Output Injection 147
List of Figures

1.1 Geometric sensing of an 8-gon ................................................. 3
1.2 Different contact motions resulting from different poses .................. 5

2.1 Sensing the pose of a polygon by taking two views. .......................... 20
2.2 A triangle sliding in a cone ................................................ 21
2.3 A convex polygon rotating in a cone ........................................ 23
2.4 Eight possible poses for a convex 6-gon bounded by three supporting lines 30
2.5 Two possible poses for a convex 6-gon inscribed in two cones ............. 32

3.1 Sensing by point sampling ...................................................... 36
3.2 Two examples whose minimum region basis sizes achieve the lower and upper bounds, respectively ................................. 38
3.3 Two reductions from Discriminating Set to Region Basis. .................... 43
3.4 An example of reduction from Independent Set to 3-IS ....................... 50
3.5 Sampling sparsely distributed random polygons/poses ....................... 52
3.6 Sampling densely distributed polygons ...................................... 55

4.1 Local observability at a state .................................................. 65

5.1 Anatomy of pushing. ............................................................. 72
5.2 A finger translating and pushing an object ................................... 73
5.3 The instantaneous rotation center of an object ................................ 77
5.4 A circular finger pushing a polygonal object .................................. 81
5.5 A shooting method for initial pose determination ............................. 87
5.6 The rolling and sliding modes under contact friction ........................ 88

6.1 A unit disk at constant velocity pushing and observing a 7-gon: the case of rolling contact ......................................................... 95
6.2 State variable trajectories vs. state estimate trajectories for the example shown in Figure 6.1 ......................................................... 96
6.3 A unit disk at constant velocity pushing and observing a 5-gon: the case of frictionless contact ......................................................... 98
6.4 State variable trajectories vs. state estimate trajectories for the example shown in Figure 6.3 ......................................................... 99
6.5 Experimental setup of pose-from-pushing ..................................... 101
## List of Tables

<table>
<thead>
<tr>
<th>Number</th>
<th>Title</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>2.1</td>
<td>Experiments on embedding a polygon in three half-planes.</td>
<td>28</td>
</tr>
<tr>
<td>2.2</td>
<td>Experiments on inscribing polygons with two cones.</td>
<td>29</td>
</tr>
<tr>
<td>2.3</td>
<td>More experiments on inscribing polygons with cones.</td>
<td>31</td>
</tr>
<tr>
<td>3.1</td>
<td>Tests on sampling sparsely distributed random polygons/poses</td>
<td>53</td>
</tr>
<tr>
<td>3.2</td>
<td>Tests on sampling densely distributed random polygons/poses</td>
<td>54</td>
</tr>
<tr>
<td>6.1</td>
<td>Simulations on observing the poses and motions of random polygons being pushed by a unit disk, assuming known contact edges</td>
<td>97</td>
</tr>
<tr>
<td>6.2</td>
<td>Simulations on observing the poses and motions of random polygons being pushed by a unit disk, assuming unknown contact edges</td>
<td>100</td>
</tr>
<tr>
<td>6.3</td>
<td>Simulations of the IP observer with the frictionless plane.</td>
<td>101</td>
</tr>
</tbody>
</table>
List of Symbols

\( \mathbb{R} \) field of real numbers
\( \mathbb{R}^2 \) Euclidean 2-space (plane)
\( \mathbb{R}^3 \) Euclidean 3-space
\( i, j, k, l, m, n \) integers
\( x, y, z \) coordinates
\( p, q \) points
\( \alpha \) curve

Part I

\( O(n) \) \( O \)-notation
\( \Theta(n) \) \( \Theta \)-notation
\( P, Q \) polygons

Chapter 2

\( \theta, \psi, \Phi \) angles
\( C \) cone
\( L \) line
\( a, b \) constants

Chapter 3

\( R \) bounded planar region
\( \Omega, \Delta \) sets of bounded planar regions
\( X, D, H \) finite sets
\( 2^X \) power set of the finite set \( X \)
\( C \) collection of subsets of a finite set
\( |C| \) size of the collection \( C \)
\( \emptyset \) empty set
\( a, x \) elements
\( S \triangle T \) symmetric difference between two sets \( S \) and \( T \)
\( H(k) \) \( k \)-th harmonic number
\( G \) graph
List of Symbols

\( V \) vertex set of a graph
\( E \) edge set of a graph
\( u, v, w \) vertices of a graph
\( e, (u,v) \) edges of a graph
\( d(u,v) \) distance between the vertices \( u \) and \( v \)

Part II

\( f, g \) vector fields, patches, etc.
\( u, v \) curve or surface parameters
\( X, Y, Z \) vector fields
\( Z_\alpha \) restriction of the vector field \( Z \) to the curve \( \alpha \)
\( \nabla_v Z \) covariant derivative of the vector field \( Z \) with respect to the tangent vector \( v \)
\( T \) unit tangent vector field on a curve
\( N \) principal normal vector field
\( B \) binormal vector field
\( \kappa \) curvature of a curve
\( \tau \) torsion of a curve
\( M \) manifold (or surface in \( \mathbb{R}^3 \))
\( U \) surface normal vector field
\( S_p \) shape operator of a surface at the point \( p \)
\( \kappa_n(u) \) normal curvature in the \( u \) direction
\( \kappa_1, \kappa_2 \) principal curvatures of a surface
\( K \) Gaussian curvature
\( H \) mean curvature
\( \kappa_g \) geodesic curvature of a curve in a surface
\( \tau_g \) geodesic torsion
\( T_p M \) tangent space at the point \( p \) of the manifold \( M \)
\( T^*_p M \) cotangent space at the point \( p \) of the manifold \( M \)
\( \phi \) one-form
\( df \) differential of the real-valued function \( f \)
\( L_X f \) Lie derivative of the function \( f \) w.r.t. the vector field \( X \)
\( L_{X_1} \cdots L_{X_n} f \) repeated Lie derivative of the function \( f \) w.r.t. the vector fields \( X_n, \ldots, X_1 \)
\( [X,Y] \) Lie bracket of the vectors fields \( X \) and \( Y \)
\( \text{ad}_X Y \) repeated Lie bracket of the vector field \( Y \) along the vector field \( X \)
\( \sigma_{z_0}(t) \) integral curve resulting from the initial state \( z_0 \)
\( Z^t \) flow of the vector field \( Z \)
\( \mathcal{O} \) observation space of a nonlinear system
\( d\mathcal{O} \) observability codistribution
\( \mathcal{B} \) object
\( \beta \) object boundary
List of Symbols

\( m \) \hspace{1em} mass \\
\( I \) \hspace{1em} angular inertia \\
\( R \) \hspace{1em} rotation matrix \\
\( g \) \hspace{1em} gravitational acceleration \\
\( u, u \) \hspace{1em} contact location on the manipulator (finger, palm) \\
\( s, s \) \hspace{1em} contact location on the object \\
\( F \) \hspace{1em} contact force on the object \\

Chapter 5

\( \rho \) \hspace{1em} radius of gyration \\
\( A \) \hspace{1em} area \\
\( v \) \hspace{1em} object velocity (in the world frame) \\
\( \omega \) \hspace{1em} object angular velocity (in the world frame) \\
\( (x_0^B, y_0^B) \) \hspace{1em} instantaneous rotation center \\
\( F \) \hspace{1em} finger \\
\( v_F \) \hspace{1em} finger velocity \\
\( r \) \hspace{1em} disk radius \\
\( h \) \hspace{1em} distance from the polygon centroid to the contact edge \\
\( \mu \) \hspace{1em} coefficient of support friction \\
\( \mu_c \) \hspace{1em} coefficient of contact friction \\
\( \Gamma \) \hspace{1em} integral of friction \\
\( f \) \hspace{1em} drift field of pushing \\
\( g_T \) \hspace{1em} tangential input field \\
\( g_N \) \hspace{1em} normal input field \\
\( \zeta \) \hspace{1em} control parameter of the Gauthier-Hammouri-Othman observer \\

Chapter 7

\( v \) \hspace{1em} object velocity (in the moving body frame) \\
\( v_B \) \hspace{1em} object velocity (in the fixed instantaneous frame) \\
\( \omega \) \hspace{1em} object angular velocity (in the body frame) \\
\( \omega_B \) \hspace{1em} object angular velocity (in the fixed instantaneous frame) \\
\( \mathcal{P} \) \hspace{1em} palm (or plane) \\
\( a_P \) \hspace{1em} palm acceleration \\
\( \Pi, \Pi_B \) \hspace{1em} frames \\
\( x, y, z \) \hspace{1em} unit vectors along coordinate axes \\
\( \psi \) \hspace{1em} rotation angle about the contact normal \\
\( R_\psi \) \hspace{1em} rotation matrix about the contact normal \\
\( \rho \) \hspace{1em} radius of curvature \\
\( V \) \hspace{1em} metric of the object surface \\
\( I_n \) \hspace{1em} \( n \times n \) identity matrix
Chapter 1

Introduction

This thesis is about sensing objects whose shapes are known in advance. Examples are industrial parts manufactured according to their geometric models. Sensing in robotics traditionally involves determining the shape and pose of a part. In this thesis we consider determining the pose as well as the motion of the part. By *pose* we mean the position as well as the orientation of the part. By *motion* we mean the velocity, angular velocity, etc., of the part. We deal with general shapes in two and three dimensions.

Object geometry affects almost every operation in a robotic task, ranging from elementary ones such as sensing and grasping to complex ones such as dextrous manipulation and assembly. For instance, without making any assumptions, we might not even be able to start segmentation of the part image, whereas knowing that the shape is convex polygonal, we can employ some simple non-vision technique such as finger probing. An understanding of the role of object geometry could enhance task planning as well as simplify hardware design.

A very important part of this understanding is on what can be sensed in a task, and on how to interpret, based on the known geometry, the sensor data to acquire information that is essential for the task. More specifically, we want to know more about what task information is encoded, possibly through some manipulation operations, in data that can be sensed, and about how to decode such information using task geometry and mechanics as a “key”. On one hand, the sensible data should be simple and easy to obtain so as to simplify the physical sensor requirement. On the other hand, such data should result in enough information for sensing to be done. Our work tries to explore the limits in both directions.

1.1 Geometry and Mechanics in Sensing

An effective sensing method should make use of its knowledge about the part shape as much as possible to attain simplicity, efficiency, and robustness. The geometric sensing algorithms to be described in Chapters 2 and 3 investigate the first two issues. There we intend to explore the extremes of hardware simplicity and computational efficiency, and the tradeoff between them.
CHAPTER 1. INTRODUCTION

Geometry and mechanics are always closely tied to each other in a task. The states (or configurations) of an object and a manipulator evolve according to mechanical laws but under the geometric constraints which they need to satisfy to continue the interaction. This interaction often yields simple information such as contact location and force, into which the task geometry is encoded by the mechanics of the manipulation. Conversely, from such encoded information we may be able to recover some unknowns about the task, especially the configuration of the object. This is the underlying principle of our observer-based sensing method to be introduced in Chapter 5. Sensing based on both geometry and mechanics may simplify the sensor hardware even more, as fewer sensors may be needed.

Part sensing and manipulation are often performed sequentially in an assembly task. Mechanics-based sensing may help integrate the two together, reducing the assembly time and cost.

In the rest of this section, we would like to discuss the sensing problems dealt with in this thesis a little further.

1.1.1 Geometric Sensing

One type of sensing employs simple sensors to obtain geometric constraints that can either immobilize a part or distinguish one possible pose from others in a finite set, and then solves a constraint satisfaction problem for the pose using an efficient algorithm. The knowledge about the part geometry is compiled into the algorithm. The size of geometric constraints not only is a measure of the hardware complexity, but also directly affects how fast the physical portion of sensing can be performed. Minimizing the size of sufficient constraints thus becomes as important as improving the running time of a sensing algorithm. Such minimization, however, cannot be achieved efficiently.

Chapters 2 and 3 describe two different approaches for sensing polygonal parts of known shape, one applicable to the case of a continuum of possible poses and the other applicable to the case of a finite number of possible poses. In the below we briefly look at the kinds of geometric constraints that will be involved in these approaches.

Perhaps the simplest geometric constraint on a polygon is incidence—when some edge touches a fixed point or some vertex is on a fixed line. For instance, Figure 1.1(a) shows an 8-gon constrained by two points \( p_1, p_2 \) and two lines \( L_1, L_2 \). The question we want to ask is: Generally, how many such constraints are necessary to fix the polygon in its real position? Note that any two such incidence constraints will confine all possible positions to a locus curve which consists of a finite number of algebraic curves parameterized by the part’s orientation. Three constraints, as long as not defined by collinear points or concurrent lines, will allow only a finite number of valid poses. These poses occur when different locus curves, each given by a pair of constraints, intersect at the same orientations. An upper bound on the number of possible poses can be analyzed. The addition of a fourth constraint is usually enough to reduce this number to one—the real pose.

If all incidences are given by lines, sensing can be viewed as inscribing the polygon into a larger polygon (not necessarily bounded) defined by these lines; if all constraints are given
by points, sensing can be viewed as placing the polygon defined by these points into the sensed polygon.

Point constraints can be created by “probing” the polygon along various directions with a tactile sensor or a range finder. Line constraints can be obtained with an angular sensor scanning across the object at exterior sites. In Chapter 2 we study the case of sensing with line constraints only, offering a very efficient algorithm to solve for all possible poses. We also derive a tight upper bound on the number of poses given three line constraints and conduct experiments to show that four line constraints are practically enough to determine the real pose.

The set of possible poses can often be reduced from a continuum to a finite number in advance by planned manipulations such as pushing or squeeze-grasping, or by sensing geometric constraints as mentioned above. More specialized sensing methods can be devised to distinguish between the remaining finite number of poses. We now view each pose as a closed set of points in the plane occupied by the part in that pose, so that sensing becomes distinguishing point sets, say, $P_1, \ldots, P_n$. An easy way is to sample several points, checking which of them are contained in the current point set (the real pose). Suppose the same 8-gon is known to have only three possible poses, as shown in Figure 1.1(b), then the real pose (shown with a solid line) can be determined by verifying that it contains both points $q_1$ and $q_2$. This can be implemented in a number of different ways, for instance, by placing light detectors underneath the point locations or by probing the point locations from above with a robot finger.

In Chapter 3 we address the problem of distinguishing finitely many polygons by point sampling. We prove that minimizing the total number of sampling points is NP-complete.

Figure 1.1: Geometric sensing of an 8-gon. (a) The case of a continuum of possible poses: Four incidence constraints $p_1, p_2, L_1$, and $L_2$ suffice to determine the pose. (b) The case of a finite number (three) of possible poses: Only the containment of two points $q_1$ and $q_2$ needs to be checked to determine the real pose.
and offer an approximation algorithm with greedy heuristic. The algorithm produces a set of sampling points whose size is within a factor of $2 \log n$ of the optimal. We also exhibit a proof demonstrating the hardness of improving this approximation ratio.

1.1.2 Mechanics-based Observation of Pose and Motion

Traditionally, sensing is conducted prior to manipulation. After the pose of a part has been determined, the manipulator performs operations such as pick-and-place, pushing, or dextrous maneuver in an open-loop way. This sequencing of a task has several drawbacks. The manipulator is hardly reactive to any inaccuracy of sensing or to any uncertainties that will likely happen during an operation, especially when it is more complicated than a simple task, say, pick-and-place. Robot vision hardly functions well when occlusions of the part happen in the middle of a task, which is often the case with dextrous manipulation. Geometric sensing methods are ideal for determining motionless poses with minimum hardware. But the hardware is often fixed and separated from the manipulator in motion. When sensing needs to be conducted multiple times in a task, multiple hardware may be needed and the sensing algorithm may become much more complicated and less efficient too. In addition, vision techniques are not good for estimating motion, whereas most geometric sensing algorithms have so far been developed for pose estimation only.

On the other hand, the mechanics of manipulation involves not only the geometry but also the motions of a part and its manipulator. The interactions between the part and the manipulator are affected by their configurations (poses and motions) and in a weak sense their contact (location and force) encodes the configuration information.

Consider the task of grasping something, say, a pen, on the table while keeping your eyes closed. Your fingers fumble on the table until one of them touches the pen and (inevitably) starts pushing it for a short distance. While feeling the contact move on the fingertip, you can almost tell which part of the pen is being touched. Assume the pushing finger is moving away from you. If the contact remains almost stable, then the middle of the pen is being touched; if the contact moves counterclockwise on the fingertip, then the right end of the pen is being touched; otherwise the left end is being touched. Immediately, a picture of the pen configuration has been formed in your head so you can coordinate other fingers to quickly close in for a grip.

The above example suggests that the human hand has some intrinsic way of exploiting the knowledge about the shape of an object and the tactile information from interacting with it, and of utilizing the mechanics of such interaction. To better illustrate this idea, Figure 1.2 shows two motions of a quadrilateral in different initial poses pushed by an ellipse under the same motion. Although the initial contacts on the ellipse were the same, the final contacts are quite far apart. Think in reverse:

1. Can we determine the pose of an object with known geometry and mechanical properties from the contact motion on a single pushing finger, or simply, from a few intermediate contact positions during the pushing?

2. Can we determine any intermediate pose of the object during the pushing?
3. Furthermore, can we estimate the motion of the object during the pushing?

The second part of the thesis, which includes Chapters 5 through 7, tries to answer these questions by looking at the above pushing problem and the problem of a smooth 3-dimensional object rolling on a moving palm. We look at the nonlinear systems that govern the dynamics of these two problems, and link sensing feasibility directly to nonlinear observability while sensing algorithms to nonlinear observers. We find that in both two problems the pose and motion of an object can be generally and locally observed from the contact information. We also describe observers in the first problem that asymptotically estimate the pose and motion from feedback.

1.2 Thesis Overview

In this section, we make brief introduction to the ensuing chapters. Chapters 2, 3, 5, and 7 include the main results of the thesis. They are independent of each other and can thus be read in any order that the reader prefers. The comprehension of Chapters 5 and 7, however, requires some background in differential geometry and nonlinear control, which is presented in Chapter 4. A reader familiar with the mathematics material may skip this chapter. Chapter 6 is about the simulation and implementation of the results in Chapter 5 and therefore has some dependency.

The work in Chapter 2 and 3 was published as [75]. Part of the work in Chapters 5 and 6 was reported earlier in [76]. And the other part is to appear in [78]. The work in Chapter 7 is to be published in [77]. Our investigation employs techniques in computational geometry,
complexity theory, mechanics, differential geometry, and nonlinear control theory.

1.2.1 Inscription

In Chapter 2, we introduce a sensing approach very applicable on a planar part which has a continuum of possible poses. In this approach, sensor readings are the supporting lines of the part that constrain its pose. A simple algorithm is then used to compute the often unique pose that satisfies all the constraints.

More specifically, the problem in this chapter involves determining the pose of a convex \( n \)-gon from a set of \( m \) supporting cones, that is, cones with both sides supporting the polygon. An algorithm with running time \( O(nm) \) that almost always reduces to \( O(n + m \log n) \) is presented to solve for all possible poses of the polygon. As a consequence, the polygon inscription problem of finding all possible poses of a convex \( n \)-gon inscribed in another convex \( m \)-gon, can be solved within the same asymptotic time bound. We prove that the number of possible poses cannot exceed \( 6n \), given \( m \geq 2 \) supporting cones with distinct vertices.

Simulation experiments demonstrate that two supporting cones are sufficient for determining the real pose of the \( n \)-gon in most cases. Our results imply that sensing in practice can be carried out by obtaining viewing angles of a planar part at multiple exterior sites in the plane (or solid angles subtended by a 3-dimensional part at different locations in the space).

1.2.2 Point Sampling

On many occasions a parts feeder will have reduced the number of possible poses of a part to a small finite set. In order to distinguish between the remaining poses of the part some simple sensing or probing operation may be used. Chapter 3 is concerned with a more general version: finding the minimum number of sensing points required to distinguish between \( n \) polygonal shapes with a total of \( m \) edges. In practice this can be carried out by embedding a series of point light detectors in a feeder tray or by using a set of mechanical probes that touch the feeder at a finite number of predetermined points.

Intuitively, each sensing point can be regarded as a binary bit that has two values ‘contained’ and ‘not contained’. So the robot senses a shape by reading out the binary representation of the shape, that is, by checking which points are contained in the shape and which are not. Thus the formalized sensing problem: Given \( n \) polygons with a total of \( m \) edges in the plane, locate the fewest points such that each polygon contains a distinct subset of points in its interior. We show that this problem is equivalent to an NP-complete set-theoretic problem introduced as Discriminating Set, and present an \( O(n^2m^2) \) approximation algorithm to solve it with a ratio of \( 2 \ln n \). Furthermore, we prove that one can use an algorithm for Discriminating Set with ratio \( c \log n \) to construct an algorithm for Set Covering with ratio \( c \log n + O(\log \log n) \). Thus the ratio \( 2 \ln n \) is asymptotically optimal unless \( \text{NP} \subset \text{DTIME}(n^{\text{poly log n}}) \), a consequence of known results on approximating Set Covering. The complexity of subproblems of Discriminating Set is also analyzed, based on their
1.2. **THESIS OVERVIEW**

relationship to a generalization of Independent Set called *3-Independent Set*.

---

1.2.3 **Pose and Motion from Contact**

So far we have addressed the use of the geometry of objects in determining their poses or distinguishing one object/pose from others. The proposed sensing methods assume the forms of geometric algorithms that compute poses from simple geometric constraints obtained by sensors.

Chapter 5 shows that task mechanics can also be made use of in sensing; in addition to an object’s pose, its motion can be properly observed by devising a feedback nonlinear observer. In this chapter, we choose one of the most fundamental manipulation tasks — pushing, and investigate the observation of a planar object with piece-wise smooth boundary being pushed by a finger.

The pushing is governed by a nonlinear system that relates through contact the object pose and motion to the finger motion. Nonlinear observability theory is employed to show that the contact information is often sufficient for the finger to determine not only the pose but also the motion of the object. Therefore a sensing strategy can be realized as an *observer* of the nonlinear dynamical system. Two observers are subsequently discussed. The first observer, based on the result of [55], has its “gain” determined by the solution of a Lyapunov-like equation. The second observer, based on Newton’s method, solves for the initial (motionless) object pose from a few intermediate contact points. Assuming the Coulomb friction model, the chapter copes with support friction in the plane and/or contact friction between the finger and the object.

---

1.2.4 **Observer Simulations and Sensor Implementation**

Chapter 6 presents simulation results on the two observers introduced in the previous chapter, and describes a simple contact sensor implementation using strain gauges. The first observer has been simulated with polygon objects and circular fingers, while the second observer has also been simulated with elliptic objects and fingers. Despite heavy load of computation due to evaluating the frictional effects on dynamics, the observers have performed with high success rates on various groups of data. The contact sensor consists of a horizontal disk, a cylindrical beam erected on the disk, and two pairs of strain gauges mounted on a cross section of the beam to measure the $x$- and $y$-components of any contact force on the disk boundary. When contact friction is small, the contact force is in a direction close to the contact normal, from which the contact location is easily obtained. Our measurements using the sensor (mounted on an Adept robot) have achieved accuracy on motionless contacts but not on contacts in motion due to the unpredictability of the contact force within the contact friction cone.
1.2.5 Local Observability of Rolling

Chapter 7 deals with local observability of a three-dimensional object rolling on a plane that is itself translating. The object is bounded by a smooth surface that makes point contact with the plane, which is free to accelerate in any direction. The plane is rough enough to allow only the rolling motion of the object. As the object rolls, the contact traces out a path in the plane, which can be detected by a tactile array sensor embedded in the plane.

Utilizing Montana’s equations for contact kinematics, we describe the kinematics and dynamics of this task by a nonlinear system, of which the output is the contact location in the plane. Then we establish a sufficient condition for the pose and motion of the object to be locally observable. This is done by decomposing the state space into the pose and angular velocity subspaces, and by later combining the sufficient conditions for the spanning of the cotangent spaces of these subspaces. The combined condition depends only on the contact geometry and on the object’s angular inertia matrix. And it is general enough to be satisfied by almost all except some degenerate shapes (such as a sphere).

1.3 Related Research

In the thesis we focus on sensing but also touch a variety of related issues in robotics such as the mechanics of basic manipulations, force and tactile sensing, parts orienting, model-based recognition, etc. The tools we apply range from computational geometry and complexity theory to differential geometry and nonlinear control.

1.3.1 Sensor Design and Geometric Sensing

Erdmann [43] proposed a method for designing sensors based on the particular manipulation task at hand. The resulting sensors satisfy a minimality property with respect to the given task goal and the available robot actions. Donald et al. [39] investigated the relationship among sensing, action, distributed resources, communication paths, and computation in the solution of robot tasks. This work provides a method for comparing disparate sensing strategies, and thus for developing minimal or redundant strategies, as desired.

Canny and Goldberg [24] introduced a reduced intricacy in sensing and control (RISC) paradigm that aims at improving the accuracy, speed and robustness of sensing by coupling simple and specialized hardware with fast and task-oriented geometric algorithms. The cross-beam sensing method developed by Wallack et al. [136] computes the orientation of a polygon from its diameters measured along three different directions, making use of a precomputed diameter function [58]. The idea of characterizing shapes with diameters and chords was also addressed earlier by Sinden [129].

For the special case that the poses are finite, an efficient method was presented by Rao and Goldberg [119]. It places a registration mark on the object so that the pose can be recognized by locating the mark position with a simple vision system. For robustness to sensor imperfections, the marked point maximizes the distance between the nearest pair
1.3. RELATED RESEARCH

among its possible locations. For the case that the number of parts is finite, the work by Govindan and Rao [60] can recognize a part with a modified parallel-jaw gripper by a sequence of grasp actions and diameter measurements. Some negative results about this projection-based sensing approach are revealed in Rao and Goldberg [120].

1.3.2 Parts Orienting

Orienting mechanical parts was early studied by Grossman and Blasgen [66]. They used a vibrating box to constrain a part to a small finite number of possible stable poses and then determined the particular pose by a sequence of probes using a tactile sensor. Inspired by their result, Erdmann and Mason [47] constructed a planner that employs sensorless tilting operations to orient planar objects randomly dropped into a tray, based on a simple model of the quasi-static mechanics of sliding. Natarajan [108] examined a similar strategy for detecting the orientations of polygonal and polyhedral objects with an analysis of the numbers of sensors sufficient and necessary for the task. Goldberg [57] showed that every polygonal part with unknown initial orientation can be oriented by a parallel-jaw gripper up to symmetry in the part’s convex hull. He constructed an algorithm with sub-cubic running time as a proof of sensorless parts orienting.

Based on Mason’s quasi-static analysis of pushing, Mani and Wilson [101] constructed edge stability maps and developed a planner using them to generate pushing sequences able to orient and position polygon parts. Brost [19] also constructed the same map, which he called the push stability diagram, but in a straightforward way. He used the diagram to plan parallel-jaw grasping of polygonal objects of non-uniform mass density and with bounded location uncertainties. Akella and Mason [3] introduced a complete open-loop (i.e., sensorless) planner that moves a polygon from a known pose to a specified final pose by pushing with a straight fence. Akella et al [2] demonstrated that a single joint robot can orient and feed polygon parts up to symmetry by pushing them on a constant speed conveyor belt.

To cope with the uncertainty in the initial pose of a workpiece and the unknown details about the contact between the workpiece and the plane, Peshkin and Sanderson [117] introduced a configuration map that describes the geometrical and physical consequence of one manipulation operation. Appropriate search techniques were then devised to find a sequence of elementary operations that are guaranteed to succeed despite uncertainty.

Donald et al. [38] studied the information structure of cooperative pushing tasks to re-orient large objects, demonstrating the equivalences between different types of sensing and communication by sensor reduction. Utilizing the theory of limit surfaces [61], Böhringer et al. [17] developed a geometric model for the mechanics of an array of microelectromechanical structures and showed how this structure can be used to uniquely align a part up to symmetry. They also offered efficient algorithms to support this new theory of sensorless, parallel manipulation.

Erdmann [45] developed a working system for orienting parts using two palms, together with frictional contact analysis tools that can predict relative slip between parts. The work demonstrates the feasibility of automatic nonprehensile palm manipulation. Zumel [141]
demonstrated that a part can be manipulated by a two degree-of-freedom palm and gave a planning algorithm based on an analysis of the mechanics of the nonprehensile contacts between the part and the palm. In view of the redundancy of fingers for a force-closure grasp in some robotic tasks, Abell and Erdmann [1] studied the problem of manipulating a polygonal object with the stable support of two frictionless contacts in a fixed gravitational field.

An early motivation of our work on mechanics-based observation came from blind grasping. The caging work by Rimon and Blake [121] is concerned with constructing the space of all configurations of a two-fingered hand controlled by one parameter that confine a given 2D object; these configurations can lead to immobilizing grasps by following continuous paths in the same space. This work requires an initial image of the object taken by a camera. Work related to caging includes Peshkin and Goldberg’s parts feeder design [115] and Brost and Goldberg’s fixture design [20]. In Chapter 5, we will be concerned with how to “feel” a known object using only one finger and how to infer its pose and motion information rather than how to constrain and grasp the object using multiple fingers.

### 1.3.3 Force and Tactile Sensing

Salisbury [127] first proposed the concept of fingertip force sensing with an approach for determining contact locations and orientations from force and moment measurements. This work was extended by Bicchi [16] who offered mathematical solutions to the problem of determining some global quantities of contact between two bodies from force and torque measurements by hard and soft ellipsoidal fingertips. Allen and Roberts [5] deployed robot fingers to obtain a number of contact points around an object and then fit (in a least squares manner) the data to a superquadric surface representation to reconstruct the object’s shape.

Solid mechanics was applied by Fearing and Hollerbach [50] to the modelling of the behavior of tactile sensors. Fearing and Binford [48] designed a cylindrical tactile sensor to determine the principal curvatures of an object from a small window of strain measurements. Based on continuum mechanics and photoelastic stress analysis, Cameron et al. [23] built a tactile sensor using a layer of photoelastic material along with its mathematical model.

Howe and Cutkosky [71] introduced dynamic tactile sensing in which sensors capture fine surface features during motion, presenting mechanical analysis and experimental performance measurements for one type of dynamic tactile sensor—the stress rate sensor. In their review of robotic touch sensing [70], Howe and Cutkosky argued that shape and force are the most important quantities measured with touch sensors. The uses of touch information in object recognition and manipulation were classified, and various sensing devices were discussed and compared.

### 1.3.4 Model-based Recognition

Model-based recognition and localization can often be regarded as a *constraint satisfaction* problem that searches for a consistent matching between sensory data (e.g., 2D) and model(s) (e.g., 3D) based on the geometric constraints between them (Grimson [64]).
1.3. RELATED RESEARCH

Horn and Ikeuchi [69] built a vision-based system that can determine the pose of a part in a randomly arranged pile by comparing a few images of the pile with a mathematical model of the part. Grimson and Lozano-Pérez [65] used tactile measurements of positions and surface normals on a 3D object to identify and locate it from a set of known 3D objects, based on the geometric constraints imposed by these tactile data. Gaston and Lozano-Pérez [54] showed how to identify and locate a polyhedron on a known plane using local information from tactile sensors that includes the position of contact points and the ranges of surface normals at these points. Motivated by an interpretation tree developed in Gaston and Lozano-Pérez [54], Siegel [128] worked on the determination the pose of an object grasped by a hand, under a situation very close to inscription.

Also using an interpretation tree search, Kemmotsu and Kanade [83] solved the pose of a polyhedron by matching a set of 3D line segments, obtained by three light-stripe range finders, to model faces; then the pose uncertainty was estimated using the covariance matrix of the endpoints of these line segments. Chen [27] used a polynomial approach to solve for the line-to-plane correspondences involved in pose determination. Based on a generalized Hough transform, Linnainmaa et al. [92] estimated the pose of a 3D object by matching point triples on the object to possibly corresponding triples in the image.

Hutchinson and Kak [73] introduced a dynamic sensor planning approach that selects sensing strategies based on current hypotheses about the identity and pose of an object (from a finite set of known objects). They focused on the problem of viewpoint and sensor-type selection and showed that sensor planning can discriminate between objects and their poses that would otherwise be indistinguishable from a fixed viewpoint by any vision-based system.

In the meantime, a variety of polygon shape descriptors (arkin et al. [7]; Mumford [107]) were analyzed theoretically and/or demonstrated experimentally to be efficient and robust to uncertainties.

Contact positions in our observer-based methods in Chapter 5 are analogous to “features” extracted from object images in object recognition. Gremban and Ikeuchi [63] considered planning sensor motion to obtain multiple observations and combined them with the known three-dimensional structure of the object to resolve recognition ambiguities.

1.3.5 Computational Geometry and Complexity Theory

Geometric algorithms for sensing unknown poses as well as unknown shapes have also been studied. Cole and Yap [33] considered “finger” probing a convex \(n\)-gon (\(n\) unknown) along directed lines and gave a procedure guaranteed to determine the \(n\)-gon with \(3n\) probes. This work was later extended by Dobkin et al. [37], who investigated the complexity of various models for probing convex polytopes in \(d\)-dimensional Euclidean space. Using a more powerful probe model that returns not only the contact points but also the normals at these points, Boissonnat and Yvinec [18] showed how to compute the exact shape of a simple polygon as long as some mild conditions about the shape are met. Li [89] gave algorithms of projection probing and line probing (similar to our inscription method in Chapter 2) that perform sufficient and necessary numbers of probes to determine a convex \(n\)-gon. Close
upper and lower bounds were derived in Lindenbaum and Bruckstein [91] on the number of composite probes to reconstruct a convex \( n \)-gon, where a composite probe comprises in parallel several supporting line probes or finger probes.

Belleville and Shermer [14] showed that the problem of deciding whether \( k \) line probes are sufficient to distinguish a convex polygon from a collection of \( n \) convex polygons is NP-complete. This result is very similar to our Theorem 6. A variation of the line-probing result due to Belleville and Shermer [14] would give us the point sampling result of Theorem 6. Arkin et al. [8] prove a similar result as well, namely, that the problem of constructing a decision tree of minimum height to distinguish among \( n \) polygons using point probes is NP-complete. This result holds even if all the polygons are convex.

Closely related work includes the research by Romanik and others on geometric testability (Romanik [122]; Romanik and Salzberg [123]; Arkin et al. [6]). In their research strategies were developed for verifying a given polygon using a series of point probes. Moreover, their research examined the testability of more general geometric objects, such as polyhedra, and develops conditions that determine whether a class of objects is (approximately) testable.

Paulos and Canny [114] studied the problem of finding optimal point probes for refining the pose of a polygonal part with known geometry from an approximate pose; they revealed that this problem is dual to the grasping problem of computing optimal finger placements and gave an efficient near-optimal solution.

Our work in Chapter 3 is closely related to finger probing. A number of researchers have looked at the problem of determining or distinguishing objects using finger probes. For a more extensive survey of probing problems and solutions, see Skiena [130].

The polygon containment problem, that is, deciding whether an \( n \)-gon \( P \) can fit into an \( m \)-gon \( Q \) under translations and/or rotations, has been studied by various researchers in computational geometry (Chazelle [25]; Baker et al. [11]; Fortune [51]; Avnaim and Boissonnat [10]). In the case where \( Q \) is convex, the best known algorithm runs in time \( O(m^2n) \) when both translations and rotations are allowed (Chazelle [25]). In Chapter 2 we deal with a special case of containment in which each edge of \( Q \) must touch \( P \); this constraint causes a reduction of the running time to \( O(mn) \), or \( O(n + m \log n) \) in practical situations.

### 1.3.6 Nonlinear Observability Theory

Hermann and Kerner [68] first studied observability using the observation space. A result due to Crouch [35] shows that an analytic system is observable if and only if the observation space distinguishes points in the state space.

Luenberger-like asymptotic observers, first constructed by Luenberger [95] for linear systems, remain likely the most commonly used observer forms for nonlinear systems today. Besides linearizing nonlinear systems, one approach of nonlinear observer design, called linearization by output injection, was independently proposed by Krener and Isidori [87] for autonomous systems and by Bestle and Zeitz [15] for time-varying systems. This approach transforms the original nonlinear system into a linear system modulo a nonlinear output injection.
1.3. RELATED RESEARCH

Necessary and sufficient conditions for the existence of such transformation for autonomous nonlinear systems without input were given in [87], and in [88] by Krener and Respondek along with a constructive algorithm. Conditions and constructive algorithms were also presented for systems with/without inputs by Xia and Gao [137]. These constructive algorithms find proper changes of coordinates from explicit solutions to certain partial differential equations involving repeated Lie brackets; hence their applicability becomes quite limited.

Gauthier, Hammouri, and Othman [55] described an observer for affine-control nonlinear systems whose "gain" is determined via the solution of an appropriate Lyapunov-like equation. Their observer is very simple: it is a copy of the original system, together with a linear corrective term that depends only on the state space dimension. Ciccarella et al. [30] proposed a similar observer whose gain vector is controlled by the properly chosen eigenvalues of a certain matrix obtained from the original system's Brunowsky canonical form, thus providing more freedom on optimizing the observer behavior. Extending the results of GHO [55], Gauthier and Kupka [56] characterized non-affine control systems that are observable under any input and construct a generic exponential observer for these systems. An approach to observer design utilizing the Lyapunov theory was introduced early on by Thau [133] but requires a good guess regarding a Lyapunov function.

Zimmer [140] designed a state estimator that conducts on-line minimization over some objective function. This observer, with provable convergence, iteratively uses Newton's method to modify its state estimate every fixed period of time. However, it intrinsically involves evaluating second order partial derivatives of the drift field of a system, thereby limiting its applicability to mostly simple nonlinear systems.

1.3.7 Contact Kinematics

Montana [106] derived a set of differential equations that govern the motion of a contact point in response to a relative motion of the objects in contact, and applied these equations to local curvature estimation and contour following. The kinematics of spatial motion with point contact was also studied by Cai and Roth [22] who assumed a tactile sensor capable of measuring the relative motion at the contact point. The special kinematics of two rigid bodies rolling on each other was treated by Li and Canny [90] in view of path planning in the contact configuration space. In the second part of this thesis, contact kinematics will be derived in terms of the absolute velocities of the objects in contact rather than their relative velocity at the contact.

Assuming rolling contact only, Kerr and Roth [84] combined the forward kinematics of multifingered hands and the contact kinematics into a system of nonlinear, time-varying ordinary differential equations which can be solved for the finger joint velocities that are necessary for achieving a desired object motion.
1.3.8 Rigid Body Dynamics

The first general discussion on the motion of a rigid body was due to Poisson in 1838, though the special case of a sphere was treated by Coriolis earlier in 1835. A few years later, friction was introduced into rigid body dynamics by Cournot in volumes 5 and 8 of *Crelle’s Journal*. In 1861, Slesser gave the equations of a rigid body constrained to roll and pivot without sliding on a horizontal plane. This method was followed by Routh who discussed the rolling of a sphere on any surface.

In his comprehensive introduction to rigid body dynamics [100], MacMillan coped with sliding in the case of linear pressure distributions over the planar base of contact, and rolling in the case of a sphere on a smooth surface.

Dynamics of sliding rigid bodies was also studied by Goyal et al. [61] using geometric methods based on the limit surface description of friction. Howe and Cutkosky [72] experimentally showed that the limit surface only approximates the force-motion relationship for sliding bodies and discussed other simplified practical models for sliding manipulation.

Observing the duality between a force and a point, Brost and Mason [21] described a graphical method for the analysis of rigid body motion with multiple frictional contacts in the plane. To geometrically represent friction between multiple rigid planar bodies in contact, including one movable, Erdmann [44] introduced configuration space friction cones to identify the set of contact forces consistent with each enumerated contact mode (and thus determine the possible motions).

The dynamic problem of predicting the accelerations and contact forces of multiple rigid bodies in contact with Coulomb friction have a unique solution if the the system Jacobian matrix has full column rank and the coefficients of friction are small enough. This result was obtained by Pang and Trinkle [113] who introduced complementarity formulations of the problem.

In the second part of the thesis we assume that the solution to rigid body dynamics has both existence and uniqueness. It is known, however, that when Coulomb friction is considered, this assumption can be violated. Löstedt [93] derived the planar equation in terms of constraint forces and gave conditions on solution consistency. Both he and Erdmann [42] constructed the sliding rod problem for which no solution exists to satisfy the Newtonian mechanics and Coulomb’s friction law simultaneously. Mason and Wang [105] revisited the same problem and demonstrated the existence of a consistent solution using an impact analysis. Dupont [40] derived a necessary and sufficient condition on the coefficient of friction for the existence and uniqueness of the forward dynamics problem.

However, coefficients of friction large enough to violate this condition are often easily encountered in tasks involving rolling or sliding contact with the environment. Baraff [12] worked on the problem of computing the contact forces between $n$ rigid bodies in contact at a finite number of points so that any interpenetration between the bodies can be prevented. He showed that deciding the consistency of a given configuration with dynamic friction is NP-complete. Although the complexity of static friction was left open in this paper, three methods for

---

1 The forward dynamics problem asks to solve for the joint motion given the input torques and the initial conditions.
approximating static friction forces were presented.

1.3.9 Quasi-Static Pushing

Mason [103] explored the mechanics of pushing using quasi-static analysis, predicting the direction in which an object being pushed rotates and plotting out its instantaneous rotation center. Sliding operations were also first identified in [103] as fundamental to manipulation, and especially to grasping. For unknown centers of friction, Alexander and Maddocks [4] reduced the problem of determining the motion of a slider under some applied force to the case of a bipod, offering analytical solutions for simple sliders.

A geometric tool very useful in the analysis of pushing tasks is the limit surface model of friction introduced to planar sliding by Goyal et al. [61]. The limit surface of a sliding object bounds the set of frictional forces and torques on the object in all possible motions (including no motion). In quasi-static pushing, the frictional wrench stays on the limit surface where the normal gives the corresponding velocity direction.

Approximating the limit surface of an object as an ellipsoid, Lynch et al. [97] designed a control system to translate the object by pushing with tactile feedback; this control system may be used for active sensing of the object’s center of mass. Lynch and Mason [98] established the possibilities of the motion of a slider even when its pusher is moving away and of slip between the two objects even with an infinite coefficient of friction. Analyses for both quasi-static and dynamic pushing are given using velocity and acceleration cones constructed over the limit surface and its dynamic analog of the slider, respectively.

Quasi-static motions are completely characterized by their associated instantaneous rotation centers. In practice the coefficient of friction and the pressure distribution are often indeterminate. Given some (e.g., constant) friction distribution, the sliding problem of solving for the motion of a slider under some applied force becomes that of determining the set of instantaneous rotation centers for all pressure distributions. For the case of a known center of friction, Peshkin and Sanderson [116] solved the sliding problem by a reduction to the case of a tripod of support, with two or three points on the boundary of the support region, and one or zero point at the rotation center. This reduction to tripods had been known earlier to Mason [102], though in a slightly different context. For the case of unknown center of friction, Alexander and Maddocks [4] proved the reduction of the problem to the case of a bipod, and offered analytical solutions for simple sliders. Lynch [99] showed how to find all feasible motions of a slider that has multiple point contacts with a pusher under an arbitrary motion.

The quasi-static analysis is applicable when the accelerations of the pusher and the slider are small and the acceleration time is short. It has been used as a basis for automatic planning of manipulation operations [102]. Mason [104] introduced a task-specific quasi-static number to quantify the comparative roles of inertial and frictional forces in pushing. To my best knowledge, no other quantitative study on the scope of quasi-static analysis has been conducted so far.
1.3.10 Dextrous Manipulation

Salisbury and Roth [125] studied the kinematic structures of articulated hands and the application of gripping forces. They identified an acceptable hand design as one that could immobilize a grasped object while also having the ability to impart arbitrary grasping forces and displacements to the object. The identification and classification of mechanisms constructible for whole arm manipulation were addressed by Salisbury [126]. Cutkosky [36] reviewed existing analytical models of grasping and compared them with human grasp selection.

Reorientations and regrasping of polygonal objects were studied in a geometric way by Fearing [49] who observed that bounded slips are valuable for adapting the fingers and objects to a stable situation. Regraspers can also be simplified into a sequence of pick-and-place operations with respect to a table. In this case the dynamic motions are excluded while only the geometric constraints between the object and the hand are considered. Tournassoud et al. [134] used backward chaining to search for a sequence of stable grasps and placements that enable a parallel-jaw gripper to achieve a desired grasp on a polyhedron.

Cole et al. [31] extended Cai and Roth’s work [22] by also incorporating the dynamics of multifingered hands and objects, as well as the test of grasp stability under Coulomb friction. They developed a control law that allows a grasped object to track a given desired trajectory in the configuration space. Cole et al. [32] presented a dynamic control law that achieves simultaneous tracking of a preplanned trajectory of a planar object together with the desired sliding motion at the fingertips. Rus [124] designed a hybrid control algorithm for finger tracking (with no friction) on a polyhedron to generate desired rotational motions.

Applying limit surfaces on soft-finger friction model and grasp stiffness matrices, Kao and Cutkosky [80] obtained differential equations of the quasi-static sliding motions of multiple fingers on a grasped object, and used these equations to control the fingers so that the object would follow a given trajectory. They [81] also conducted experiments on manipulating a card in the plane with two soft fingertips pressing down in the normal direction. These results show that quasi-static sliding analyses can be used for motion planning with soft sliding fingertips, provided that the average speed is low and the average coefficient of friction and normal force are accurately known.

Trinkle and Paul [135] exploited the problem of achieving a force-closure grasp of a two-dimensional object by frictionless sliding at the contacts between the object and an articulated hand. The kinematics and dynamics of the manipulation of a planar object sliding on a multifingered jig was treated by Yashima et al. [139]. Omata and Nagata [111] gave a search algorithm to plan for a sequence of finger repositionings that reorient a grasped object.
Part I

Geometric Sensing
Chapter 2

Polygon Inscription

In this chapter we will study the problem of detecting the pose of a convex polygon in the plane by taking views of the polygon from multiple exterior sites. The shape of the polygon is assumed to be known in advance, but the pose of the polygon can be arbitrary. Each view results in a cone formed by the two outermost occluding rays starting from the viewing site. The cone in turn imposes a constraint on the possible poses of the polygon: the polygon must be contained in the cone and make contact with both its sides. A containment in which every edge of the containing object touches the contained object is called an inscription, so we will say that the polygon is inscribed in the cone. Such constraints imposed by individual views together allow only a small number of possible poses of the polygon, which often reduces to one. For example, Figure 2.1 illustrates two views taken of a convex 6-gon \( P \) in some unknown pose from sites \( o_1 \) and \( o_2 \), respectively: The two cones \( C_1 \) and \( C_2 \) thus formed determine the real pose of \( P \), and this pose can be solved using the algorithm presented later in Section 2.1.4.

The above sensing approach appears to be simple, but to make it efficient and to minimize the cost of sensing hardware, we would like to take as few views as possible. This leads us to the main question of this chapter: How many views are sufficient in the general case in order to determine the pose of a convex \( n \)-gon?\(^1\)

The answer to the above question is “two”, and to argue this answer we will go through several steps, each of which occupies a separate section: Section 2.1 describes how to compute the set of possible poses for a convex polygon inscribed in multiple cones and derives an upper bound on the number of possible poses for two-cone inscription in particular; Section 2.2 empirically demonstrates that in spite of the upper bound, two cones turn out to be sufficient in most cases to uniquely determine the pose of an inscribed polygon. Section 2.3 further discusses the extensions of this method and proposes a general sensing scheme: sensing by inscription.

\(^1\)It should be noted that there exist cases in which the pose of a convex \( n \)-gon cannot be uniquely determined, no matter how many views are taken. This happens only if the polygon preserves self-congruence over certain rotations. (It is not hard to see that in such a case the rotation angle must be a multiple of \( \frac{2\pi}{k} \), where \( k \) is a positive integer that divides \( n \).) However, all congruent poses are usually considered as the same in the real applications.
2.1 Multi-Cone Inscription

To simplify the presentation, let us agree throughout Section 2.1 that all angles take values in the half-open interval $[0, 2\pi)$. In accordance with this agreement, any expression $E$ on angles equated with or assigned to an angle $\theta$ in formulas such as $\theta = E$ or $\theta \leftarrow E$ will be regarded as “$E \mod 2\pi$”, even though we do not mention so explicitly. (However, this does not apply to conditions such as $E_1 = E_2$ where both sides are expressions on angles.) Moreover, intervals for angle values whose left end points are greater than right end points are allowed; for example, an interval $[\psi_1, \psi_2]$, where $0 \leq \psi_2 < \psi_1 < 2\pi$, is understood as the interval union $[\psi_1, 2\pi) \cup [0, \psi_2]$.

2.1.1 A Triangle Sliding in a Cone

We first deal with the case of a triangle in a cone, not only because it is the simplest, but also because the case of a polygon, as we will see later, can be decomposed into subcases of triangles. Let $\triangle p_0p_1p_2$ be a triangle inscribed in an upright cone $C$ with angle $\phi$ and vertex $o$, where $0 < \phi < \pi$, making contacts with both sides of the cone at vertices $p_1$ and $p_2$ respectively. What is the locus of vertex $p_0$ as edge $p_1p_2$ slides against the two sides of the cone?

Two different situations can occur with this inscription: (1) $p_0$ is outside $\triangle p_1op_2$, and (2) $p_0$ is inside (only when $\angle p_1p_0p_2 \geq \phi$) (Figure 2.2). Assume that $\triangle p_0p_1p_2$ may degenerate into any one of its edges but not a point; writing $\psi_1 = \angle p_0p_1p_2$, $\psi_2 = \angle p_0p_2p_1$, $r = \|p_0p_2\|$ and $s = \|p_1p_2\|$, this degeneracy is taken into account by the following constraints:

$$0 \leq \psi_1, \psi_2 < \pi, \quad s \geq 0, \quad \text{and} \quad \begin{cases} \quad d, & \text{if } s = 0 \text{ or } \psi_2 > 0; \\ 0 \leq r \leq s, & \text{otherwise.} \end{cases}$$

Let us set up a coordinate system with the origin at $o$ and the $y$ axis bisecting angle $\phi$, as shown in Figure 2.2. Then the orientation of $\triangle p_0p_1p_2$ can be denoted by the angle $\gamma$.
Figure 2.2: A triangle sliding in a cone. Vertices $p_1$ and $p_2$ move along the two sides of cone $C$. The locus of $p_0$ is an elliptic curve (possibly degenerating into a line segment) parameterized by the angle $\gamma$ between directed edge $\overrightarrow{p_2p_1}$ and the $x$ axis. There are two different cases: (a) $p_0$ is above edge $p_1p_2$; (b) $p_0$ is below edge $p_1p_2$.

between vector $\overrightarrow{p_2p_1}$ and the $x$ axis. The range of valid $\gamma$ values can be easily determined as

$$\max\left(\frac{\pi}{2} + \phi, \frac{\pi}{2} - \phi + \psi_2\right) \leq \gamma \leq \min\left(\frac{3\pi}{2} - \phi, \frac{3\pi}{2} + \phi - \psi_1\right)$$

in case (1), and as $\gamma = \gamma' \mod 2\pi$ in case (2), where

$$\frac{3\pi}{2} + \phi + \psi_2 \leq \gamma' \leq 2\pi + \frac{\pi}{2} - \phi - \psi_1.$$

For any valid $\gamma$, there exists a unique pose of the triangle in cone $C$; this allows us to parameterize the locus $(x, y)$ of $p_0$ by $\gamma$.

Cases (1) and (2) yield similar results differing only by a “+” or “−” sign, so we will treat them together. We begin with writing out the following equations for the locus of $p_0$:

$$x = r \cos(\gamma - \psi_2) \pm \|op_2\| \sin \frac{\phi}{2},$$

$$y = r \sin(\gamma - \psi_2) + \|op_2\| \cos \frac{\phi}{2},$$

where $\|op_2\| = \pm \frac{\cos(\gamma \pm \frac{\phi}{2})}{\sin \frac{\phi}{2}}$. Here the notation “±” means “+” in case (1) and “−” in case (2) and the notation “±” means just the opposite. Several steps of manipulation on the above equations plus a detailed subsequent analysis on the ranges of angles reveal the locus of $p_0$ as described below.
Namely, as edge \( p_1 p_2 \) slides in the cone, \( p_0 \) traces out an elliptic curve \( \alpha \) with implicit equation

\[
ax^2 \pm bxy + cy^2 = d,
\]

where

\[
\begin{align*}
a &= r^2 - rs \frac{\sin(\frac{\phi}{2} \pm \psi_2)}{\sin \frac{\phi}{2}} + \frac{s^2}{2 - 2 \cos \phi}; \\
b &= \frac{(2r \cos \psi_2 - s)s}{\sin \phi}; \\
c &= r^2 - rs \frac{\cos(\frac{\phi}{2} \pm \psi_2)}{\cos \frac{\phi}{2}} + \frac{s^2}{2 + 2 \cos \phi}; \\
d &= \left(r(s \sin(\frac{\phi}{2} \pm \psi_2) - \sin \phi)\right)^2.
\end{align*}
\]

Furthermore, if the orientation \( \gamma \) changes monotonically within its range, \( p_0 \) moves monotonically along \( \alpha \) except when \( \alpha \) degenerates into a line segment. In that degenerate case, \( p_0 \) moves along a segment of a line through the cone vertex \( o \) and with equation

\[
\left\{ \begin{array}{ll}
\cos \frac{\phi}{2} x \mp \sin \frac{\phi}{2} y = 0, & \text{if } r = 0 \text{ (and thus } \psi_2 = 0) ; \\
\cos(\frac{\phi}{2} \pm \psi_2) x \pm \sin(\frac{\phi}{2} \pm \psi_2) y = 0, & \text{if } r \neq 0, s \neq 0, \text{ and } \frac{r}{s} = \frac{\sin(\phi \pm \psi_2)}{\sin \phi},
\end{array} \right.
\]

crossing the same point at any two valid orientations \( \gamma_1 \neq \gamma_2 \) with \( \gamma_1 + \gamma_2 = 2\pi \pm \phi \) when the line segment assumes the first equation, or with \( \gamma_1 + \gamma_2 = 2\pi - \phi + 2\psi_2 \) in case (1) and with \( \gamma_1 + \gamma_2 = 2\psi_2 + \phi \) in case (2) when it assumes the second equation.

Interestingly, we observe that in case (1) if \( r \neq 0 \) and \( \frac{r}{s} = \frac{\sin(\phi \pm \psi_2)}{\sin \phi} \), the second line equation reduces to \( x = 0 \) when \( \phi = 2\psi_2 \) in case (1); but this reduction does not occur in case (2) because \( \frac{r}{s} = \frac{\sin(\phi + \psi_2)}{\sin \phi} \) implies that \( 0 < \phi + 2\psi_2 < 2\pi \), thereby establishing \( \sin(\frac{\phi}{2} + \psi_2) > 0 \). This observation reflects a small asymmetry between the two cases.

### 2.1.2 One-Cone Inscription

Now consider the case that a convex \( n \)-gon \( P \) with vertices \( p_0, p_1, \ldots, p_{n-1} \) in counterclockwise order is inscribed in a cone \( C \). Let us choose the same coordinate system as in Section 2.1.1. Then the pose of \( P \) is uniquely denoted by the locus of some vertex, say \( p_0 \), and the angle \( \theta \) between the \( x \) axis and some directed edge, say \( \overrightarrow{p_0 p_1} \). Clearly, any orientation \( \theta \) gives rise to a unique pose of \( P \); so we can compute the locus of \( p_0 \) as a function of \( \theta \) over \([0, 2\pi)\). Let \( p_l \) and \( p_r \) be the two vertices currently incident on the left and right sides of cone \( C \), respectively (Figure 2.3). As long as \( p_l \) and \( p_r \) remain incident on these two sides, respectively, the problem reduces to the case of \( \triangle p_0 p_l p_r \) sliding in cone \( C \) except that the locus of \( p_0 \) (an elliptic curve) now needs to be parameterized by \( \theta \), instead of \( \gamma \), which
2.1. MULTI-CONE INSCRIPTION

Figure 2.3: A convex polygon $P$ rotating in a cone. The pose of $P$ is denoted by the position of vertex $p_0$ and the angle $\theta$ between directed edge $p_0p_1$ and the $x$ axis. The space of orientations $[0, 2\pi)$ is partitioned into closed intervals, each defining an elliptic curve that describes the corresponding locus of $p_0$.

we had used before. This is easy, for we observe that when $p_l \neq p_r \neq p_0$,

$$\gamma = \begin{cases} \theta + \angle p_0 p_r + \angle p_0 p_l p_i - \pi, & \text{if } p_0 \text{ is above edge } p_l p_r; \\ \theta + \angle p_0 p_l + \angle p_0 p_r p_i - \pi, & \text{otherwise}. \end{cases}$$

The three special cases $p_l = p_0$, $p_r = p_0$ and $p_l = p_r$ can be handled by substituting $\pi$ for $\angle p_1 p_0 p_r$ and $\angle p_1 p_0 p_l$ in the first case, $\pi$ for $\angle p_1 p_0 p_l$ and $\angle p_1 p_0 p_r$ for $\angle p_0 p_l p_r$ in the second case, and $0$ for $\angle p_0 p_l p_i$ (or $\angle p_0 p_r p_i$) in the third case.

More observations show that the entire range $[0, 2\pi)$ of orientations can be partitioned into a sequence of closed intervals, within each of which the vertices $p_l$ and $p_r$ incident on cone $C$ are invariant.

We present a linear-time algorithm that computes the above orientation intervals as well as the corresponding elliptic curves describing the locus of $p_0$. The algorithm rotates the polygon counterclockwise in the cone, generating a new interval whenever one (or both) of the incident vertices $p_l$ and $p_r$ changes; the new incident vertex (or vertices) is determined by a comparison between angle $\phi$ and the angle intersected by the two rays $p_{i-1}p_i$ and $p_{i}p_{i+1}$.

Let $[\theta_{\min}, \theta_{\max}]$ denote the current interval, and let $\varphi_i$ denote the interior angle $\angle p_{i-1}p_i p_{i+1}$ for $0 \leq i \leq n - 1$. (For convenience, arithmetic operations performed on the subscripts of vertices or internal angles are regarded as followed by a “$\mod n$” operation; for example, $p_{-1}$ is identified with $p_{n-1}$ and $p_n$ with $p_0$.) In the algorithm, $\Phi_{\text{left}}$ and $\Phi_{\text{right}}$ keep track of the angle between $\overrightarrow{p_{i-1}p_{i+1}}$ and $\overrightarrow{p_0p_i}$ and the angle between $\overrightarrow{p_{i}p_{i+1}}$ and $\overrightarrow{p_0p_{i+1}}$, respectively. The algorithm proceeds as follows:

Algorithm 1
**Step 2** The current orientation interval has its left end point at \( \theta_{\text{min}} \). Output the elliptic curve (now parameterized by \( \theta \)) resulting from sliding edge \( p_i p_r \) in cone \( C \).

Next determine the right end point \( \theta_{\text{max}} \) of the current interval. Let \( \psi \) be the angle intersected by rays \( p_{l-1} p_i \) and \( p_r p_{r-1} \); set \( \psi \leftarrow \text{nil} \) if they do not intersect. There are three different cases:

**Case 1** \( \psi < \phi \) or \( \psi = \text{nil} \). [Advance \( p_r \) clockwise to the next vertex.] Set \( \Phi_{\text{right}} \leftarrow \Phi_{\text{left}} + \pi - \varphi_r, \theta_{\text{max}} \leftarrow \Phi_{\text{right}} + \pi - \varphi_r, \) and \( r \leftarrow r - 1 \).

**Case 2** \( \psi > \phi \). [Advance \( p_l \).] Set \( \Phi_{\text{left}} \leftarrow \Phi_{\text{left}} + \pi - \varphi_l, \theta_{\text{max}} \leftarrow \Phi_{\text{left}} + \frac{3\pi}{2} + \frac{\varphi_l}{2}, \) and \( l \leftarrow l - 1 \).

**Case 3** \( \psi = \phi \). [Advance both \( p_l \) and \( p_r \).] Set \( \Phi_{\text{left}} \leftarrow \Phi_{\text{left}} + \pi - \varphi_l, \Phi_{\text{right}} \leftarrow \Phi_{\text{right}} + \pi - \varphi_r, \theta_{\text{max}} \leftarrow \Phi_{\text{left}} + \frac{3\pi}{2} + \frac{\varphi_l}{2}, l \leftarrow l - 1, \) and \( r \leftarrow r - 1 \).

Output the current interval \([\theta_{\text{min}}, \theta_{\text{max}}]\). Set \( \theta_{\text{min}} \leftarrow \theta_{\text{max}} \) and repeat Step 2 until \( r \) changes from 1 to 0.

The number of intervals produced by the above algorithm cannot exceed \( 2n \), because each loop of Step 2 advances either \( p_r \) to \( p_{r-1} \), or \( p_l \) to \( p_{l-1} \), or both to \( p_{r-1} \) and \( p_{l-1} \), respectively, and because there are \( 2n \) vertices in total (\( n \) each for \( p_l \) and \( p_r \)) to advance before returning to the initial incident vertices \( p_0 \) and \( p_1 \).

We can easily apply the above algorithm for the general case in which the vertex of cone \( C \) is at an arbitrary point \((x_0, y_0)\) and the axis of the cone forms an angle \( \theta_0 \) with the \( y \) axis. Each generated interval \([\psi_1, \psi_2]\) now needs to be right-shifted to \([\psi_1 + \theta_0, \psi_2 + \theta_0]\). Let \( q = p_r \) and \( q' = p_l \) if \( p_0 \) is above edge \( p_l p_r \), and let \( q = p_l \) and \( q' = p_r \) otherwise. Then the corresponding locus of \((x, y)\) of \( p_0 \) is determined as, assuming \( p_l \neq p_r \neq p_0 \),

\[
x = \left(-\|qp_0\| \cdot \cos \angle p_1 p_0 q + \|p_l p_r\| \cdot k_c \cdot \frac{\sin \left( \frac{\phi}{2} + \theta_0 \right)}{\sin \phi} \right) \cos \theta \\
+ \left(\|qp_0\| \cdot \sin \angle p_1 p_0 q - \|p_l p_r\| \cdot k_s \cdot \frac{\sin \left( \frac{\phi}{2} + \theta_0 \right)}{\sin \phi} \right) \sin \theta + x_0;
\]

\[
y = \left(-\|qp_0\| \cdot \sin \angle p_1 p_0 q \pm \|p_l p_r\| \cdot k_c \cdot \frac{\cos \left( \frac{\phi}{2} + \theta_0 \right)}{\sin \phi} \right) \cos \theta \\
- \left(\|qp_0\| \cdot \cos \angle p_1 p_0 q \pm \|p_l p_r\| \cdot k_s \cdot \frac{\cos \left( \frac{\phi}{2} + \theta_0 \right)}{\sin \phi} \right) \sin \theta + y_0,
\]

where \( k_c = \cos (\angle p_1 p_0 q + \angle p_0 q q' - \theta_0 \mp \frac{\phi}{2}) \) and \( k_s = \sin (\angle p_1 p_0 q + \angle p_0 q q' - \theta_0 \mp \frac{\phi}{2}) \). Here \( \pm \) and \( \mp \) denote \( + \) and \( - \), or \( - \) and \( + \), according as \( p_0 \) is above or below edge \( p_l p_r \).
2.1. MULTI-CONE INSCRIPTION

2.1.3 Upper Bounds

The preceding subsection tells us that the set of possible poses for a convex polygon inscribed in one cone can be described by a continuous and piecewise elliptic curve defined over orientation space $[0, 2\pi)$. We call this curve the locus curve for the inscription. This subsection will show that only finite possible poses exist for a convex $n$-gon $P$ inscribed in two cones, so long as the vertices of the cones do not coincide. An upper bound on the number of possible poses can be obtained straightforwardly by intersecting two locus curves, each resulting from the inscription of $P$ in one cone.

Claim 1 There exist no more than $8n$ possible poses for a convex $n$-gon $P$ inscribed in two cones $C_1$ and $C_2$ with distinct vertices.

Proof Let $p_0, \ldots, p_{n-1}$ be the vertices of $P$ in counterclockwise order; then a pose of $P$ can be represented by the location of $p_0$ as well as the angle $\theta$ between directed edge $\overrightarrow{p_0p_1}$ and the $x$ axis. Let $\alpha_1(\theta)$ and $\alpha_2(\theta)$ be the two locus curves for the inscriptions of $P$ in cones $C_1$ and $C_2$, respectively. We need only to show that $\alpha_1$ and $\alpha_2$ meet at most $8n$ times, that is, they pass through common points at no more than $8n$ values of $\theta$.

It is known that each $\alpha_i$ consists of at most $2n$ elliptic curves defined over a sequence of intervals that partition $[0, 2\pi)$. Intersecting these two sequences of intervals gives a partition that consists of at most $4n$ intervals. Within each interval the possible orientations (hence the possible poses) of $P$ can be found by computing where the corresponding pair of elliptic curves meet. According to the last subsection, this pair of curves may be written in the parameterized forms

$$(a_{ix} \cos \theta + b_{ix} \sin \theta + x_i, a_{iy} \cos \theta + b_{iy} \sin \theta + y_i),$$

for $i = 1, 2$. Here $(x_i, y_i)$ is the vertex of cone $C_i$, and $a_{ix}, b_{ix}, a_{iy}, b_{iy}$ are constants determined by $P$ and $C_i$. Using the condition $(x_1, y_1) \neq (x_2, y_2)$, we suppose $x_1 \neq x_2$ without loss of generality, and let $\delta = \sqrt{(a_{ix} - a_{2x})^2 + (b_{ix} - b_{2x})^2}$. Then it is not hard to show that these two curves do not meet if $|x_1 - x_2| > \delta$. Otherwise they may meet only at

$$\theta = \psi_2 - \psi_1 \quad \text{and} \quad \theta = \pi - \psi_2 - \psi_1,$$

where $\psi_1 = \text{atan}(\frac{a_{ix} - a_{2x}}{\delta}, \frac{b_{ix} - b_{2x}}{\delta})$ and $\psi_2 = \text{sin}^{-1}\frac{x_2 - x_1}{\delta}$. □

The upper bound $8n$ is not tight: A lower one can be obtained even without using two cones to constrain the polygon. Notice in the proof above that the bound came from a partition of orientation space $[0, 2\pi)$ into up to $4n$ intervals that combined the individual pose constraints imposed by the two cones. Therefore, an improvement on that bound must require a different partitioning of $[0, 2\pi)$. To see this, we regard each cone as the intersection of two half-planes and decompose its constraint on the polygon into two constraints introduced by the half-planes independently.

A polygon $P$ is said to be embedded in a half-plane if $P$ is contained in the half-plane and supported by its bounding line. Thus, $P$ is inscribed in a cone if and only if it is
embedded in the two half-planes defining the cone by intersection. Two cones with distinct vertices together provide three or four half-planes, of which at least three have nonconcurrent bounding lines (i.e., bounding lines that do not pass through a common point). Such three half-planes are indeed enough to bound the number of possible poses of $P$ within $6n$.

**Theorem 2** There exist no more than $6n$ possible poses for a convex $n$-gon $P$ embedded in three half-planes with nonconcurrent bounding lines; furthermore, this upper bound is tight.

**Proof** Let $L_1$, $L_2$, and $L_3$ be the bounding lines of the three half-planes respectively. We can assume that these lines are not all parallel; otherwise it is easy to see that no feasible pose for $P$ exists. So suppose $L_1$ and $L_2$ intersect; their corresponding half-planes form a cone in which $P$ is inscribed. Let the orientation of $P$ be represented by the angle $\theta$ between the $x$ axis and some directed edge of $P$. Then orientation space $[0, 2\pi)$ is partitioned into at most $2n$ intervals according to which pair of vertices are possibly on $L_1$ and $L_2$, respectively. In the mean time, it is also partitioned into exactly $n$ intervals according to which vertex is possibly on $L_3$. Intersecting the intervals in these two partitions yields a finer partition of $[0, 2\pi)$ that consists of at most $3n$ intervals, each containing orientations at which $P$ is to be supported by $L_1$, $L_2$, and $L_3$ at the same three vertices.

Let us look at one such interval, and let $p_i$, $p_j$, and $p_k$ be the vertices of $P$ on $L_1$, $L_2$, and $L_3$, respectively whenever a possible orientation exists in the interval. The possible orientations occur exactly where $L_3$ crosses an elliptic curve $\alpha(\theta)$ traced out by $p_k$ when sliding $p_i$ and $p_j$ on $L_1$ and $L_2$, respectively. Now we prove that this interval contains at most two possible orientations. Note $\alpha$ does not degenerate into a point, because the case $p_i = p_j = p_k$ will never happen, given $L_1$, $L_2$, $L_3$ are not concurrent. Therefore, $\alpha$ is either an elliptic segment monotonic in $\theta$ or a line segment that attains any point for at most two $\theta$ values (Section 2.1.1). In both cases, it is clear that $\alpha$ crosses $L_3$ for no more than two $\theta$ values. Thus there are at most $6n$ possible poses in orientation space $[0, 2\pi)$.

Appendix A gives an example in which a polygon can actually have $6n$ poses when embedded in three given half-planes, thereby proving the tightness of this upper bound.

When two of the three nonconcurrent bounding lines are parallel, we can similarly derive a less and tight upper bound $4n$, using the same interval partitioning technique. We omit the details of the derivation here.

Since any two cones with distinct vertices are formed by three or four half-planes with at least three nonconcurrent bounding lines, and since embedding a polygon in three half-planes with nonconcurrent bounding lines is equivalent to inscribing it in any two cones determined by intersecting a pair of the half-planes, we immediately have

**Corollary 3** There exist at most $6n$ possible poses for a convex $n$-gon inscribed in two cones with distinct vertices, and this upper bound is tight.

Would more half-planes (or cones) further reduce the number of possible poses for an embedded polygon to be asymptotically less than $n$? The answer is no. For example, an embedded regular $n$-gon will always have $kn$ possible poses, where $1 \leq k \leq 6$, no matter how many half-planes are present. The experimental results in Section 2.2 will show that two cones (or four half-planes) are usually sufficient to determine a unique pose for the polygon.
2.1.4 An Algorithm for Inscription

With the results in the previous subsections, we here present an algorithm that computes all possible poses for a convex \( n \)-gon \( P \) to be inscribed in \( m \) cones \( C_1, \ldots, C_m \), where \( m \geq 2 \). (The vertices of these cones are assumed to be distinct.) Let \( p_0, \ldots, p_{n-1} \) be the vertices of \( P \) in counterclockwise order.

Algorithm 2

**Step 1** [Compute an initial set of poses w.r.t. two cones.] Solve for all possible poses of \( P \) when inscribed in cones \( C_1 \) and \( C_2 \) (use Algorithm 1 and see the proof of Claim 1), and let set \( S \) consist of the resulting poses (already sorted by orientation). Set \( i \leftarrow 3 \).

**Step 2** [Verify with the remaining cones.] If \( i = m+1 \) or \( S = \emptyset \), then terminate. Otherwise go to Step 3 if \( |S| = 1 \) or \( |S| = 2 \). Otherwise apply Algorithm 1 to generate the locus curve \( \alpha_i(\theta) \) for the inscription of \( P \) in \( C_i \). Sequentially verify whether each pose in \( S \) is on \( \alpha_i(\theta) \), deleting from \( S \) those poses that are not. Set \( i \leftarrow i + 1 \) and repeat Step 2.

**Step 3** [More efficiently verify one or two poses.] For each pose in \( S \), let polygon \( P' \) be \( P \) in that pose and do the following: For \( i \leq j \leq m \), construct the supporting cone \( C'_j \) of \( P' \) at the vertex of cone \( C_j \); if there exists some \( C'_j \) that does not coincide with \( C_j \), then delete the corresponding pose from \( S \).

When the above algorithm terminates, set \( S \) will contain all possible poses for the inscription. Corollary 3 shows that there are at most \( 6n \) poses in \( S \) after Step 1. Since the supporting cone of \( P \) from a point can be constructed in time \( O(\log n) \) using binary search (Preparata and Shamos [118]), the running time of the algorithm is \( O((k-1)n + (m-k+1)\log n) \) (i.e., \( O(mn) \) in the worst case), where \( k \) is the value of variable \( i \) when leaving Step 2. However, the simulations in Section 2.2 will demonstrate that \( k = 3 \) almost always holds; hence Step 2 will almost never get executed more than once, reducing the running time to \( O(n + m \log n) \).

Since a convex \( m \)-gon \( Q \) is naturally the intersection of \( m \) cones, each with a vertex of \( Q \) as its vertex and with the internal angle at that vertex as its apex angle, we can use Algorithm 2 to compute all possible inscriptions of \( P \) in \( Q \) with the same time cost. This problem shall be called the *polygon inscription* problem, which can be regarded as another version of multi-cone inscription because the intersection of multiple cones is always a polygon (possibly unbounded or empty).

2.2 Simulations

The first set of experiments were conducted to find out how many possible poses usually exist for a polygon embedded in three half-planes with nonconcurrent bounding lines. The results are summarized in Table 2.1.

Seven groups of convex polygons were tested. The first six groups consisted of convex hulls generated over 10, 100, and 1000 random points successively and for each number in
two kinds of uniform distributions: inside a square and inside a circle, respectively. It can be seen in the table that the polygons in these groups had a wide range (3–43) of sizes (i.e., numbers of vertices), but their shapes were not arbitrary enough, approaching either a square or a circle when large numbers of random points were used. So, we introduced the last group of data, which consisted of polygons generated by a method called circular march, which outputs the vertices of a convex polygon as random points inside a circle in counterclockwise order. The size of a polygon in this group was randomly chosen between 3 and 15.

Given a convex polygon, three supporting lines, each bounding a half-plane on the side of the polygon, were generated according to the uniform distribution; namely, with probability \( \frac{\pi - \phi_i}{2\pi} \), each line passed through vertex \( p_i \) with internal angle \( \phi_i \). An additional check was performed to ensure that these lines were not attached to the same vertex of the polygon. The number of possible poses for the polygon to be embedded in these generated half-planes was then computed, and the summarized results for all group are listed in the last two columns of the table.

Table 2.1 tells us that three half-planes are insufficient to limit all possible poses of an embedded polygon to a unique one, namely, the real pose; in fact the table suggests that a linear (in the size of the polygon) number of possible poses will usually exist. We can see in the table that despite the appearances of cases with one or two possible poses, the ratio between the mean of numbers of possible poses and mean polygon size lies in the approximate range 0.43–0.93, decreasing very slowly as the mean polygon size in a group increases. These results tend to support a conjecture that in the average case there exist \( O(n) \) possible poses for a convex \( n \)-gon embedded in three half-planes with nonconcurrent bounding lines.

The above conjecture may be very difficult to prove. However, a plausible explanation for the experimental results can be sought. Recall, a polygon with three half-planes defines
2.2. SIMULATIONS

A partition of orientation space $[0, 2\pi)$ into at most $3n$ intervals, each containing orientations that would allow the same three incident vertices whenever a possible pose exists at that orientation. The feasible orientations in each interval occur when one supporting line crosses the locus curve of its associated incident vertex. The locus curve results from moving the other two incident vertices along their supporting lines. As these curves (for all intervals) may often cluster together, the likelihood that they get crossed $O(n)$ times in total by the first supporting line is quite large. This happened particularly often during the experiments when a vertex coincided with an intersection of two supporting lines (Figure 2.4).

The purpose of the second set of experiments was to study how many poses usually exist for a convex polygon inscribed into two or more cones with distinct vertices. We first tested with two cones using the same source of random data generated in the way we did for the first set of experiments, and the results are shown in Table 2.2. Since a polygon was always generated inside a square (circle), the cone vertices were chosen as random points uniformly distributed between this boundary and a larger square (circle). The ratio between these two squares (circles) was set uniformly to be $\frac{1}{2}$ for all seven groups of data.

In contrast to Table 2.1, Table 2.2 tells us that two cones allow a unique pose of an inscribed polygon in most cases. In each group of tests, only cases with one or two poses occurred, and the mean of possible poses stayed very close to 1, independent of the mean polygon size. (It is not hard for us to see that the percentage of two-pose cases was very low, in the range of 0% to 3.6% for the first six groups of data. The 10.6% for the seventh group was a bit high but expected, because polygons generated by circular march were more likely to be in a certain shape that would often incur two possible poses, as we will discuss later.)

Tests were also conducted with 3–10 cones on reproduced data of four of the seven groups, while the other experiment parameters were kept the same. As shown in Table 2.3, the means of possible poses did not decrease dramatically as compared to those in Table 2.2.

<table>
<thead>
<tr>
<th># tests</th>
<th>data source</th>
<th># poly vertices</th>
<th># possible poses</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>range</td>
<td>mean</td>
</tr>
<tr>
<td>10000</td>
<td>10 sq.</td>
<td>3–10</td>
<td>5.9493</td>
</tr>
<tr>
<td>10000</td>
<td>10 cir.</td>
<td>3–10</td>
<td>6.1113</td>
</tr>
<tr>
<td>1000</td>
<td>100 sq.</td>
<td>6–18</td>
<td>12.002</td>
</tr>
<tr>
<td>1000</td>
<td>100 cir.</td>
<td>9–21</td>
<td>15.138</td>
</tr>
<tr>
<td>1000</td>
<td>1000 sq.</td>
<td>10–26</td>
<td>18.073</td>
</tr>
<tr>
<td>1000</td>
<td>1000 cir.</td>
<td>26–45</td>
<td>33.665</td>
</tr>
<tr>
<td>1000</td>
<td>cir. mar.</td>
<td>3–15</td>
<td>9.009</td>
</tr>
</tbody>
</table>

Table 2.2: Experiments on inscribing polygons with two cones.
Figure 2.4: Eight possible poses for a convex 6-gon bounded by three supporting lines (as taken from a sample run). The first one represents the real pose whose supporting lines as shown were generated randomly; the remaining seven represent all other poses consistent with the supporting line constraints. Notice in this example that three of the eight poses occurred when a vertex of the polygon coincided with an intersection of two supporting lines.
The experimental result that two nonincident cones usually allow a unique pose of an inscribed convex $n$-gon $P$ has in fact a very intuitive explanation. As mentioned before, two such cones generally provide four half-planes, some three of which will limit the number of possible poses of $P$ to at most $6n$. Let polygons $P_1, \ldots, P_m$, where $m \leq 6n$, represent $P$ in all possible poses, respectively, when embedded in the three half-planes; then those $P_i$ corresponding to the final possible poses must be supported by the bounding line $L$ of the fourth half-plane. So the probability that a two-pose case occurs is no more than the probability that $L$ passes through a vertex of $P_i$ and a vertex of $P_j$, for $i \neq j$. Note the vertices of $P_1, \ldots, P_m$ together occupy $\Theta(mn)$ points in the plane in general, only one of which must lie on $L$. If no two of these vertices coincide, the probability that $L$ passes through two vertices of different polygons is zero (assuming that $L$ is independent of the other three bounding lines), which means that a two-pose case almost never occurs in this situation. Otherwise suppose two vertices of $P_i$ and $P_j$, respectively, are at the same point $p$ for some $i \neq j$; then the probability that $L$ passes through $p$ is $\Theta(\frac{1}{nm})$. This is $\Theta(\frac{1}{n^2})$ in the average case, as suggested that $m = \Theta(n)$ by the first set of experiments. Since in the usual case there exist only a constant number of such coincident vertex pairs, the probability $\Theta(\frac{1}{n^2})$ is an approximate upper bound on how often two-pose cases occur. This bound turns out to be consistent with the percentages of two-pose cases in Table 2.2.

It was observed during the experiments that a large number of two-pose cases occurred
when both cones happened to be supporting the polygon at the same pair of vertices (Figure 2.5). The two possible orientations differed by $\pi$, and each supporting vertex in one pose coincided with the other in the other pose. This situation often happened when the distance between one pair of vertices of the polygon was much larger than the distance between any other pair of vertices, or when the sites were far away from the polygon (as evidenced by the high percentages of two-pose cases in the last two groups of tests in Table 2.3).

2.3 Summary

The analyses and simulations in this chapter have laid out the bases for a general sensing scheme applicable to planar objects with known shapes. The scheme, termed sensing by inscription, determines the pose of an object by finding its inscription in a polygon of geometric constraints derived from the sensory data. An implementation may use a rotary sensor or a linear CCD array combined with a diverging lens to obtain the necessary constraints. In particular, two supporting cones are often enough to detect the real pose of a polygonal object. In real situations, if two (or more) possible poses arise from a two-cone inscription, they can be distinguished by point sampling, which is to be introduced in Chapter 3.\(^2\)

Although only the inscription of a convex polygon is treated, the extensions to an arbitrary polygon and a polyhedron should be straightforward. In the first case, the convex hull of the polygon can be used instead. In the second case the 3D cones are defined by visible vertices as well as edges. However, the extension to a closed and piecewise smooth curve needs further study. The technique can also be applied in object recognition: A finite set of polygons should be generally distinguishable by inscription.

Robustness of inscription can be realized by allowing some tolerance for intersecting

\(^2\)Notice that cases with more than two possible poses never occurred in our experiments on two-cone inscription.
2.3. SUMMARY

locus curves; that is, an intersection of two locus curves is considered to be a real pose if it is attained on these curves at orientations that differ by an amount less than the tolerance.

Future work would involve the design of specialized cone sensors or other sensors suited for inscription. Ken Goldberg suggested the use of a rotating light beam which can yield multiple inscribing cones as a part moves down a conveyor belt. Another direction would be to conduct an investigation of a theoretical framework for incorporating sensing uncertainties into the inscription algorithms.
Chapter 3

The Complexity of Point Sampling

Often the possible poses in which a part settles on the assembly table are not of a continuum but of a small number, either reduced by a parts feeder or limited by the mechanical constraints imposed by a sequence of planned manipulations. See the tray-tilting method (Erdmann and Mason [47]) and the squeeze-grasping method (Brost [19]) for examples of the latter case. This fact allows the implementation of more effective sensing mechanisms. The efficiency of such sensing mechanisms depends on both the time cost of the physical operations and the time complexity of the algorithms involved. Consequently, minimizing one or both of these two factors has become an important aspect of sensor design.

To illustrate the goals of this section, consider a polygonal part resting on a horizontal assembly table. The table is bounded by vertical fences at its bottom left corner, as shown in Figure 3.1. Pushing the part toward that corner will eventually cause the part to settle in one of the 12 stable poses listed in the figure. (Note that to reach a stable pose, both fences must be in contact with some vertices of the part, while at least one fence must be in contact with no less than two vertices.) To distinguish between these 12 poses, the robot has marked 4 points on the table beforehand, so it can infer the pose from which marks are covered by the part and which are not.\(^1\)

The above “shape recovery” method is named sensing by point sampling, as a loose analogy to the reconstruction of band limited functions by sampling on a regular grid in signal processing. To save the expense of sampling, the robot wants to mark as few points as possible. The problem: How to compute a minimum set of points to be marked so that parts of different types and poses can be distinguished from each other by this method?\(^2\)

\(^1\)This can be implemented in multiple ways, such as placing light detectors in the table; probing at the points; or if the robot has a vision system, taking a scene image and checking the corresponding pixel values.

\(^2\)Some very basic background in computational geometry and NP-completeness is assumed for the reading of this chapter, even though no formal prerequisites are needed.
Figure 3.1: Sensing by point sampling. (a) The 12 possible stable poses of an assembly part after pushing, along with 4 sampling points (optimal by Lemma 1) found by our implementation of the approximation algorithm to recognize these poses, where dashed line segments are fences perpendicular to the plane. (b) The planar subdivision formed by these poses that consists of 610 regions.
3.1 The Formal Problem

Consider \( n \) simple polygons \( P_1, \ldots, P_n \) in the plane, not necessarily disjoint from each other. We wish to locate the minimum number of points in the plane such that no two polygons \( P_i \) and \( P_j, i \neq j \), contain exactly the same points. To avoid ambiguities in sensing, we require that none of the located points lie on any edge of \( P_1, \ldots, P_n \). The planar subdivision formed by \( P_1, \ldots, P_n \) divides the plane into one unbounded region, some bounded regions outside \( P_1, \ldots, P_n \), called the “holes”, and some bounded regions inside. (For example, the 12 polygons in Figure 3.1(a) form the subdivision in Figure 3.1(b) that consists of 610 regions, none of which is a hole.) Immediately we make two observations: (1) Points on the edges of the subdivision, in the interior of the unbounded region, or in “holes” do not need to be considered as locations; and (2) for each bounded (open) region inside some polygon, only one point needs to be considered.

Let \( \Omega \) denote the set of bounded regions in the subdivision that are contained in at least one of \( P_1, \ldots, P_n \). Each polygon \( P_i, 1 \leq i \leq n \), is partitioned into one or more such regions; we write \( R \subseteq P_i \) when a region \( R \) is contained in polygon \( P_i \). A region basis for polygons \( P_1, \ldots, P_n \) is a subset \( \Delta \subseteq \Omega \) such that

\[
\{ R \mid R \in \Delta \text{ and } R \subseteq P_i \} \neq \{ R \mid R \in \Delta \text{ and } R \subseteq P_j \},
\]

for \( 1 \leq i \neq j \leq n \); that is, each \( P_i \) contains a distinct collection of regions from \( \Delta \). A region basis \( \Delta^* \) of minimum cardinality is called a minimum region basis. Thus, the problem of sensing by point sampling becomes the problem of finding a minimum region basis \( \Delta^* \). We will call this problem Region Basis and focus on it throughout the section. The following lemma gives the upper and lower bounds for the size of such \( \Delta^* \).

**Lemma 4** A minimum region basis \( \Delta^* \) for \( n \) polygons \( P_1, \ldots, P_n \) satisfies \( \lceil \log n \rceil \leq |\Delta^*| \leq n - 1 \).

**Proof** To verify the lower bound \( \lceil \log n \rceil \), note that each of the \( n \) polygons must contain a distinct subset of \( \Delta^* \); so \( n \leq 2^{|\Delta^*|} \), the cardinality of the power set \( 2^{\Delta^*} \).

To verify the upper bound \( n - 1 \), we incrementally construct a region basis \( \Delta \) of size at most \( n - 1 \). This construction is similar to Natarajan [108]'s Algorithm 2. Initially, \( \Delta = \emptyset \). If \( n > 1 \), without loss of generality, assume \( P_1 \) has the smallest area. Then there exists some region \( R_1 \in \Omega \) outside \( P_1 \). Split \( \{P_1, \ldots, P_n\} \) into two nonempty subsets, one including those \( P_i \) containing \( R_1 \) and the other including those not; and add \( R_1 \) into \( \Delta \). Recursively split the resulting subsets in the same way, and at each split, add into \( \Delta \) its defining region (as we did with \( R_1 \)) if this region is not already in \( \Delta \), until every subset eventually becomes a singleton. The \( \Delta \) thus formed is a region basis. Since there are \( n - 1 \) splits in total and each split adds at most one region into \( \Delta \), we have \( |\Delta| \leq n - 1 \). \( \square \)

Figure 3.2 gives two examples for which \( |\Delta^*| = \lceil \log n \rceil \) and \( |\Delta^*| = n - 1 \), respectively. Therefore these two bounds are tight.

We can view all the bounded nonhole regions as elements of \( \Omega \) and all the polygons \( P_1, \ldots, P_n \) as subsets of \( \Omega \). Then a region basis \( \Delta \) is a subset of \( \Omega \) that can discriminate
Figure 3.2: Two examples whose minimum region basis sizes achieve the lower bound \( \lceil \log n \rceil \) and the upper bound \( n - 1 \), respectively. Bounded regions in the examples are labelled with numbers. 

(a) For \( 1 \leq i \leq n \) polygon, \( P_i \) is defined to be the boundary of the union of regions \( \lceil \log n \rceil + 1, \ldots, \lceil \log n \rceil + i \), and all regions \( k \) with \( 1 \leq k \leq \lceil \log n \rceil \) such that the \( k \)th bit of the binary representation (radix 2) for \( i - 1 \) is 1. Thus, \( \Delta^* = \{1, 2, \ldots, \lceil \log n \rceil\} \). 

(b) The polygons \( P_1, \ldots, P_n \) contain each other in increasing order: \( \Delta^* = \{2, 3, \ldots, n\} \).
subsets $P_1, \ldots, P_n$ by intersection. Hence the Region Basis problem can be rephrased as: Find a subset of $\Omega$ of minimum size whose intersections with any two subsets $P_i$ and $P_j$, $1 \leq i \neq j \leq n$, are not equal. The general version of this set-theoretic problem, in which $\Omega$ stands for an arbitrary finite set and $P_1, \ldots, P_n$ stand for arbitrary subsets of $\Omega$, we call Discriminating Set. We have thus reduced Region Basis to Discriminating Set, and the former problem will be solved once we solve the latter one.

Let us analyze the amount of computation required for the geometric preprocessing to reduce Region Basis to Discriminating Set. Let $m$ be the total size of $P_1, \ldots, P_n$ (i.e., the sum of the number of vertices each polygon has); trivially $m \geq 3$. Then the planar subdivision these polygons define has at most $s$ vertices, where $3 \leq s \leq \binom{m}{2}$. By Euler’s relation on planar graphs, the number of regions and the number of edges are upper bounded by $2s - 4$ and $3s - 6$, respectively. So we can construct the planar subdivision either in time $O(m \log m + s)$ using an optimal algorithm for intersecting line segments by Chazelle and Edelsbrunner [26], or in time $O(s \log m)$ using a simpler plane sweep version by Nievergelt and Preparata [109]. To obtain the set of regions each polygon contains, we need only to traverse the portion of the subdivision bounded by that polygon, which takes time $O(s)$. It follows that the reduction to Discriminating Set can be done in time $O(m \log m + ns)$, or $O(nm^2)$ in the worst case.

Here is a short summary of the structure of the rest of Section 3.1: Section 3.2 proves the NP-completeness of Discriminating Set; based on this result, Section 3.3 establishes an equivalence between Discriminating Set and Region Basis, hence proving the latter problem NP-complete; Section 3.4 presents an $O(n^2m^2)$ approximation algorithm for Region Basis with ratio $2 \ln n$ and shows that further improvement on this ratio is hard; Section 3.5 closes up the complexity analysis of various subproblems of Discriminating Set and introduces a family of related NP-complete problems called $k$-Independent Sets; the simulation results in Section 3.6 show that sensing by point sampling is mostly applicable for dense pose distributions in the plane. We have implemented our approximation algorithm and have tested it on both real data taken from mechanical parts and random data extracted from the arrangements of random lines. The algorithm works very well in practice.

### 3.2 Discriminating Set

Given a collection $C$ of subsets of a finite set $X$, suppose we want to identify these subsets just from their intersections with some subset $D \subseteq X$. Thus, $D$ must have distinct intersection with every member of $C$; that is,

$$D \cap S \neq D \cap T, \quad \text{for all } S, T \in C \text{ and } S \neq T.$$  

We call such a subset $D$ a **discriminating set** for $C$ with respect to $X$. From a different point of view, each element $x \in D$ can be regarded as a binary “bit” that, to represent any subset $S \subseteq X$, gives value 1 if $x \in S$ and value 0 otherwise. In such a way $D$ represents an encoding scheme for subsets in $C$. 
Below we show that the problem of finding a minimum discriminating set is NP-complete. As usual, we consider the decision version of this minimization problem:

**Discriminating Set (D-Set)**
Let \( C \) be a collection of subsets of a finite set \( X \) and \( l \leq |X| \) a non-negative integer. Is there a discriminating set \( D \subseteq X \) for \( C \) such that \( |D| \leq l? \)

Our proof of the NP-completeness for D-Set uses a reduction from Vertex Cover (VC), which determines if a graph \( G = (V, E) \) has a cover of size not exceeding some integer \( l \geq 0 \) (i.e., a subset \( V' \subseteq V \) that, for each edge \( (u, v) \in E \), contains either \( u \) or \( v \)). The reduction is based on a key observation that for any three finite sets \( S_1, S_2, \) and \( S_3 \),

\[
S_1 \cap S_2 \neq S_1 \cap S_3 \quad \text{if and only if} \quad S_1 \cap (S_2 \triangle S_3) \neq \emptyset,
\]

where \( \triangle \) denotes the operation of symmetric difference (i.e., \( S_2 \triangle S_3 = (S_2 \setminus S_3) \cup (S_3 \setminus S_2) \)).

**Theorem 5** Discriminating Set is NP-complete.

**Proof** That D-Set \( \in \text{NP} \) is trivial.
Next we establish VC \( \preceq_p \text{D-Set} \); that is, there exists a polynomial-time reduction from VC to D-Set. Let \( G = (V, E) \) and integer \( 0 < l \leq |V| \) be an instance of VC. We need to construct a D-Set instance \((X, C)\) such that the collection \( C \) of subsets of \( X \) has a discriminating set of size \( l' \) or less if and only if \( G \) has a vertex cover of size \( l \) or less.

The construction uses the component design technique described by Garey and Johnson [53]. It is rather natural for us to begin by including every vertex of \( G \) in set \( X \), and assigning each edge \( e = (u, v) \) a subset \( S(e) \) in \( C \) that contains at least \( u \) and \( v \); in other words, we have \( V \subset X \) and

\[
\{u, v\} \subset S(e) \in C, \quad \text{for all } e = (u, v) \in E.
\]

To ensure that any discriminating set \( D \) for \( C \) contains at least one of \( u \) and \( v \) from subset \( S(e) \), we add an auxiliary subset \( A_e \) into \( C \) that consists of some new elements not in \( V \), and in the meantime define

\[
S(e) = \{u, v\} \cup A_e.
\]

Hence \( S(e) \triangle A_e = \{u, v\} \); and \( D \cap \{u, v\} \neq \emptyset \) follows directly from \( D \cap S(e) \neq D \cap A_e \). Since any discriminating set \( D' \) for \( \{A_e \mid e \in E\} \) can also distinguish between \( S(e_1) \) and \( S(e_2) \) and between \( S(e_1) \) and \( A_{e_2} \) for any \( e_1, e_2 \in E \) and \( e_1 \neq e_2 \), \( D' \) unioned with a vertex cover for \( G \) becomes a discriminating set for \( C \). Conversely, every discriminating set \( D \) for \( C \) can be split into a discriminating set for \( \{A_e \mid e \in E\} \) and a vertex cover for \( G \).

The \( m = |E| \) auxiliary subsets should be constructed in a way such that we can easily determine the size of their minimum discriminating sets in order to set up the entire D-Set instance. There is a simple way: We introduce \( m \) elements \( a_1, a_2, \ldots, a_m \not\in V \) into \( X \) and define subsets \( A_e \), for \( e \in E \), to be

\[
\{a_1\}, \{a_2\}, \ldots, \{a_m\},
\]
where the order of mapping does not matter. It is clear that there are \( m \) minimum discriminating sets for the above subsets: \( \{a_1, \ldots, a_m\} \setminus \{a_i\}, 1 \leq i \leq m \).

Setting \( l' = l + m - 1 \), we have completed our construction of the D-Set instance as

\[
X = V \cup \{a_1, \ldots, a_m\}; \quad a_1, \ldots, a_m \notin V;
\]

\[
C = \{ S(e) \mid e \in E \} \cup \{ A_e \mid e \in E \}.
\]

The construction can be carried out in time \( O(|V| + |E|) \). We omit the remaining task of verifying that \( G \) has a vertex cover of size at most \( l \) if and only if \( C \) has a discriminating set of size at most \( l + m - 1 \).

One thing about this proof is worthy of note. All subsets in \( C \) constructed above have at most three elements. This reveals that D-Set is still NP-complete even if \( |S| \leq 3 \) for all \( S \in C \), a stronger assertion than Theorem 5. The subproblem where all \( S \in C \) have \( |S| \leq 1 \) is obviously in \( P \), for an algorithm can simply count \( |C| \) in linear time and then answer “yes” if \( l \geq |C| - 1 \) and “no” if \( 0 \leq l < |C| - 1 \). For the remaining case in which all \( S \in C \) have \( |S| \leq 2 \), we will prove in Section 3.5 that the NP-completeness still holds. However, the proof will be a bit more involved than the one we just gave under no restriction on \( |S| \).

There would seem to be connections between our work and the concept of VC-dimension often used in learning theory. For instance, in this section we develop the notion of a “discriminating set” to distinguish different polygons. The concept of a discriminating sets bears some resemblance to the idea of shattered sets associated with VC-dimension. However, discriminating sets and shattered sets are different. A minimum discriminating set is the smallest set of points that uniquely identifies every object in a set of objects, whereas VC-dimension is the size of the largest set of points shattered by the set of objects. Thus, the VC-dimension of a finite class gives a lower bound on the size of a minimum discriminating set. For dense polygon distributions, the two cardinalities may be the same, namely, \( \log n \), where \( n \) is the number of polygons. For sparsely distributed polygons, the two cardinalities are different. For instance, the VC-dimension can be 1, while the minimum discriminating set has size \( n - 1 \) (Figure 3.2).

### 3.3 Region Basis

Now that we have shown the NP-completeness of D-Set, the minimum region basis cannot be computed in polynomial time through the use of an efficient algorithm for D-Set, because no such algorithm would exist unless \( P = NP \). This conclusion, nevertheless, leads us to conjecture that the minimization problem Region Basis is also NP-complete. Again we consider the decision version:

**Region Basis (RB)**

Given \( n \) polygons \( P_1, \ldots, P_n \) and integer \( 0 \leq l \leq n - 1 \), does there exist a region basis \( \Delta \) for the planar subdivision \( \Omega \) formed by \( P_1, \ldots, P_n \) such that \( |\Delta| \leq l \)?

The condition \( 0 \leq l \leq n - 1 \) above is necessary because we already know from Lemma 4 that a minimum region basis has size at most \( n - 1 \).
Consider a mapping $\mathcal{F}$ from the set of RB instances to the set of D-Set instances that maps regions to elements and polygons to subsets in a one-to-one manner. Every RB instance is thus mapped into an equivalent D-Set instance, as pointed out in Section 3.1. We claim that $\mathcal{F}$ is not onto. Suppose $\mathcal{F}$ were onto. Then the elements of each subset in a D-Set instance must correspond to regions in some RB instance. The union of these regions must be a polygon, and this polygon must map to the subset given in the D-Set instance. However, this is not always possible. Consider a D-Set instance generated from a nonplanar graph such that each edge is a subset containing its two vertices as only elements. No RB instance can be mapped to such a D-Set instance. For if there were such an RB instance, the geometric dual of the planar subdivision it defines would contain a planar embedding for the original nonplanar graph. This is an impossibility, and hence we have a contradiction.

Thus, the set of RB instances constitutes a proper subset of the set of D-Set instances; in other words, RB is isomorphic to a subproblem of D-Set. Therefore, the NP-completeness of RB does not follow directly from that of D-Set established earlier. Fortunately, however, D-Set has an equivalent subproblem that is isomorphic to a subproblem of RB under $\mathcal{F}$. That isomorphism provides us with the NP-completeness of Region Basis.

**Theorem 6** Region Basis is NP-complete.

**Proof** That RB $\in$ NP is easy to verify, based on the fact mentioned in Section 3.1 that the number of regions in the planar subdivision is at most quadratic in the total size of the polygons.

Let $(X, C)$ be a D-Set instance, where

$$
X = \{x_1, x_2, \ldots, x_m\}; \\
C = \{S_1, S_2, \ldots, S_n\} \subseteq 2^X.
$$

Without loss of generality, we make two assumptions

$$
\bigcup_{i=1}^{n} S_i = X \quad \text{and} \quad \bigcap_{i=1}^{n} S_i = \emptyset,
$$

because elements contained in none of the subsets or contained in all subsets can always be removed from any discriminating set of $(X, C)$. Now add in a new element $a \notin X$ and consider the D-Set instance $(X \cup \{a\}, C')$, where $C' = \{S_i \cup \{a\} \mid 1 \leq i \leq n\}$. Clearly $(X \cup \{a\}, C')$ and $(X, C)$ have the same set of irreducible discriminating sets,\(^3\) and hence they are considered equivalent.

The planar subdivision defined by the constructed RB instance for $(X \cup \{a\}, C')$ takes the configuration shown in Figure 3.3(a): A rectangular region is divided by a horizontal line segment into two identical regions among which the bottom region is named $R(a)$; the top region is further divided, this time by vertical line segments, into $2m - 1$ identical regions among which the odd numbered ones, from left to right, are named $R(x_1), \ldots, R(x_m)$.

\(^3\)A discriminating set $D$ is said to be irreducible if no subset $D' \subset D$ can be a discriminating set.
3.4 APPROXIMATION

Figure 3.3: Two reductions from Discriminating Set to Region Basis.

respectively. Remove those \( m - 1 \) unnamed regions on the top. For \( 1 \leq i \leq n \), define polygon \( P_i \) to be the boundary of the union of all regions \( R(x) \), \( x \in S_i \cup \{a\} \). It should be clear that \( P_i \) is indeed a polygon; and the two assumptions guarantee that \( P_1, \ldots, P_n \) form the desired subdivision. Note that the subdivision consists of \( m + 1 \) rectangular regions and \( 4m + 2 \) vertices. All can be computed in time \( \Theta(m) \), given the coordinates of the four vertices of the bounding rectangle. Thus, the reduction takes time \( \Theta(\sum_{i=1}^n |S_i|) \).

It is clear that \( C \) has a discriminating set of size \( l \) or less if and only if there is a region basis of the same size for \( P_1, \ldots, P_n \). Hence we have proved the NP-completeness of RB.

The above proof implies that we may regard Discriminating Set and Region Basis as equivalent problems. Note that the polygons \( P_1, \ldots, P_n \) in Figure 3.3 are not convex; will Region Basis become P when all the polygons are convex? This question is answered by the following corollary.

**Corollary 7** Region Basis remains NP-complete even if all the polygons are convex.

**Proof** This proof is the same as the proof of Theorem 6 except that we use the planar subdivision shown in Figure 3.3(b). (The vertices of the subdivision partition an imaginary circle (dotted line in the figure) into \( 2n \) equal arcs.)

3.4 Approximation

Sometimes we can derive a polynomial-time approximation algorithm for the NP-complete problem at hand from some existing approximation algorithm for another NP-complete problem by reducing one problem to the other. In fewer cases, where the reduction preserves
the solutions, namely, every instance of the original problem and its reduced instance have the same set of solutions, any approximation algorithm for the reduced problem together with the reduction will solve the original problem. The problem to which we will reduce Discriminating Set is Hitting Set:

**Hitting Set**

Given a collection $C$ of subsets of a finite set $X$, find a minimum hitting set for $C$ (i.e., a subset $H \subseteq X$ of minimum cardinality such that $H \cap S \neq \emptyset$ for all $S \in C$).

Karp [82] shows Hitting Set to be NP-complete by a reduction from Vertex Cover. The reducibility from D-Set to Hitting Set follows a key fact we observed when proving Theorem 5: The intersections of a finite set $D$ with two finite sets $S$ and $T$ are not equal if and only if $D$ intersects their symmetric difference $S \triangle T$. Given a D-Set instance, the corresponding Hitting Set instance is constructed simply by replacing all the subsets with their pairwise symmetric differences. Thus, every discriminating set of the original D-Set instance is also a hitting set of the constructed instance, and vice versa.

The approximability of Hitting Set can be studied through another problem, Set Covering:

**Set Covering**

Given a collection $C$ of subsets of a finite set $X$, find a minimum cover for $X$ (i.e., a subcollection $C' \subseteq C$ of minimum size such that $\bigcup_{S \in C'} S = X$).

This problem is also shown to be NP-complete by Karp [82] using a reduction from Exact Cover by 3-Sets. A greedy approximation algorithm for this problem due to Johnson [79] and Lovász [94] guarantees to find a cover $\hat{C}$ for $X$ with ratio

$$\frac{|\hat{C}|}{|C^*|} \leq H(\max_{S \in C} |S|) \quad \text{or simply} \quad \frac{|\hat{C}|}{|C^*|} \leq \ln |X| + 1,$$

where $C^*$ is a minimum cover and $H(k) = \sum_{i=1}^{k} \frac{1}{i}$, known as the $k$th harmonic number. The algorithm works by selecting, at each stage, a subset from $C$ that covers the most remaining uncovered elements of $X$. We refer the reader to Chvátal [29] for a general analysis of the greedy heuristic for Set Covering.

Hitting Set and Set Covering are duals to each other—the roles of set and element in one problem just get switched in the other. More specifically, let a Hitting Set instance consist of some finite set $X$ and a collection $C$ of its subsets; its dual Set Covering instance then consists of a set $\bar{C}$ and a collection of its subsets $\bar{X}$ where

$$\bar{C} = \{ \bar{S} \mid S \in C \} \quad \text{and} \quad \bar{X} = \{ \bar{x} \mid x \in X \},$$
and where each subset $\bar{x}$ is defined as

$$\bar{x} = \{ \bar{S} \mid S \in C \text{ and } S \ni x \}.$$  

Intuitively speaking, the element $x \in X$ “hits” the subset $S \in C$ in the original instance if and only if the subset $\bar{x}$ “covers” the element $\bar{S} \in \bar{C}$ in the dual instance. Thus, it follows that $H \subseteq X$ is a hitting set for $C$ if and only if $H = \{ \bar{x} \mid x \in H \}$ is a cover for $\bar{C}$. Hence the corresponding greedy algorithm for Hitting Set selects at each stage an element that “hits” the most remaining subsets. It is clear that the approximation ratio for Hitting Set becomes $H(\max_{x \in X} |\{ S \mid S \in C \text{ and } S \ni x \}|)$ or $\ln |C| + 1$.  

As a short summary, the greedy heuristic on a Discriminating Set instance $(X, C)$ works by finding a hitting set for the instance $(X, \{ S \triangle T \mid S, T \in C \})$. Since an element can appear in at most $\left\lceil \frac{n^2}{4} \right\rceil$ such pairwise symmetric differences, where $n = |C|$, the approximation ratio attained by this heuristic is $\ln \left[ \frac{n^2}{4} \right] + 1 < 2 \ln n$. The same ratio is attained for Region Basis by the heuristic that selects at each step a region discriminating the most remaining pairs of polygons, where $n$ is now the number of polygons.

The greedy algorithm for Set Covering (dually for Hitting Set) can be carefully implemented to run in time $O(\sum_{S \in C} |S|)$ (Cormen et al. [34]). The reduction from a D-Set instance $(X, C)$ to a Hitting Set instance takes time $O(|C|^2 \max_{S \in C} |S|)$. Combining the time complexity of the geometric preprocessing in Section 3.1, we can easily verify that Region Basis can be solved in time $O(nm^2 + n^2m^2) = O(n^2m^2)$, where $n$ and $m$ are the number and size of polygons, respectively.

In the remainder of this subsection we establish the hardness of approximating D-Set and hence Region Basis. Both problems allow the same approximation ratio, since the reductions from one to another do not change the number of subsets (or polygons) in an instance. First we should note that the ratio bound $H(\max_{S \in C} |S|)$ of the greedy algorithm for Set Covering is actually tight; an example that makes the algorithm achieve this ratio for arbitrarily large $\max_{S \in C} |S|$ is given in Johnson [79].

Next we present a reverse reduction from Hitting Set to D-Set to show that an algorithm for D-Set with approximation ratio $c \log n$ can be used to obtain an algorithm for Hitting Set with ratio $c \log n + O(\log \log n)$, where $c > 0$ is any constant and $n$ is the number of subsets in an instance. Afterwards we will apply some recent results on the hardness of approximating Set Covering (and thus Hitting Set).

**Lemma 8** For any $c > 0$, if $c \log n$ is the approximation ratio of Discriminating Set, then Hitting Set can be approximated with ratio $c \log n + O(\log \log n)$.

---

4According to this definition, $\bar{x} = \bar{y}$ may hold for two different elements $x \neq y$. In this case only one subset is included in $\bar{x}$.

5This definition also establishes the duality between D-Set and a known NP-complete problem called Minimum Test Set (Garey and Johnson [53]). Given a collection of subsets of a finite set, Minimum Test Set asks for a minimum subcollection such that exactly one from each pair of distinct elements is contained in some subset from this subcollection.

6Kolaitis and Thakur [86] syntactically define a class of NP-complete problems with logarithmic approximation algorithms that contains Set Covering and Hitting Set, and show that Set Covering is complete for the class.
CHAPTER 3. THE COMPLEXITY OF POINT SAMPLING

Proof Suppose there exists an algorithm $A$ for D-Set with approximation ratio $c \log n$. Let $(X, C)$ be an arbitrary instance of Hitting Set, where $C = \{S_1, \ldots, S_n\} \subseteq 2^X$, and let $n = |C|$. To construct a D-Set instance, we first make $f(n)$ isomorphic copies $(X_1, C_1), \ldots, (X_f(n), C_f(n))$ of $(X, C)$ such that $X_i \cap X_j = \emptyset$ for $1 \leq i \neq j \leq f(n)$. Here $f$ is an as yet undetermined function of $n$ upper bounded by some polynomial in $n$. Now consider the enlarged Hitting Set instance $(X', C') = (\bigcup_{i=1}^{f(n)} X_i, \bigcup_{i=1}^{f(n)} C_i)$. Every hitting set $H'$ of $(X', C')$ has $H' = \bigcup_{i=1}^{f(n)} H_i$, where $H_i$ is a hitting set of $(X_i, C_i)$, $1 \leq i \leq n$; so from $H'$ we can obtain a hitting set $H$ of $(X, C)$ with $|H| \leq |H'|/f(n)$ merely by taking the smallest one of $H_1, \ldots, H_f(n)$.

Next we introduce a set $A$ consisting of new elements $a_1, a_2, \ldots, a_{\log(nf(n))} \not\in X'$ and for $1 \leq i \leq nf(n)$ define auxiliary sets $A_i$:

$$A_i = \{ a_j \mid 1 \leq j \leq \log(nf(n)) \text{ and the } j\text{th bit of the binary representation of } i-1 \text{ is } 1 \}.$$ 

It is not hard to see that $\{a_1, \ldots, a_{\log(nf(n))}\}$ must be a subset of any discriminating set for $A_1, \ldots, A_{nf(n)}$; therefore, it is the minimum one for these auxiliary sets. The constructed D-Set instance is then defined to be $(X'', C'')$, where

$$X'' = X_1 \cup \cdots \cup X_f(n) \cup \{a_1, a_2, \ldots, a_{\log(nf(n))}\};$$

$$C'' = \{ T \cup A_{(i-1)n+j} \mid T \in C_i \text{ and } T \cong S_j \} \cup \{ A_1, \ldots, A_{nf(n)} \}.$$ 

It is easy to verify that every discriminating set of $(X'', C'')$ is the union of $A$ and a hitting set of $(X', C')$.

Now run algorithm $A$ on the instance $(X'', C'')$ and let $D$ be the discriminating set found. Then

$$\frac{|D|}{|D^*|} \leq c \log(|C''|)$$

$$= c \log(2nf(n)),$$

where $D^*$ is a minimum discriminating set. From the construction of $(X'', C'')$ we know that $D = H_1 \cup \cdots \cup H_{f(n)} \cup A$ and $D^* = H^*_1 \cup \cdots \cup H_{f(n)}^* \cup A$, where for $1 \leq i \leq n$, $H_i$ and $H_i^*$ are some hitting set and some minimum hitting set of $(X_i, C_i)$, respectively. Let $H_k$ satisfy $|H_k| = \min_{i=1}^{f(n)} |H_i|$ and thus let $H \cong H_k$ be a hitting set of $(X, C)$. Also, let $H^*$ with $|H^*| = |H^*_1| = \cdots = |H_{f(n)}^*|$ be a minimum hitting set of $(X, C)$. Then

$$\frac{|D|}{|D^*|} = \frac{\sum_{i=1}^{f(n)} |H_i| + |A|}{\sum_{i=1}^{f(n)} |H_i^*| + |A|} \geq \frac{f(n) \cdot |H| + \log(nf(n))}{f(n) \cdot |H^*| + \log(nf(n))}.$$ 

Combining the two inequalities above generates:

$$\frac{|H|}{|H^*|} \leq c \log(2nf(n)) + \frac{(c \log(2nf(n)) - 1) \cdot \log(nf(n))}{f(n) \cdot |H^*|}.$$
3.5. MORE ON DISCRIMINATING SET

\[ c \log(2nf(n)) + \frac{(c \log(2nf(n)) - 1) \cdot \log(nf(n))}{f(n)} \]
\[ = c \log n + \left[ c + c \log f(n) + \frac{(c \cdot (1 + \log n + \log f(n)) - 1) \cdot (\log n + \log f(n))}{f(n)} \right]. \]

Setting \( f(n) = \log^2 n \), all terms in the brackets can be absorbed into \( O(\log \log n) \) after simple manipulations on asymptotics (Graham et al. [62]); thus we have

\[ \frac{|H|}{|H^*|} \leq c \log n + O(\log \log n). \]

Though Set Covering has been extensively studied since the mid 1970s, essentially nothing on the hardness of approximation was known until very recently. The results of Arora et al. [9] imply that no polynomial approximation scheme exists unless \( P = NP \). Based on recent results from interactive proof systems and probabilistically checkable proofs and their connection to approximation, several asymptotic improvements on the hardness of approximating Set Covering have been made. In particular, Lund and Yannakakis [96] showed that Set Covering cannot be approximated with ratio \( c \log n \) for any \( c < \frac{1}{4} \) unless \( NP \subset DTIME(n^{poly \log n}) \); Bellare et al. [13] showed that approximating Set Covering within any constant is \( NP \)-complete, and approximating it within \( c \log n \) for any \( c < \frac{1}{8} \) implies \( NP \subset DTIME(n^{\log \log n}) \). Based on their results and by Lemma 8, we conclude on the same hardness of approximating D-Set and Region Basis:

**Theorem 9** Discriminating Set and Region Basis cannot be approximated by a polynomial-time algorithm with ratio bound \( c \log n \) for any \( c < \frac{1}{4} \) unless \( NP \subset DTIME(n^{poly \log n}) \), or for any \( c < \frac{1}{8} \) unless \( NP \subset DTIME(n^{\log \log n}) \).

Following the above theorem, the ratio \( 2 \ln n \approx 1.39 \log n \) of the greedy algorithm for D-Set remains asymptotically optimal if \( NP \) is not contained in \( DTIME(n^{poly \log n}) \).

It should be noted that Arkin et al. [8] also exhibit a greedy approximation algorithm for constructing such a decision tree. Their result is similar to our approximation algorithm, with a similar ratio bound. The difference is that our greedy algorithm seeks to minimize the total number of probe points rather than the tree height.\(^7\)

### 3.5 More on Discriminating Set

Now let us come back to where we left the discussion on the subproblems of D-Set in Section 3.2; it has not been settled whether D-Set remains NP-complete when every subset

\(^7\)It is easy to give an example for which a minimum height decision tree uses more than minimum number of total probes, while a decision tree with minimum number of total probes does not attain the minimum height. Consider the problem of discriminating sets \( \{a, b, a', e'\}, \{a, a'\}, \{b, b', d'\}, \{c, b'\}, \{d, c'\}, \) and \( \emptyset \), which can be viewed as probing a collection of polygons by the later transformation technique in the proof of Theorem 6. The decision tree using minimum probes \( a, b, c, d \) always has height 4, and the decision tree using probes \( a', b', c', d', e' \) can achieve minimum height 3.
S in the collection $C$ satisfies $|S| \leq 2$. We now prove that this subproblem is NP-complete.

Here we look at a special case of this subproblem, namely, a “subsubproblem” of D-Set and subject it to two restrictions: (1) $\emptyset \in C$ and (2) $|S| = 2$ for all nonempty subsets $S \in C$. Let us call this special case $0-2$ D-Set. If 0-2 D-Set is proven to be NP-complete, so will be the original subproblem.

It is quite intuitive to understand a 0-2 D-Set instance in terms of a graph $G = (V, E)$, where $V = X$, the finite set of which every $S \in C$ is a subset, and

$$E = \{ (u, v) \mid \{u, v\} \in C \}.$$ 

In other words, each element of the set $X$ corresponds to a vertex in $G$ while each subset, except $\emptyset$, corresponds to an edge. Clearly this correspondence from all 0-2 D-Set instances to all graphs is one-to-one. Since any discriminating set $D$ for $C$ has

$$D \cap S \neq D \cap \emptyset = \emptyset, \quad \text{for all } S \in C \text{ and } S \neq \emptyset,$$

$D$ must be a vertex cover for $G$. Let $d(u, v)$ be the distance (i.e., the length of the shortest path) between vertices $u, v$ in $G$ (or $\infty$ if $u$ and $v$ are disconnected). A 3-independent set in $G$ is a subset $I \subseteq V$ such that $d(u, v) \geq 3$ for every pair $u, v \in I$. The following lemma captures the dual relationship between a discriminating set for $C$ and a 3-independent set in $G$.

**Lemma 10** Let $X$ be a finite set and $C$ a collection of $\emptyset$ and two-element subsets of $X$. Let $G = (X, E)$ be a graph with $E = \{ (u, v) \mid \{u, v\} \in C \}$. Then a subset $D \subseteq X$ is a discriminating set for $C$ if and only if $X \setminus D$ is a 3-independent set in $G$.

**Proof** Let $D$ be a discriminating set for $C$. Assume there exist two distinct elements (vertices) $u, v \in X \setminus D$ such that $d(u, v) < 3$. We immediately have $(u, v) \notin E$, since $D$ must be a vertex cover in $G$; so $d(u, v) = 2$. Hence there is a third vertex, say $w$, that is connected to both $u$ and $v$; furthermore, $w \in D$ holds, since the edges $(u, w)$ and $(v, w)$ must be covered by $D$. However, now we have $D \cap \{u, w\} = D \cap \{v, w\} = \{w\}$, a contradiction to the fact that $D$ is a discriminating set.

Conversely, suppose $X \setminus D$ is a 3-independent set in $G$, for some subset $D \subseteq X$. Then $D$ must be a vertex cover. Suppose it is not a discriminating set for $C$. Then there exist two distinct subsets $S_1, S_2 \in C$ such that $D \cap S_1 = D \cap S_2 = \{w\}$, for some $w \in S$. Writing $S_1 = \{u, w\}$ and $S_2 = \{v, w\}$, we have $d(u, v) = 2$; but in the meantime $u, v \in X \setminus D$. A contradiction again. \qed

This lemma tells us that the NP-completeness of 0-2 D-Set, and therefore of our remaining open subproblem of D-Set, follows if we can show the NP-completeness of 3-Independent Set. 3-Independent Set is among a family of problems defined, for all integers $k > 0$, as follows:

**k-Independent Set (k-IS)**

Given a graph $G = (V, E)$ and an integer $0 < l \leq |V|$, is there a $k$-independent set of size at least $l$, that is, is there a subset $I \subseteq V$ with $|I| \geq l$ such that $d(u, v) \geq k$ for every pair $u, v \in I$?
Thus 2-IS is the familiar NP-complete Independent Set problem. We will see in Appendix B that every problem in this family for which \( k > 3 \) is also NP-complete. To avoid too much divergence from 0-2 D-Set, let us focus on 3-IS only here.

**Lemma 11** 3-Independent Set is NP-complete.

**Proof** It is trivial that 3-IS \( \in \text{NP} \). To show NP-hardness, we reduce Independent Set (2-IS) to 3-IS. Let \( G = (V, E) \) and \( 0 < l \leq |V| \) form an instance of Independent Set. A graph \( G' \) is constructed from \( G \) in two steps. In the first step, we introduce a “midvertex” \( w_{u,v} \) for each edge \((u, v) \in E\), and replace this edge with two edges \((u, z_{u,v})\) and \((z_{u,v}, v)\). In the second step, an edge is added between every two midvertices that are adjacent to the same original vertex. More formally, we have defined \( G' = (V', E') \) where

\[
V' = V \cup \{ z_{u,v} \mid (u, v) \in E \};
\]

\[
E' = \{ (z_{u,v}, u) \mid (u, v) \in E \} \cup \{ (z_{u,v}, z_{u,w}) \mid (u, v) \neq (u, w) \in E \}.
\]

Two observations are made about this construction. First, it has the property that \( d'(u, v) = d(u, v) + 1 \) holds for any pair of vertices \( u, v \in V \), where \( d \) and \( d' \) are the two distance functions in \( G \) and \( G' \) respectively. This equality can be verified by contradiction. Next, if \((u, v) \in E\), then any two midvertices \( z_{u,x} \) and \( z_{v,y} \) have

\[
d'(z_{x,x}, z_{y,y}) \leq d'(z_{x,x}, z_{y,y}) + d'(z_{y,y}, z_{v,y}) \leq 2;
\]

\[
d'(z_{u,x}, z_{v,y}) = d'(z_{u,x}, z_{u,v}) + d'(z_{u,v}, z_{v,y}) \leq 2;
\]

\[
d'(z_{u,x}, z_{v,y}) = d'(z_{u,x}, z_{u,v}) + d'(z_{u,v}, z_{v,y}) \leq 2.
\]

Note strict “<”s appear in the above three inequalities when \( x = v \) or \( y = u \) and in the first inequality when \( x = y \). It is not difficult to see that the entire reduction can be done in time \( O(|V|^3) \). Figure 3.4 illustrates an example of the reduction.

We claim that \( G \) has an independent set \( I \) of size at least \( l \) if and only if \( G' \) has a 3-independent set \( I' \) of the same size. Suppose \( I \) with \( |I| \geq l \) is an independent set in \( G \). Then \( I \) is also a 3-independent set in \( G' \). This follows from our first observation. Conversely, suppose \( I' \) with \( |I'| \geq l \) is a 3-independent set in \( G' \). Then the set \( I \), produced by replacing each midvertex \( z_{u,v} \in I' \) with either \( u \) or \( v \), is an independent set in \( G \). To see this, assume there exist two vertices \( u, v \in I \) such that \( d(u, v) = 1 \). Thus, \( d'(u, v) = d(u, v) + 1 = 2 < 3 \); so either \( u \) or \( v \), or both, must have replaced some midvertices in \( I' \). Let \( s, t \in I' \) be the two vertices corresponding to \( u \) and \( v \) before the replacement, respectively; that is, \( s = u \) or \( z_{u,x} \) and \( t = v \) or \( z_{v,y} \) for some \( x, y \in V \). According to our second observation, we always have \( d'(s, t) \leq 2 \). Thus we have reached a contradiction, since \( s, t \in I' \). That \(|I'| = |I| \geq l \) is easy to verify in a similar way.

Combining Lemmas 10 and 11, we have the NP-completeness of 0-2 D-Set; this immediately resolves the complexity of our remaining subproblem of D-Set:

**Theorem 12** D-Set remains NP-complete even if \(|S| \leq 2 \) for all \( S \in C \).
CHAPTER 3. THE COMPLEXITY OF POINT SAMPLING

Figure 3.4: An example of reduction from Independent Set to 3-IS. (a) An instance of Independent Set. (b) The constructed 3-IS instance: α vertices are added to increase the distances between the original vertices by exactly one.

3.6 Simulations

For geometric preprocessing, we implemented the plane sweep algorithm by Nievergelt and Preparata [109]. We modified the original algorithm so that the containing polygons of each swept region are maintained and propagated along during the sweeping.\(^8\) The greedy approximation algorithm for Set Covering was implemented with a linked list to attain the running time \(O(\sum_{S \in C} |S|)\). All code was written in Common Lisp and was run on a Sparcstation IPX.

We discuss simulation results on random polygons. These simulations empirically study how the number of sampling points varies with the “density” of polygons in the plane. The results suggest that the point sampling approach is most effective at sensing polygonal objects that have highly overlapping poses. Experiments on a Zebra robot are underway, and the results will be presented in the near future.

To generate random polygons, we precomputed an arrangement of a large number (such as 100) of random lines using a topological sweeping algorithm (Edelsbrunner and Guibas [41]). A random polygon was extracted as the first “valid” cycle during a random walk on this line arrangement, which was then randomly scaled, rotated, and translated. By “valid” we mean that the number of vertices in the cycle was no less than some small random integer. This constraint was introduced merely to allow a proper distribution of polygons of various sizes, for otherwise triangles and quadrilaterals would be generated with high probabilities according to our observations. In a sample run, a group of 1000 polygons generated (by this

\(^8\)This implementation has the same worst-case running time as a different version described in Section 3.1 that obtains the containment information by traversing the planar subdivision after the sweeping. However, the implemented version is usually more efficient in practice.
3.6. SIMULATIONS

method) from an arrangement of 100 random lines had sizes in the range 3–30, with mean 5.545 and standard deviation 3.225.

All random polygons (or all random poses of a single polygon) in a test were bounded by some square, so that the “density” (i.e., the degree of overlap) of these polygons mainly depended on their number as well as on the ratio between their average area and the size of the bounding square. Since polygons were generated randomly, the average area could be viewed as approximately proportional to the square of the average perimeter. The configuration of each polygon, say $P$, was assumed to obey a “uniform” distribution inside the square. More specifically, the orientation of $P$ was first randomly chosen from $[0, 2\pi)$; the position of $P$ was then randomly chosen from a rectangle inside the square consisting of all feasible positions at that orientation.\(^9\)

To be robust against sensor noise, the sampling point of every region in the region basis was selected as the center of a maximum inscribed circle in that region. In other words, this sampling point had the maximum distance to the polygon bounding that region. It is not difficult to see that such a point must occur at a vertex of the generalized Voronoi diagram inside the polygon, also called its internal skeleton or medial axis function.\(^10\) Also, for sensing robustness, regions with area less than some threshold were not considered at the stage of region basis computation.\(^11\) Although this thresholding traded off the completeness of sampling, it almost never resulted in the failure of finding a region basis once the threshold was properly set.

The first two groups of six tests gave a sense of the number of sampling points required when polygons are sparsely distributed in the plane. The results are summarized in Table 3.1. Every test in group (a) was conducted on distinct (i.e., non-congruent), random polygons with perimeters between $\frac{1}{4}$ and $\frac{3}{4}$ of the width of the bounding square; every test in group (b) was conducted on distinct poses of a single polygon with perimeter equal to $\frac{5}{8}$ of the width of the square. The scenes of the last tests from these two groups are displayed in Figure 3.5.

Without any surprise, the number of sampling points found were around half of the number of polygons, for all twelve tests in Table 3.1. This supports the fact that, for $n$ sparsely distributed polygons in the plane, the minimum number of sampling points turns out to be $\Theta(n)$. As we can see from Figure 3.5, in such a situation every polygon intersects at most a few or, more precisely, no more than some constant number of, other polygons. In other words, the number of polygon pairs distinguishable by any single region in the planar subdivision is $\Theta(n)$; but there are $\lfloor \frac{n^2}{4} \rfloor$ such pairs in total! Thus, sensing by point sampling

\(^9\)If the diameter of $P$ is greater than the width of the square, then not every orientation is necessarily feasible. However, this situation was avoided in our simulations.

\(^10\)The construction of the internal skeleton of a polygon is a special case of the construction of the generalized Voronoi diagram for a set of line segments for which $O(n \log n)$ algorithms were given by Kirkpatrick [85], Fortune [52], and Yap [138]. A linear time algorithm on finding the medial axis of a simple polygon was recently given by Chin et al. [28]. Since the maximum region size for a region basis turned out to be very small in the simulations, we implemented only an $O(n^4)$ brute force algorithm.

\(^11\)We thresholded on the region area rather than the radius of a maximum inscribed circle merely to avoid the inefficient computation on the latter for all the regions in the planar subdivision.
Figure 3.5: Sampling 100 sparsely distributed random polygons/poses. (a) The scene of the last test from group (a) in Table 3.14: There are 1422 regions in the planar subdivision and 51 sampling points (drawn as dots) to discriminate the 100 polygons. (b) The scene of the last test from group (b) in Table 3.1: There are 1643 regions in the planar subdivision and 61 sampling points to discriminate the 100 poses.
3.6. SIMULATIONS

### Table 3.1: Tests on sampling sparsely distributed random polygons/poses.

The twelve tests were divided into two groups: (a) All polygons in each test were distinct, with perimeters between $\frac{1}{4}$ and $\frac{3}{4}$ times the width of the bounding square. (b) All polygons in each test represented distinct random poses of a same polygon. The polygon perimeter was uniformly $\frac{5}{8}$ times the side length of the square for all six tests in the group.

<table>
<thead>
<tr>
<th># polys</th>
<th># regions</th>
<th># sampling points</th>
</tr>
</thead>
<tbody>
<tr>
<td>50</td>
<td>320</td>
<td>31</td>
</tr>
<tr>
<td>60</td>
<td>500</td>
<td>37</td>
</tr>
<tr>
<td>70</td>
<td>594</td>
<td>41</td>
</tr>
<tr>
<td>80</td>
<td>783</td>
<td>46</td>
</tr>
<tr>
<td>90</td>
<td>973</td>
<td>47</td>
</tr>
<tr>
<td>100</td>
<td>1422</td>
<td>51</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th># polys</th>
<th># regions</th>
<th># sampling points</th>
</tr>
</thead>
<tbody>
<tr>
<td>50</td>
<td>362</td>
<td>36</td>
</tr>
<tr>
<td>60</td>
<td>609</td>
<td>34</td>
</tr>
<tr>
<td>70</td>
<td>741</td>
<td>34</td>
</tr>
<tr>
<td>80</td>
<td>1061</td>
<td>39</td>
</tr>
<tr>
<td>90</td>
<td>1125</td>
<td>49</td>
</tr>
<tr>
<td>100</td>
<td>1643</td>
<td>61</td>
</tr>
</tbody>
</table>

(a) (b)

is inefficient in a situation with a large number of sparsely distributed polygons.

The next two groups of six tests were on polygons much more densely distributed in the plane, and the results are given in Table 3.2. In these two groups of tests, we used a bounding square with side length only $\frac{1}{4}$ of the side length of the one used in the two test groups in Table 3.1. Every test in group (a) was conducted on distinct polygons with perimeter in the range $\frac{1}{2}$–2 times the side length of the bounding square. All tests in group (b) were distinct poses of the same polygon used in the last test of group (b) in Table 3.1. Again, the scenes of the last tests from groups (a) and (b) are shown in Figure 3.6.

All twelve tests in the above two groups except the last one in group (a) found sampling points at most twice the lower bound $\lceil \log n \rceil$, while the first test in group (b) found exactly $\lceil \log n \rceil$ sampling points. The data in group (b) were more densely distributed than the data in group (a) in that any pair of poses intersected. Since an extremely dense distribution of polygons may cause numerical instabilities in the plane sweep algorithm, smaller numbers of polygons were tested in these two groups than were tested in the first two groups. The results of these two groups of tests show that the sampling strategy is very applicable to sensing densely distributed polygons.
### Table 3.2: Tests on sampling densely distributed random polygons/poses, divided into two groups (a) and (b). The width of the bounding square was reduced to $\frac{1}{4}$ times the width of the square used in groups (a) and (b) in Table 3.1. In group (a) all polygons in each test were distinct with perimeter in the range between $\frac{1}{2}$ and 2 times the width of the bounding square. In group (b) all polygons in each test were distinct poses of the same polygon as in Figure 3.5(b).
Figure 3.6: Sampling densely distributed polygons. The bounding square has width $\frac{1}{4}$ times the width of the one shown in Figure 3.5. (a) The scene of the last test from group (a) in Table 3.2: There are 50 distinct polygons that form a planar subdivision with 2678 regions and that can be discriminated by 13 sampling points. (b) The scene of the last test from group (b) in Table 3.2: There are 40 distinct poses of the polygon from Figure 3.5(b), which form a planar subdivision with 4955 regions and can be discriminated by 9 sampling points.
3.7 Summary

The sensing approach, termed *sensing by point sampling*, distinguishes a finite set of poses by examining the containment of a number of points. It also works for objects with more general boundaries, though constructing the planar subdivision may become computationally expensive. Robustness mainly depends on the topological invariance of the planar subdivision. In this sense it is often more robust to consider only regions whose maximum inscribed circles have radii greater than a certain threshold.

In practice, light detectors may be placed underneath the precomputed sampling locations on the surface of an assembly table or in a tray. The actual embedding can be avoided if the surface is transparent. In this case light detectors can be easily reconfigured given different sets of parts and poses, which ensures the modularity of sensing.

Another implementation strategy is to mechanically probe the sampling points using a robot. Robustness of sampling points is more important since even a slight probe may disturb the object’s pose.
Part II

Observing Pose and Motion through Contact
Chapter 4

Mathematical Basics

This chapter reviews some basics in differential geometry and nonlinear control which are needed in Chapters 5 and 7 only. The reader may skip to Chapter 5 at this point if he is familiar with the intended material.

4.1 Elements of Differential Geometry

In this section we assume the reader's knowledge about tangent vectors, tangent spaces, and vector fields, thereby not going over these definitions. For an elementary introduction to differential geometry, we refer the reader to O'Neill [112]; for a comprehensive introduction, we refer to Spivak's five volume series that begins with [131].

We use the following conventions on notation in the second part of the thesis. Every vector $x$ is a column vector written as $(x_1, \ldots, x_n)^T$ for some variables $x_1, \ldots, x_n$. The derivative of a vector function $x(t) = (x_1(t), \ldots, x_n(t))^T$ with respect to $t$ is denoted by $\frac{dx}{dt} = (\frac{dx_1}{dt}, \ldots, \frac{dx_n}{dt})^T$. The gradient of a scalar function $y(x)$, where $x = (x_1, \ldots, x_n)^T$, is a row vector $\frac{\partial y}{\partial x} = (\frac{\partial y}{\partial x_1}, \ldots, \frac{\partial y}{\partial x_n})$. The partial derivative of a vector field $f(x) = (f_1(x), \ldots, f_m(x))^T$ with respect to vector $x = (x_1, \ldots, x_n)^T$ is a $m \times n$ matrix, given by $(\frac{\partial f}{\partial x})_{ij} = \frac{\partial f_i}{\partial x_j}$.

4.1.1 Curves

Let $I$ be an open interval in $\mathbb{R}$. A curve in $\mathbb{R}^3$ is a differentiable function $\alpha : I \to \mathbb{R}^3$ from $I$ into $\mathbb{R}^3$. For each $u \in I$, the velocity of $\alpha$ at $u$ is the tangent vector $\alpha'(u)$. A curve $\alpha(u)$ is regular provided its speed $\|\alpha'(u)\|$ is not zero for each $u \in I$. If $\alpha(u)$ is a regular curve in $\mathbb{R}^3$, then there exists a reparametrization $\beta(s)$ of $\alpha$ such that $\beta$ has unit speed, that is, $\|\beta'(s)\| = 1$ for each $s \in I$.

The unit tangent field on a unit-speed curve $\beta(s)$ from $I$ to $\mathbb{R}^3$ is defined to be $T = \beta'$. The function $\kappa : s \mapsto \|T'(s)\|$ for all $s \in I$ is called the curvature of $\beta$. Also defined on $\beta$ are the principal normal vector field $N = T'/\kappa$ and the binormal vector field $B = T \times N$. We call $T, N, B$ the Frenet frame field on $\beta$. The function $\tau : I \to \mathbb{R}$ given by $B' = -\tau N$ is
called the torsion of $\beta$. The Frenet formulas hold for the derivatives of $T, N, B$:

$$
T' = \kappa N; \\
N' = -\kappa T + \tau B; \\
B' = -\tau N.
$$

The osculating plane at the point $\beta(s)$ of a unit-speed curve $\beta$ is the plane orthogonal to the binormal vector $B(s)$. If $\kappa(s) > 0$, there exists one and only circle $\gamma$ such that

$$
\gamma(0) = \beta(s), \quad \gamma'(0) = \beta'(s), \quad \text{and} \quad \gamma''(0) = \beta''(s).
$$

The circle $\gamma$ is called the osculating circle, its center and radius the center and radius of curvature of $\beta$ at $\beta(s)$, respectively.

### 4.1.2 Surfaces

A mapping $f$ from an open set $D \subset \mathbb{R}^2$ to $\mathbb{R}^3$ is regular provided that at each point $p \in D$ the Jacobian matrix $\partial f / \partial p$ has rank 2. A coordinate patch (or a patch) $g : D \rightarrow \mathbb{R}^3$ is a one-to-one regular mapping from an open set $D \subset \mathbb{R}^2$ to $\mathbb{R}^3$. A patch $g : D \rightarrow \mathbb{R}^3$ is called a proper patch provided its inverse function $g^{-1} : g(D) \rightarrow D$ is continuous.

Let $g : D \rightarrow \mathbb{R}^3$ be a coordinate patch. For each point $(u_0, v_0) \in D$ the curve $g(u, v_0)$ is called the $u$-parameter curve at $v = v_0$, and the curve $g(u_0, v)$ is called the $v$-parameter curve at $u = u_0$.

A subset $M$ of $\mathbb{R}^3$ is a surface in $\mathbb{R}^3$ provided for each point $p \in M$ there exists a proper patch $g$ whose image contains a neighborhood of $p$.

Let $p$ be a point of a surface $M$ in $\mathbb{R}^3$. The set of all tangent vectors to $M$ at $p$ is called the tangent plane of $M$ at $p$ and denoted by $T_p(M)$. A vector field $V$ on a surface $M$ is a differentiable function that assigns to each point $p$ of $M$ a tangent vector $V(p)$ to $\mathbb{R}^3$ at $p$.

A vector field $V$ is said to be a tangent vector field provided $V(p)$ is tangent to $M$ at each point $p$ of $M$. A vector field $V$ is said to be a normal vector field provided each vector $V(p)$ is normal to $M$. A surface $M$ in $\mathbb{R}^3$ is said to be orientable provided there exists a normal vector field $Z$ on $M$ that is nonzero at each point of $M$.

### 4.1.3 Normal and Gaussian Curvatures

Let $Z$ be a vector field on a surface $M$ in $\mathbb{R}^3$, $v$ a tangent vector to $M$, and $\alpha$ a curve in $M$ that has initial velocity $\alpha'(0) = v$. Let $Z_{\alpha} : t \mapsto Z(\alpha(t))$ be the restriction of $Z$ to $\alpha$.

Then the covariant derivative of $Z$ with respect to $v$ is defined to be $\nabla_v Z = (Z_{\alpha})'(0)$. Let $p$ be a point of $M$ and $U$ a unit normal vector field on a neighborhood of $p$ in $M$. The shape operator of $M$ at $p$ is a function $S_p : v \mapsto -\nabla_v U$ for each tangent vector $v$ to $M$ at $p$.

The shape operator $S_p$ is a symmetric linear operator that maps the tangent plane $T_p(M)$ to itself.

Let $u$ be a unit tangent vector to $M$ at a point $p$. The normal curvature of $M$ in the $u$ direction is given by $\kappa_u(u) = S(u) \cdot u$. The normal section of $M$ at $p$ in the $u$ direction is a
4.1. ELEMENTS OF DIFFERENTIAL GEOMETRY

A curve cut from $M$ by a plane containing $\mathbf{u}$ and the surface normal $\mathbf{U}(p)$; hence its curvature is the normal curvature $\kappa_n(\mathbf{u})$. The maximum and minimum values of the normal curvature $\kappa_n(\mathbf{u})$ are called the principal curvatures of $M$ at $p$ and denoted by $\kappa_1$ and $\kappa_2$, respectively. The directions in which $\kappa_1$ and $\kappa_2$ occur are called the principal directions of $M$ at $p$. Vectors in these directions are called the principal vectors of $M$ at $p$.

Let $\kappa_1, \kappa_2$ and $\mathbf{e}_1, \mathbf{e}_2$ be the principal curvatures and vectors of $M$ at $p$, with $\mathbf{e}_1 \times \mathbf{e}_2$ being the (oriented) normal. The normal curvature of $M$ in the direction $\mathbf{u} = \cos \theta \mathbf{e}_1 + \sin \theta \mathbf{e}_2$ is

$$\kappa_n(\mathbf{u}) = \kappa_1 \cos^2 \theta + \kappa_2 \sin^2 \theta.$$

A point $p$ of $M$ is umbilic provided the normal curvature $\kappa_n(\mathbf{u})$ is constant on all unit tangent vectors $\mathbf{u}$ to $M$ at $p$. If $p$ is a nonumbilic points, $\kappa_1 \neq \kappa_2$, then there are exactly two principal directions and they are orthogonal.

The Gaussian curvature $K$ at a point $p$ of $M$ is the determinant of the shape operator $S_p$ and the mean curvature is the function $H = \frac{1}{2} \text{trace } S_p$. They can be also expressed in terms of the principal curvatures by $K = \kappa_1 \kappa_2$ and $H = \frac{1}{2}(\kappa_1 + \kappa_2)$.

A surface $M$ is flat provided its Gaussian curvature is zero, and minimal provided its mean curvature is zero.

4.1.4 Curves in a Surface

Let $\alpha$ be a unit-speed curve in a surface $M \subset \mathbb{R}^3$. The Darboux frame field is defined by the unit tangent $T$ of $\alpha$, the surface normal $\mathbf{U}$ restricted to $\alpha$, and $V = \mathbf{U} \times T$. As in the case of the Frenet frame, the derivatives of $T, U, V$ can be represented in terms of $T, U, V$ in the Darboux frame:

$$T' = \kappa_g V + \kappa_n U;$$
$$V' = -\kappa_g T + \tau_g U;$$
$$U' = -\kappa_n T - \tau_g V;$$

where $\kappa_n$ is the normal curvature of $M$, $\kappa_g$ and $\tau_g$ are the geodesic curvature and torsion of $\alpha$, respectively. Obviously we have $\kappa = \sqrt{\kappa_n^2 + \kappa_g^2}$, where $\kappa$ is the curvature of $\alpha$.

Let $\kappa_1, \kappa_2$ and $\mathbf{e}_1, \mathbf{e}_2$ be the principal curvatures and vectors of $M$ at $p$, with $\mathbf{e}_1$ and $\mathbf{e}_2$ positively oriented. The geodesic torsion in the direction $\mathbf{u} = \cos \theta \mathbf{e}_1 + \sin \theta \mathbf{e}_2$ is

$$\tau_g = (\kappa_2 - \kappa_1) \sin \theta \cos \theta.$$

A regular curve $\alpha$ in $M$ is a principal curve (or line of curvature) provided that its velocity $\alpha'$ always points in a principal direction. A regular curve $\alpha$ is principal if and only if $\tau_g = 0$.

A regular curve $\alpha$ in $M$ is an asymptotic curve provided its velocity $\alpha'$ always points in a direction in which $\kappa_n = 0$.

A curve $\alpha$ in $M$ is a geodesic provided its acceleration $\alpha''$ is always normal to $M$. A curve $\alpha$ is geodesic if and only if $\kappa_g = 0$. 
4.1.5 Patch Computations

A patch \( f : D \rightarrow M \) is orthogonal provided its two partial derivatives are orthogonal to each other, that is, \( f_u \cdot f_v = 0 \) for each \((u, v) \in D\). An orthogonal patch \( f : D \rightarrow M \) is principal provided \( S(f_u) \cdot f_v = S(f_v) \cdot f_u = 0 \), where \( S \) is the shape operator of \( M \). The parameter curves of a principal patch \( f \) are lines of curvature.

If a point \( p \) on a surface \( M \) is not umbilic, then there exists a one-to-one and regular mapping \( f : U \rightarrow M \) on an open set \( U \subset \mathbb{R}^2 \) with \( p \in f(U) \), whose parameter curves are lines of curvature.

Let \( f : D \rightarrow M \) be an orthogonal patch in surface \( M \). The normalized Gauss frame at a point \( f(u, v) \) is the coordinate frame with origin at \( f(u, v) \) and coordinate axes\(^1\)

\[
\begin{align*}
x(u, v) &= \frac{f_u(u, v)}{\|f_u(u, v)\|}; \\
y(u, v) &= \frac{f_v(u, v)}{\|f_v(u, v)\|}; \\
z(u, v) &= x(u, v) \times y(u, v).
\end{align*}
\]

The shape operator \( S \) with respect to \( x, y \) is

\[
S = (x, y)^T \left(-\nabla_x z, -\nabla_y z\right) = (x, y)^T \left(-\frac{z_u}{\|f_u\|}, -\frac{z_v}{\|f_v\|}\right)
\]

At a point \( f(u, v) \in D \) the geodesic curvatures \( \kappa_{gu} \) and \( \kappa_{gv} \), of the \( u \)-parameter curve and the \( v \)-parameter curve, respectively, are given as

\[
\begin{align*}
\kappa_{gu} &= y \cdot \nabla_x x = y \cdot \frac{x_u}{\|f_u\|}; \\
\kappa_{gv} &= -x \cdot \nabla_y y = y \cdot \frac{x_v}{\|f_v\|}. 
\end{align*}
\]

4.1.6 Manifolds, Cotangent Bundles, Codistributions

An \( n \)-dimensional manifold \( M \) is a set furnished with a collection \( C \) of abstract patches (one-to-one functions \( f : D \rightarrow M \), \( D \) an open set in \( \mathbb{R}^n \)) such that

1. \( M \) is covered by the images of the (abstract) patches in the collection \( C \).

\(^1\)We assume the coordinate frame is everywhere right-handed.
2. For any two patches $f$ and $g$ in $C$, the composite functions $g^{-1}f$ and $f^{-1}g$ are differentiable and defined on open sets in $\mathbb{R}^n$.

A surface of $\mathbb{R}^3$ is just a two-dimensional manifold. Tangent vectors, tangent spaces, vector fields on an $n$-dimensional manifold are defined in the same way as in the special case $n = 2$. We need only replace $i = 1, 2$ by $i = 1, 2, \ldots, n$.

Let $M$ be a manifold and $T^*_pM$ its tangent space at a point $p$ of $M$. The dual space of $T^*_pM$, denoted $T^*_pM$, is called the cotangent space of $M$ at $p$. More specifically, $T^*_pM$ consists of all linear functions on $T^*_pM$. An element of $T^*_pM$ is called a cotangent vector. The cotangent bundle of a manifold $M$ is defined as

$$T^*M = \bigcup_{p \in M} T^*_pM.$$ 

A codistribution $D$ on a manifold $M$ assigns to each point $p \in M$ a linear subspace $D(p)$ of the cotangent space $T^*_pM$.

A one-form $\phi$ on $M$ is a map that assigns to each point $p \in M$ a cotangent vector $\phi(p) \in T^*_pM$. The gradient of a real-valued function $f$ on $M$ is a one-form $df$ called the differential of $f$.

### 4.1.7 Lie Derivatives

Let $M$ be an $n$-dimensional manifold. The Lie derivative of a function $h : M \to \mathbb{R}$ along a vector field $X$ on $M$, denoted by $L_Xh$, is the directional derivative $dh(X) = dh \cdot X$, where the one-form $dh$ is the differential of $h$. We use the notation $L_{X_1}L_{X_2} \cdots L_{X_l}h$ for the repeated Lie derivative $L_{X_1}(L_{X_2}(\ldots(L_{X_l}h)\ldots))$ with respect to the vector fields $X_1, \ldots, X_2, X_1$ on $M$.

The Lie bracket of two vector fields $X$ and $Y$ on $M$ at a point $p$ of $M$ is a vector field defined as

$$[X,Y](p) = \frac{\partial Y}{\partial p}(p)X(p) - \frac{\partial X}{\partial p}(p)Y(p).$$

The ad-notation is used for repeated Lie brackets:

$$\text{ad}^0_X Y = Y;$$
$$\text{ad}^j_X Y = [X, \text{ad}_{X}^{j-1} Y], \quad \text{for } j > 0.$$ 

The bracket $[X,Y]$ can be interpreted in some sense as the “derivative” of the vector field $Y$ along the vector field $X$. It is therefore also denoted as $L_XY$, the Lie derivative of $Y$ along $X$.

Let $\phi$ be a one-form defined on $M$. Lie differentiation also acts on $\phi$ in the following way:

$$L_X(\phi)(p) = \left(\frac{\partial \phi^T}{\partial p}(p)X(p)\right)^T + \phi(p)\frac{\partial X}{\partial p}(p).$$
The introduced three kinds of Lie differentiation are related by the following Leibniz-type formula:

\[ L_X(\phi \cdot Y) = L_X\phi \cdot Y + \phi \cdot [X,Y]. \]

When \( \phi \) is a differential \( dh \) for some real function \( h \) on \( M \), \( L_X \) and \( d \) commute:

\[ L_X(dh) = d(L_X h), \]

and the Leibniz formula becomes

\[ L_{[X,Y]} h = L_X L_Y h - L_Y L_X h. \]

### 4.2 Nonlinear Control

The theoretical foundation of our work comes from the part of control theory concerned with the observability and observers of nonlinear systems. For a general introduction to nonlinear control theory, we refer the reader to Isidori [74] and Nijmeijer and van der Schaft [110]. Related results to our work will be mentioned as we proceed later in Sections 4.2.1 and 4.2.2.

#### 4.2.1 Observability of a Nonlinear System

Consider a smooth affine (or input-linear) control system together with an output map:

\[
\begin{align*}
\dot{x} &= f(x) + \sum_{i=1}^{m} u_i g_i(x), \\
y &= h(x),
\end{align*}
\]

where \( x = (x_1, \ldots, x_n)^T \) is the state in a smooth \( n \)-dimensional manifold \( M \subseteq \mathbb{R}^n \) (called the state space manifold), \( f, g_1, \ldots, g_m \) are smooth vector fields on \( M \), and \( h = (h_1, \ldots, h_k)^T : M \rightarrow \mathbb{R}^k \) is the smooth output map of the system. Here \( f \) is called the drift vector field, and \( g_1, \ldots, g_m \) are called the input vector fields. In the system, \( u_1, \ldots, u_m \) are the inputs, called the controls, over time whose Cartesian product range \( U \) defines the system’s input space. At state \( x \), \( f(x) \) is a tangent vector to \( M \) representing the rate of change of \( x \) without any input, while \( g_j(x) \) for \( 1 \leq j \leq m \) is a tangent vector showing the rate of such change under unit input of \( u_j \).

Throughout we are only concerned with the class of controls \( U \) that consists of piecewise constant functions that are continuous from the right.\(^2\) We call these controls admissible. The system with constant controls, or no input fields, equivalently, is said to be autonomous.

Denote by \( y(t, x_0, u) \), \( t \geq 0 \), the output function of the system with initial state \( x_0 \) and under control \( u \). Two states \( x_1, x_2 \in M \) are said to be indistinguishable (denoted by \( x_1 I x_2 \)) if for every admissible control \( u \) the output functions \( y(t, x_1, u) \) and \( y(t, x_2, u) \), \( t \geq 0 \) are

\(^2\)So that \( U \) is closed under concatenation.
4.2. NONLINEAR CONTROL

identical on their common domain of definition. The system is observable if \( x_1 I x_2 \) implies \( x_1 = x_2 \).

To derive a condition on nonlinear observability, the above definition of “observable” is localized in the following way. Let \( V \subset M \) be an open set containing states \( x_1 \) and \( x_2 \). These two states are said to be \( V \)-indistinguishable, denoted by \( x_1 I_V x_2 \), if for any \( T > 0 \) and any constant control \( u : [0, T] \rightarrow U \) such that \( x(t, x_1, u), x(t, x_2, u) \in V \) for all \( 0 \leq t \leq T \), it follows that \( y(t, x_1, u) = y(t, x_2, u) \) for all \( 0 \leq t \leq T \) on their common domain of definition. The system is locally observable at \( x_0 \) if there exists a neighborhood \( W \) of \( x_0 \) such that for every neighborhood \( V \subset W \) of \( x_0 \) the relation \( x_0 I_V x_1 \) implies that \( x_0 = x_1 \). The system is called locally observable if it is locally observable at every \( x_0 \in M \). Figure 4.1 illustrates local observability for the case of one output function.

The observation space \( \mathcal{O} \) of system (4.1) is the linear space (over \( \mathbb{R} \)) of functions on \( M \) that includes \( h_1, \ldots, h_k \), and all repeated Lie derivatives

\[
L_{X_1}L_{X_2} \cdots L_{X_l}h_j, \quad j = 1, \ldots, k, \ l = 1, 2, \ldots
\]

where \( X_i \in \{ f, g_1, \ldots, g_m \} \), \( 1 \leq i \leq l \). It is not difficult to show that \( \mathcal{O} \) is also the linear space of functions on \( M \) that includes \( h_1, \ldots, h_k \), and all repeated Lie derivatives

\[
L_{Z_1}L_{Z_2} \cdots L_{Z_l}h_j, \quad j = 1, \ldots, k, \ l = 1, 2, \ldots
\]

where

\[
Z_i(x) = f(x) + \sum_{j=1}^{m} u_{ij} g_j(x), \quad (4.2)
\]

for some point \( u_i = (u_{i1}, \ldots, u_{im}) \in U \).
The observation space shall be better understood with the notion of integral curve. Given a nonlinear system

\[ \dot{z} = Z(z), \]

defined by some vector field \( Z \) on the state space \( M \), the integral curve \( \sigma_{z_0}(t) \) is the solution of the system satisfying the initial condition \( \sigma_{z_0}(0) = z_0 \). For every bounded subset \( M_1 \subset M \), there exists an interval \( (t_1, t_2) \ni 0 \) on which the integral curve \( \sigma_{z_0}(t) \) is well-defined for all \( t \in (t_1, t_2) \). This allows us to introduce on \( M_1 \) a set of maps, called the flow,

\[ Z^t : M_1 \to M, \quad t \in (t_1, t_2), \quad z_0 \mapsto \sigma_{z_0}(t). \]

Now choose inputs of system (4.1) such that it is driven by a sequence of vector fields \( Z_1, \ldots, Z_l \) of form (4.2) for small time \( t_1, \ldots, t_l \), respectively. The outputs of the system at time \( t_1 + \cdots + t_l \) are

\[ h_i(Z_{t_l}^l \circ Z_{t_{l-1}}^{l-1} \circ \cdots \circ Z_{t_1}^1(x_0)), \quad \text{for } i = 1, \ldots, k. \]

Differentiate these outputs sequentially with respect to \( t_l, t_{l-1}, \ldots, t_1 \) at \( t_l = 0, t_{l-1} = 0, \ldots, t_1 = 0 \) yields \( L_{Z_1}L_{Z_2} \cdots L_{Z_l}h_i(x_0) \), for \( i = 1, \ldots, k \). Hence we see that the observation space in fact consists of the output functions and their derivatives along all possible system trajectories (in infinitesimal time).

The *observability codistribution* at state \( x \in M \), denoted by \( dO(x) \), is defined as:

\[ dO(x) = \text{span}\{dH(x) \mid H \in \mathcal{O}\}. \]

**Theorem 13 (Hermann and Krener)** System (4.1) is locally observable at state \( x_0 \in M \) if \( \dim dO(x_0) = n \).

The equation \( \dim dO(x_0) = n \) is called the *observability rank condition*. Proofs of the above theorem can be found in [68] and [110, pp. 95–96]. Basically, to distinguish between a state and any other state in its neighborhood, it is necessary to consider not only the output functions but also their derivatives along all possible system trajectories. The rank condition ensures the existence of \( n \) output functions and/or derivatives that together define a diffeomorphism on some neighborhood of the state, which in turn ensures that the state is locally distinguishable.

### 4.2.2 Nonlinear Observers

An *observer* of a nonlinear system is a new system whose state estimates the state of the original system by converging to it. The input of the observer consists of the input as well as the output of the original system. A number of different approaches exist for constructing nonlinear observers.
4.2. NONLINEAR CONTROL

State Space Linearization

Consider the nonlinear system (4.1) again. One approach of observer design is to locally linearize the system using state space transformation. More specifically, in some neighborhood of a state \( x_0 \), the approach looks for a change of coordinates \( z = Z(x) \) with \( Z(x_0) = 0 \) that will transform the system into a linear system

\[
\dot{z} = Az + Bu, \quad y = Cz, \quad (4.3)
\]

where \( A, B, C \) are matrices of dimensions \( n \times n, n \times m, \) and \( k \times n, \) respectively, and satisfy the Kalman rank condition for observability\(^3\)

\[
\text{rank} \begin{pmatrix} C \\ CA \\ \vdots \\ CA^{n-1} \end{pmatrix} = n.
\]

For the linear system (4.3) a Luenberger observer \([95]\) is known to exist in the form of

\[
\dot{\hat{z}} = (A - KC)\hat{z} + Ky + Bu,
\]

where \( K \) is some constant \( n \times k \) matrix; and its error of estimate \( e = \hat{z} - z \) satisfies the nonlinear system

\[
\dot{e} = (A - KC)e. \quad (4.4)
\]

Since \( A \) and \( C \) satisfy the Kalman rank condition, \( K \) can be chosen such that \( A - KC \) has all of its eigenvalues in the open left half-plane; in this case \( \lim_{t \to \infty} e(t) = 0 \). In fact, any arbitrary exponential decay of the error \( e \) can be achieved with a proper choice of \( K \).

The following theorem offers the necessary and sufficient conditions under which such coordinate transformation exists.

**Theorem 14 (Nijmeijer)** Consider the nonlinear system (4.1) near a point \( x_0 \) with \( f(x_0) = 0 \) and \( h(x_0) = 0 \). There exists a coordinate transformation of the system into a minimal (controllable \([68]\) and observable) linear system if and only if the following three conditions hold on a neighborhood \( V \) of \( x_0 \):

1. \( \text{dim} \left( \text{span}\{ \text{ad}_j^i g_i(x) \mid i = 1, \ldots, m, j = 0, \ldots, n - 1 \} \right) = n \), for all \( x \in V \).

2. \( \text{rank} \begin{pmatrix} \frac{dh(x)}{dx} \\ \frac{dL_f h(x)}{dx} \\ \vdots \\ \frac{dL_f^{n-1} h(x)}{dx} \end{pmatrix} = n \), for all \( x \in V \).

---

\(^3\)The Kalman rank condition for a linear system is the counterpart of the observability rank condition for a nonlinear system as given in Theorem 13.
3. \( L_X, \ldots L_X, h_j(x) = 0 \), for all \( x \in V \), \( j = 1, \ldots, k \), \( l \geq 2 \) and \( X_1, \ldots, X_l \in \{ f, g_1, \ldots, g_m \} \) with at least two \( X_i \)'s different from \( f \).

Here \( L_j h(x) = (L_j h_1(x), \ldots, L_j h_k(x))^T \) is a slight abuse of notation. This theorem still holds without the assumptions \( f(x_0) = 0 \) and \( h(x_0) = 0 \) except that the linear dynamics and output equations would be \( \dot{z} = Az + Bu + c \) and \( y = Cz + d \) instead, for some constant vectors \( c \) and \( d \).

**Nonlinear Output Injection**

The method transforms the original nonlinear system into a linear system modulo a nonlinear output injection. More specifically, this approach looks for a change of coordinates on a neighborhood \( V \) of some state \( x_0 \):

\[
\begin{align*}
    z &= Z(x, u), \\ 
    w &= W(y),
\end{align*}
\]

under which system (4.1) will be transformed into

\[
\begin{align*}
    \dot{z} &= Az + \gamma(y, u), \\ 
    w &= Cz.
\end{align*}
\]

For system (4.6), a Luenberger observer exists in the form of

\[
\dot{\hat{z}} = (A - KC)\hat{z} + Kw + \gamma(y, u)
\]

with the same error dynamics (4.4). In other words, the new coordinates make it possible to “read off” the dynamics of the observer.

In particular, the following theorem defines the class of autonomous systems with single outputs that can be subjected to linearization by output injection.

**Theorem 15 (Krener and Isidori)** A single output nonlinear system

\[
\begin{align*}
    \dot{x} &= f(x) \\ 
    y &= h(x)
\end{align*}
\]

around some state \( x_0 \) is equivalent to a system of the form

\[
\begin{align*}
    \dot{z} &= Az + \gamma(y) \\ 
    y &= Cz + d
\end{align*}
\]

under a change of coordinates \( z = S(x) \), where \((C, A)\) is observable and \( d \) is some constant, if and only if the following two conditions hold on a neighborhood of \( x_0 \):

1. \( \dim \left( \text{span} \{dh(x), L_f dh(x), \ldots, L_f^{n-1} dh(x)\} \right) = n \), where \( n \) is the state space dimension,
2. the vector field $g$ defined by

$$L_g L^k_f(h) = \begin{cases} 0, & 0 \leq k < n - 1, \\ 1, & k = n - 1, \end{cases}$$

satisfies

$$[g, \text{ad}^k_f](x) = 0, \quad k = 1, 3, \ldots, 2n - 3.$$ 

An Observer Based on Lyapunov-like Equations

This result is due to Gauthier, Hammouri, and Othman [55].

**Theorem 16 (Gauthier, Hammouri, and Othman)** Consider the single output nonlinear (and analytic) system

$$\begin{align*}
\dot{x} &= f(x), \\
y &= h(x),
\end{align*}$$

(4.8)

defined on $n$-dimensional state space manifold $M$. Suppose the following two conditions hold:

1. the mapping $Z : x \mapsto z = \begin{pmatrix} h(x) \\ L_f h(x) \\ \vdots \\ L_f^{n-1} h(x) \end{pmatrix}$ is a diffeomorphism on $M$,

2. $L^n_f h(x)$ can be extended from $M$ to $\mathbb{R}^n$ by a $C^\infty$ function that is globally Lipschitzian on $\mathbb{R}^n$.

Let $C = (1, 0, \ldots, 0)$. Let $A = (a_{ij})$ be an $n \times n$ matrix with $a_{ij} = 1$ if $i = j - 1$ and 0 if $i \neq j - 1$, and $S_\infty(\zeta)$ be the $n \times n$ matrix that satisfies the equation

$$-\zeta S_\infty - A^T S_\infty - S_\infty A + C^T C = 0,$$

(4.9)

where $\zeta$ is some large enough constant. Then the system

$$\begin{align*}
\dot{x} &= f(\bar{x}) - \left(h(\bar{x}) - y\right) \frac{\partial Z^{-1}}{\partial z} \left(Z(\bar{x})\right) S^{-1}_\infty C^T
\end{align*}$$

(4.10)

is an observer for (4.8) with error dynamics

$$\|\bar{x}(t) - x(t)\| \leq K(\zeta) e^{-\frac{\zeta}{4}} \|\bar{x}(0) - x(0)\|,$$

where $K(\zeta)$ is some constant.
The proof of the above theorem given in [55] is based on standard Lyapunov arguments. The parameter $\zeta$ controls the speed of the observer. The matrix $S_\infty(\zeta) = (s_{ij})$ is the limit of the stationary solution of $\dot{S}_t(\zeta) = -\zeta S_t(\zeta) - A^T S_t(\zeta) - S_t(\zeta)A + C^T C$ as $t \to \infty$, with the initial value $S_0(\zeta)$ being any symmetric positive definite matrix.\(^4\) The matrix $S_\infty$ can be determined by starting from its first row and column simultaneously and progressing to higher ordinal pairs of rows and columns. We observe $s_{11} = -\frac{1}{\zeta}$ and let $s_{0j} = s_{j0} = 0$, for $j = 1, 2, \ldots, n$. Then the remaining entries of $S_\infty$ satisfy a three-term recurrence relation:

$$s_{ij} = -\frac{1}{\zeta}(s_{i-1,j} + s_{i,j-1}), \quad i > 1 \text{ or } j > 1.$$  

The observer (4.10) is a copy of the original system (4.8) with a corrective term that does not depend on system (4.8) but only on the dimension and the desired speed $\zeta$.

The GHO observer for a general nonlinear system (4.1) with inputs is a copy of the original system plus the error corrective term given in (4.10). To have such an observer, not only must conditions 1 and 2 in the above theorem hold for the drift system $\dot{x} = f(x)$, but also the original system must be observable for any input.

---

\(^4\)So is $S_t(\zeta)$ for $t > 0$ symmetric positive definite.
Chapter 5

Pose and Motion from Contact

This chapter will study sensing approaches that take advantage of not only known geometry but also task mechanics as well. More specifically, we would like to answer the following questions:

1. *Can we determine the pose of an object with known geometry and mechanical properties from the contact motion on a single pushing finger, or simply, from a few intermediate contact positions during the pushing?*

2. *Can we determine any intermediate pose of the object during the pushing?*

3. *Furthermore, can we estimate the motion of the object during the pushing?*

We will give affirmative answers to the above questions in the general case. To accomplish this, we will characterize pushing as a system of nonlinear differential equations based on its dynamics. As shown in Figure 5.1, the state of the system will include the configurations (positions, orientations, and velocities) of the finger and object during the pushing at any time instant. The system input will be the acceleration of the finger. The system output will be the contact location on the finger subject to the kinematics of contact. This output will be fed to nonlinear observers, which server as the sensing algorithms, to estimate the object’s pose and motion.

Section 5.1 copes with the dynamics of pushing and the kinematics of contact, deriving a system of differential equations that govern the object and contact motions and resolving related issues such as friction and initial object motion; Section 5.2 applies nonlinear control theory to verify the soundness of our sensing approach to be proposed, establishing the local observability of this dynamical pushing system from the finger contact; Section 5.3 describes two nonlinear observers which estimate the object pose (and motion) at any instant and at the start of pushing, respectively, and which require different amounts of sensor data; Section 5.4 extends the results to incorporate contact friction between the finger and the object.

In this chapter, the cross product of two vectors (e.g., $\alpha \times v$) is treated as a scalar wherever ambiguity would not arise. A scalar in a cross product (e.g., the angular velocity
\( \omega \) in \( \omega \times \beta \) acts as a vector of equal magnitude and orthogonal to the plane. To avoid any ambiguity, the notation \( \dot{\cdot} \) means differentiation with respect to time, while the notation \( \dot{\cdot}' \) means differentiation with respect to some curve parameter. For example, \( \dot{\alpha} = \alpha' \dot{u} = \frac{d\alpha}{du} \frac{du}{dt} \) gives the velocity of a point moving on a curve \( \alpha(u) \).

5.1 Motion of Contact

Throughout the chapter we consider the two-dimensional problem of a translating finger \( \mathcal{F} \) pushing an object \( \mathcal{B} \). The coefficient of support friction, that is, friction between \( \mathcal{B} \) and the plane, is everywhere \( \mu \). For simplicity, let us assume uniform mass pressure distributions of \( \mathcal{B} \). Let us also assume frictionless contact between \( \mathcal{F} \) and \( \mathcal{B} \) at present and deal with contact friction exclusively in Section 5.4. Let \( \mathbf{v}_\mathcal{F} \) be the velocity of \( \mathcal{F} \), known to \( \mathcal{F} \)'s controller, \( \mathbf{v} \) and \( \omega \) the velocity and angular velocity of \( \mathcal{B} \), respectively, all in the world coordinate frame (Figure 5.2).

Let \( \mathcal{F} \)'s boundary be a smooth curve \( \alpha \) and \( \mathcal{B} \)'s boundary be a piecewise smooth closed curve \( \beta \) such that \( \alpha(u) \) and \( \beta(s) \) are the two points in contact in the local frames of \( \mathcal{F} \) and \( \mathcal{B} \), respectively. Following convention, moving counterclockwise along \( \alpha \) and \( \beta \) increases \( u \) and \( s \), respectively. Assume that one curve segment of \( \beta \) stays in contact with \( \alpha \) throughout the pushing.

That \( \mathcal{F} \) and \( \mathcal{B} \) maintain contact imposes a velocity constraint:

\[
\mathbf{v}_\mathcal{F} + \alpha' \dot{u} = \mathbf{v} + \omega \times R\beta + R\beta' \dot{s},
\]  

(5.1)
where $R(\theta) = \begin{pmatrix} \cos \theta & -\sin \theta \\ \sin \theta & \cos \theta \end{pmatrix}$ is the rotation matrix associated with the orientation $\theta$ of $B$, which is determined by the orientation of $F$, $u$, and $s$. Newton’s and Euler’s equations on rigid body dynamics are stated as:

\[ F + \int_B -\mu \eta g \hat{v}_p \, dp = m \dot{v}, \]  
\[ R\beta \times F + \int_B Rp \times (-\mu \eta g \hat{v}_p) \, dp = I \dot{\omega}, \]

where $F$ is the contact force acting on $B$, $g$ the acceleration of gravity, $m$ the mass, $\eta$ the mass density, and $I$ the angular inertia about $O$ (all of $B$). Here $v_p = v + \omega \times Rp$ is the velocity of $p \in B$ and $\dot{v}_p = v_p/\|v_p\|$ its direction.\(^1\)

With no friction at the contact point, $F$ acts along the inward normal of $B$:

\[ F \cdot R\beta' = 0; \]
\[ R\beta' \times F > 0. \]  

Finally, the normals of $F$ and $B$ at the contact are opposite to each other; equivalently, we have

\[ \alpha' \times R\beta' = 0, \]

and $\alpha' \cdot R\beta' < 0$ (which always holds).

Given the finger motion $v_F$, there are seven equations (5.1), (5.2), (5.3), (5.4), and (5.6) with seven variables $u, s, \omega, v$, and $F$.\(^2\) From these equations, we are now ready to derive the differential equations for $u, s, \omega, v$.  

---

\(^1\)That $F$ is translating implies either $v \neq 0$ or $\omega \neq 0$ after the pushing starts. So $v_p$ can vanish over at most one point $p \in B$, which will vanish in the integrals in equations (5.2) and (5.3).

\(^2\)Note that equations (5.1) and (5.2) and variables $v$ and $F$ are each counted twice.
Let $\mathbf{a}_\mathcal{F}$ be the acceleration of $\mathcal{F}$, $A = \int_B d\mathbf{p} = \frac{m}{\eta}$ and $\rho = \sqrt{\frac{I}{m}}$ the area and radius of gyration of $\mathcal{B}$, respectively, and $\Gamma = \int_B R\beta' \times (R\mathbf{p} \times \hat{v}_p) + (\beta' \cdot \beta)\hat{v}_p d\mathbf{p}$ an integral associated with support friction. We have

**Theorem 17** Consider the pushing system described by (5.1)–(5.6). In the non-degenerate case the points of contact evolve according to

\[
\dot{u} = \frac{- (\alpha' \cdot R\beta')^2 \omega + (\alpha' \times R\beta') (\alpha' \cdot (\mathbf{v} + \omega \times R\beta - \mathbf{v}_\mathcal{F}))}{(\alpha' \cdot R\beta')(\alpha'' \times R\beta') + \|\alpha'\|^2(\alpha' \times R\beta')},
\]

(5.7)

\[
\dot{s} = \frac{- \|\alpha'\|^2 (\alpha' \cdot R\beta') \omega + (\alpha'' \times R\beta') (\alpha' \cdot (\mathbf{v} + \omega \times R\beta - \mathbf{v}_\mathcal{F}))}{(\alpha' \cdot R\beta')(\alpha'' \times R\beta') + \|\alpha'\|^2(\alpha' \times R\beta')},
\]

(5.8)

and the object’s angular acceleration and acceleration are

\[
\dot{\omega} = \frac{\dot{\mathbf{a}}' \times (\mathbf{v}_\mathcal{F} - \mathbf{v}) + \mathbf{a}' \times \mathbf{a}_\mathcal{F} - (\dot{\mathbf{a}}'' \cdot R\beta + \mathbf{a}' \cdot (\omega \times R\beta + R\beta') \hat{s}) \omega + \frac{\mu g}{A\beta' \cdot \beta} \alpha' \times \Gamma}{\alpha' \cdot R(\beta + \frac{\rho^2}{\beta' \cdot \beta'})},
\]

(5.9)

\[
\dot{\mathbf{v}} = \frac{A\rho^2 \hat{\omega} \times R\beta' - \mu g \Gamma}{A\beta' \cdot \beta}.
\]

(5.10)

**Proof** Taking the dot products of $\alpha'$ with both sides of (5.1) and rearranging terms thereafter, we obtain

\[
\|\alpha'\|^2 \dot{u} - (\alpha' \cdot R\beta') \dot{s} = \alpha' \cdot (\mathbf{v} + \omega \times R\beta - \mathbf{v}_\mathcal{F}).
\]

Next differentiate both sides of (5.6):

\[
(\alpha'' \times R\beta') \dot{u} + (\alpha' \times R\beta'') \dot{s} + (\alpha' \cdot R\beta') \omega = 0.
\]

Immediately, we solve for $\dot{u}$ and $\dot{s}$ from the two equations above and obtain (5.7) and (5.8).

Now we move on to derive the differential equations for $\mathbf{v}$ and $\omega$. First take the cross products of $R\beta'$ with both sides of (5.3), eliminating the resulting term that contains $F \cdot R\beta'$ and substituting (5.2) in after term expansion:

\[
-(\beta' \cdot \beta) m \dot{\mathbf{v}} - \mu g \Gamma = R\beta' \times I\hat{\omega}.
\]

Here the term

\[
\Gamma = \int_B R\beta' \times (R\mathbf{p} \times \hat{v}_p) + (\beta' \cdot \beta)\hat{v}_p d\mathbf{p},
\]

(5.11)

when multiplied by $\mu g$, combines the dynamic effects of friction. Thus we can write $\dot{\mathbf{v}}$ in the form of (5.10).

---

3The degenerate case will be classified and discussed in Section 5.1.1.
5.1. MOTION OF CONTACT

Taking the cross products of $\alpha'$ with both sides of (5.1) and cancelling the term $\alpha' \times R\beta'$ according to (5.6), we have after a few more steps of term manipulation

$$\alpha' \times (v_F - v) = (\alpha' \cdot R\beta)\omega.$$  \hfill (5.12)

Differentiating both sides of (5.12) yields

$$\dot{\alpha}'' \times (v_F - v) + \alpha' \times (a_F - \dot{v}) = \left(\dot{\alpha}\alpha' \cdot R\beta + \alpha' \cdot (\omega \times R\beta + R\beta' \dot{s})\right)\omega + (\alpha' \cdot R\beta)\dot{\omega}. \hfill (5.13)$$

Finally, substituting (5.10) in (5.13) gives us (5.9).

Substitute (5.7) and (5.8) into (5.9) and the resulted differential equation into (5.10). We have thus obtained the differential equations of $\omega$ and $v$ which, along with (5.7) and (5.8), form a system of ordinary differential equations (ODEs). This system is numerically solvable for $u, s, \omega$, and $v$. Without any ambiguity, we also let (5.9) and (5.10) refer to their corresponding differential equations.

The motion of $B$ is independent of its mass density $\eta$, as seen from (5.7), (5.8), (5.9), and (5.10) or directly from (5.2) and (5.3).

If $\alpha$ and $\beta$ are convex unit-speed curves with curvatures $\kappa_\alpha$ and $\kappa_\beta$ at the contact point, respectively, such that $\kappa_\alpha + \kappa_\beta \neq 0$, then equations (5.7) and (5.8) are simplified to

$$\dot{u} = \frac{\omega + \kappa_\beta \alpha' \cdot (v + \omega \times R\beta - v_F)}{\kappa_\alpha + \kappa_\beta}; \hfill (5.14)$$

$$\dot{s} = -\frac{\omega + \kappa_\alpha \alpha' \cdot (v + \omega \times R\beta - v_F)}{\kappa_\alpha + \kappa_\beta}. \hfill (5.15)$$

For example, let $\alpha$ be a circle with radius $r$ and $\beta$ a polygon. Hence $\kappa_\alpha = \frac{1}{r}$ and $\kappa_\beta = 0$. We have $\dot{u} = \frac{\omega}{\kappa_\alpha} = \omega r$. During a push of time $\Delta t$, the contact moves an arc of length

$$\int_0^{\Delta t} \dot{u} \, dt = r \int_0^{\Delta t} \omega \, dt = r \Delta \theta$$

on $\alpha$, which can be immediately verified from the tangency between $\alpha$ and $\beta$.

5.1.1 Degenerate Case

Our derivation of the differential equations (5.7)–(5.10) is correct only if the denominators on their right hand sides do not vanish. By parameterizing $\alpha$ and $\beta$ as unit-speed curves with curvatures $\kappa_\alpha$ and $\kappa_\beta$, respectively, we can easily verify that the denominator in (5.7) and (5.8) vanishes only when one of the curves is concave at the contact with $\kappa_\alpha = \kappa_\beta$, a situation that almost never happens.

\footnote{The term $\kappa_\alpha + \kappa_\beta$ below is the 2-dimensional case of the relative curvature form introduced by Montana [106].}

\footnote{Here $\kappa_\beta$ is the curvature of the polygon edge in contact with $\alpha$. We assume that the finger will not be in contact with any vertex during the pushing.}
The vanishing of the denominator \( \beta' \cdot \beta \) on the right in (5.10) implies that the contact force \( F \) passes through the center of mass of \( B \), yielding zero torque. If \( \beta' \cdot \beta \neq 0 \), the denominator in (5.9) does not vanish for we have

\[
\beta' \cdot \beta = 0 \Rightarrow \beta' \cdot \beta + \frac{\beta'^2}{\beta' \cdot \beta} \beta' \neq 0
\]

\[
\Rightarrow \beta' \cdot \left( \beta + \frac{\beta'^2}{\beta' \cdot \beta} \beta' \right) \neq 0
\]

\[
\Rightarrow R\beta' \cdot R \left( \beta + \frac{\beta'^2}{\beta' \cdot \beta} \beta' \right) \neq 0
\]

\[
\Rightarrow \alpha' \cdot R \left( \beta + \frac{\beta'^2}{\beta' \cdot \beta} \beta' \right) \neq 0;
\]

Hence \( \beta' \cdot \beta = 0 \), or equivalently, \( R\beta \times F = 0 \), remains the only degenerate condition.

**Corollary 18** In the pushing system described by (5.1), (5.2), (5.3), (5.4), (5.5), and (5.6), the object's angular acceleration and acceleration in the degenerate case \( \beta' \cdot \beta = 0 \) are given as

\[
\dot{\omega} = -\frac{\mu g \int_B R\dot{p} \times \dot{v}_p \, dp}{A \beta^2},
\]

\[
\dot{v} = -\left( R\beta' \times \left( \dot{v}_\alpha'' \times (v_F - v) + \alpha' \times a_F - (\dot{\alpha} \alpha'' \cdot R\beta + \alpha' \cdot (\omega \times R\beta + R\beta' \dot{s})) \right) \right) + \frac{\mu g \int_B \dot{v}_p \, dp \cdot R\beta'}{A \alpha'}. \tag{5.17}
\]

**Proof** Equation (5.16) follows directly from (5.3) and \( R\beta \times F = 0 \). Taking the dot products of \( R\beta' \) with both sides of (5.2) we obtain

\[
\dot{v} \cdot R\beta' = -\frac{\mu g \int_B \dot{v}_p \, dp \cdot R\beta'}{A}.
\]

Taking the cross products of \( R\beta' \) with both sides of (5.13) and substituting the above equation and \( \alpha' \cdot R\beta = 0 \) in yields (5.17).

It is not difficult to verify that the limits of \( \dot{\omega} \) and \( \dot{v} \) given by equations (5.9) and (5.10), respectively, as \( \beta' \cdot \beta \to 0 \), are equal to their degenerate forms given by equations (5.16) and (5.17), respectively.

**5.1.2 Integral of Support Friction**

To numerically integrate (5.7)–(5.10), it is necessary to evaluate the integral \( \Gamma \) given by (5.11) which represents the role of support friction in dynamics. Two-dimensional numerical integration of \( \Gamma \) is usually very slow. However, by choosing proper polar coordinates we can
5.1. MOTION OF CONTACT

Figure 5.3: The instantaneous rotation center \((-v_y^B/\omega, v_x^B/\omega)^T\) of object \(B\) with velocity \((v_x^B, v_y^B)^T\) and angular velocity \(\omega \neq 0\) (about its center of mass \(O\)). The integral \(\Gamma\) is evaluated in polar coordinates with respect to the i.r.c..

reduce the evaluation of \(\Gamma\) to one variable integration, and if \(B\) is polygonal, obtain the closed form of \(\Gamma\).

When the motion of \(B\) is pure translation \((\omega = 0)\), the evaluation is easy:

\[
\Gamma = (\beta' \cdot \beta) A \hat{v}.
\]  

(5.18)

So we focus our discussion on the case \(\omega \neq 0\). The integral \(\Gamma\) can be evaluated in the polar coordinates with respect to the instantaneous rotation center of \(B\) introduced below.

Let us first express \(\Gamma\) in terms of \(B\)'s moving body frame at its center of mass \(O\):

\[
\Gamma = R \beta' \times \int_B p \times \hat{v}_p^B \, dp + (\beta' \cdot \beta) R \int_B \hat{v}_p^B \, dp,
\]  

(5.19)

where \(v_p^B = R^{-1}v + \omega \times p\) is the velocity at \(p \in B\) in the body frame. At the moment, \(B\) is rotating about the point \((x_0^B, y_0^B)^T = \omega \times (v_x^B, v_y^B)^T / \omega^2 = (-v_y^B / \omega, v_x^B / \omega)^T\), called the instantaneous rotation center (i.r.c.), as shown in Figure 5.3. For convenience and clarity, we only illustrate the case where \(B\) is convex. The evaluation should be straightforwardly generalized to the case where \(B\) is concave.

Any ray at angle \(\phi\) from the i.r.c. has at most two intersections \((\phi, r_1(\phi))^T\) and \((\phi, r_2(\phi))^T\), \(r_1(\phi) < r_2(\phi)\), with the object boundary. Every point \(p\) on the ray is moving along the same direction \(\hat{v}_p^B = (-\sin \phi, \cos \phi)^T\) if \(\omega > 0\), or along the direction \(\hat{v}_p^B = (\sin \phi, -\cos \phi)^T\) if \(\omega < 0\). The two subintegrals given in (5.19) now reduce to one-variable integrals in the polar coordinates:

\[
I_1 = \int_B \hat{v}_p^B \, dp
\]  

\(^6\)If the i.r.c. is in the interior of \(B\), let \(\phi_1 = 0, \phi_2 = 2\pi\), and \(r_1(\phi) = 0\).
\[ I_2 = \int_B \mathbf{p} \times \dot{\mathbf{v}}_p \, dp. \]

\[ = \int_{\phi_1}^{\phi_2} \int_{r_1(\phi)}^{r_2(\phi)} \left( x_0^B \cos \phi + y_0^B \sin \phi \right) \left( r_2^2(\phi) - r_1^2(\phi) \right) \frac{2}{3} \, d\phi \]

For polygonal shapes, the closed forms of \( I_1 \) and \( I_2 \) are given in Appendix C; for most other shapes, these integrals can only be evaluated numerically.

### 5.1.3 Initial Motion

In order to numerically integrate equations (5.7), (5.8), (5.9), (5.10), it is necessary to determine the initial accelerations \( \dot{\mathbf{v}} \) and \( \dot{\omega} \) of \( B \) from the finger acceleration \( \mathbf{a}_F \) and the configurations of \( F \) and \( B \).

At the start of pushing, both the finger \( F \) and the object \( B \) are motionless; that is, we have

\[ \mathbf{v}(0) = \mathbf{v}_0 = 0, \quad \omega(0) = \omega_0 = 0, \quad \text{and} \quad \mathbf{v}_F(0) = 0. \]

Plugging the above into (5.7) and (5.8) yields the initial velocities of the contact points:

\[ \dot{u}(0) = 0 \quad \text{and} \quad \dot{s}(0) = 0. \]

In the degenerate case where the contact normal \( N \) passes through the center of mass \( O \), the initial angular acceleration and acceleration follow easily from (5.16) and (5.17), respectively:

\[ \dot{\omega}_0 = -\frac{\mu g \int_B \mathbf{p} \times N \, dp}{A \rho^2} \]

\[ = -\frac{\mu g \int_B \mathbf{p} \, dp \times N}{A \rho^2} \]

\[ = 0; \]

\[ \dot{\mathbf{v}}_0 = -\frac{R \beta' \times (\alpha' \times \mathbf{a}_F) + \mu g (N \cdot R \beta') \alpha'}{\alpha' \cdot R \beta'} \]

\[ = \mathbf{a}_F - \frac{\mathbf{a}_F \cdot R \beta'}{\alpha' \cdot R \beta'} \alpha'. \]

\[ ^7\text{In this section, our focus is on simulating the start of pushing. Hence we temporarily assume the initial configuration of } B \text{ is known. Later in Section 5.3.2, we will see how to use such simulation to solve for the initial pose of } B \text{ using Newton’s method.} \]
5.1. MOTION OF CONTACT

Here we write $\dot{\omega}(0) = \dot{\omega}_0$ and $\dot{v}(0) = \dot{v}_0$.

In the non-degenerate case, we have $\dot{\omega}_0 \neq 0$. The frictional force $f_p$ at point $p \in B$ is opposed to the direction of relative motion [59], which, at the start of pushing, is the direction of the acceleration

$$\dot{v}_p(0) = \dot{v}_0 + \dot{\omega}_0 \times p + \omega_0 \times (\omega_0 \times p)$$

$$= \dot{v}_0 + \dot{\omega}_0 \times p$$

$$= \dot{\omega}_0 \left( \frac{\dot{v}_0}{\omega_0} + 1 \times p \right).$$

By a simple argument, the sign of $\dot{\omega}_0$ must agree with its sign were there no friction; hence it is easily determined. Consequently, $\hat{\dot{v}}_p(0)$, $f_p$, and

$$\Gamma_0 = R\beta' \times \int_B R p \times \hat{\dot{v}}_p(0) \, dp + (\beta' \cdot \beta) \int_B \hat{\dot{v}}_p(0) \, dp$$

become functions of $\frac{\dot{v}}{\omega_0}$. Thus (5.10) can be rewritten as

$$\dot{v}_0 = \frac{A\rho^2 \dot{\omega}_0 \times R\beta' - \mu g \Gamma_0(\frac{\dot{v}}{\omega_0})}{A\beta' \cdot \beta} \quad (5.21)$$

at $t = 0$. Meanwhile, it follows from (5.13) that

$$\dot{\omega}_0 = \frac{\alpha' \times a_F}{\alpha' \cdot R\beta + \alpha' \times \frac{\dot{v}_0}{\omega_0}} \quad (5.22)$$

Dividing both sides of (5.21) by $\dot{\omega}_0$ and substituting (5.22) in, we get the following equation in $\frac{\dot{v}_0}{\omega_0}$:

$$\frac{\dot{v}_0}{\dot{\omega}_0} = \frac{A\rho^2 \times R\beta' - \mu g \Gamma_0(\frac{\dot{v}}{\omega_0}) - \frac{\alpha' \cdot R\beta + \alpha' \times \frac{\dot{v}_0}{\omega_0}}{\alpha' \cdot a_F}}{A\beta' \cdot \beta} \quad (5.23)$$

Equation (5.23) is solvable for $\frac{\dot{\omega}_0}{\omega_0}$ by Newton’s method using the value of $\frac{\dot{v}}{\omega_0}$ for $\mu = 0$, obtainable directly from (5.9) and (5.10), as an initial estimate. Intuitively, the method iterates until the computed acceleration and angular acceleration agree with what would be yielded under Newton’s law by the finger acceleration and the frictional force, the latter of which in turn depends on the accelerations themselves under Coulomb’s law of friction.

The partial derivative $\partial \Gamma_0 / \partial \frac{\dot{\omega}_0}{\omega_0}$ required for the iterations can be evaluated numerically or using its closed form given in Appendix C.2 when $B$ is polygonal. Hence $\dot{\omega}_0$ and $\dot{v}_0$ are determined from (5.21) and (5.22).
5.1.4 Contact Breaking

The only constraint that was left out in the derivation of the differential equations (5.7)–(5.10) is inequality (5.5). This constraint, however, is used for checking when the contact between the finger and the object breaks. More specifically, the contact breaks when $R_{\beta'} \times F < 0$.

5.2 Local Observability

In the previous section we saw that the kinematics of contact and the dynamics of pushing are together determined by a system of nonlinear ordinary differential equations (5.7)–(5.10). A state of this nonlinear system consists of $u$ and $s$, which determine the contact locations on the finger and on the object, respectively, the object’s angular velocity $\omega$, velocity $v$, and orientation $\theta$; the input is the finger’s acceleration $a_F$, generated by the controller of the finger; and the output is $u$, reported by a tactile sensor mounted on the finger. The sensing task becomes to “observe” $s$ from $u$, which, as suggested by the system equations, is no easier than to “observe” the whole state of the system.

In this section we shall study local observability of one instantiation of the above system in which the finger is circular and the object is polygonal. This type of pushing is representative in real manipulation tasks. First of all, we introduce the notion of nonlinear observability as well as a theorem about local observability; next, we show that the instantiation is locally observable. It will then not be difficult to see that these results can generalize to many other finger and object shapes.

5.2.1 The Disk-Polygon System

Now we study the case in which finger $\mathcal{F}$ is a disk bounded by $\alpha = r(\cos \frac{u}{r}, \sin \frac{u}{r})^T$ and object $\mathcal{B}$ is a simple polygon. The interior of one edge $e$ of $\mathcal{B}$ maintains contact with $\mathcal{F}$ throughout the pushing.\(^8\) We assume that $e$ is known since local observability is concerned, and since a sensing strategy can hypothesize all edges of $\mathcal{B}$ as the contact edge and verify them one by one. Let $h$ be the distance from the centroid $O$ of $\mathcal{B}$ to $e$. Choose $s$ as the signed distance from the contact to the intersection of $e$ and its perpendicular through $O$ such that $s$ increases monotonically while moving counterclockwise (with respect to $\mathcal{B}$’s interior) on $e$. See Figure 5.4. The orientation of $\mathcal{B}$ is $\theta = u/r - \pi/2$.\(^9\) The tangent and normal of $\mathcal{F}$ at the contact are $T = \alpha' = (-\sin \frac{u}{r}, \cos \frac{u}{r})^T$ and $N = r\alpha'' = -(\cos \frac{u}{r}, \sin \frac{u}{r})^T$, respectively. The system is governed by the following nonlinear equations as special cases of (5.7)–(5.10),

---

\(^8\)This is easily realizable in a real pushing scenario.

\(^9\)Given a different contact edge $e_1$ it follows $\theta = u/r - \pi/2 + \theta_{e_1}$ for some constant $\theta_{e_1}$. 
5.2. LOCAL OBSERVABILITY

Figure 5.4: A circular finger pushing a polygonal object.

respectively:\textsuperscript{10,11}

\begin{align*}
\dot{u} &= \omega r, \\
\dot{s} &= T \cdot (v - v_F) - \omega(r + h), \\
\dot{\omega} &= \frac{s}{s^2 + \rho^2} \left( \omega^2(r + h) - 2\omega T \cdot (v - v_F) - T \times a_F \right) - \frac{\mu g}{A(s^2 + \rho^2)} T \times \Gamma, \\
\dot{v} &= \frac{s}{s^2 + \rho^2} \left( N \cdot a_F - \omega^2(r + h) + 2\omega T \cdot (v - v_F) \right) N - \frac{\mu g}{As} (T \cdot \Gamma) T \\
&\quad - \frac{s}{s^2 + \rho^2} \frac{\mu g}{A} (N \cdot \Gamma) N,
\end{align*}

where

\[ \Gamma = sR_I + I_2 N \]  \hspace{1cm} (5.25)

is the integral of friction reduced from (5.19).

Note that (5.24) in fact subsumes the degenerate case \((s = 0)\) discussed in Section 5.1.1. From (5.24) we have at \(s = 0\)

\[ \dot{\omega} = -\frac{\mu g}{A\rho^2} I_2, \]

\[ \dot{v} = -\left( \omega^2(r + h) - 2\omega T \cdot (v - v_F) - T \times a_F \right) N - \frac{\mu g}{A} (T \cdot R_I) T, \]

which can also be derived from equations (5.16) and (5.17). We will refer to (5.24) and its future variations as the \textit{disk-polygon} system.

Of all the variables and constants in system (5.24), only the height \(h\) of the contact edge (from the polygon’s center of geometry) and the contact location \(s\) on the edge encode

\textsuperscript{10}These equations also apply to the degenerate case \(s = 0\) in Section 5.1.1.

\textsuperscript{11}These equations assume that \(O\) and the disk center are on different sides of \(e\). Otherwise the term \(r + h\) in the equations for \(\dot{s}, \dot{\omega}, \dot{v}\) need to be replaced by \(r - h\).
the geometry of the contact. Suppose the polygon assumes the degeneracy that two of its edges have the same height. Then every pair of points on these two edges, respectively, and with the same $s$ value, would result in exactly the same system behavior. The system cannot distinguish between such pair of contact points, or subsequently, the corresponding two different poses, just from the disk contact $u$.

The relative orientation of the polygon to the disk, determined by $u$, appears in the equations for $\dot{s}$, $\dot{\omega}$, and $\dot{v}$, thereby in both system kinematics and dynamics. The relative position, determined by $s$, however, appears only in the system dynamics. That $s$ does not directly affect the kinematics is due to that local geometry on the contact edge is everywhere the same, with zero curvature. However, this is not true for curved objects.

To apply Theorem 13 to show that system (5.24) is locally observable, we first need to rewrite it into the form (4.1) of an affine system. For convenience, we express $v$ in terms of the Frenet frame at the disk contact defined by the tangent $T$ and normal $N$: $v = (v_T, v_N)^T$, where $v_T = v \cdot T$ and $v_N = v \cdot N$. Also express the disk velocity $v_F$ and acceleration $a_F$ in the same frame as $(v_F^T, v_F^N)^T$ and $(a_F^T, a_F^N)^T$, respectively. We find that $v_N$ depends on $s$, $\omega$, and $v_F^N$ by taking the dot product of $N$ with the velocity constraint (5.1):

$$v_N = v_F^N + s\omega.$$  

From the above equation, $\frac{d\Gamma}{dt} = \omega N$, and (5.24) we have

$$\dot{v}_T = \frac{d(v \cdot T)}{dt} = v \cdot \frac{dT}{dt} + \dot{v} \cdot T$$

$$= \omega v_N - \frac{\mu g}{A_T} \Gamma_T$$

$$= \omega v_F^N + s\omega^2 - \frac{\mu g}{A_T} \Gamma_T,$$

where $\Gamma_T = \Gamma \cdot T$.

System (5.24) is now rewritten as

$$\dot{x} = f(x) + a_F^T g_T(x) + a_F^N g_N(x). \tag{5.26}$$

The state $x$ of the system becomes $(u, s, \omega, v_T, v_F^T, v_F^N)^T$ with six variables in total; the inputs are the acceleration components $a_F^T$ and $a_F^N$ along the contact tangent and normal, respectively; and the output is a triple $y = (u, v_F^T, v_F^N)^T$. The drift and input fields are given by

$$f(x) = \begin{pmatrix}
\omega r \\
\frac{s}{s^2 + \rho^2} \left( \omega^2 (r + h) - 2\omega (v_T - v_F^T) \right) - \frac{\mu g}{A(s^2 + \rho^2)} \Gamma_N \\
\omega v_F^N + s\omega^2 - \frac{\mu g}{A_T} \Gamma_T \\
0 \\
0
\end{pmatrix},$$

$$g_T(x) = \begin{pmatrix}
v_T - v_F^T - \omega(r + h) \\
0 \\
0
\end{pmatrix},$$

$$g_N(x) = \begin{pmatrix}
0 \\
0 \\
0
\end{pmatrix}.$$
5.2. LOCAL OBSERVABILITY

\[
\begin{align*}
g_T(x) &= (0,0,0,1,0)^T, \\
g_N(x) &= \left(0,0,-\frac{s}{s^2 + \rho^2},0,0,1\right)^T.
\end{align*}
\]

**Theorem 19** The disk-polygon system (5.26) is locally observable.

**Proof** By Theorem 13 it suffices to show that the observability codistribution \(dO\) has rank 6 at every state. Here the observation space \(O\) consists of the outputs \(u, v_{\mathcal{F}_T}, v_{\mathcal{F}_N}\) and their repeated Lie derivatives. We choose from \(O\) the following functions and write out their differentials:

\[
\begin{align*}
du &= (1,0,0,0,0,0); \\
dv_{\mathcal{F}_T} &= (0,0,0,0,1,0); \\
dv_{\mathcal{F}_N} &= (0,0,0,0,0,1);
\end{align*}
\]

\[
\begin{align*}
L_f u &= du \cdot f = \omega r, \\
L_g L_f u &= -r \frac{s}{s^2 + \rho^2}, \\
L_g L_g L_f u &= (r + h) \frac{s(s^2 - \rho^2)}{(s^2 + \rho^2)^3}, \\
L_g L_f L_g L_f u &\big|_{s=\rho} = \left(0, \frac{r(r + h)}{4\rho^4}, 0, 0, 0, 0\right).
\end{align*}
\]

Thus \(du, dv_{\mathcal{F}_T}, dv_{\mathcal{F}_N}, dL_f u,\) and \(dl_{\mathcal{F}_N} L_f L_g L_f u\) (or \(dl_{\mathcal{F}_N} L_f L_g L_f u\) when \(s = \rho\)) span the cotangent bundle of the space of all possible 5-tuples \((u, s, \omega, v_{\mathcal{F}_T}, v_{\mathcal{F}_N})^T\). It suffices to find one more function in \(O\) whose partial derivative with respect to \(v_T\) will not vanish.

Such a task is quite easy, for we have

\[
\begin{align*}
\frac{\partial(L_f L_g L_f u)}{\partial v_T} &= r \frac{s^2 - \rho^2}{(s^2 + \rho^2)^2}; \\
\frac{\partial}{\partial v_T}(L_f L_g L_f L_g L_f u) \bigg|_{s=\rho} &= \frac{r(r + h)}{4\rho^4}.
\end{align*}
\]

In summary, the observability codistribution \(dO\) is spanned by \(du, dv_{\mathcal{F}_T}, dv_{\mathcal{F}_N}, dL_f u, dl_g L_f u\) and \(dl_g L_g L_f u\) (or \(dl_g L_f L_g L_f u\) and \(dl_f L_g L_f L_g L_f u\) when \(s = \rho\)) and thus attains full rank.

The above proof in fact constructs several control sequences which, when applied for infinitesimal amounts of time, will distinguish between different states in any neighborhood. Assuming \(s \neq \eta\), one of the functions \(u, v_{\mathcal{F}_T}, v_{\mathcal{F}_N}, L_f u, L_g L_f u, L_f L_g L_f u\) and \(L_f L_g L_f u\) must have different values in any two different states close enough as guaranteed by the observability rank condition. Note that \(L_f u\) is in fact the differential output under zero control. Since
$L_{gN} L_f u$ may be written as $\frac{1}{2} L_{f+gN} L_f u - \frac{1}{2} L_{f-gN} L_f u$, one of these two functions must distinguish the two states if $L_{gN} L_f u$ does. Obviously, $L_{f+gN} L_f u$ (or $L_{f-gN} L_f u$) is realizable in an arbitrarily small amount of time by the control sequence starting with zero control and ending with $a_{\mathcal{F}_N} = 1$ (or $-1$).\footnote{Note $a_{\mathcal{F}_N}$ can be slightly negative without causing the break of contact. The value $-1$ is just for making this point but not necessarily small enough sometimes.} The case with function $L_f L_{gN} L_f u$ is similar.

Moreover, the proof reveals the relative “hardness” of observing the state variables, especially $u, s,$ and $\omega$. The disk contact $u$ constitutes the system output and thus is the easiest to observe. The angular velocity $\omega$ of the polygon needs to be obtained from the first order derivative of $u$. The polygon contact $s$, the hardest of the three to observe, requires a Lie derivative of the second order or above, which is obtained using two or more controls.

Support friction does not affect the local observability of the disk-polygon system, as none of the differentials chosen in the proof to span $d \mathcal{O}$ involve the integral $\Gamma$ or any of its partial derivatives.

The proof makes use of the input vector field $g_N$ but not $g_T$, which suggests that pushing along a tangential direction is unnecessary for the purpose of local observability. Intuition tells us that pushing along the contact normal will more likely helps the disk observe the polygon.

We can indeed treat the variables $v_{\mathcal{F}_T}$ and $v_{\mathcal{F}_N}$ as ordinary functions. This will reduce the state space dimension by 2. The above proof can easily carry over with a simple modification.

We conjecture that the autonomous version of the system (under $a_{\mathcal{F}} = 0$) is locally observable at all except a finite number of states. Although it seems much more difficult to prove the linear independence of $du, dv_{\mathcal{F}_T}, dv_{\mathcal{F}_N}, dL_fu, dL^2_fu, \text{ and } dL^3_fu$ at every state, this conjecture will be supported by our simulation results later in Chapter 6.

### 5.3 Pose Observers

With local observability, we can view sensing strategies as nonlinear observers for the disk-polygon system (5.26) or for the general pushing system (5.7)–(5.10). Luenberger-like asymptotic observers [95] for nonlinear systems are often designed through linearization. The disk-polygon system (5.26), however, cannot be linearized by Nijmeijer’s procedure in Theorem 14, for we have that

$$L_{gN} L_f L_{gN} L_f u = r(r + h) \frac{s(s^2 - \rho^2)}{(s^2 + \rho^2)^3},$$

violating condition 3 of the theorem. This negative result moves us away from constructing an observer through linearization.

Another approach of observer design is linearization by output injection introduced in Section 4.2.2. The necessary conditions for a nonlinear system to admit linear observer error dynamics are rather restrictive and hardly satisfied by the disk-polygon system, let alone system (5.7)–(5.10). In Appendix E we will apply Theorem 15 to show that linearization by output injection is not even applicable to perhaps the simplest version of the disk-polygon...
system satisfying conditions: (1) no support friction \((\mu = 0)\); (2) constant finger velocity \((a_T = 0)\); and (3) rolling contact (to be introduced later in Section 5.4.1). Even if the necessary conditions for linearization hold, it is still quite burdensome (and sometimes impossible) to find explicit solutions to partial differential equations involving repeated Lie brackets on which the desired coordinate transformation must be based.

### 5.3.1 A Gauthier-Hammouri-Othman Observer

The observer design technique we shall apply for designing an observer for the disk-polygon system (5.26) is the GHO procedure described in Theorem 16.

Getting back to the disk-polygon system (5.26), we now need to consider only \(u, s, \omega, \text{ and } v_T\) as state variables. The drift and input fields reduce from (5.27) to

\[
\begin{align*}
\mathbf{f}(\mathbf{x}) &= \begin{pmatrix} \omega r \\ \frac{r}{s^2 + \rho^2} \left( \omega^2 (r + h) - 2 \omega (v_T - v_T) \right) - \frac{\mu g}{A(s^2 + \rho^2)} \Gamma_N \\ \omega v_{\mathcal{F}N} + s \omega^2 - \frac{\mu g}{A} \Gamma_T \\ u \end{pmatrix}, \\
\mathbf{g}_T(\mathbf{x}) &= (0, 0, 0, 0)^T, \\
\mathbf{g}_N(\mathbf{x}) &= \begin{pmatrix} 0, 0, -\frac{s}{s^2 + \rho^2}, 0 \end{pmatrix}^T.
\end{align*}
\]

With \(u\) being the system’s only output, the new coordinates under map \(\chi\) consist of \(u\) and its Lie derivatives, up to the third order:

\[
\begin{pmatrix} u \\ s \\ \omega \\ v_T \end{pmatrix} \xrightarrow{\chi} \begin{pmatrix} u \\ \omega r \\ rL_{f}\omega \\ rL_{f}^2\omega \end{pmatrix}.
\]

Generally, for all except at most a finite number of states, \(du, rd\omega, rdL_{f}\omega,\) and \(rdL_{f}^2\omega\) are linearly independent, which implies that the map \(\chi\) is locally diffeomorphic. The Jacobian of the inverse transformation \(\chi^{-1}\) is then the inverse of the Jacobian of \(\chi\) [131, p. 2-17]:

\[
\frac{\partial \chi^{-1}}{\partial \mathbf{x}} = \begin{pmatrix} \frac{du}{\partial \mathbf{x}} \\ \frac{rd\omega}{\partial \mathbf{x}} \\ \frac{rdL_{f}\omega}{\partial \mathbf{x}} \\ \frac{rdL_{f}^2\omega}{\partial \mathbf{x}} \end{pmatrix}^{-1}.
\]

The differential \(dL_{f}\omega\) consists of the following partial derivatives:

\[
\frac{\partial L_{f}\omega}{\partial u} = 2 \omega \frac{s}{s^2 + \rho^2} \frac{v_{\mathcal{F}N}}{r} - \frac{\mu g}{A(s^2 + \rho^2)} \left( \frac{\partial \Gamma}{\partial u} \cdot N - \frac{\Gamma_T}{r} \right);\]
\[
\frac{\partial L_f\omega}{\partial s} = \frac{\rho^2 - s^2}{(s^2 + \rho^2)^2} \left( \omega^2 (r + h) - 2\omega (v_T - v_{\bar{r}T}) \right) + \frac{2\mu g s}{A(s^2 + \rho^2)} \Gamma_N - \frac{\mu g}{A(s^2 + \rho^2)} \frac{\partial \Gamma}{\partial s} \cdot N;
\]
\[
\frac{\partial L_f\omega}{\partial \omega} = \frac{2s}{s^2 + \rho^2} \left( \omega (r + h) - (v_T - v_{\bar{r}T}) \right) - \frac{\mu g}{A(s^2 + \rho^2)} \frac{\partial \Gamma}{\partial \omega} \cdot N;
\]
\[
\frac{\partial L_f\omega}{\partial v_T} = -2\omega \frac{s}{s^2 + \rho^2} - \frac{\mu g}{A(s^2 + \rho^2)} \left( \frac{\partial \Gamma}{\partial v_T} \right) \cdot N,
\]

where the closed form of \( d\Gamma \) (on \( u, s, \omega, v_T \)) is given in Appendix C.2. The differential \( dL_f^2\omega \), however, involves second-order partial derivatives of \( \Gamma \) whose closed forms is too complicated to obtain. Hence we choose to evaluate \( dL_f^2\omega \) numerically.

Solve equation (4.9) under \( n = 4 \) and take the inverse of the solution:

\[
S_\infty = \begin{pmatrix}
\frac{1}{\zeta} & -\frac{1}{\zeta^2} & \frac{1}{\zeta^3} & \frac{1}{\zeta^4} \\
-\frac{1}{\zeta} & \frac{2}{\zeta^2} & \frac{3}{\zeta^3} & \frac{4}{\zeta^4} \\
\frac{1}{\zeta} & -\frac{3}{\zeta^2} & \frac{6}{\zeta^3} & \frac{10}{\zeta^4} \\
-\frac{1}{\zeta} & \frac{4}{\zeta^2} & -\frac{10}{\zeta^3} & \frac{20}{\zeta^4}
\end{pmatrix}
\]

and

\[
S_\infty^{-1} = \begin{pmatrix}
4\zeta & 6\zeta^2 & 4\zeta^3 & \zeta^4 \\
6\zeta^2 & 14\zeta^3 & 11\zeta^4 & 3\zeta^5 \\
4\zeta^3 & 11\zeta^4 & 10\zeta^5 & 3\zeta^6 \\
\zeta^4 & 3\zeta^5 & 3\zeta^6 & \zeta^7
\end{pmatrix}.
\]

Finally, from (4.10) and (5.28) we obtain a GHO observer for frictionless contact:

\[
\begin{pmatrix}
\dot{\bar{u}} \\
\dot{s} \\
\dot{\bar{\omega}} \\
\dot{v}_T
\end{pmatrix} = \begin{pmatrix} f(\bar{u}, \bar{s}, \bar{\omega}, v_T) - (a_F \cdot N(\bar{u}))g_N(s) - (1 \ 0 \ 0 \ 0 \ \rho dL_f\omega(\bar{u}, \bar{s}, \bar{\omega}, v_T) - \rho dL_f^2\omega(\bar{u}, \bar{s}, \bar{\omega}, v_T) \right)^{-1} (4\zeta \ 6\zeta^2 \ 4\zeta^3 \ \zeta^4) (\bar{u} - u).
\]

(5.29)

It should be noted that we did not verify condition 2 in Theorem 16. The Lie derivative \( L_f^2u \) is generally not extendable to a globally Lipschitzian function. However, \( L_f^2u \) is locally Lipschitzian. So the observer should work well as long as the state estimate is close to the real state and its trajectory does not exit the local neighborhood in which the Lipschitz condition holds. This will be supported by the simulation results in Section 6.1.

### 5.3.2 The Initial Pose Observer

The asymptotic observer presented in Section 5.3.1 has two drawbacks. First, for finger and object shapes other than disks and polygons, the computation of the Lie derivatives and the Jacobian may become a burden. Second, the observer requires a sequence of contact locations on the finger to be sensed, which may cause difficulties in sensor implementation.

One sensing strategy is to observe the initial object pose. The state of the pushing system at any time will then be determined from equations (5.7)–(5.10) given that the finger’s pose and velocity during the pushing are known (to the controller). The initial object pose is determined by the initial contact position \( s_0 \) on the object boundary. So is the contact
5.3. POSE OBSERVERS

Figure 5.5: A shooting method for initial pose determination. (a) Two different finger contact motions resulting from the initial object poses \( s_0 = a \) and \( s_0 = b \), respectively; (b) possible finger contacts at time \( t_1 \) resulting from any initial pose \( s_0 \in [a, b] \) constitute a curve \( g(s_0) = u(t_1; s_0) \). The initial pose observer works by intersecting \( g(s_0) \) with the line \( u(t_1) = u_1 \) to determine the real initial object pose \( s_0^* \).

position \( u(t) \) on the finger. This fact leads to our second observer which is a variation of the shooting method for the integration of ordinary differential equations. This observer is for system (5.7)–(5.10) in which the finger and the object have general planar shapes.

For each initial object contact \( s_0 \), there is a unique finger contact trajectory \( u(t) \equiv u(t; s_0), t \geq 0 \), as the solution to the differential equations (5.7)–(5.10) with the initial values including \( s_0 \). Note \( u(0; s_0) = u_0 \) must hold, where \( u_0 \) is the initial finger contact. Let the finger sense a second contact position \( u_1 \) at time \( t_1 > 0 \). Then the problem reduces to finding a zero \( s_0^* \) of the function \( u(t_1; s_0) - u_1 \). Figure 5.5 depicts the contact curves resulting from two different initial object contacts, together with the curve segment \( g(s_0) = u(t_1; s_0) \) representing all possible finger contacts at time \( t_1 \) resulting from any initial object contact in between.

The root \( s_0^* \) of \( u(t_1; s_0) - u_1 \) can be obtained iteratively by Newton’s method for solving nonlinear equations. Each evaluation of this function now involves solving a separate initial value problem for the system (5.7)–(5.10) given the value of \( s_0 \) at the present iteration step.

The initial pose (IP) observer is also local and therefore subject to how close the estimate on \( s_0 \) at the start of iteration is from the real pose \( s_0^* \). To globalize sensing, we provide Newton’s method with multiple guesses of \( s_0 \) along the object boundary. This may yield multiple solutions to \( u(t_1; s_0) = u_1 \), as we will see in the simulation results in Section 6.2. However, such ambiguities can often be resolved by detecting a third contact \( u_2 \) on the finger at time \( t_2 > t_1 \) and verifying against \( u_2 \) the finger contacts at \( t_1 \) resulting from all ambiguous \( s_0 \) values.
5.4 Contact Friction

This section extends the results in the previous sections to include contact friction between the finger and the object. Now we need to consider two modes of contact: rolling and sliding, according as whether the contact force lies inside the contact friction cone or on one of its two edges. Each mode is hypothesized and solved; then the obtained contact force is verified with the contact friction cone for consistency. This hypothesis-and-test approach is quite common in solving multi-rigid-body contact problem with Coulomb friction. (See, for instance, Haug et al. [67].)

5.4.1 Rolling

When rolling contact occurs, the contact force $F$ may lie anywhere inside the contact friction cone. Let $\mu_c$ be the coefficient of contact friction. Constraint (5.4) for frictionless contact must now be replaced by

$$R(\frac{\pi}{2} + \phi)R\beta' \times F < 0 < R(\frac{\pi}{2} - \phi)R\beta' \times F,$$

(5.30)

where $\phi = \tan^{-1} \mu_c$ is the half angle of the contact friction cone and $R, \beta, F$ are defined in Section 5.1. Furthermore, the two points in contact, fixed on $\alpha$ and $\beta$, respectively, must have the same instantaneous velocity; that is,

$$v_F = v + \omega \times R\beta.$$

(5.31)
Subtracting (5.31) from the velocity constraint (5.1) on contact maintenance yields
\[ \alpha' \dot{u} = R \beta' \dot{s}. \] (5.32)

We are now ready to set up the contact and object motion equations for rolling.

**Proposition 20** In the problem of a translating finger pushing an object considered in Section 5.1, assume contact friction between the finger and the object as well. In addition to the notation of Section 5.1, let \( \mu_c \) and \( \phi = \tan^{-1} \mu_c \) be the coefficient and the angle of contact friction, respectively. When the object is rolling along the finger boundary, the pushing system is determined by (5.2), (5.3), (5.6), (5.30), (5.31), and (5.32). The contact and object motions satisfy
\[ \dot{u} = \frac{-\omega (\alpha' \cdot R \beta')^2}{(\alpha' \cdot R \beta')(\alpha'' \times R \beta') + \| \alpha' \|^2 (\alpha' \times R \beta')}, \] (5.33)
\[ \dot{s} = \frac{-\omega}{(\alpha' \cdot R \beta')(\alpha'' \times R \beta') + \| \alpha' \|^2 (\alpha' \times R \beta')} \times \| \beta \|^2 + \rho^2, \] (5.34)
\[ \dot{\omega} = \frac{R \beta \times a_F - (\beta \cdot \beta') \omega \dot{s} + \frac{1}{2} \mu \eta g \int_B (\beta - p) \times \dot{v}_p \, dp}{\| \beta \|^2 + \rho^2}, \] (5.35)
\[ v = v_F - \omega \times R \beta. \] (5.36)

**Proof** Equations (5.33) and (5.34) are just the special cases of equations (5.7) and (5.8), respectively, under the rolling constraint (5.31).

Differentiate both sides of (5.31):
\[ a_F = \dot{v} + \dot{\omega} \times R \beta - \omega^2 R \beta + \omega \times R \beta' \dot{s}. \] (5.37)

Meanwhile, substituting Newton’s equation (5.2) into Euler’s equation (5.3) and manipulate the resulting terms to obtain
\[ R \beta \times \dot{v} = \frac{1}{m} \left( I \dot{\omega} + \mu \eta g \int_B R(p - \beta) \times \dot{v}_p \, dp \right). \] (5.38)

Taking the cross products of \( R \beta \) with both sides of (5.36) and plugging (5.37) in, we have after a few steps of term expansion:
\[ R \beta \times a_F = \frac{1}{m} \left( I \dot{\omega} + \mu \eta g \int_B R(p - \beta) \times \dot{v}_p \, dp \right) + \| \beta \|^2 \dot{\omega} + (\beta \cdot \beta') \omega \dot{s}, \] from which (5.35) immediately follows.

To investigate local observability in the presence of contact friction, we look at the same problem of a disk pushing a polygon considered before. In fact, local observability for the case of rolling can be established more easily. Under rolling contact, \( v \) depends on \( u, s, \omega \):
\[ v = v_F - \omega \times R \beta \]
\[ = v_F - \omega \times (hN - sT) \]
\[ = v_F + \omega (hT + sN). \] (5.38)
Subsequently, a state can be denoted by $x = (u, s, \omega)^T$.\textsuperscript{13} And the dynamical system (5.26) has simpler drift and input fields:

$$
\mathbf{f} = \begin{pmatrix}
\omega r \\
-r s \omega^2 + \frac{\mu g}{A} \int_B R(\beta - p) \times \hat{v}_p \, dp \\
\frac{\omega r}{s^2 + h^2 + \rho^2}
\end{pmatrix},
$$

$$
\mathbf{g}_T = \begin{pmatrix}
0, 0, -\frac{h}{s^2 + h^2 + \rho^2} \end{pmatrix}^T,
$$

$$
\mathbf{g}_N = \begin{pmatrix}
0, 0, -\frac{s}{s^2 + h^2 + \rho^2} \end{pmatrix}^T.
$$

We leave to the reader the task of verifying that the differentials $du, dL_f u$, and $dL_{gT} L_f u$ (or $dL_{gN} L_f u$ if $s = 0$) are linearly independent.

**Theorem 21** The disk-polygon system (5.39) with rolling contact between the disk and the polygon and under support friction in the plane is locally observable.

The GHO observer for the rolling case has the form

$$
\begin{pmatrix}
\dot{\hat{u}} \\
\dot{\hat{s}} \\
\dot{\hat{\omega}}
\end{pmatrix} = \mathbf{f}(\hat{u}, \hat{s}, \hat{\omega}) + \left( \mathbf{a}_F \cdot T(\hat{u}) \right) \mathbf{g}_T(\hat{s}) + \left( \mathbf{a}_F \cdot N(\hat{u}) \right) \mathbf{g}_N(\hat{s})
$$

$$
- \begin{pmatrix}
1 & 0 & 0 \\
0 & 0 & r \\
r dL_f \omega(\hat{u}, \hat{s}, \hat{\omega})
\end{pmatrix}^{-1} \begin{pmatrix}
3 \zeta \\
3 \zeta^2 \\
\zeta^3
\end{pmatrix} (\hat{u} - u).
$$

The derivation of (5.40) is similar to that of (5.29) and given in Appendix D.

### 5.4.2 Sliding

When sliding contact occurs, $F$ must lie along one edge of the contact friction cone that makes an obtuse angle with the sliding direction. Constraint (5.4) must be accordingly replaced by

$$
F \cdot R(\theta \pm \phi) \beta' = 0,
$$

where “$\pm$” is determined by the sliding direction, which is hypothesized. Rewrite constraint (5.41) as $F \cdot R \tilde{\beta}' = 0$ where $\tilde{\beta}' = R(\pm \phi) \beta'$. Also denote $\tilde{\beta}' = R(\pm \phi) \beta'$ and $\hat{\Gamma} = \int_B R \tilde{\beta}' \times (R p \times \hat{v}_p) + (\tilde{\beta}' \cdot \beta) \hat{v}_p \, dp$. The differential equations governing contact and object motions are similar to those under no contact friction given in Section 5.1).

---

\textsuperscript{13}Unlike in the case of frictionless contact, here $v_{F_T}$ and $v_{F_N}$ are not involved in the dynamics of rolling. So they are not considered as state variables.
5.5. SUMMARY

Proposition 22 In the problem of a translating finger pushing an object considered in Section 5.1, assume contact friction between the finger and the object as well. In addition to the notation of Section 5.1, let \( \mu_c \) and \( \phi = \tan^{-1} \mu_c \) be the coefficient and the angle of contact friction, respectively. When the object is sliding along the finger boundary, the pushing system is determined by (5.1), (5.2), (5.3), (5.5), (5.6), and (5.41). The contact motions still follow (5.7) and (5.8), while the object’s angular acceleration and acceleration satisfy

\[
\omega = \frac{\dot{u} \alpha'' \times (v_F - v) + \alpha' \times a_F - (\dot{u} \alpha'' \cdot R\beta + \alpha' \cdot (\omega \times R\beta + R\beta'))}{A\beta' \cdot \beta},
\]

\[
\alpha' \cdot \dot{R} \left( \beta + \frac{\rho^2}{\beta' \cdot \beta'} \right).
\]

(5.42)

\[
\dot{v} = \frac{A\rho^2 \omega \times R\beta' - \mu g \Gamma}{A\beta' \cdot \beta}.
\]

(5.43)

Proof Analogous to the proof of Theorem 17. \( \square \)

The resemblance of equations (5.42) and (5.43) to equations (5.9) and (5.10) suggests the reasoning on local observability for the disk-polygon system in the sliding case to resemble the proof of Theorem 19. We write the system in this case into the form of (5.26) and obtain its drift and input fields:

\[
f = \begin{pmatrix}
\omega r \\
v_T - v_{F_T} - \omega (r + h) \\
0 \\
0
\end{pmatrix},
\]

\[
g_N = \begin{pmatrix}
0 \\
0 \\
s \cos \phi \pm h \sin \phi \\
\mp s^2 \cos \phi \pm h s \sin \phi + \rho^2 \cos \phi \\
0 \\
1
\end{pmatrix},
\]

and \( g_T \) as given in (5.27), where “\pm” stands for “+” for left sliding of the polygon and “−” for right sliding, and “\_” for some complicated terms. Involved calculations will reveal that \( du, dv_{F_T}, dv_{F_N}, dLf_u, dLg_nL_fu \) (or \( dLg_nL_fL_gnL_fu \)), and \( dL_fL_gnL_fL_fu \) (or \( dL_fL_gnL_fL_gnL_fu \)) again span the observability codistribution unless \( \tan \phi = \frac{r + h}{\rho} \).

Theorem 23 The disk-polygon system with sliding contact between the disk and the polygon and under support friction in the plane is locally observable if \( \tan \phi \neq \frac{r + h}{\rho} \).

5.5 Summary

This chapter introduces a sensing approach based on nonlinear observability theory that makes use of one-finger tactile information. The approach determines the pose of a known
CHAPTER 5. POSE AND MOTION FROM CONTACT

A planar object by pushing it with a finger that can feel the contact motion on the fingertip. It also obtains the object motion during the pushing. Both the finger and the object are assumed to have piecewise smooth boundaries.

We derive a system of nonlinear differential equations governing contact and object motions from geometric and velocity constraints, as well as from the dynamics of pushing. The state of this system includes the pose and motion of the object while its output is the moving contact position on the fingertip. We establish the local observability of the system for the case of a disk pushing a polygon, a result that can generalize to many other finger and object shapes. We also show that the object pose just before pushing is almost always locally observable. These results form the underlying principles of our sensing algorithms, which essentially are observers of the nonlinear dynamical system.

Based on the result of [55], we construct an asymptotic nonlinear observer and demonstrate it by simulations. This observer is a composition of a copy of the original system with an error corrective term constructed over the system output and the solution of a Lyapunov-like equation. It is capable of asymptotically correcting any local error in estimating the object pose and motion. The observer can accept a sequence of fingertip contacts starting at any time during a push. Such an on-line property makes the observer quite flexible but it needs a good sensor to provide continually more than just a few contact data.

We also present a nonlinear observer based on Newton’s method. It determines the initial resting pose of the object (and thus any pose from then on) from a small number of intermediate contact positions on the fingertip. In designing this observer, we view pushing as a mapping from the one-dimensional set of initial object poses to the set of contact motions on the fingertip; and sensing just as its inverse mapping.

Both support friction in the plane and contact friction between the object and the finger have been taken into account. Appendix C offers an algorithm for analytically evaluating the integrals which combine the effects of friction with the dynamics of polygonal objects, as well as the first order partial derivatives of these integrals. It also shows how to determine the initial acceleration and angular acceleration of a motionless object under pushing in the presence of support friction only.
Chapter 6

Observer Simulations and Sensor Implementation

We simulated the GHO observer and the IP observer by the fourth-order Runge-Kutta integration with a stepsize corresponding to 0.01 second (0.01s) real time. The object data in our simulations included polygons and ellipses,\(^1\) all of which were randomly generated.\(^2\)

Our proofs of the local observability of the disk-polygon system and its variations in Sections 5.2.1, 5.4.1, and 5.4.2 did not rely on support friction. This suggests that friction would hardly affect the observer’s performance. So we set the coefficient of support friction to be uniformly 0.3. This number was also consistent with the measurements in our experiments, which are to be discussed in the next section. The finger accelerations and velocities used in the simulations are easily achievable on an Adept robot. For convenience, only constant finger accelerations were used.

The simulation code was written in Lisp and run on a Sparcstation 20. The major load of computation turned out to have come from the evaluations of the integrals of friction and their partial derivatives (with respect to the object pose and velocities). To speed up, these integrals and their first order partial derivatives were evaluated via closed forms when the object was a polygon. In such a case, each evaluation took time linear in the number of the polygon vertices (see Appendix C for the algorithm). For instance, evaluating \(\Gamma\) given by (5.19) for a 7-gon took 0.183s and evaluating its partial derivatives took 1.118s; while evaluating \(\Gamma\) and its partial derivatives for a triangle took only 0.067s and 0.412s, respectively.

Section 6.1 presents the results of simulating the GHO observer. Section 6.2 presents the results of simulating the initial pose observer, demonstrating that three intermediate contact points often suffice to determine the initial pose for the fingers and objects tested. Section 6.3 shows some preliminary experimental results. Section 6.4 briefly describes the implementation of a “planar finger” capable of detecting any contact position on its boundary. Section 6.5 discusses sensing ambiguities due to the global object shape.

\(^1\)The latter shapes are for the initial pose observer only.
\(^2\)The polygons were constructed by taking random walks on the arrangement of a large number of random lines precomputed by a topological sweeping algorithm [41].
We simulated two versions of the GHO observer for the disk-polygon system: (5.29) in the case of frictionless contact ($\mu_c = 0$) between the disk and the polygon and (5.40) in the case of rolling contact. The first version has four state variables: the disk contact $u$, the edge contact $s$ (which determines the polygon’s pose), the polygon’s angular velocity $\omega$, and the tangential component of the polygon’s velocity $v_T$. The second version has only three: $u$, $s$, $\omega$. The case of sliding contact was not simulated mainly because it is very similar to the case of frictionless contact except its nonlinear system is more complicated.

The magnitude of the control parameter $\zeta$ of the GHO observer directly affects its performance. When $\zeta$ is too small, the observer would either converge its estimate to the real state very slowly or not converge at all. In this case, the error correction would be dominated by the original system’s drift field such that it may not be enough to drive the estimate to some neighborhood of the real state where it can converge. On the other hand, when $\zeta$ is too large, the error correction would dominate the original system, causing the state estimate to change dramatically and often to diverge. Based on numerous trials, we chose $\zeta = 10$ in our simulations.

The disk radius was normalized to 1cm in all simulations. All time measurements will refer to how long the pushing would have taken place in the real world rather than how long the computation took.\footnote{Simulating 0.8s observation of a pushed quadrilateral with rolling contact took 232s, while simulating 0.66s observation of the same quadrilateral with frictionless contact took 1012s.}

To get an idea of the observer’s behavior, let us look at a simple example of a 7-gon being pushed by the unit disk and rolling on its boundary (see Figure 6.1). The trajectories of $u$, $s$, $\omega$ and their estimates $\tilde{u}$, $\tilde{s}$, $\tilde{\omega}$ are shown in Figure 6.2. Since the disk contact $u$ is also the output, its estimate $\tilde{u}$ converges faster than the estimates of other state variables. However, this had caused the following problem in many other instances we simulated: The feedback $\tilde{u} - u$ that drives the observer’s error corrective term, would usually diminish fast and become ineffective before other estimates can be corrected. To remedy this problem, our observer turns off error correction in the last 0.04s of every 0.1s interval of pushing so that the error $\tilde{u} - u$ would accumulate a bit for the corrective term to become effective again at the start of the next 0.1s interval. This scheme has turned out to be quite effective at driving other state variable estimates toward convergence.

We first conducted tests assuming known contact edges. In each test, a state and an estimate were randomly generated over the ranges of the state variables.\footnote{The range of $u$ in terms of the polar angle with respect to the disk center was set to be the interval $[80, 100]$ (degrees); the range of $s$ was determined from the contact edge; the ranges of $\omega$ and $v_T$ were set as $[-1, 1]$ (rad/s) and $[-0.4, 0.4]$ (cm/s$^2$), which were based on the velocity range of the Adept robot and on our simulation data of pushing.} The test would be regarded as a success as soon as the difference between the state and its estimate had become negligible for a period of time;\footnote{The length of the period can be arbitrarily set but should be large enough. It was chosen as 0.2s in our simulations.} it would be regarded as a failure if one of the state

\begin{itemize}
    \item \textit{CHAPTER 6. OBSERVER SIMULATIONS AND SENSOR IMPLEMENTATION}
    \item \textbf{6.1 On the GHO Observer}
    \item We simulated two versions of the GHO observer for the disk-polygon system: (5.29) in the case of frictionless contact ($\mu_c = 0$) between the disk and the polygon and (5.40) in the case of rolling contact. The first version has four state variables: the disk contact $u$, the edge contact $s$ (which determines the polygon’s pose), the polygon’s angular velocity $\omega$, and the tangential component of the polygon’s velocity $v_T$. The second version has only three: $u$, $s$, $\omega$. The case of sliding contact was not simulated mainly because it is very similar to the case of frictionless contact except its nonlinear system is more complicated.
    \item The magnitude of the control parameter $\zeta$ of the GHO observer directly affects its performance. When $\zeta$ is too small, the observer would either converge its estimate to the real state very slowly or not converge at all. In this case, the error correction would be dominated by the original system’s drift field such that it may not be enough to drive the estimate to some neighborhood of the real state where it can converge. On the other hand, when $\zeta$ is too large, the error correction would dominate the original system, causing the state estimate to change dramatically and often to diverge. Based on numerous trials, we chose $\zeta = 10$ in our simulations.
    \item The disk radius was normalized to 1cm in all simulations. All time measurements will refer to how long the pushing would have taken place in the real world rather than how long the computation took.$^3$
    \item To get an idea of the observer’s behavior, let us look at a simple example of a 7-gon being pushed by the unit disk and rolling on its boundary (see Figure 6.1). The trajectories of $u$, $s$, $\omega$ and their estimates $\tilde{u}$, $\tilde{s}$, $\tilde{\omega}$ are show in Figure 6.2. Since the disk contact $u$ is also the output, its estimate $\tilde{u}$ converges faster than the estimates of other state variables. However, this had caused the following problem in many other instances we simulated: The feedback $\tilde{u} - u$ that drives the observer’s error corrective term, would usually diminish fast and become ineffective before other estimates can be corrected. To remedy this problem, our observer turns off error correction in the last 0.04s of every 0.1s interval of pushing so that the error $\tilde{u} - u$ would accumulate a bit for the corrective term to become effective again at the start of the next 0.1s interval. This scheme has turned out to be quite effective at driving other state variable estimates toward convergence.
    \item We first conducted tests assuming known contact edges. In each test, a state and an estimate were randomly generated over the ranges of the state variables.$^4$ The test would be regarded as a success as soon as the difference between the state and its estimate had become negligible for a period of time;$^5$ it would be regarded as a failure if one of the state
Figure 6.1: A disk of radius 1cm at constant velocity 5cm/s pushing a 7-gon while observing its pose and motion. The snapshots are taken every 0.1s. Contact friction between the polygon and the disk is assumed to be large enough to allow only the rolling on the disk edge. The edge of the polygon in contact is assumed to be known. The coefficient of support friction is 0.3. (a) The scene of pushing for 0.71s. (b) The imaginary scene as “perceived” by the observer during the same time period. The observer constantly adjusts its estimates of the polygon’s pose and motion based on the moving contact on the disk boundary until they converge to the real pose and motion. Although the real contact and its estimate were about 4.5cm apart on the contact edge at the start of estimateion, the error becomes negligible in 0.56s.
Figure 6.2: State variable trajectories vs. state estimate trajectories for the example shown in Figure 6.1. The sampling rate is 100Hz. Variable $u$ gives the polar angle (scaled by the disk radius 1cm) of the contact from the disk center. Variable $s$ measures the (signed) distance from the contact point to the intersection of the contact edge with its perpendicular through the polygon’s center of geometry; it has the range $[-5.82, 2.90]$ (cm). Variable $\omega$ is the polygon’s angular velocity. These three state variables have estimates $\tilde{u}$, $\tilde{s}$, and $\tilde{\omega}$, respectively. (c) The trajectories of $u$ and $\tilde{u}$. (d) the trajectories of $s$ and $\tilde{s}$; and (e) the trajectories of $\omega$ and $\tilde{\omega}$. Note that $\tilde{u}$ and $\tilde{\omega}$ converge faster than $\tilde{s}$.
6.1. ON THE GHO OBSERVER

<table>
<thead>
<tr>
<th>Type of Pushing</th>
<th>$\alpha_F$ (cm$^2$/s)</th>
<th>No. of Tests</th>
<th>Successes</th>
</tr>
</thead>
<tbody>
<tr>
<td>Frictionless</td>
<td>0</td>
<td>50</td>
<td>42</td>
</tr>
<tr>
<td>Contact</td>
<td>2.5</td>
<td>50</td>
<td>44</td>
</tr>
<tr>
<td>Rolling</td>
<td>0</td>
<td>500</td>
<td>457</td>
</tr>
<tr>
<td></td>
<td>2.5</td>
<td>500</td>
<td>456</td>
</tr>
</tbody>
</table>

Table 6.1: Simulations on observing the poses and motions of random polygons being pushed by the unit disk, assuming the contact edges were known. Two versions of the GHO observers were simulated: (5.29) for the case of no friction between a polygon and the unit disk; and (5.40) for the case where the polygon is rolling on the disk boundary.

variables had gone out of its range repeatedly or there was no success after a long period of observation.\(^6\)

Table 6.1 summarizes the results with known contact edges. There are four groups of data, each representing a different combination of contact mode and disk acceleration. As the table indicates, the finger acceleration $\alpha_F$ did not affect the observer’s performance. This seems to be in contradiction with our resort to the use of the normal input field $g_N$, driven by the normal acceleration $\alpha_{FN}$, in the proof of Theorem 19 on local observability. Nevertheless, the use of $g_N$ serves to simplify the construction of an algebraic proof of the observability rank condition. We might have used the drift field $f$ only in the proof, except the rank condition would be very hard to establish.

Figure 6.3(a) shows a simulation example in which a 5-gon making frictionless contact with the unit disk translating at constant velocity. Figure 6.3(b) plots the ‘polygon motion’ as understood by the observer from the contact motion along the disk boundary. In 0.6s (real time), the observer is able to locate the contact point (thereby determining the pose of the 5-gon) as well as to estimate its velocity and angular velocity. The trajectories of the state variables $u, s, \omega, v_T$ paired with the trajectories of their estimates $\tilde{u}, \tilde{s}, \tilde{\omega}, \tilde{v}_T$ are shown in Figure 6.4 (c), (d), (e), (f), respectively.

Since in every test the estimated contact point, given by $s$, was randomly chosen on the contact edge, it could be far from the real contact point. Yet, the results in Table 6.1 seem to suggest that the local GHO observer has “globalness”, at least within one edge.

We also observed that the disk contact estimate $\tilde{u}$ and the angular velocity estimate $\tilde{\omega}$ always converged very fast, and the tangential velocity estimate $\tilde{v}_T$ almost always converged. The pose estimate $\tilde{s}$, however, was always part of the divergence whenever it occurred. This phenomenon agrees with our previous discussion following the proof of Theorem 19 on the relative “hardness” of observing different state variables of the disk-polygon system.

\(^6\)In the simulations, we set this “long period” as 2s.
CHAPTER 6. OBSERVER SIMULATIONS AND SENSOR IMPLEMENTATION

Figure 6.3: A disk of radius 1cm at constant velocity 5cm/s pushing and observing a 5-gon. The contact between the disk and the polygon is assumed to be frictionless. (a) The scene of pushing for 0.6s. (b) The imaginary scene as “perceived” by the observer (5.29). The real contact and its estimate were about 7.84cm apart on the edge at the start of estimation. The error became negligible in about 0.5s. Figure 6.4 details the convergence of the estimates of the pose, velocity, and angular velocity of the polygon during the push.

In the real situation, only the finger contact \( u \) is known. In other words, the contact edge, the contact location \( s \) on the edge, and the velocities \( \omega \) and \( v_T \) are all unknown. Accordingly, we modify the observer as follows. The observer generates for each edge of the polygon being pushed a state estimate that hypothesizes the edge as in contact. Then it simulates the push starting with these estimates in parallel for a short period of time.\(^7\) Assuming that the estimate hypothesizing the correct contact edge will likely have converged to the real state by now, the observer then turns off its error correction and continues the simulation of the remaining possible state trajectories. The estimate is chosen from the trajectory that outlasts all the others in having its \( \tilde{u} \) stay negligibly close to the the observed disk contact \( u \). The observer fails if all estimates have gone out of their ranges in the first period, or the obtained contact estimate, including the edge and the location on the edge, is incorrect.

Table 6.2 shows the test results with unknown contact edges. A high percentage of the failures reported in the table were due to incorrect contact edges. To explain this, recall in the disk-polygon system (5.24) that the only parameters reflecting the contact geometry are the distance (or height) \( h \) from the contact edge to the polygon’s center of geometry (or mass) and the signed distance \( s \) from the contact to where the edge intersects its perpendicular from this center. Contact points on different edges of (approximately) the same height and with (approximately) the same \( s \) can thus result in (approximately) the same behavior of the disk-polygon system. Finding a wrong contact edge is therefore expected to happen

\(^7\)This length of this period was based on the average convergence time in Table 6.1.
Figure 6.4: State variable trajectories vs. state estimate trajectories for the example shown in Figure 6.3. Variables $u$, $s$, and $\omega$ are as specified in Figure 6.2. Variable $v_T$ is the projection of the velocity of $P$ onto the contact tangent. Variable $s$ has the range $[-5.82, 2.90]$ (cm). The four state variable estimates are $\tilde{u}$, $\tilde{s}$, $\tilde{\omega}$, and $\tilde{v}_T$, respectively. Note that $\tilde{u}$, $\tilde{\omega}$, $\tilde{v}_T$ converge faster than $\tilde{s}$. 
### Table 6.2: Simulations of the GHO observers (5.29) and (5.40) on finding the poses and motions of random polygons being pushed by a unit disk, assuming unknown contact edges.

<table>
<thead>
<tr>
<th>Type of Pushing</th>
<th>(a_\tau) (cm(^2)/s)</th>
<th>No. of Tests</th>
<th>No. of Successes</th>
<th>No. Ratio</th>
<th>Time (s)</th>
<th>Contact Edge Divergence</th>
</tr>
</thead>
<tbody>
<tr>
<td>Frictionless</td>
<td>0</td>
<td>50</td>
<td>26</td>
<td>52%</td>
<td>0.733</td>
<td>9 8 7</td>
</tr>
<tr>
<td>Contact</td>
<td>2.5</td>
<td>50</td>
<td>26</td>
<td>52%</td>
<td>0.760</td>
<td>9 9 6</td>
</tr>
<tr>
<td>Rolling</td>
<td>0</td>
<td>500</td>
<td>294</td>
<td>58.8%</td>
<td>0.893</td>
<td>32 122 52</td>
</tr>
<tr>
<td>Rolling</td>
<td>2.5</td>
<td>500</td>
<td>287</td>
<td>57.4%</td>
<td>0.891</td>
<td>35 129 49</td>
</tr>
</tbody>
</table>

There were three types of observation failures, shown from left to right in the three columns under the “Failures” title bar: (1) the observer found the correct contact edge but not the correct contact point; (2) the observer found the incorrect contact edge; (3) the observer diverged on all initial estimates.

often when the polygon’s center of geometry is approximately equidistant to the real contact edge and to another edge. In fact, the failures due to incorrect contact edges that we had observed individually were predominantly of this type.

### 6.2 On the IP Observer

Simulations were conducted for three types of pushing: ellipse(finger)-ellipse(object), line-ellipse, and ellipse-polygon. No contact friction was assumed in these simulations.

Closed forms of integral \(\Gamma\) exist for polygons but not for ellipses. On a Sparcstation 20, one evaluation of \(\Gamma\) takes about 2s for an ellipse. The computation of initial accelerations as in Section 5.1.3 takes about 1.6s for a hexagon and 25s for an ellipse.

During a push, the initial, the final, and one intermediate contact positions on the finger were recorded, along with the times when the contact reached these positions. The initial pose observer in Section 5.3.2 computed possible resting poses of the object which, under the push, would cause the contact to move to the intermediate position on the fingertip at the recorded time. More specifically, the algorithm guessed a number of initial contacts on the object, and called the Newton-Raphson routine.\(^8\) The final contact position was then used to further eliminate infeasible poses.

Table 6.3 shows the test results under no support friction. These results support our conjecture in Section 5.3.2 that the object pose can often be determined from three instantaneous contacts on the finger during a push.

The slow numerical evaluation of integral \(\Gamma\) prohibits us from conducting large number

---

\(^8\)In the experiments, 10 guesses were taken for an ellipse and 3 guesses for each edge of a polygon.
### 6.3 Preliminary Experiments

Later we conducted some experiments with an Adept 550 robot. The “finger” in our experiments was a plastic disc held by the robot gripper. The disk edge was marked with angles from the disk center so a contact position could be read by flesh eyes. Plastic polygonal parts of different material were used as objects. A plywood surface served as the supporting plane for pushing. Figure 6.5 shows the experimental setup.

Simulation and experimental results on pushing were found to agree closely (Figure 6.6), with slight discrepancies mainly due to shape uncertainties and non-uniform properties of the disk, the parts, and the plywood, all handmade.

<table>
<thead>
<tr>
<th>Finger</th>
<th>Object</th>
<th>No. of Tests</th>
<th>Successes</th>
</tr>
</thead>
<tbody>
<tr>
<td>ellipse</td>
<td>ellipse</td>
<td>1000</td>
<td>978</td>
</tr>
<tr>
<td>line</td>
<td>ellipse</td>
<td>1000</td>
<td>975</td>
</tr>
<tr>
<td>ellipse</td>
<td>poly</td>
<td>200</td>
<td>189</td>
</tr>
</tbody>
</table>

Table 6.3: Simulations of the IP observer with the frictionless plane.

Figure 6.5: Experimental setup of pose-from-pushing. The coefficient of contact friction between the part and the disk (finger) was small (measured to be 0.213).

of tests on elliptic objects under support friction. Simulations under friction were only performed on polygons, for which closed forms of \( \Gamma \) exist. The 105 tests took about 65 hours, yielding 94 successes, 11 failures and ambiguities.
Figure 6.6: Simulations versus experiments on a triangular part. The graphs show the final sensor contact $u_1$ as a function of the initial part contact $s_0$, which here measured the distance from a vertex of the part counterclockwise along the boundary. The part boundary was discretized into a finite set of locations $\{s_0\}$. For each such $s_0$ we performed a numerical simulation and a physical experiment. The same disk motion lasting 0.75 seconds was used in each of these simulations and experiments. The initial contact $u_0$ on the pushing disk was always at 90 degrees from its center. The case where $s_0$ was 11cm in the physical experiment is illustrated by the dotted lines: Four feasible poses were found by the simulator from contact position $u_1 = 105$ degrees after the push. These four poses were later distinguished (and the real pose was thus determined) using a second contact position $u_2$ determined after a second identical push by the disk.
6.4 Sensor Implementation

We have built a “finger” with tactile capability using four strain gauges as shown in Figure 6.7. The strain gauges are mounted near the top of a vertical stainless steel beam and connected to an Omega PC plug-in card to form two Wheatstone half bridges. The lower end of the beam is attached a disk which serves as the “finger”. A contact with the disk would result in the bending of the beam, which would be detected by the strain gauges. The components of the contact force exerted on the disk boundary along the $x$ and $y$ axes of the disk, respectively, can then be calculated. When contact friction is small enough, the contact force measured by the gauges would point along the disk normal at the contact, thereby indicating the contact location on the disk boundary.

The sensor is sensitive enough to detect force in microstrains with a frequency over 2000 Hz. It reports the contact in terms of its polar angle with respect to the disk center. After calibration, the sensed static contacts (in 1000 readings) constantly have a mean within one degree away from the real contact and a standard deviation of less than 0.5 degree. For
Figure 6.8: Situations where sensing ambiguities arise: (a) The object is translating, thereby no contact motion; (b) different curve segments along the object boundary may also result in the same contact motion on the finger.

As demonstrated by the simulation results, three contact positions \( u_0, u_1, \) and \( u_2 \) on a pushing finger \( \mathcal{F} \) often uniquely determine the initial pose \( s_0 \) of a pushed object \( \mathcal{B} \). However, there exist worst cases in which \( s_0 \) cannot be determined from even the entire contact motion \( u(t) \).

One such case arises when the finger’s velocity \( v_\mathcal{F} \) always points along an inward contact normal that passes through \( \mathcal{B} \)’s center of mass \( O \) (see Figure 6.8(a)). In this case, object \( \mathcal{B} \) translates and \( u(t) = u_0 \) for all \( t > 0 \). Every point on \( \mathcal{B} \)’s boundary \( \beta \) with its inward normal passing through \( O \) could be initially in contact with \( \mathcal{F} \), thereby corresponding to a

6.5 Ambiguities of Sensing

As demonstrated by the simulation results, three contact positions \( u_0, u_1, \) and \( u_2 \) on a pushing finger \( \mathcal{F} \) often uniquely determine the initial pose \( s_0 \) of a pushed object \( \mathcal{B} \). However, there exist worst cases in which \( s_0 \) cannot be determined from even the entire contact motion \( u(t) \).

One such case arises when the finger’s velocity \( v_\mathcal{F} \) always points along an inward contact normal that passes through \( \mathcal{B} \)’s center of mass \( O \) (see Figure 6.8(a)). In this case, object \( \mathcal{B} \) translates and \( u(t) = u_0 \) for all \( t > 0 \). Every point on \( \mathcal{B} \)’s boundary \( \beta \) with its inward normal passing through \( O \) could be initially in contact with \( \mathcal{F} \), thereby corresponding to a
6.5. AMBIGUITIES OF SENSING

possible initial pose.

Sensing ambiguity can also occur with different curve segments along \( \beta \). As we saw in the example of a disk pushing a polygon, contact points on different edges of the same height from the polygon’s center of geometry may result in exactly the same contact motion on the disk as well as the same polygon motion.

More generally, suppose the finger configuration, the initial finger contact \( u_0 \), and the finger motion \( v_F \) are known and do not change. Let \( \beta_1 \) be the portion of the object boundary that will be in contact with the finger during the push, parameterized as a unit-speed curve in its initial orientation with respect to the center of mass location \( O_1 \). Equations (5.14), (5.15), (5.9), and (5.10) then determine the contact motion \( u(t) \). Meanwhile, it is known that \( \beta_1 \) is determined, up to rotation and translation, by its curvature function \( \kappa_{\beta_1} \) [131]. Assuming now \( u(t) \) is known but \( \kappa_{\beta_1} \) is not, the same equations conversely determine the curvature function \( \kappa_{\beta_1} \), thereby the curve segment \( \beta_1 \) (in its initial pose). We can therefore construct a different curve segment \( \beta_2 \) along the object boundary that would also result in contact motion \( u(t) \) under the same push. Now choose the center of mass to be at a different location \( O_2 \) close enough to \( O_1 \). The curvature function \( \kappa_{\beta_2} \) determined from \( u(t) \) will be different from \( \kappa_{\beta_1} \). Thus \( \kappa_{\beta_2} \) defines a different curve segment \( \beta_2 \). Transform \( \beta_1 \) until \( O_1 \) coincides with \( O_2 \). Join \( \beta_1 \) (under the transformation) and \( \beta_2 \) with some curve segments to form a closed curve \( \beta \) whose center of geometry (mass) is at \( O_2 \). See Figure 6.8(b). Hence at least two possible poses of \( \beta \) would result in the same sensed contact motion.

\[ \text{This can be seen from (5.14) and initially } R_{\beta_1} \neq R_{\beta_2} \text{ due to } O_1 \neq O_2. \]
Chapter 7

Rolling on a Moving Plane — Mechanics and Local Observability

In Chapters 5 and 6 we investigated the local observability and the observers for a two-dimensional manipulation task that involves a finger pushing an object in the plane. As we saw there, geometry and mechanics are closely tied to each other. In a task, the states (or configurations) of an object and its manipulator evolve under the laws of mechanics and subject to the geometric constraints of contact. Such interaction often yields simple information into which the geometry and motions of the object and the manipulator are encoded by the mechanics of the manipulation. The contact between the object and the manipulator is a form of interaction through which information about the object is conveyed to the manipulator implicitly. We showed that such information can be used by a proper nonlinear observer for recovering the object pose and motion in the 2-dimensional pushing task.

In this chapter we continue our study of local observability, focusing on three-dimensional tasks. We intend to look into the same type of information: contacts between objects and manipulators. As we shall see, surface geometry around the contact plays the main role in kinematics. It affects not only how fast an object moves relative to a manipulator but also how fast it rotates, in response to a relative motion between the object and the manipulator. Meanwhile, dynamics deals with the rate of change of such relative motion under gravity and contact force (or simply, under the controlled manipulator motion).

We refer to the work by Montana [106] and by Cai and Roth [22] for detailed treatments of the kinematics of contact between smooth shapes. We also refer to MacMillan’s book [100] for a comprehensive introduction to dynamics of rigid bodies that blends analytic methods well with geometric intuition.

The task chosen for our study involves a horizontal plane, or a palm, that can translate in arbitrary directions, and a smooth object that can only roll on the palm. As shown in Figure 7.1, the palm is trying to control the object’s motion by executing certain motion of its own. The state of the object, comprising its pose and motion, however, is unknown at the moment. So the palm needs to determine the state before purposefully manipulating the object. We refer to this state estimation problem as palm sensing.
Can the object pose and motion be observed by the palm, or at least locally? As the object rolls, its contact traces out a curve as function of time in the plane (see Figure 7.1). Suppose the palm is covered by an array of tactile cells that are able to detect the contact location at any time instant. We would like to know if this curve of contact contains enough information for the palm to know about the configuration of the object. Our approach is to study local observability of rolling from the contact curve. Through this investigation we hope to touch on the more general issue of mechanics-based sensing and information retrieval in robotic tasks.

Section 7.1 derives a nonlinear system from the kinematics and dynamics of rolling that describes the object’s motion as well as the contact motions upon the surfaces of the object and palm. Section 7.2 establishes a sufficient condition on local observability for the configuration of the rolling object. This condition depends only on the object’s angular inertia matrix and local shape around the contact.

### 7.1 The Rolling Motion

The motion of an object rolling on a plane without slipping is subject to two constraints: (1) The point in contact has zero velocity relative to the plane; and (2) the object has no rotation about the contact normal. In this section we shall study the rolling motion of an object and the resulting motions of the contact on the object and in the plane. We shall derive a system of nonlinear differential equations that describes the object’s kinematics and dynamics. Such system will be given in terms of the object’s body frame. We shall see that the kinematics are affected by the local geometry at the contact, while the dynamics are affected by the position of the contact and the object’s angular inertia.

Before we begin, the problem needs to be formally defined. As shown in Figure 7.1, it
7.1. THE ROLLING MOTION

concerns an object $\mathcal{B}$ (with or without any initial velocity) moving on a horizontal plane $\mathcal{P}$, which is translating at velocity $\mathbf{v}_P$. In order to maintain contact, the plane acceleration $\dot{\mathbf{v}}_P$ must not exceed the gravitational acceleration $g$ downward. The friction between $\mathcal{P}$ and $\mathcal{B}$ is assumed to be large enough to allow only the rolling motion of $\mathcal{B}$ (without slipping).

For the sake of simplicity and clarity, we make a few additional assumptions:

1. Object $\mathcal{B}$ has uniform mass distribution.
2. Object $\mathcal{B}$ is bounded by an orientable surface.
3. The contact stays in one proper patch $\beta$ in the surface of $\mathcal{B}$ during the period of time when local observability is concerned.
4. The patch $\beta$ is convex and has positive Gaussian curvature everywhere.
5. The patch $\beta$ is principal.

The third assumption makes sense because local observability is concerned with an infinitesimal amount of time. The fourth assumption restricts the contact to be a point. The fifth assumption is justified because every point in a surface has a neighborhood that can be reparametrized as a principal patch. This fact is stated in Section 4.1 and proved in [132, pp. 320-323].

7.1.1 Kinematics of Rolling

To describe the motion of $\mathcal{B}$ and the motions of contact on both $\mathcal{P}$ and $\mathcal{B}$, we here set up several coordinate frames. Let $\mathbf{o}$ be the center of mass of $\mathcal{B}$. Let the coordinate frame $\Pi_B$ as shown in Figure 7.2 be centered at $\mathbf{o}$ and defined by $\mathcal{B}$’s principal axes $\mathbf{x}_B$, $\mathbf{y}_B$, and $\mathbf{z}_B$. The angular inertia matrix $I$ with respect to $\Pi_B$ is thus diagonal. Frame $\Pi_B$ is moving constantly relative to the plane $\mathcal{P}$ due to the rolling of $\mathcal{B}$.

A frame $\Pi_P$ with axes $\mathbf{x}_P$, $\mathbf{y}_P$, and $\mathbf{z}_P$ is attached to the plane $\mathcal{P}$ such that axis $\mathbf{z}_P$ is an upward normal to $\mathcal{P}$. The world coordinate frame, denoted by $\Pi_W$, has the same orientation as frame $\Pi_P$.

It is often convenient to describe the motion of $\mathcal{B}$ in terms of its body frame $\Pi_B$ by velocity $\mathbf{v}$ and angular velocity $\mathbf{\omega}$. In the meantime, denote by $\mathbf{v}_B$ and $\mathbf{\omega}_B$ the velocities of $\mathcal{B}$ relative to the fixed frame that currently coincide with $\Pi_B$. Hence $\mathbf{v}_B$ and $\mathbf{v}$ are identical at the moment, so are $\mathbf{\omega}_B$ and $\mathbf{\omega}$. But their derivatives have quite different meanings, as will be discussed in Section 7.1.2.

The contact point on $\mathcal{B}$ is denoted by $\mathbf{q} = \beta(s) = \beta(s_1, s_2)$ in $\Pi_B$. Since $\beta$ is a principal patch, the normalized Gauss frame $\Pi$ at $\mathbf{q}$ is well-defined by axes

$$
\mathbf{x} = \frac{\beta_{s_1}}{\|\beta_{s_1}\|},
\mathbf{y} = \frac{\beta_{s_2}}{\|\beta_{s_2}\|}.
$$
Figure 7.2: The coordinate frames for rolling. The body frame $x_B y_B z_B$ of object $B$ defined by its principal axes; the frame $x_p y_p z_p$ of plane $P$; the frame $x y z$ at the contact $q$ on $B$; the frame $x_p y_p z_p$ at the contact $p$ in $P$; and the world frame $x_W y_W z_W$. The frames $x_p y_p z_p$, $x_p y_p z_p$, and $x_W y_W z_W$ have the same orientation.

$$z = \frac{\beta_s \times \beta_s}{\left\| \beta_s \right\| \left\| \beta_s \right\|}.$$  

all with respect to the body frame $\Pi_B$. The parameters $s$ of $\beta$ are chosen such that $z$ is the outward normal. The orientation of frame $\Pi$ relative to the body frame $\Pi_B$ is thus given by the $3 \times 3$ rotation matrix

$$R = (x, y, z).$$  

Meanwhile, the contact point in the plane $P$ is denoted by $p = (u, 0)^T = (u_1, u_2, 0)^T$ in frame $\Pi_p$. We attach to $p$ a frame $\Pi_p$ (with axes $x_p, y_p, z_p$) with the same orientation as frames $\Pi_P$ and $\Pi_W$. Let $\psi$ be the angle of rotation needed to align axis $x$ with axis $x_p$ (see Figure 7.2). The matrix

$$R_{xy} = \begin{pmatrix} \cos \psi & -\sin \psi \\ -\sin \psi & \cos \psi \end{pmatrix}$$

therefore relates axes $x, y$ to axes $x_p, y_p$. Accordingly, $\Pi$ is related to $\Pi_p$ by the $3 \times 3$ rotation matrix

$$R_{pq} = \begin{pmatrix} R_{xy} & 0 \\ 0 & -1 \end{pmatrix}.$$
7.1. THE ROLLING MOTION

Consequently, the orientation of the body frame $\Pi_B$ relative to the contact frame $\Pi_p$ is given by the matrix

$$
R_{po} = R_{pq}R^T_p
= \begin{pmatrix}
R_\phi & 0 \\
0 & -1
\end{pmatrix}
(x, y, z)^T
= \begin{pmatrix}
R_\phi (x, y)^T \\
-z^T
\end{pmatrix}.
$$

(7.2)

Now we see that the orientation of $\mathcal{B}$ relative to $\mathcal{P}$ (and thereby to the world frame $\Pi_W$) is completely determined by $s$ and $\psi$, which represent the three degrees of freedom of $\mathcal{B}$ in order to maintain contact with $\mathcal{P}$. The velocity and angular velocity of $\mathcal{B}$ in $\Pi_W$ are given by $R_{po}v$ and $R_{po}\omega$, respectively.

The contact kinematics depends on the relative motion between the two contact frames $\Pi$ and $\Pi_p$. Denote by $(\omega_x, \omega_y, \omega_z)^T$ the angular velocity of $\Pi$ relative to $\Pi_p$ and in terms of $\Pi$. Since $\Pi$ is fixed relative to $\Pi_B$ and the angular velocity of $\Pi_B$ relative to $\Pi_p$ is $\omega$, we have

$$
\begin{pmatrix}
\omega_x \\
\omega_y \\
\omega_z
\end{pmatrix}
= R^T\omega = (x, y, z)^T\omega.
$$

(7.3)

However, $\omega_z = 0$ due to rolling without slipping. This imposes a constraint on the angular velocity $\omega$:

$$
z \cdot \omega = 0.
$$

(7.4)

Also due to rolling the relative velocity of $\Pi$ to $\Pi_p$ is zero:

$$
v_x = v_y = v_z = 0.
$$

Therefore the absolute velocity of $\mathbf{q}$ equals $v_P$; in other words,

$$
R_{po}(v_B + \omega_B \times \beta) = v_P,
$$

(7.5)

or equivalently,

$$
R_{po}(v + \omega \times \beta) = v_P.
$$

(7.6)

Thus the velocity $v$ is determined as

$$
v = R^T_{po}v_P - \omega \times \beta.
$$

(7.7)

The local geometry of $\mathcal{B}$ and $\mathcal{P}$ at the contact plays a major role in the contact kinematics. We will utilize Montana’s kinematic equations of contact [106] that relates the contact
motions to the relative motion of the two frames (in this case \( \Pi \) and \( \Pi_p \)) at the contact. First of all, we compute the shape operator, geodesic curvatures, and metric of \( \mathcal{B} \) as

\[
S = (\mathbf{x}, \mathbf{y})^T(-\nabla_x z, -\nabla_y z)
\]

\[
= \begin{pmatrix}
\kappa_1 & 0 \\
0 & \kappa_2
\end{pmatrix},
\]

\[
K_g = \mathbf{y}^T(\nabla_x x, \nabla_y x)
\]

\[
= (\kappa_g_1, \kappa_g_2),
\]

\[
V = \begin{pmatrix}
\|\beta_s_1\| & 0 \\
0 & \|\beta_s_2\|
\end{pmatrix},
\]

and those corresponding invariants of \( \mathcal{P} \) as

\[
S_P = 0,
\]

\[
K_{gP} = 0,
\]

\[
V_P = I_2,
\]

where \( \kappa_1 \) and \( \kappa_2 \) are the principal curvatures of \( \beta \) at \( q \), \( \kappa_g_1 \) and \( \kappa_g_2 \) the geodesic curvatures of the principal curves at \( q \), and \( I_2 \) the \( 2 \times 2 \) identity matrix. The kinematics of rolling follow from the above forms, (7.4), and Montana’s equations:

\[
\dot{\mathbf{u}} = R_\psi S^{-1}(\mathbf{y}, -\mathbf{x})^T \mathbf{\omega};
\]

(7.8)

\[
\dot{s} = V^{-1} S^{-1}(\mathbf{y}, -\mathbf{x})^T \mathbf{\omega};
\]

(7.9)

\[
\dot{\psi} = K_g S^{-1}(\mathbf{y}, -\mathbf{x})^T \mathbf{\omega}.
\]

(7.10)

Since the patch \( \beta \) has Gaussian curvature \( K = \det S \neq 0 \), the shape operator \( S \) is invertible.

Equations (7.8), (7.9), and (7.10) shall be better understood in the following geometric way. Let \( \rho_1 \) and \( \rho_2 \) be the radii of curvature of the normal sections in the principal directions \( x \) and \( y \) at the contact \( q \), respectively. From the convexity of \( \beta \) we have \( \rho_1 = -\frac{1}{\kappa_1} \) and \( \rho_2 = -\frac{1}{\kappa_2} \). For simplicity, let us assume the principal curves to be unit-speed; that is, \( \|\beta_{s_1}\| = \|\beta_{s_2}\| = 1 \). Hence equation (7.9) reduces to

\[
\dot{s} = I_2 \begin{pmatrix}
\rho_1 & 0 \\
0 & \rho_2
\end{pmatrix} \begin{pmatrix}
-\omega_y \\
\omega_x
\end{pmatrix}
\]

\[
= \begin{pmatrix}
-\rho_1 \omega_y \\
\rho_2 \omega_x
\end{pmatrix}.
\]

Figure 7.3 shows the normal sections of \( \beta \) in the \( x \) and \( y \) directions. Since \( q \) has zero relative velocity to \( p \), the angular velocity component \( \omega_y \) generates a contact velocity component of \( -\omega_y \rho_1 \) along the \( s_1 \)-parameter curve at \( q \), while the component \( \omega_x \) generates a contact
7.1. THE ROLLING MOTION

Figure 7.3: The normal sections of a pure rolling object $B$'s surface in the principal directions $x$ and $y$ at the contact. The radii of curvature are $\rho_1$ and $\rho_2$, respectively. The angular velocities at the centers of curvature about $y$ and $x$ are $\omega_y$ and $\omega_x$, respectively. It is clear that the velocities of contact in $x$ and $y$ are $-\rho_1 \omega_y$ and $\rho_2 \omega_x$, respectively.

velocity component of $\omega_x \rho_2$ along the $s_2$-parameter curve at $q$. Under rolling, the velocity $\dot{u}$ is related to the velocity $\dot{s}$ by matrix $R_\psi$. Equation (7.10), now written as,

$$\dot{\psi} = (\kappa_{gx}, \kappa_{gy}) \dot{s},$$

has a similar explanation from the definition of geodesic curvature.

7.1.2 Dynamics of Rolling

To apply Newton’s second law, we need to use a fixed reference frame with respect to which the velocity and angular velocity can be properly differentiated. Let the reference frame be the fixed frame that instantaneously coincides with frame $\Pi_B$. Therefore we shall use the absolute velocities $\dot{v}_B$ and $\dot{\omega}_B$ in this fixed frame.

Although $v$ and $\omega$ are relative velocities in terms of the moving frame $\Pi_B$, it is not difficult to derive that

$$\dot{v}_B = \omega \times v + \dot{v};$$
$$\dot{\omega}_B = \omega \times \omega + \dot{\omega} = \dot{\omega}.$$  \hspace{1cm} (7.11)

(7.12)

Let $F$ be the contact force on object $B$. The dynamics of the object obey Newton’s and Euler’s equations:

$$F + mgz = m\dot{v}_B = m(\dot{v} + \omega \times v);$$
$$\beta(s) \times F = I\dot{\omega}_B + \omega_B \times I\omega_B = I\dot{\omega} + \omega \times I\omega.$$  

The reader may either try a derivation himself or refer to MacMillan [100, pp. 175-176].
Immediately from the above equations we can eliminate the contact force \( F \) which may be anywhere inside the contact friction cone:

\[
\beta \times m(\dot{v} + \omega \times v - gz) = I \dot{\omega} + \omega \times I \omega. \tag{7.13}
\]

Next, differentiate the kinematic constraint (7.5) and plug (7.11) and (7.12) in:

\[
(R_{po} \omega) \times R_{po}(v + \omega \times \beta) + R_{po}(\dot{v} + \omega \times v + \dot{\omega} \times \beta + \omega \times (\omega \times \beta + \dot{\beta})) = a_P,
\]

where \( a_P \) is the acceleration of the plane \( P \). The above equation is simplified to

\[
2\omega \times (v + \omega \times \beta) + \dot{v} + \dot{\omega} \times \beta + \omega \times \dot{\beta} = R^T_{po} a_P, \tag{7.14}
\]

since

\[
(R_{po} \omega) \times R_{po}(v + \omega \times \beta) = R_{po}(\omega \times (v + \omega \times \beta)).
\]

From (7.14) we obtain \( \dot{v} \) and plug it into (7.13). After several steps of symbolic manipulation, we have

\[
\dot{\omega} = D^{-1}\left( \beta \times R^T_{po} a_P - \beta \times (gz + \omega \times v + 2\omega \times (\omega \times \beta) + \omega \times \dot{\beta}) - \omega \times \frac{I}{m} \omega \right), \tag{7.15}
\]

where

\[
\begin{align*}
D &= \frac{I}{m} + \|\beta\|^2 I_3 - \beta \beta^T, \tag{7.16} \\
v &= R^T_{po} v_P - \omega \times \beta, \quad \text{from (7.6).}
\end{align*}
\]

In the above \( I_3 \) is the \( 3 \times 3 \) identity matrix. The matrix \( D \) is positive definite, following that the angular inertia matrix \( I \) is positive definite. More specifically, for any \( 3 \times 1 \) vector \( e \neq 0 \), we have

\[
e^T D e = e^T \left( \frac{I}{m} + \|\beta\|^2 I_3 - \beta \beta^T \right) e
\]

\[
= e^T \frac{I}{m} e + \|\beta\|^2 \|e\|^2 - (e \cdot \beta)^2
\]

\[
> 0
\]

**Theorem 24** The nonlinear system consisting of equations

\[
\begin{align*}
\dot{u} &= R_{\psi} S^{-1}(y, -x)^T \omega, \\
\dot{s} &= V^{-1} S^{-1}(y, -x)^T \omega, \\
\dot{\psi} &= K_{g} S^{-1}(y, -x)^T \omega, \\
\dot{\omega} &= D^{-1} \beta \times R^T_{po} a_P - D^{-1}\left( \beta \times (gz + \omega \times v + 2\omega \times (\omega \times \beta) + \omega \times \dot{\beta}) + \omega \times \frac{I}{m} \omega \right),
\end{align*} \tag{7.17}
\]

governs the object motion as well as the contact motions on the object and in the plane.
7.2. Local Observability of Rolling

Both the rolling constraints (7.4) and (7.6) were used in deriving system (7.17) and are therefore implicitly included in the system. Nevertheless, \( v \) is a redundant state variable of the system as it depends on \( \omega \) and \( s \); and (7.4) induces another redundancy among \( \omega_1, \omega_2, \) and \( \omega_3. \)

A state of the nonlinear system (7.17) has eight (scalar) variables including \( u = (u_1, u_2)^T, s = (s_1, s_2)^T, \psi, \) and \( \omega = (\omega_1, \omega_2, \omega_3)^T. \) Together they determine the object’s position and orientation relative to frame \( \Pi_P. \) Thus given the plane motion, they determine the object’s motion relative to frame \( \Pi_S. \) The one redundancy among \( \omega_1, \omega_2, \omega_3 \) will be taken care of when we investigate the local observability of system (7.17) in the next section. Note there is no need to treat \( v_P \) as a state variable since it is known.

The output of the system is \( u = (u_1, u_2)^T. \) The input of the system is \( a_P = (a_{P_1}, a_{P_2}, a_{P_3})^T \) whose components control the input fields

\[
\begin{align*}
\begin{pmatrix} h_1 \\ h_2 \\ h_3 \end{pmatrix} &= \begin{pmatrix} 0 \\ \tilde{h}_1 \\ \tilde{h}_2 \\ \tilde{h}_3 \end{pmatrix}, \quad \begin{pmatrix} 0 \\ \tilde{h}_1 \\ \tilde{h}_2 \\ \tilde{h}_3 \end{pmatrix}, \quad \text{and} \quad \begin{pmatrix} 0 \\ \tilde{h}_1 \\ \tilde{h}_2 \\ \tilde{h}_3 \end{pmatrix},
\end{align*}
\]

respectively, where the vector fields \( \tilde{h}_1, \tilde{h}_2, \tilde{h}_3 \) together define a \( 3 \times 3 \) matrix:

\[
\begin{align*}
(\tilde{h}_1, \tilde{h}_2, \tilde{h}_3) &= D^{-1} \beta \times R_{po}^T \\
&= D^{-1} \beta \times (x, y) R_{\psi}, -z.
\end{align*}
\]

Denote by \( f \) the drift field of the system, which is composed of the right sides of the equations (7.17) when \( a_P = 0. \) It measures the rate of state change under zero input.

## 7.2. Local Observability of Rolling

In the previous section we studied the kinematics and dynamics of rolling and derived the nonlinear system (7.17) that describes the object and contact motions. The unknowns include the point of contact \( u \) in the plane, the point of contact on the object as determined by \( s, \) the rotation \( \psi \) of the object about the contact normal, and the object’s angular velocity \( \omega. \) These variables constitute a state in system (7.17)’s state space manifold \( M. \) This section will look into whether knowing \( u \) is sufficient for locally determining \( s, \psi, \) and \( \omega. \)

Our approach is to investigate the local observability of system (7.17). Denote by \( \xi \) the current state of rolling so that \( \xi = (u, s, \psi, \omega)^T. \) Essentially, we need to determine if the observability codistribution \( d\mathcal{O} \) at \( \xi \) equals to the cotangent space \( T^*_\xi M \) spanned by the differentials \( du, ds, d\psi, d\omega \) at \( \xi. \) The codistribution \( d\mathcal{O} \) consists of the differentials of \( u_1 \) and \( u_2 \) and their higher order Lie derivatives with respect to \( f, h_1, h_2, h_3. \) The linear space \( \mathcal{O} \) spanned by these functions is called the observation space of the system.

We shall decompose the cotangent space \( T^*_\xi M \) at state \( \xi \) into subspaces \( T^*_\xi M_u, T^*_\xi M_s, T^*_\xi M_\psi, T^*_\xi M_\omega, \) with bases \( du, ds, d\psi, d\omega, \) respectively. Obviously, \( T^*_\xi M_u \) is spanned since \( du \in d\mathcal{O}. \) First, we shall show that the cotangent space \( T^*_\xi M_\omega \) is spanned. Second, we shall derive a sufficient condition about the contact geometry that guarantees the cotangent space
$T^*_\xi M_s$ to be spanned. Third, we shall extend this sufficient condition for the spanning of $T^*_\xi M_s \times T^*_\xi M_\psi$. Finally, we shall argue that the differentials chosen from the observability codistribution $dO$ to span these subspaces will span $T^*_\xi M$ under the combined sufficient conditions.

### 7.2.1 Angular Velocity

As mentioned in Section 7.1.2, one of the angular velocity components $\omega_1, \omega_2, \omega_3$ is a redundant state variable under the rolling constraint (7.4). Let the coordinates of $x, y, z$ in frame $\Pi_B$ be $(x_1, x_2, x_3), (y_1, y_2, y_3), (z_1, z_2, z_3)$, respectively. Without loss of generality, we assume $z_3 \neq 0$ in some neighborhood of the state. Immediately from (7.4)

$$\omega_3 = -\frac{z_1 \omega_1 + z_2 \omega_2}{z_3}. \quad (7.19)$$

The new system rewritten from system (7.17) using (7.19) has seven state variables $u, s, \psi, \omega_1, \omega_2$. The new observation space equals the old one (denoted $O$), except that every appearance of $\omega_3$ is now replaced with (7.19). For this reason, we will not distinguish the new observation space from $O$ unless taking the partial derivatives of functions in $O$ with respect to $\omega_1$ and $\omega_2$, on which $\omega_3$ has become dependent.

The cotangent space $T^*_\xi M_\omega$ is a two-dimensional and spanned by $d\omega_1(\xi)$ and $d\omega_2(\xi)$. To show the spanning of $T^*_\xi M_\omega$ by differentials in the codistribution $dO$, it suffices to prove that the Jacobian matrix $\frac{\partial L_f u}{\partial (\omega_1, \omega_2)}$ has rank 2, where $L_f u = (L_f u_1, L_f u_2)^T$. This Jacobian is given as

$$\frac{\partial L_f u}{\partial (\omega_1, \omega_2)} = \frac{\partial L_f u}{\partial \omega} \frac{\partial \omega}{\partial (\omega_1, \omega_2)}$$

$$= \frac{\partial \dot{u}}{\partial \omega} \frac{\partial \omega}{\partial (\omega_1, \omega_2)}$$

$$= R_\psi S^{-1} \begin{pmatrix} y_1 & y_2 & y_3 \\ -x_1 & -x_2 & -x_3 \end{pmatrix} \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix} \begin{pmatrix} -\frac{x_1}{z_3} & -\frac{x_2}{z_3} & -\frac{x_3}{z_3} \\ \frac{y_1}{z_3} & \frac{y_2}{z_3} & \frac{y_3}{z_3} \end{pmatrix}$$

$$= R_\psi S^{-1} \begin{pmatrix} \frac{-x_1}{z_3} & \frac{x_1}{z_3} \\ \frac{-x_2}{z_3} & \frac{-x_2}{z_3} \end{pmatrix}.$$

Computing the determinant of the Jacobian is straightforward:

$$\det \left( \frac{\partial L_f u}{\partial (\omega_1, \omega_2)} \right) = \det R_\psi \det S^{-1} \frac{x_1 y_2 - x_2 y_1}{z_3^2}$$

$$= \frac{1}{K z_3},$$

where $K \neq 0$ is the Gaussian curvature at the contact point $q$. 


7.2. LOCAL OBSERVABILITY OF ROLLING

Proposition 25 The cotangent space $T^*_\xi \mathcal{M}_\omega$ is spanned by the differentials $dLf_u$.

This lemma states that rolling induces a one-to-one mapping from the space of angular velocities to the space of contact velocities in the plane for constant $s$ and $\psi$. Accordingly, there exists some control that can distinguish between any two different (but close enough) angular velocities of the object in the same pose.

7.2.2 Contact on the Object

We now proceed to determine whether the cotangent space $T^*_s \mathcal{M}_s$ is spanned by the differentials in $d\mathcal{O}$. This time we choose from $\mathcal{O}$ six (scalar) functions $L_1, L_f u, L_2, L_f u, L_3, L_f u$ comprising the following $2 \times 3$ matrix:

$$
Q_\psi = (L_1, L_f u, L_2, L_f u, L_3, L_f u)
= R_\psi S^{-1}(y, -x)^T D^{-1}\beta \times ((x, y)R_\psi, -z). \tag{7.20}
$$

The cotangent space $T^*_s \mathcal{M}_s$ is spanned provided the partial derivatives $\frac{\partial Q_\psi}{\partial s_1}$ and $\frac{\partial Q_\psi}{\partial s_2}$, viewed as two 6-dimensional vectors, are linearly independent. Introducing a $2 \times 3$ matrix:

$$
Q = S^{-1}(y, -x)^T D^{-1}\beta \times R, \tag{7.21}
$$

where $R = (x, y, z)$ as given by (7.1), we have

Lemma 26 The partial derivatives $\frac{\partial Q_\psi}{\partial s_1}$ and $\frac{\partial Q_\psi}{\partial s_2}$ are linearly dependent if and only if the partial derivatives $\frac{\partial Q}{\partial s_1}$ and $\frac{\partial Q}{\partial s_2}$ are linearly dependent.

Proof Suppose there exist $c_1$ and $c_2$, not both zero, such that

$$
c_1 \frac{\partial Q_\psi}{\partial s_1} + c_2 \frac{\partial Q_\psi}{\partial s_2} = 0.
$$

Expand this equation into two:

$$
c_1 R_\psi \frac{\partial}{\partial s_1}(S^{-1}(y, -x)^T D^{-1}\beta \times (x, y))R_\psi 
+ c_2 R_\psi \frac{\partial}{\partial s_2}(S^{-1}(y, -x)^T D^{-1}\beta \times (x, y))R_\psi = 0; \tag{7.22}
$$

$$
c_1 R_\psi \frac{\partial}{\partial s_1}(S^{-1}(y, -x)^T D^{-1}\beta \times (-z)) 
+ c_2 R_\psi \frac{\partial}{\partial s_2}(S^{-1}(y, -x)^T D^{-1}\beta \times (-z)) = 0. \tag{7.23}
$$

Multiply (7.22) by $R_\psi$ on both the left and the right and (7.23) by $-R_\psi$ on the left; and merge the two resulting equations into one:

$$
c_1 \frac{\partial}{\partial s_1}(S^{-1}(y, -x)^T D^{-1}\beta \times (x, y, z)) + c_2 \frac{\partial}{\partial s_2}(S^{-1}(y, -x)^T D^{-1}\beta \times (x, y, z)) = 0.
$$
Hence $\partial Q/\partial s_1$ and $\partial Q/\partial s_2$ are linearly dependent.

Conversely.

Thus we will focus the investigation on the linear independence of the partial derivatives $\partial Q/\partial s_1$ and $\partial Q/\partial s_2$. Unfortunately, for certain shape $\beta$, these two partial derivatives are linearly dependent. For example, if $\beta$ is a sphere with radius $r$, one can verify that\(^\text{2}\)

$$Q = \begin{pmatrix} -\frac{5}{7} & 0 & 0 \\ 0 & -\frac{5}{7} & 0 \end{pmatrix}.$$ 

The partial derivatives of $Q$ are zero and obviously linearly dependent. In this example, the contact point can be anywhere on the sphere whatever path the sphere rolls along in the supporting plane.

But objects of more general shape are to our real interest. So we aim at establishing some condition on the linear independence of $\partial Q/\partial s_1$ and $\partial Q/\partial s_2$ that can be satisfied by most shapes.

To compute these two partial derivatives, we first obtain the partial derivatives of $x, y, z$:

$$\frac{\partial}{\partial s_1}(x, y, z) = \|\beta_s\|(x, y, z) \begin{pmatrix} 0 & -\kappa g_1 & -\kappa_1 \\ \kappa g_1 & 0 & 0 \\ \kappa_1 & 0 & 0 \end{pmatrix};$$

$$\frac{\partial}{\partial s_2}(x, y, z) = \|\beta_s\|(x, y, z) \begin{pmatrix} 0 & -\kappa g_2 & 0 \\ \kappa g_2 & 0 & -\kappa_2 \\ 0 & \kappa_2 & 0 \end{pmatrix}.$$ 

Let $A = D^{-1}\beta\times$, where

$$\beta\times = \begin{pmatrix} 0 & -\beta_3 & \beta_2 \\ \beta_3 & 0 & -\beta_1 \\ -\beta_2 & \beta_1 & 0 \end{pmatrix}$$

with $\beta = (\beta_1, \beta_2, \beta_3)^T$ in frame $\Pi_B$ is the skew-symmetric matrix representing cross product. Then $DA = \beta\times$. Differentiate this equation with respect to $s_1$ ($s_2$, respectively) on both sides and solve for the partial derivatives of $A$:

$$\frac{\partial}{\partial s_1}(D^{-1}\beta\times) = D^{-1}\left(\|\beta_s\|\times - \frac{\partial D}{\partial s_1}D^{-1}\beta\times\right);$$

$$\frac{\partial}{\partial s_2}(D^{-1}\beta\times) = D^{-1}\left(\|\beta_s\|\times - \frac{\partial D}{\partial s_2}D^{-1}\beta\times\right).$$

\(^2\)Use, for instance, the parametrization $\beta = r(\cos s_1 \cos s_2, -\cos s_1 \sin s_2, \sin s_1)^T$. 
7.2. LOCAL OBSERVABILITY OF ROLLING

With all the above partial derivatives we can write out the partial derivatives of $Q$:

\[
\frac{\partial Q}{\partial s_1} = \left( \frac{\partial S^{-1}}{\partial s_1}(y, -x)^T - \|\beta_{s_1}\| S^{-1}(\kappa_{g_1} x, \kappa_{g_1} y + \kappa_1 z)^T \right) D^{-1} \beta \times R \\
+ S^{-1}(y, -x)^T D^{-1} \left( -\frac{\partial D}{\partial s_1} D^{-1} \beta \times R \right) \\
+ \|\beta_{s_1}\| \left( (0, z, -y) + \beta \times (\kappa_{g_1} y + \kappa_1 z, -\kappa_{g_1} x, -\kappa_1 x) \right); \quad (7.24)
\]

\[
\frac{\partial Q}{\partial s_2} = \left( \frac{\partial S^{-1}}{\partial s_2}(y, -x)^T - \|\beta_{s_2}\| S^{-1}(\kappa_{g_2} x - \kappa_2 z, \kappa_{g_2} y)^T \right) D^{-1} \beta \times R \\
+ S^{-1}(y, -x)^T D^{-1} \left( -\frac{\partial D}{\partial s_2} D^{-1} \beta \times R \right) \\
+ \|\beta_{s_2}\| \left( (-z, 0, x) + \beta \times (\kappa_{g_2} y, -\kappa_{g_2} x + \kappa_2 z, -\kappa_2 y) \right). \quad (7.25)
\]

It seems very difficult to say anything about the linear independence of $\partial Q/\partial s_1$ and $\partial Q/\partial s_2$ directly from their complicated forms above. So we intend to find a matrix $A$ and a sufficient condition under which $(\partial Q/\partial s_1)A$ and $(\partial Q/\partial s_2)A$ will be linearly independent. The same condition will then ensure the linear independence between $\partial Q/\partial s_1$ and $\partial Q/\partial s_2$. The matrix of our choice is $R^T \beta$, which, when multiplied on (7.24) and (7.25) to the right, eliminates those terms in $\partial Q/\partial s_1$ and $\partial Q/\partial s_2$ that have $\beta \times R$ on the right. We can further simplify the remaining terms in the products using the following equations:

\[
(0, z, -y)^T R^T = x; \\
(-z, 0, x)^T R^T = y; \\
(\kappa_{g_1} y + \kappa_1 z, -\kappa_{g_1} x, -\kappa_1 x)^T R^T = \kappa_{g_1} z \times -\kappa_1 y \times; \\
(\kappa_{g_2} y, -\kappa_{g_2} x + \kappa_2 z, -\kappa_2 y)^T R^T = \kappa_{g_2} z \times + \kappa_2 x \times.
\]

In the end, we obtain

\[
\frac{\partial Q}{\partial s_1} R^T \beta = \|\beta_{s_1}\| S^{-1}(y, -x)^T D^{-1} \beta \times b_1; \quad (7.26)
\]

\[
\frac{\partial Q}{\partial s_2} R^T \beta = \|\beta_{s_2}\| S^{-1}(y, -x)^T D^{-1} \beta \times b_2; \quad (7.27)
\]

where

\[
b_1 = -x + (\kappa_{g_1} z - \kappa_1 y) \times \beta; \quad (7.28)
\]

\[
b_2 = -y + (\kappa_{g_2} x + \kappa_2 z) \times \beta. \quad (7.29)
\]

**Lemma 27** The vectors $(\partial Q/\partial s_1) R^T \beta$ and $(\partial Q/\partial s_2) R^T \beta$ are linearly dependent if and only if $\beta \cdot (Iz) = 0$ or $\det(b_1, b_2, \beta) = 0$. 
Proof We need to conduct a few steps of reasoning:

\[
\frac{\partial Q}{\partial s_1} R^T \beta \quad \text{and} \quad \frac{\partial Q}{\partial s_2} R^T \beta \quad \text{are linearly dependent}
\]

if and only if

\[
(y, -x)^T D^{-1} \beta \times b_1 \quad \text{and} \quad (y, -x)^T D^{-1} \beta \times b_2 \quad \text{are linearly dependent}
\]

if and only if

\[
D^{-1} \beta \times b_1, \quad D^{-1} \beta \times b_2, \quad \text{and} \quad z \quad \text{are linearly dependent}
\]

if and only if

\[
\beta \times b_1, \quad \beta \times b_2, \quad \text{and} \quad Dz \quad \text{are linearly dependent},
\]

where

\[
Dz = \left( \frac{I}{m} + \|\beta\|^2 I_3 - \beta \beta^T \right) z = \frac{I}{m} z + \beta \times (z \times \beta).
\]

There are two cases: (1) \( \beta \cdot (Iz) = 0 \) and (2) \( \beta \cdot (Iz) \neq 0 \). In case (1), \( \beta \times b_1, \beta \times b_2, Dz \) are linearly dependent. In case (2), they are linearly dependent if and only if \( \beta \times b_1 \) and \( \beta \times b_2 \) are linearly dependent, which happens if and only if \( \det(b_1, b_2, \beta) = 0 \). \( \square \)

Combining Lemmas 26 and 27, we arrive at a sufficient condition for the spanning of \( T^*_s M_s \).

Proposition 28 The cotangent space \( T^*_s M_s \) is spanned by the differentials of six functions \( L_h L_f u, L_{h_2} L_f u, L_{h_3} L_f u \), all in the observation space of system (7.17), if (1) \( \beta \cdot (Iz) \neq 0 \) and (2) \( \det(b_1, b_2, \beta) \neq 0 \).

When \( \beta \) is a sphere, \( b_1 = b_2 = 0 \), violating condition (2) in Proposition 28. For general \( \beta \), conditions (1) and (2) in Proposition 28 are satisfied at all but at most an one-dimensional set of contact points.

Up until now we have selected from the observation space \( \mathcal{O} \) a total of 10 scalar functions: \( u, L_f u, L_{h_2} L_f u, L_{h_3} L_f u \), and \( L_{h_3} L_f u \). The first two functions depend on \( u \) only, while the last six functions on only \( s \) and \( \psi \). Hence we have shown that the cotangent space \( T^*_s M_u \times T^*_s M_s \times T^*_s M_\omega \) is spanned unless \( \beta \cdot (Iz) = 0 \) or \( \det(b_1, b_2, \beta) = 0 \).

7.2.3 Rotation about the Contact Normal

The rolling object has three degrees of freedom: the point of contact on the object \( s \) and the rotation \( \psi \) of the object about the contact normal. The last proposition states that the contact point alone can be distinguished under a sufficient condition. Such condition
7.2. LOCAL OBSERVABILITY OF ROLLING

essentially ensures the differentials of \( L_{h_{1}}L_{f}u \), \( L_{h_{2}}L_{f}u \), and \( L_{h_{3}}L_{f}u \) to span the cotangent space \( T_{x}^{*}M_{x} \). Now we shall show that these differentials indeed span the cotangent space \( T_{x}^{*}M_{x} \times T_{x}^{*}M_{\psi} \) except for special shapes such as a sphere. That is, we shall show that the 2\times3 matrices \( \partial Q_{\psi}/\partial s_{1} \), \( \partial Q_{\psi}/\partial s_{2} \), and \( \partial Q_{\psi}/\partial \psi \) are linearly independent for general shape \( \beta \).

Let \( Q_{12} \) be the 2\times2 matrix formed by the first two columns in \( Q \):

\[
Q_{12} = S^{-1}(y, -x)^{T}D^{-1}\beta \times (x, y).
\]

Construct a 2\times3 matrix:

\[
\tilde{Q} = \begin{pmatrix} 0 & -1 \\ 1 & 0 \end{pmatrix} Q + Q_{12} \begin{pmatrix} 0 & 1 & 0 \\ -1 & 0 & 0 \end{pmatrix}.
\]

**Lemma 29** The partial derivatives \( \partial Q_{\psi}/\partial s_{1} \), \( \partial Q_{\psi}/\partial s_{2} \), \( \partial Q_{\psi}/\partial \psi \) are linearly dependent if and only if \( \tilde{Q} \) and the partial derivatives \( \partial Q/\partial s_{1} \), \( \partial Q/\partial s_{2} \) are linearly dependent.

The proof of Lemma 29 is similar to that of Lemma 26, and makes use of the following equations:

\[
R_{\psi} \frac{\partial R_{\psi}}{\partial \psi} = \begin{pmatrix} 0 & -1 \\ 1 & 0 \end{pmatrix};
\]

\[
\frac{\partial R_{\psi}}{\partial \psi} R_{\psi} = \begin{pmatrix} 0 & 1 \\ -1 & 0 \end{pmatrix}.
\]

Lemma 29, along with Propositions 25 and 28, suggest the weak dependency, if any, of system (7.17)’s local observability on the object’s rotation angle \( \psi \).

In this section, we assume that \( \partial Q/\partial s_{1} \) and \( \partial Q/\partial s_{2} \) are linearly independent. We want to determine if \( \tilde{Q} \) is linearly independent of them. If not always, we would like to seek some sufficient condition on the object shape \( \beta \). Let us start by offering a rather restrictive necessary condition for \( \tilde{Q} \) to vanish.

**Lemma 30** The matrix \( \tilde{Q} \) is equal to 0 only if the following three conditions all hold:

\[
\beta \times z = 0;
\]

\[
\kappa_{1}x^{T}D^{-1}x = \kappa_{2}y^{T}D^{-1}y;
\]

\[
x^{T}D^{-1}y = 0.
\]

**Proof** There are two cases: (1) \( \beta \times z \neq 0 \), and (2) \( \beta \times z = 0 \).

In the first case, we have \( D^{-1}\beta \times z \neq 0 \) due to that \( D^{-1} \) is positive definite. The third column of \( \tilde{Q} \) is not zero, for we know:

\[
\begin{pmatrix} 0 & -1 \\ 1 & 0 \end{pmatrix} S^{-1}(y, -x)^{T}D^{-1}\beta \times z = 0
\]
Multiply both sides of (7.31) by $\beta \times z = 0$

if and only if

$$ (y, -x)^T D^{-1} \beta \times z = 0 $$

if and only if

$$ D^{-1} \beta \times z = cz, \quad \text{for some } c \neq 0 $$

if and only if

$$ \beta \times z = cDz, \quad \text{for some } c \neq 0, $$

only if

$$ 0 = z \cdot (\beta \times z) = z^T Dz. $$

But $z^T Dz > 0$, as $D$ is positive definite.

In the second case, the third column of $Q$ is zero. Since $\beta = \|\beta\|z$, we have

$$ Q_{12} = \|\beta\|S^{-1}(y, -x)^T D^{-1} z \times (x, y) $$

$$ = \|\beta\|S^{-1}(y, -x)^T D^{-1}(y, -x). $$

The $2 \times 2$ matrix $(y, -x)^T D^{-1}(y, -x)$ is positive definite, so we write

$$ Q_{12} = \|\beta\|S^{-1}\begin{pmatrix} a & b \\ b & d \end{pmatrix}, \quad a > 0 \text{ and } ad - b^2 > 0. $$

With the above form of $Q_{12}$ we are able to simplify the first two columns of $\tilde{Q}$:

$$ \begin{pmatrix} 0 & -1 \\ 1 & 0 \end{pmatrix} Q_{12} + Q_{12} \begin{pmatrix} 0 & 1 \\ -1 & 0 \end{pmatrix} = \|\beta\|\begin{pmatrix} \frac{b}{\kappa_1} - \frac{b}{\kappa_2} \\ \frac{a}{\kappa_1} - \frac{d}{\kappa_2} \end{pmatrix}. $$

The above $2 \times 2$ matrix is zero if and only if $b = 0$ and $a = \frac{d}{\kappa_2}$, which imply conditions (7.30).

Suppose the necessary conditions in Lemma 30 does not hold. Then $\tilde{Q} \neq 0$. Suppose $\partial Q/\partial s_1$ and $\partial Q/\partial s_2$ are linearly independent. The matrices $\partial Q/\partial s_1$, $\partial Q/\partial s_2$, and $\tilde{Q}$ are linearly dependent if and only if there exist $c_1, c_2$, not all zero, such that

$$ c_1 \frac{\partial Q}{\partial s_1} + c_2 \frac{\partial Q}{\partial s_2} = \begin{pmatrix} 0 & -1 \\ 1 & 0 \end{pmatrix} Q + Q_{12} \begin{pmatrix} 0 & 1 & 0 \\ 1 & 0 & 0 \end{pmatrix}. $$

Multiply both sides of (7.31) by $R^T \beta$ on the right, plug (7.26) and (7.27) in, and then multiply both sides of the resulting equation by $S$ on the left:

$$ c_1\|\beta_s\|(y, -x)^T D^{-1} \beta \times b_1 + c_2\|\beta_s\|(y, -x)^T D^{-1} \beta \times b_2 = (y, -x)^T D^{-1} \beta \times (\beta \times z). $$

Solving the above equation for $c_1$ and $c_2$ and substituting the solutions into (7.31), we end up with a system of third order partial differential equations of $\beta$, four of which are independent. We postulate that this PDE system has at most one solution.

**Proposition 31** Assume the differentials $dL_{b_1} L_f u$, $dL_{b_2} L_f u$, and $dL_{x_3} L_f u$ span the cotangent space $T^*_x M_s$. Furthermore, assume the necessary condition in Lemma 30 does not hold. The same differentials will also span the cotangent space $T^*_x M_s \times T^*_x M_\psi$ except for at most one (local) shape of $\beta$. 
7.3. SUMMARY

7.2.4 Sufficient Condition for Local Observability

To summarize, we have chosen from the observation space $\mathcal{O}$ a total of ten functions: two scalar functions $\mathbf{u}$, which are the system outputs; two first order Lie derivatives $L_f \mathbf{u}$, which involve $\omega_1, \omega_2, s_1, s_2, \psi$; and six second order Lie derivatives $L_{h_1} L_f \mathbf{u}, L_{h_2} L_f \mathbf{u}, L_{h_3} L_f \mathbf{u}$, which involve $s_1, s_2, \psi$ only. The differentials of these functions (except $L_f \mathbf{u}$) live in orthogonal subspaces of the cotangent space $T^*_\xi M$. If these subspaces are spanned, given that $T^*_\xi M_\omega$ is spanned by $dL_f \mathbf{u}$, $T^*_\xi M$ is also spanned and local observability of system (7.17) follows.

Combining Proposition 25 through Proposition 31, we obtain a sufficient condition for the local observability of rolling:

**Theorem 32** The rolling system (7.17) is locally observable except for at most one object shape $\beta$ if

$$\text{det}(\mathbf{b}_1, \mathbf{b}_2, \beta) \neq 0 \quad (7.32)$$

and

$$\beta \cdot (Iz) \neq 0, \quad \text{when } \beta \times z \neq \mathbf{0}; \quad (7.33)$$

$$\kappa_1 \mathbf{x}^T D^{-1} \mathbf{x} \neq \kappa_2 \mathbf{y}^T D^{-1} \mathbf{y} \quad \text{or} \quad \mathbf{x}^T D^{-1} \mathbf{y} \neq 0, \quad \text{when } \beta \times z = \mathbf{0}, \quad (7.34)$$

where $\mathbf{b}_1$ and $\mathbf{b}_2$ are given by (7.28) and (7.29), respectively.\(^3\)

Condition (7.32) depends on the differential geometry of contact. It is only violated by at most a one-dimensional set of points in a general patch $\beta$. Condition (7.33) states that the contact normal transformed by the angular inertia matrix should not be perpendicular the contact location vector. It is only violated by at most a one-dimensional set of points in a general patch. Condition (7.34), only for $\beta \times z = \mathbf{0}$, is hardly seen to be violated by any shape other than a sphere.

7.3 Summary

We deal with local observability of a three-dimensional object rolling on a translating plane. The object is bounded by a smooth surface that makes point contact with the plane, which is free to accelerate in any direction. The plane is rough enough to allow only the pure rolling motion of the object. As the object rolls, the contact traces out a path in the plane, which can be detected by a tactile array sensor embedded in the plane.

Utilizing Montana’s equations for contact kinematics, we describe the kinematics and dynamics of this task by a nonlinear system, of which the output is the contact location in the plane. Then we establish a sufficient condition for the pose and motion of the object to be locally observable. This is done by decomposing the cotangent space at the current state

---

\(^3\)Note that $\beta \times z = \mathbf{0}$ implies $\beta \cdot (Iz) \neq 0$ given that $I$ is positive definite.
into orthogonal subspaces associated with the object’s pose and angular velocity, respectively, and by later combining the sufficient conditions on the spanning of these subspaces. The combined condition depends only on the object’s contact geometry and angular inertia matrix. And it is general enough to be satisfied almost everywhere on any object surface that does not assume certain degeneracies as a sphere does.

One direction of future work will be to design a nonlinear observer that can asymptotically estimate the pose and motion of a rolling object from the path of contact. Further along this line of research will be to implement (tactile) sensors and mount them on a palm to estimate the poses and motions of real objects.

Another direction is to study the controllability of the rolling object on a palm and to combine the result with the present local observability result for developing parts orienting and dextrous manipulation strategies. For example, the object is dropped onto the palm which first executes a sequence of motions to find out the object’s pose and then perform another sequence of motion move the object to the desired configuration.
Chapter 8

Conclusions

This thesis exploited geometry and mechanics in robotic tasks to devise simple, efficient, and dynamic strategies for sensing objects that have known shapes, and possibly, uniform mechanical properties (on mass distribution, pressure distribution, etc) as well. We showed how to compute an object’s static position and orientation from geometric constraints obtainable with simple sensors, and how to estimate its motion during a manipulation based on kinematics and dynamics. The work provided us with new insights into the interactions among various types of information (such as positions, orientations, velocities, forces, etc.) relevant to a robotic task; and into the representation of such information in various forms of sensible data. Most importantly, the thesis revealed that a large portion of the information about a manipulation task is indeed encoded, through manipulation itself, in simple tactile data such as contact; and demonstrated that such information can be decoded, at least locally, by means of nonlinear observers.

A sensing algorithm is viewed as the part of a sensing strategy that interprets the data acquired by physical sensors to derive the desired information about an object or a task. Therefore the power of a sensing algorithm is complementary to that of physical sensing. Such an algorithm may take the form of a computational algorithm making use of the object geometry, a nonlinear observer based on task mechanics, or something else. Just as geometric sensing in Chapters 2 and 3 resorted to computational algorithms and complexity theory, dynamic sensing in Chapters 5 and 7 applied nonlinear control techniques.

With known information such as the object geometry, we need to decide on how much extra information is to be obtained by physical sensor(s). This part of the information will be the sensor data. Then we design an algorithm that interprets the sensor data and computes the wanted information (such as pose, motion, etc) that otherwise would have to be acquired with more complicated hardware. However, the aforementioned two steps of sensing strategy design need not be sequential. What kind of physical data to be used is subject to the sensor availability; it is also influenced by the task to be performed and by the computational difficulty in interpreting such data. We may prefer one sensor over another for a variety of reasons: the former is more reliable and faster; it would make the design of a sensing algorithm easier; it would enable a faster sensing algorithm; it would be more suitable for the task execution; and so on. Coupled with different forms of sensor data, a
spectrum of sensing algorithms, varying as wide as from computational geometry algorithms to nonlinear control systems, can be developed.

Let us illustrate the above idea with a revisit to a simple and familiar example. It is known that the pose of a planar object, denoted by its position \((x, y)\) and orientation \(\theta\), needs at least three and often four geometric constraints to be fully constrained. Some previous work such as [24] and [75] described pose determination strategies using sufficient geometric constraints obtained by sensors. The algorithms employed in these strategies are simple and purely geometric.

What happens if sensors are unable to provide enough geometric constraints? In the case of a motionless finger in contact with a planar object, only two geometric constraints are imposed on the object pose: the contact location and the opposition of the fingertip and object normals at the contact. The role of pushing (as described in Chapter 5) is to generate a family of contact points evolving out of time,\(^1\) three of which would often be enough to disambiguate the initial pose. Since motion is now involved, the sensing algorithm is expected to become more complicated than purely geometric: In Chapter 5 it was an observer constructed over the system of nonlinear differential equations that governs pushing.

As a matter of fact, we can even waive one more geometric constraint for the sake of simplicity. Let the finger degenerate into a point and let the output of the resulting system be the magnitude of force exerted on this point during a push. The above principle of sensing is expected to still work. The set of possible initial poses are now two-dimensional under the incidence constraint induced by the point finger. Therefore we generally need at least four intermediate contacts to single out the real contact motion and subsequently to determine the real initial pose.

The above discussion touches the issue on the roles of geometry and of mechanics in the design of a sensing strategy. A manipulation operation enables a simple sensor to capture the evolution of certain geometric constraint(s) over time, which may give rise to enough information for a task that would otherwise require more powerful sensing hardware. In this sense, exploiting the mechanics of a task can help simplify sensing hardware as well as integrate sensing into manipulation (e.g., grasping).

A good sensing strategy should exploit as much as possible the sensors available as well as the geometry and mechanics of the task. The design of a sensing strategy should take into account not only how efficient a sensing algorithm can be developed but also how well the algorithm can be integrated into the execution of a task. Here we would like to refer the reader to Erdmann’s methodology [46] on task-specific and action-based sensor design. In the remainder of this chapter, we shall summarize the thesis contributions and address unresolved issues related to geometric sensing and mechanics-based observation.

\(^1\)It should be noted that all normal constraints except the initial one are redundant.
8.1 Issues on Geometric Sensing

In the context of geometric sensing, we view the expense of sensing as the combination of the sensor cost and the time cost. The latter part in turn includes the time to perform physical operations as well as the time for a geometric algorithm to process the obtained sensor data. The size (or number) of geometric constraints is directly related to the sensor cost and/or the time cost of physical sensing. For example, if we use rotary sensors to implement the inscription strategy described in Chapter 2, the number of needed sensors would be equal to the number of inscribing cones, which has been demonstrated to be two in most cases.\(^2\) The similar would be true if we embed light detectors to implement the point sampling strategy introduced in Chapter 3. However, if we resort to mechanical probes as an implementation of the same strategy, the number of sampling points would then directly reflect the probing time.

Chapter 2 aimed at addressing the simplicity, efficiency, and ambiguity of determining the poses of known objects from geometric constraints. We chose a problem in which a simple polygon has a three-dimensional set of possible poses, and showed how its real pose can be derived from four geometric constraints of the simplest kind — incidence. The inscription algorithm is very simple and achieves subquadratic running time in practice. Although ambiguities in the worst case can become linear in the number of polygon vertices, they seldom arose in our simulations, or were of a much smaller magnitude\(^3\) even if they did.

The study of the point sampling approach in Chapter 3 focused on the computational complexity issues surrounding the cost minimization of physical sensing. The sampling approach applies to the situation where only a finite number of possible poses (and objects) exist and they are known. It can be executed very fast or be very economical in terms of hardware expense when the possible poses/objects have a dense distribution. The geometric constraint to our interest in this chapter was containment, that is, which among a number of precomputed locations are occupied by the real pose. Geometric sensing was viewed as an optimization problem. Like many other combinatorial optimization problems, this optimal sensing problem is not expected to be tractable in polynomial time. The NP-completeness result led us to look for suboptimal solutions. We offered an approximation algorithm with a provable ratio and showed that such ratio is hard to improve.

Through all the above study, we have linked geometric sensing directly to constraint satisfaction. The constraints are the data obtained by physical sensors. They must be at least enough for either locally constraining the pose of an object or disambiguating the real pose from a finite set of possible poses. Although upper bounds on sensing ambiguities may be sought through worst case analyses, simulations often give more realistic pictures. When ambiguities arise, in the case of inscription for instance, a supplementary approach such as point sampling could be employed to resolve them.

One major issue is to minimize the size of geometric constraints used for sensing, which directly affects the efficiency and cost of sensing. As we have exhibited, optimal sensing

\(^2\)No motion of the object is assumed here. With a conveyor belt, we can use one rotary sensor to generate multiple inscribing cones of a part moving down the belt.

\(^3\)In our tests on two-cone inscription, instances with more than two ambiguous poses never appeared.
problems bear the nature of many other combinatorial optimization problems that they cannot be solved efficiently on some occasions. Suboptimal sensing strategies shall then be used instead.

We have barely touched the robustness of either inscription or sampling. Due to imperfections of physical sensing, the obtained inscription cone or the probed point locations have uncertainties. To what degree such uncertainties will affect the pose computation shall be assessed in order for these geometric sensing approaches to really become practical. This line of research is worthy of future endeavor. The implementation of geometric sensors is another possible future direction.

8.2 Mechanics-based Observation

The second part of the thesis revealed local observability of the state of an object under manipulation, given a small amount of information yielded by the interaction between the object and its manipulator. Contact motion was considered by us for it compactly encodes a range of information about the object and manipulator, which includes their local geometry, velocities, contact forces, etc. In the thesis, contact was generated by pushing or palm translation.

Chapter 5 considered a problem of planar pushing in which the object’s pose and motion are unknown to the pusher. We proved that, in the case of a disk pushing a polygon, not only the polygon’s pose but also its motion are locally observable from the movement of the contact point on the disk boundary, so long as the polygon and disk maintain contact. This local observability result is expected to generalize to other object and pusher shapes as well. Two procedures, called observers and based on nonlinear control theory, were described for pose and motion estimation. These observers were simulated in Chapter 6 where the implementation of a force sensor capable of sensing contact was also described.

Although in establishing the above results we made some simplifying assumptions such as uniform pressure distribution, our results imply that the information about a manipulation task is often hidden in simple tactile data and could be properly revealed by exploiting the mechanics and designing a nonlinear state observer.

Chapter 7 extended the study of local observability to three dimensions. It considered the rolling of a smooth convex object on a horizontal plane (or palm) that can accelerate translationally in any direction. The conclusion was that the object’s state is locally observable unless the local geometry around the contact satisfies one of three degenerate conditions listed in Theorem 32. Local observability under any of these degenerate conditions remains open due to the difficulty in establishing the observability rank condition; but this does not concern general object shapes or contact points. One interesting remark is that the object’s rolling motion does not seem to affect its own local observability.

\[4\text{To deal with support friction in the plane.}\]
\[5\text{We know at least that a sphere, which satisfies two of the three degenerate conditions, is not locally observable.}\]
8.2. MECHANICS-BASED OBSERVATION

The aforementioned results on local observability and observers are expected to generalize to other manipulation tasks which have sound mechanical models and in which contact or other types of interaction can be recorded.

Inspired by the way a human hand touches, the work in Part II may serve as a primitive step in exploring interactive sensing in manipulation tasks. From a more general perspective, it presents a method for acquiring geometric and dynamical information about a task from a small amount of tactile data, with the application of nonlinear observability theory.

Our approach may be quite applicable in model-based object recognition. We would like to identify an object from a set of known objects with one push. Here we could first hypothesize the object and then verify it using one of the introduced observers — for instance, the hypothesis is simply determined as “correct” if the observer converges.

Observer-based sensing has quite a few advantages over computer vision in tasks involving known objects. First, an observer relies on tactile information, thereby not subject to situations such as occlusion. Second, it makes use of the object and manipulator geometry, eliminating unnecessary and costly image processing and shape reconstruction. Third, sensing becomes interactive and adjustable to errors based on feedback. Fourth, the closed loop nature of an observer helps integrate sensing into the task execution. Last, an observer can be used to estimate a range of information including poses, velocities, forces, etc.

In the below, we discuss local observability a little further, and address some open issues of observer-based sensing, which may serve as good directions of future research.

8.2.1 Remarks on Local Observability

The angular velocity of an object is relatively easier to “observe” from the contact motion (on a manipulator) than its pose. In both the pushing task and the rolling task, described in Chapters 5 and 7, respectively, the real angular velocity can be locally distinguished from other possibilities by looking at the contact velocity directly. (In the case of a disk pushing a polygon, for instance, the polygon’s angular velocity can be directly obtained from the contact velocity. See (5.24)). To distinguish the contact position on the object boundary, which determines the object’s location, we have to look at the acceleration of the contact on the disk or in the palm (i.e., the second order Lie derivative of the output). When your fingertip pushes a small object, the first sense you get is how fast the object is rotating on the fingertip rather than where on the object the fingertip is touching.

In establishing local observability, we made use of input vector fields. In the disk-polygon problem, the input fields are controlled by the disk acceleration; in the rolling task, they are controlled by the palm acceleration. The use of controls was merely to avoid the technical difficulty in establishing the observability rank condition from repeated Lie derivatives of the output with respect to very complicated drift fields. It would be no surprise if local observability still holds under zero control, that is, when the disk or the plane translates at constant speed. This was in fact demonstrated by simulations in Chapter 6. On such an occasion, the observation is viewed as “harder” due to the involvement of higher order Lie derivatives to establish the observability rank condition.
8.2.2 Observer-related Issues

There are a number of unresolved issues surrounding pose and motion observers that are worthy of future study.

Observer Globalization

The first issue is about the locality of the observer’s initial estimate of a state. As suggested by the local observability proof of Theorem 19 and observed in our simulations in Chapter 6, neither the output of finger contact nor the object motion seem to affect the outcome of the Gauthier-Hammouri-Othman observer much. For instance, in many simulation examples, the GHO observer succeeded with various finger motions and control parameters, whereas in a few others, it failed for all chosen finger motions and control parameters. Also, the sufficient condition for local observability given in Chapter 7 involves only the contact position on an object and its local geometry. All the above suggests that the finger and object geometry affect local observability and an observer’s behavior the most. So it seems quite reasonable for us to use the object contact only as a locality measure. In other words, we measure how far the estimated object contact is from the real contact.

How local is local enough? An analytic study has to delve into the theory of ordinary differential equations and is beyond the scope of this thesis. Meanwhile, we seem unable to draw any satisfactory empirical conclusion from the simulation results in Chapter 6. The difficulty is especially reflected in the choice of an appropriate parameter value to control the error corrective term of a GHO observer. If the parameter is too small, the error corrective term would be dominated by the dynamics of the original system, making the convergence slow or impossible; if the parameter is too large, the state estimate would very likely evolve out of the local neighborhood, causing a failure.

Nevertheless, a parallel strategy can be adopted to “globalize” the GHO observer and the initial pose observer. More specifically, run a few observers in parallel, each with a different combination of estimate of the object contact and control parameter value. If no observer succeeds within a certain amount of time, it is regarded as a failure.\(^6\)

To characterize the role of geometry on the pose observers and to study the interaction between the finger shape and the object shape thus seem very important. Any major progress is expected to help us find good sensor shapes that could enhance the observer performance significantly.

Efficiency

How to make the pose observers run fast? In order to have any real application, the observers have to run in real time. Currently, the GHO observers for a disk pushing a polygon run one to a few hundred times as slow on a Sparcstation 20 as the real pushing, which is very inadequate. The bulk of time is currently spent on the numerical evaluation of the

\(^6\)An observer in our simulations was regarded to have succeeded if the difference between the simulated output and the real output has become negligible and remained so for a while.
integral of support friction. Similar inefficiency caused by friction also occurs in computing contact forces between rigid bodies in computer graphics (see [12]). A deeper understanding of the observer behavior could result in some speedup, though the bottleneck seems to be mechanics-based simulation. The initial pose observer, however, can be efficiently implemented by a table lookup method.

Non-uniform Pressure Distributions

The observer-based sensing also applies to the situation of planar pushing where pressure distribution is non-uniform but known. It is not applicable, however, when the pressure distribution is unknown. In the latter situation, quasi-static analysis [103] can be used to predict the rotation direction and center of the object being pushed but not the object’s angular velocity or velocity. It is not clear how dynamics based observation could be applied though.
Appendix A

A Worst Case of Inscription

In this appendix we construct an example where a convex \( n \)-gon embedded in three half-planes can attain \( 6n \) possible poses.

Symmetry plays a central role in the construction. Let the three half-planes form an equilateral triangle \( \triangle q_0 q_1 q_2 \) by intersection; and let \( P \) be a convex polygon with vertices \( p_0, p_1, \ldots, p_{n-1} \) in counterclockwise order such that \( p_0 \) is the center of an arc of radius \( r \) and measure \( \alpha \), and \( p_1, \ldots, p_{n-1} \) together divide this arc into \( n - 2 \) equal pieces. The reason for choosing this particular shape of \( P \) is that we expect to obtain \( 2(n-1) \) poses by positioning \( p_0 \) at each of \( q_0, q_1, \) and \( q_2 \) and rotating \( P \) about \( p_0 \) inside \( \triangle q_0 q_1 q_2 \) such that \( p_1, \ldots, p_{n-1} \) will each touch the opposite edge in \( \triangle q_0 q_1 q_2 \) exactly twice during the rotation. Figure A.1(a) illustrates the first of a sequence of \( 2(n-1) \) poses resulting from counterclockwise rotation about \( q_0 \). By symmetry we already have \( 6(n-1) \) poses in total, called poses of the \textit{first type}. The remaining \( 6 \) poses of the \textit{second type} are symmetric to each other, attained when vertices \( p_0, p_1, \) and \( p_{n-1} \) are on different edges of \( \triangle q_0 q_1 q_2 \) (Figure A.1(b)). So our task is to derive the conditions on radius \( r \) and measure \( \alpha \) of the arc that realize these two types of poses.

Let us look at poses of the first type. Let \( d > 0 \) be the altitude of \( \triangle q_0 q_1 q_2 \) and \( \beta \) an acute angle with \( \cos \beta = \frac{d}{r} \), and suppose \( p_0 \) coincides with \( q_0 \). We require that \( d < r \) and \( \alpha < \alpha + \beta < \frac{\pi}{6} \) so that \( p_1 \) in the initial pose is not on edge \( q_0 q_1 \) in order to allow poses of the second type, and \( p_{n-1} \) lies to the left of the midpoint on edge \( q_1 q_2 \) in order to be incident on this edge twice as \( P \) rotates counterclockwise. Thus, we have the following conditions on \( \alpha \) and \( r \):

\[
0 < \alpha < \frac{\pi}{6} \quad \text{and} \quad \cos \left( \frac{\pi}{6} - \alpha \right) < \frac{d}{r} < 1.
\]

With the above constraints, we still need to make sure that \( P \) is indeed above edge \( q_1 q_2 \) when each vertex \( p_i, 1 \leq i \leq n-1 \), becomes incident on this edge during the rotation. In fact, it suffices to ensure that vertices \( p_{i-1} \) and \( p_{i+1} \) are above edge \( q_1 q_2 \). Figure A.1(c) illustrates the case in which \( p_i \) lies to the left of the midpoint on edge \( q_1 q_2 \). It is clear that \( p_{i-1} \) is above edge \( q_1 q_2 \); \( p_{i+1} \) is above edge \( q_1 q_2 \) if and only if \( 2\beta < \angle p_0 p_i p_{i+1} = \frac{\alpha}{n-2} \); that
Figure A.1: A worst case example of inscription. The three half-planes form an equilateral triangle $\triangle q_0q_1q_2$ by intersection and the convex $n$-gon $P$ has vertex $p_0$ at the center of an arc equally divided by the remaining vertices $p_1, \ldots, p_{n-1}$ of $P$. The conditions derived in this appendix on radius $r$ and measure $\alpha$ of the arc guarantee $6n$ different poses in which $P$ is inscribed in $\triangle q_0q_1q_2$: (a) $6(n - 1)$ poses of $P$ result from rotations about $p_0$ positioned at vertices $q_0$, $q_1$, and $q_2$, respectively. (b) Six other poses result when $p_0$, $p_1$, and $p_{n-1}$ are incident on different edges of $\triangle q_0q_1q_2$. (c) illustrates a pose of the first type such that vertex $p_i$, $2 \leq i \leq n - 2$, is on edge $q_iq_2$ during the rotation about $q_0$. 
is,

\[
\frac{d}{r} > \cos \frac{\alpha}{2(n-2)}.
\]

Starting at the initial pose shown in Figure A.1(a), slide \(p_0\) down along edge \(q_0q_2\) and \(p_{n-1}\) left along edge \(q_1q_2\) until \(p_1\) touches edge \(q_0q_1\); then we get the pose in Figure A.1(b). Observe that \(p_0\) is in the interior of edge \(q_0q_2\) and closer to \(q_0\) than \(q_2\). The other five poses of the second type follow by symmetry.

Hence \(P\) has \(6n\) feasible poses when inscribed in \(\triangle q_0q_1q_2\) if the following two conditions are satisfied:

\[
0 < \alpha < \frac{\pi}{6};
\]

\[
d < r < \min \left( \frac{d}{\cos \left(\frac{\pi}{6} - \alpha\right)}, \frac{d}{\cos \left(\frac{\alpha}{2(n-2)}\right)} \right).
\]

In fact, the above conditions are also necessary for the given shapes of \(P\) and \(\triangle q_0q_1q_2\).
Appendix B

$k$-Independent Sets

We extend Lemma 11 to all $k$-IS with $k > 3$: They are NP-complete as well. The proof we will present is indeed a generalization of the proof of Lemma 11; it will again construct a $k$-IS instance with graph $G'$ from an instance of Independent Set with graph $G$ by local replacement. In the proof, each vertex $v$ in $G$ will be replaced by a simple path $P_v$ of fixed length (depending only on $k$) that has $v$ in the middle and an equal number of auxiliary vertices on each side; and each edge $(u, v)$ will be replaced by four edges connecting the two end vertices on $P_u$ with the two end vertices on $P_v$, either directly or through a “midvertex”. More intuitively speaking, all shortest paths between pairs of vertices in $G$, if they exist, get elongated in $G'$ to such a degree that (1) $(u, v)$ is an edge in $G$ if and only if the distance between vertices $u$ and $v$ in $G'$ is less than $k$; and (2) any two vertices $u'$ and $v'$ in $G'$ with a distance of at least $k$ can be easily mapped to two non-adjacent vertices in $G$. The first condition ensures that any given independent set in $G$ will be a $k$-independent set in $G'$, while the second condition ensures the construction of an independent set in $G$ from any given $k$-independent set in $G'$.

Lemma 33 $k$-Independent Set is NP-complete for all integers $k > 3$.

Proof Given an instance of Independent Set as a graph $G = (V, E)$ and a positive integer $l \leq |V|$, a $k$-IS instance is constructed by two consecutive substitutions. A path

$$P_v = \begin{cases} v_{k-3} \ldots v_1 v v_2 \ldots v_{k-2}, & \text{if } k \text{ even;} \\ v_{k-4} \ldots v_1 v v_2 \ldots v_{k-3}, & \text{if } k \text{ odd,} \end{cases}$$

first substitutes for vertex $v \in V$, where $v_1, \ldots, v_{k-3}$ (and $v_{k-2}$ when $k$ is even) are auxiliary vertices. And then a set of four edges

$$E_{u,v} = \begin{cases} \{(u_{k-3}, v_{k-3}), (u_{k-3}, v_{k-2}), \\ (u_{k-2}, v_{k-3}), (u_{k-2}, v_{k-2})\}, & \text{if } k \text{ even;} \\ \{(u_{k-4}, z_{u,v}), (u_{k-3}, z_{u,v}), \\ (v_{k-4}, z_{u,v}), (v_{k-3}, z_{u,v})\}, & \text{if } k \text{ odd,} \end{cases}$$
substitute for each edge \((u, v) \in E\), where \(z_{u,v}\) is an introduced midvertex. Figure B.1 shows two subgraphs after applying the above substitutions on edge \((u, v) \in E\), for \(k\) even and odd, respectively.

We can easily verify that for any pair of vertices \(x\) on \(P_u\) and \(y\) on \(P_v\), both \(k\) even and odd, we have \(d'(x, y) \leq d(u, v) = k - 1 < k\) if \((u, v) \in E\), where \(d'\) is the distance function defined on \(G'\). On the other hand, if \((u, v) \notin E\), we have \(d'(u, v) \geq k\) when \(k\) is even and \(d'(u, v) \geq k + 1\) when \(k\) is odd. Thus, an independent set \(I\) in \(G\) is also a \(k\)-independent set in \(G'\). Conversely, suppose \(I'\) with \(|I'| \geq l\) is a \(k\)-independent set in \(G'\). We substitute \(u \in V\) for every auxiliary vertex \(u_i \in I\) on path \(P_u\), and \(u\) or \(v\) for every midvertex \(z_{u,v} \in I\) when \(k\) is odd. Let \(I\) be the set after this substitution. It needs to be shown that \(I\) is an independent set in \(G\) and \(|I| = |I'| \geq l\). This is obvious for the case that \(k\) is even. When \(k\) is odd, however, the situation is a bit more complicated due to the possible occurrences of those \(z\) vertices in \(I'\). We observe, for any \(z_{u,v}, z_{u',v'}, x\) on path \(P_w\) where \(u, v, u', v', w \in V\),

\[
d'(z_{u,v}, x) \leq \frac{k - 3}{2} + 3 < k, \quad \text{if } w = u \text{ or } v, \text{ or } (u, w) \in E, \text{ or } (v, w) \in E; \\
d'(z_{u,v}, z_{u',v'}) \leq 4 < k, \quad \text{if } (u, u') \in E.
\]

In fact, these two conditions guarantee that \(I\) is an independent set in \(G\) and \(|I| = |I'|\), which we leave for the reader to verify.

The reduction can be done in time \(O(k|V| + |E|)\), which reduces to \(O(|V| + |E|)\) if \(k\) is treated as a constant, in contrast to the time \(O(|V|^3)\) required for the reduction from Independent Set to 3-IS. This time reduction is due to the fact that midvertices corresponding to the same vertex in \(V\) no longer have edges between each other.

Since 1-IS can be easily solved by comparing \(|V|\) and \(l\), we are now ready to sum up the complexity results on this family of problems in the following theorem.

**Theorem 34** \(k\)-Independent Set is in \(P\) if \(k = 1\) and \(NP\)-complete for all \(k \geq 2\).
Appendix C

Integral \( \Gamma \) of Support Friction
(Disk-Polygon)

We assume \( \omega \neq 0 \) in this appendix for otherwise the integral of friction \( \Gamma \) is given by equation (5.18), hence trivial to evaluate. The object under consideration is a polygon and the finger is a disk. As in Figure 5.4, the contact point is at (signed) distance \( s \) away from the intersection of the contact edge and its perpendicular through the polygon centroid. The integral \( \Gamma \) is given by (5.25), where \( N = -(\cos \frac{u}{r}, \sin \frac{u}{r})^T \) is the contact normal, \( I_1 \) and \( I_2 \) are two integrals defined by (5.20) in the object’s body frame. In this appendix we will obtain \( I_1 \) and \( I_2 \) as well as all partial derivatives of \( \Gamma \).

C.1 Integrals \( I_1 \) and \( I_2 \)

Let \( B \) be a simple \( n \)-gon with vertices \( p_1, p_2, \ldots, p_n \); let \( q = (x_0^B, y_0^B)^T \) be its instantaneous rotation center. For each edge \( p_i p_{i+1} \), \( i = 1, 2, \ldots, n \), define

\[
    c_i = \begin{cases} 
    1, & \text{if the ray from } q \text{ to the midpoint of } p_i p_{i+1} \text{ is exiting } B; \\
    -1, & \text{otherwise}. 
    \end{cases}
\]

Then any integration over \( B \) is decomposed into \( n \) integrations over triangles \( \triangle q p_i p_{i+1} \), \( i = 1, \ldots, n \), respectively. The sum of the obtained integrals, each weighted by \( c_i \), respectively, is the desired integral over \( B \) (see Figure C.1(a)).

So we need only to look at the integrations over one such triangle, defined by \( q \) and, say, edge \( p_1 p_2 \). A point on \( p_1 p_2 \), in polar coordinates \((r, \phi)\) with respect to \( q \) satisfies

\[
    r(\phi) = -\frac{p_1 \times p_2 + q \times (p_1 - p_2)}{c} \\
    = \frac{a \cos \phi + b \sin \phi}{c},
\]

1Note that \( p_{n+1} \) is identified with \( p_1 \).
Figure C.1: Evaluating an integral over a simple polygon. For each polygon edge integrate over the triangular region defined by the instantaneous rotation center \( (x_0^B, y_0^B) \) and the edge; then add the integral to or subtract it from the total, according as the rays from the i.r.c. are leaving or entering the polygon.
where $a, b, \text{ and } c$ are constants dependent only on $q, p_1, \text{ and } p_2$.

Let $\phi_1$ and $\phi_2$ be the polar angles of $p_1$ and $p_2$, respectively. Without loss of generality, assume $\phi_1 < \phi_2$. Let us define three definite integrals $J_1, J_2, \text{ and } J_3$:

\[
J_1 = \int_{\phi_1}^{\phi_2} \frac{\sin \phi}{(a \cos \phi + b \sin \phi)^2} d\phi; \\
J_2 = \int_{\phi_1}^{\phi_2} \frac{\cos \phi}{(a \cos \phi + b \sin \phi)^2} d\phi; \\
J_3 = \int_{\phi_1}^{\phi_2} \frac{1}{(a \cos \phi + b \sin \phi)^3} d\phi.
\]

for which the closed forms can be obtained.\(^2\) We are ready to represent $I_1$ and $I_2$ using the above integrals:

\[
I_1 = \pm \int_{\phi_1}^{\phi_2} \frac{r^2(\phi)}{2} \left( \frac{-\sin \phi}{\cos \phi} \right) d\phi \\
= \pm \int_{\phi_1}^{\phi_2} \frac{c^2}{2(a \cos \phi + b \sin \phi)^2} \left( \frac{-\sin \phi}{\cos \phi} \right) d\phi \\
= \pm \frac{c^2}{2} \left( -J_1 \right) J_2; \\
I_2 = \pm \int_{\phi_1}^{\phi_2} \left( x_0^B \cos \phi + y_0^B \sin \phi \right) \frac{r^2(\phi)}{2} + \frac{r^3(\phi)}{3} d\phi \\
= \pm \int_{\phi_1}^{\phi_2} \left( x_0^B \cos \phi + y_0^B \sin \phi \right) \frac{c^2}{2(a \cos \phi + b \sin \phi)^2} + \frac{c^3}{3(a \cos \phi + b \sin \phi)^3} d\phi \\
= \pm \left( \frac{c^2}{2} \left( x_0^B J_2 + y_0^B J_1 \right) + \frac{c^3}{3} J_3 \right).
\]

Here the notation “$\pm$” means “$+$” if $\omega > 0$ and “$-$” if $\omega < 0$.

It follows that the computation of $I_1$ and $I_2$ takes time $O(n)$.

### C.2 Partial Derivatives of $\Gamma$

Next, we derive the partial derivatives of $\Gamma$ with respect to state variables $u, s, \omega, \text{ and } v_T$. Here only the case $\omega \neq 0$ will be considered. In the degenerate case $\omega = 0$, $\Gamma$ has a simpler

\(^2\)Introduce a constant $\psi$ with $\sin \psi = a/\sqrt{a^2 + b^2}$ and $\cos \psi = b/\sqrt{a^2 + b^2}$. For example,

\[
J_1 = \frac{1}{a^2 + b^2} \int_{\phi_1}^{\phi_2} \sin(\phi + \psi) \cos \psi - \cos(\phi + \psi) \sin \psi \frac{d\phi}{\sin^2(\phi + \psi)} \\
= \frac{1}{a^2 + b^2} \left( \cos \psi \int_{\phi_1 + \psi}^{\phi_2 + \psi} \frac{1}{\sin \zeta} d\zeta - \sin \psi \int_{\phi_1 + \psi}^{\phi_2 + \psi} \frac{\cos \zeta}{\sin^2 \zeta} d\zeta \right)
\]

where the two integrals on the right hand side of the last line of the equation above have closed forms.
form (5.18) for which partial derivatives can be easily obtained.

From equation (5.25) $\Gamma$ depends on $u, s, \omega, v$, and $\theta$, among which the following dependencies exist:

\[ v = v_T T + v_N N; \]
\[ v_N = v_{RN} + s \omega = v_{RN} N + s \omega; \]
\[ \theta = \frac{u}{r} - \frac{\pi}{2} + \theta_0, \]

where $\theta_0$ is some constant angle determined by the edge of contact. Subsequently, the integral of friction assumes several different forms depending on which variables are chosen:

\[ \Gamma = sRI_1 + I_2 N \]
\[ = \gamma(u, s, \omega, v(u, v_T), \theta(u)) \]
\[ = \gamma_1(u, s, \omega, v_T,v_N(u, s, \omega)) \]
\[ = \gamma_2(u, s, \omega, v_T). \]

Thus we can write out the partial derivatives of $\Gamma$ on $u, s, \omega, v_T$ in terms of the partial derivatives of $\gamma$ with respect to $u, s, \omega, v$, and $\theta$:

\[ \frac{\partial \Gamma}{\partial u} = \frac{\partial \gamma_2}{\partial u} \]
\[ = \frac{\partial \gamma_1}{\partial u} + \frac{\partial \gamma_1}{\partial v_N} \frac{\partial v_N}{\partial u} \]
\[ = \frac{\partial \gamma_1}{\partial u} - \frac{v_{RN} \partial \gamma_1}{r \partial v_N} \]
\[ = \left( \frac{\partial \gamma}{\partial u} + \frac{\partial \gamma}{\partial v} \frac{\partial v}{\partial u} + \frac{\partial \gamma}{\partial \theta} \frac{\partial \theta}{\partial u} \right) - \frac{v_{RN}}{r} \left( \frac{\partial \gamma}{\partial v} \frac{\partial v}{\partial v_N} \right) \]
\[ = \frac{\partial \gamma}{\partial u} + \frac{\partial \gamma}{r \partial v} \left((v_T - v_{RN}) N - v_N T\right) + \frac{\partial \gamma}{r \partial \theta}; \]
\[ \frac{\partial \Gamma}{\partial s} = \frac{\partial \gamma}{\partial s} + \omega \frac{\partial \gamma}{\partial v}; \]
\[ \frac{\partial \Gamma}{\partial \omega} = \frac{\partial \gamma}{\partial \omega} + s \frac{\partial \gamma}{\partial v}; \]
\[ \frac{\partial \Gamma}{\partial v_T} = \frac{\partial \gamma}{\partial v} T. \]

Below we obtain the partial derivatives of $\gamma$ with respect to $u, s, \omega, v$, and $\theta$. We have

\[ \frac{\partial \gamma}{\partial u} = -\frac{I_2 T}{r}; \]
\[ \frac{\partial \gamma}{\partial s} = RI_1. \]
Given the sign of $\omega$, $I_1$ and $I_2$ depend only on the location of the instantaneous rotation center $q$, which in turn depends on $\omega, v,$ and $\theta$:

$$q = \begin{pmatrix} x_0^B \\ y_0^B \end{pmatrix} = \begin{pmatrix} -v_y^B / \omega \\ v_x^B / \omega \end{pmatrix} = \frac{1}{\omega} \begin{pmatrix} \sin \theta & -\cos \theta \\ \cos \theta & \sin \theta \end{pmatrix} \begin{pmatrix} v_x \\ v_y \end{pmatrix}.$$ 

Accordingly, $\Gamma$ has yet another form

$$\Gamma = \gamma_3 \left( u, s, \theta, x_0^B(\omega, v, \theta), y_0^B(\omega, v, \theta) \right).$$

The remainder of the task is in fact reduced to evaluating the partial derivatives of $\gamma_3$ with respect to $q$ since

$$\frac{\partial \gamma}{\partial \omega} = \frac{1}{\omega^2} \frac{\partial \gamma_3}{\partial q} \begin{pmatrix} v_y^B \\ -v_x^B \end{pmatrix} = \frac{1}{\omega^2} \frac{\partial \gamma_3}{\partial q} \begin{pmatrix} -\sin \theta & \cos \theta \\ -\cos \theta & -\sin \theta \end{pmatrix} \begin{pmatrix} v_x \\ v_y \end{pmatrix};$$

$$\frac{\partial \gamma}{\partial v} = \frac{1}{\omega} \frac{\partial \gamma_3}{\partial q} \begin{pmatrix} \sin \theta & -\cos \theta \\ \cos \theta & \sin \theta \end{pmatrix};$$

$$\frac{\partial \gamma}{\partial \theta} = \frac{s dR}{d\theta} \frac{\partial I_1}{\partial q} + \frac{1}{\omega} \frac{\partial \gamma_3}{\partial q} \begin{pmatrix} \cos \theta & \sin \theta \\ -\sin \theta & \cos \theta \end{pmatrix} \begin{pmatrix} v_x \\ v_y \end{pmatrix}. $$

Since

$$\frac{\partial \gamma_3}{\partial q} = sR \frac{\partial I_1}{\partial q} + \frac{\partial I_2}{\partial q} N,$$

we need to evaluate the partial derivatives of $I_1$ and $I_2$ on $x_0^B$ and $y_0^B$, which is the subject of Appendix C.3.

### C.3 Partial Derivatives of $I_1$ and $I_2$ on I.R.C.

As in Appendix C.1, we partition the convex hull of $q$ and the polygon into triangles, each defined by $q$ and one edge. We differentiate the integrals over each triangle, weigh the derivatives by 1 or $-1$, and add these derivatives for all triangles together. Again we illustrate by differentiating the integrals over a triangle $\triangle p_1p_2q$ due to edge $p_1p_2$ (see again Figure C.1). For clarity as well as convenience, we rewrite integrals $I_1$ and $I_2$ in equations (C.1) as (assuming $\omega > 0$):

$$I_j = \int_{\phi_1}^{\phi_2} \Psi_j(x_0^B, y_0^B, \phi) \, d\phi, \quad \text{for } j = 1, 2.$$
Then it follows that
\[
\frac{\partial I_j}{\partial q} = \Psi_j(x_0^R, y_0^R, \phi_2) \frac{\partial \phi_2}{\partial q} - \Psi_j(x_0^R, y_0^R, \phi_1) \frac{\partial \phi_1}{\partial q} + \int_{\phi_1}^{\phi_2} \frac{\partial \Psi_j}{\partial q} \, d\phi.
\]

Denote \((x_i^R, y_i^R)^T\) by \(p_i\). First, we obtain the partial derivative of the polar angles \(\phi_1\) and \(\phi_2\):
\[
\frac{\partial \phi_i}{\partial q} = \frac{1}{\|p_i - p_0\|^2} (y_i^R - y_0^R, x_i^R - x_0^R).
\]

Next, we have
\[
\frac{\partial \Psi_1}{\partial q} = r(\phi) \left( -\sin \phi \quad \cos \phi \right) \frac{\partial r}{\partial q},
\]
\[
\frac{\partial \Psi_2}{\partial q} = \frac{r^2(\phi)}{2} \left( \cos \phi, \sin \phi \right) + \left( q \cdot \left( \cos \phi, \sin \phi \right) \right) r(\phi) \frac{\partial r}{\partial q} + r^2(\phi) \frac{\partial r}{\partial q}.
\]

In the above
\[
\frac{\partial r}{\partial q} = -\frac{1}{\left( \frac{\cos \phi}{\sin \phi} \right) \times (p_1 - p_2)} \left( \binom{1}{0} \times (p_1 - p_2), \binom{0}{1} \times (p_1 - p_2) \right)
\]
\[
= \frac{1}{(a \cos \phi + b \sin \phi)} (c_1, c_2),
\]
where \(a, b, c_1,\) and \(c_2\) are constants dependent only on \(p_1\) and \(p_2\). Therefore the integrals of \(\Psi_i\) over \(\phi\) can be represented by integrals \(J_1, J_2,\) and \(J_3\) defined in Appendix C.1:
\[
\int_{\phi_1}^{\phi_2} \frac{\partial \Psi_1}{\partial q} \, d\phi = c \binom{-J_1}{J_2} (c_1, c_2);
\]
\[
\int_{\phi_1}^{\phi_2} \frac{\partial \Psi_2}{\partial q} \, d\phi = c \left( J_1, J_2, J_3 \right) \begin{pmatrix} c_1 y_0^B & c/2 + c_2 y_0^B \\ c/2 + c_1 x_0^B & c_2 x_0^B \\ c_1^2 & c_2^2 \end{pmatrix}.
\]

Note that \(c\) is a constant introduced in Appendix C.1.
Appendix D

Observer for Rolling (Disk-Polygon)

For rolling, the integral of friction becomes
\[
\Gamma_r = \int_B R(\beta - p) \times \dot{v}_p \, dp
\]
\[
= \beta \times I_1 - I_2,
\]
\[
= \gamma_r(s, \omega, v(u, s, \omega), \theta(u))
\]

where \(v\) is given by (5.38) and \(I_1\) and \(I_2\), defined by (5.20), are computed in Appendix C.1. The coordinate transformation is given by

\[
\begin{pmatrix}
u \\
s \\
\omega
\end{pmatrix}
\xrightarrow{\chi}
\begin{pmatrix}
u \\
\omega_r
\end{pmatrix}
\]

The Jacobian of the inverse transformation \(\chi^{-1}\) is equal to the inverse of the Jacobian of \(\chi\), that is,

\[
\frac{\partial \chi^{-1}}{\partial \mathbf{x}} = \left(\begin{array}{c}
du/rd\omega \\
rdL_f\omega
\end{array}\right)^{-1}.
\]

The differentials \(du\) and \(d\omega\) are trivial. The differential \(dL_f\omega\) involves the partial derivatives of \(\Gamma_r\). To apply the chain rule we need the partial derivatives of \(\gamma_r\) on \(s, \omega, v, \theta\) as well as the partial derivatives of \(v\) and \(\theta\) on \(u, s, \omega\). The first group of partial derivatives is easy to evaluate given \(\frac{\partial I_1}{\partial q}\) and \(\frac{\partial I_2}{\partial q}\), where \(q\) is the instantaneous rotation center (see Appendix C.3). The second group is listed in the below:

\[
\frac{\partial v}{\partial u} = \frac{\omega}{r}(hN - sT);
\]
\[
\begin{align*}
\frac{\partial v}{\partial s} &= \omega N; \\
\frac{\partial v}{\partial \omega} &= hT + sN; \\
\frac{\partial \theta}{\partial u} &= \frac{1}{r}.
\end{align*}
\]

Finally, solve equation (4.9) under \( n = 3 \) and obtain the inverse of the solution:

\[
S_{\infty} = \begin{pmatrix}
\frac{1}{\zeta} & -\frac{1}{\zeta^2} & \frac{1}{\zeta^3} \\
-\frac{1}{\zeta^2} & \frac{2}{\zeta^3} & -\frac{3}{\zeta^4} \\
\frac{1}{\zeta^3} & -\frac{3}{\zeta^4} & \frac{6}{\zeta^5}
\end{pmatrix}
\quad \text{and} \quad
S_{\infty}^{-1} = \begin{pmatrix}
3\zeta & 3\zeta^2 & \zeta^3 \\
3\zeta^2 & 5\zeta^3 & 2\zeta^4 \\
\zeta^3 & 2\zeta^4 & \zeta^5
\end{pmatrix}.
\]

Plugging all the above into (4.10) gives us the observer (5.40) in the case of rolling.
Appendix E

Infeasibility of Linearization by Output Injection

We show that linearization by output injection in Theorem 15 does not apply to a simple version of the disk-polygon system. In this version, no support friction exists in the plane; the finger has constant velocity; and the object is rolling along the finger boundary. The system equations are given as (cf. Proposition 20)

\[
\begin{align*}
\dot{u} &= \omega r; \\
\dot{s} &= -\omega r; \\
\dot{\omega} &= \frac{rs\omega^2}{h^2 + \rho^2 + s^2}.
\end{align*}
\]

(E.1)

Claim 35 System (E.1) cannot be transformed into the form of (4.7) by a change of coordinates.

Proof Writing

\[
\begin{align*}
x &= \begin{pmatrix} u \\ s \\ \omega \end{pmatrix} \quad \text{and} \quad f &= \begin{pmatrix} \omega r \\ -\omega r \\ rs\omega^2 \\ (h^2 + \rho^2 + s^2) \end{pmatrix},
\end{align*}
\]

we first obtain the differentials of several Lie derivatives of the output \( u \), up to the second order:

\[
\begin{align*}
du &= (1, 0, 0); \\
dL_fu &= (0, r, 0); \\
dL^2_fu &= \left(0, \frac{h^2 + \rho^2 - s^2}{(h^2 + \rho^2 + s^2)^2} r^2 \omega^2, \frac{2r^2 s\omega}{h^2 + \rho^2 + s^2} \right).
\end{align*}
\]

Condition 1 in Theorem 15 is indeed satisfied unless \( s^2 = h^2 + \rho^2 \) or \( \omega = 0 \).
However, condition 2 in the theorem is violated. To see this, the linear system of equations

\[
\begin{align*}
L_g u &= 0, \\
L_g L_f u &= 0, \\
L_g L_f^2 u &= 1,
\end{align*}
\]

has solution

\[
g = \begin{pmatrix}
0 \\
\frac{(h^2 + \rho^2 + s^2)^2}{r^2 \omega^2 (h^2 + \rho^2 - s^2)} \\
0
\end{pmatrix}.
\]

But we have

\[
\text{ad}_f g = \begin{pmatrix}
0 \\
\frac{4s(-2(h^2 + \rho^2)^2 - (h^2 + \rho^2)s^2 + s^4)}{r \omega (h^2 + \rho^2 - s^2)^2} \\
\frac{-1}{r}
\end{pmatrix};
\]

\[
[g, \text{ad}_f g] = \begin{pmatrix}
0 \\
\frac{2(h^2 + \rho^2 + s^2)^2(-5(h^2 + \rho^2)^3 + 9(h^2 + \rho^2)^2 s^2 - 15(h^2 + \rho^2)s^4 + 3s^6)}{r^3 \omega^3 (h^2 + \rho^2 - s^2)^4} \\
0
\end{pmatrix}.
\]

Apparently, \([g, \text{ad}_f g]\) does not vanish on any open set.
Bibliography


