Vision-Based Control on Lie groups

with Application to Needle Steering

by

Vinutha Kallem

A dissertation submitted to The Johns Hopkins University in conformity with the

requirements for the degree of Doctor of Philosophy.

Baltimore, Maryland

June, 2008

Abstract

This thesis presents vision-based control algorithms for systems evolving on Lie groups. The thesis consists of two parts: (1) task-induced symmetry and reduction and its application to needle steering and (2) kernel-based visual servoing.

The core of this thesis is motivated by image-guided control of flexible bevel-tip needles. Image guidance promises to improve targeting accuracy and broaden the scope of medical procedures performed with needles. We build upon a previously proposed non-holonomic kinematic model of flexible bevel-tip needle steering in which the needle is inserted and rotated at its base in order to steer it in six degrees of freedom. As a first step for control, we show that the needle tip can be automatically guided to a planar slice of tissue as it is inserted by a physician; our approach keeps the physician in the loop to control insertion speed. The distance of the needle tip position from the plane of interest is used to drive an observer-based feedback controller. We prove that the complete six degree-of-freedom pose of the needle tip can be estimated from just the three-dimensional needle tip position measurements over time. This enables us to develop dead-beat and asymptotic observers to recover needle-tip orientation for control.

ABSTRACT

The task of driving a needle tip to a desired plane induces symmetry resulting in a reduced system which greatly simplifies controller and observer design. We propose a method to perform such reduction for generic nonholonomic kinematic systems on Lie groups with left-invariant vector fields. This technique is used to develop controllers for curve-following of a unicycle and subspace-following in needle steering. These subspace controllers for needle steering are designed to work in conjunction with subspace planners for the needle tip to reach a desired location in human tissue. We show that this task-induced reduction lifts to mechanical systems as well.

In the second part of the thesis, we present kernel-based visual servoing algorithms. In visual servoing, the goal is to control the motion of the robot/scene such that a set of image features converge to a known constellation; this requires tracking these feature points in every frame. Moving away form the traditional visual servoing approaches that have treated tracking and control as two isolated problems, kernel-based visual servoing paradigm fuses tracking and control by removing the need to explicitly track features in a scene. In this method, a weighted average of the image (or its transform) is used as the signal to the controller; the weighting function is a smooth kernel and the weighted average is called the kernel measurement. Using smooth kernel functions, we design, develop, and test controllers to navigate a robot to reach a desired goal in the three translational and roll degrees of freedom for an eye-in-hand configuration. This work provides a new framework to design vision-based controllers on natural images and their formal stability characterization using Lyapunov theory.

ABSTRACT

Advisor: Prof. Noah J. Cowan

Readers: Prof. Dong Eui Chang, Prof. Gregory S. Chirikijian, Prof. Gregory D. Hager, and

Prof. Allison M. Okamura

Acknowledgements

First, I would like to thank my advisor, Noah Cowan. During my four years at Hopkins, he has been a great teacher, mentor and a friend. Noah is always there for his students and is only a short walk across the corridor. It is not always that someone will give you individual lessons on manifolds and geometry (and em dashes and en dashes) like I have gotten in our research meetings. These meetings have shaped my thinking and my approach to research. I am grateful for his encouragement and faith in me during my dissertation work. None of this thesis would have been possible without his guidance and mentoring.

I would like to express gratitude to my thesis committee: Dong Eui Chang, Greg Chirikjian, Greg Hager and Allison Okamura for their time and insights. I have benefited greatly from my interactions with them. I am specially indebted to Dong Eui Chang for his invaluable lessons on geometric mechanics. I am also grateful to my collaborators Maneesh Dewan and John Swensen, whom I had the pleasure of working with on kernel-based visual servoing, the topic of Chapter 6 of this thesis.

Special thanks are also due to Rene Vidal, Louis Whitcomb and Joel Spruck for being a part of my Graduate Board Oral examination committee and for the courses I have taken

ACKNOWLEDGEMENTS

ACKNOWLEDGEMENTS

Contents

CONTENTS

CONTENTS

CONTENTS

List of Figures

LIST OF FIGURES

Chapter 1

Introduction

Vision is one of the most important sensors in animals and humans and also in robotic systems. Vision-based control has applications in a wide variety of robotic tasks ranging from automated highways, navigation of unmanned vehicles, factory manufacturing, automated homes to medical interventions. In this work, we present vision-based control algorithms for robotic systems evolving on Lie groups. This thesis consists of two parts: (1) task-induced symmetry and reduction and its application to needle steering and (2) kernel-based visual servoing. In both these problems, the sense of vision is enabled through cameras, and Lyapunov theory is used to show the stability of the controllers.

The core of this thesis is motivated by the design of feedback controllers for flexible bevel-tip needle insertions into human tissue. When a flexible-bevel tip needle is inserted into tissue, due to the geometry of the tip asymmetric cutting forces are generated on the tip thus bending the needle. It has been shown that the needle follows a circular arc and

if the base of the needle outside the tissue is rotated, the bevel direction reorients and the needle bends in a new direction. This needle insertion has been modeled as a 6 degree-of-freedom (DOF) nonholonomic system [85] with two inputs, namely the *forward insertion* and the *rotation at the base of the needle shaft*. As a first step to controlling this system, we consider the task of driving the needle to a desired plane. The feedback signal we use is the tip position. We show that this task induces symmetry and therefore a reduction in the configuration space, thus simplifying the design of an observer–controller pair to achieve the task. Numerical simulations demonstrate the stability and robustness of the controller in the face of parametric uncertainty and measurement noise. Physical experiments with bevel-tip Nitinol needles inserted into a transparent tissue phantom under stereo image guidance validate the effectiveness of our controller.

The phantom tissue used in our laboratory experiments is transparent and thereby enables the use of a pair of stereo cameras as our imaging modality. In contrast in the operating room, the imaging modality would be 3D ultrasound, biplane fluoroscopy, MRI or CT scans or some combination of these. Using any of these imaging modalities, the needle tip position can be measured, but since the needle shaft diameter is approximately 1mm, it is challenging to directly measure the orientation of the needle. However, we show that with just the tip position measurements, the full 6-DOF pose of the needle is observable. We develop dead-beat and asymptotic observers to estimate the needle orientation.

Inspired by this application to needle steering, we generalize the notion of task-induced symmetry and reduction to other systems. Typically, the symmetries that one exploits

to obtain dimensional reduction in a mechanical system are intrinsic to the mechanical system at hand, e.g. invariance of the system's Lagrangian to some group of motions. Here, we consider symmetries that arise from an extrinsic control task, rather than the intrinsic structure of the configuration space, constraints, or system dynamics. We illustrate this technique with several examples, such as curve-following of a planar cart and subspace-following of flexible bevel-tip needles. In these examples, the reduction in configuration space enables us to design essentially global feedback controllers on the reduced systems. We also demonstrate how the proposed technique dovetails with Lagrangian reduction for second order systems.

In the second part of this thesis, we develop controllers for *Kernel-Based Visual Servoing* (KBVS). Traditionally in visual servoing, a set of feature points are tracked and these points are used either directly or indirectly as feedback signals to develop control algorithms. This approach separates the problem into two subproblems, namely tracking and control. In this work, we aim to eliminate the need for explicit tracking by combining the two subproblems. We define the weighted sum of the image (or its transform) with a smooth weighting function (called the *kernel*) as the kernel measurement of the image. We then design visual servoing algorithms that drive the kernel measurement to that measured at the goal location. We develop such algorithms for an "eye-in-hand" configuration over four of the possible six degrees of freedom, namely the three degrees of translation, and the roll about the optical axis of the camera. We implement these controllers on a 6-DOF industrial robot and present experimental results.

In summary, the contributions of the thesis are as follows:

- we design image-based controllers for steering flexible bevel-tip needles and implement the controllers in tissue phantom;

- we show that for the purpose of driving the needle to a desired plane, a reduced order system suffices for the purposes of control and estimation;

- we develop an asymptotic observer to estimate the full 6-DOF pose from the 3D position measurement when the needle is being driven to a desired plane;

- we generalize task-induction symmetry and reduction to kinematic systems on Lie groups with left-invariant vector fields and then extend it to other mechanical systems;

- we develop kernel-based visual servoing controllers, a new paradigm where we eliminate explicit feature tracking to navigate a robot to a desired location in three translational and roll degrees of freedom.

1.1 Organization of thesis

The thesis is organized as follows. In the rest of this chapter we review some of the concepts from geometric mechanics that are used in the thesis; readers familiar with these concepts can skip this section. In Chapter 2 we present an introduction of needle steering and a review of its nonholonomic kinematic model. In Chapter 3 we discuss the challenges

in feedback control of flexible bevel-tip needles and present an image-guided observer-controller to drive the needle to a desired 2D plane. We observe that to perform planar control, only three of the six degrees of freedom of the system need to be considered. In Chapter 4 we discuss the observability of the full 6-DOF pose of the needle from just its tip position measurements and present dead-beat and asymptotic observers. We generalize the concept of task-induced reduction to generic kinematic systems on Lie groups in Chapter 5 and extend it to mechanical systems as well. In Chapter 6 we present the KBVS algorithm and experiments performed on a 6-DOF industrial robot. Finally, in Chapter 7 we discuss future work and open questions resulting from this research.

1.2 Publications from the thesis

Parts of the thesis have been previously presented in the following publications [42–45]:

- V. Kallem, and N. J. Cowan. "Image-Guided Control of Flexible Bevel-Tip Needles" in *IEEE International Conference on Robotics and Automation (ICRA)*, Rome, Italy, 2007, pp 3015–2020.

- V. Kallem, D. E. Chang, and N. J. Cowan. "Task-Induced Symmetry and Reduction in Kinematic Systems with Application to Needle Steering" in *IEEE/RSJ Intelligent Robots and Systems (IROS)*, San Diego, USA, October 2007, pp. 3302–3308.

- V. Kallem, M. Dewan, J. P. Swensen, G. D. Hager, and N. J. Cowan. "Kernel-Based Visual Servoing" in *IEEE/RSJ Intelligent Robots and Systems (IROS)*, San Diego,

USA, October 2007, pp. 1975–1980.

- V. Kallem, and N. J. Cowan. "Image Guidance of Flexible Bevel-Tip Needles", *IEEE Transactions on Robotics*. Accepted.

1.3 Preliminaries

In this section, we review some of the key concepts from special Euclidean groups and fiber bundles that are used in this thesis. For more detail, see [8, 40, 49, 60, 64].

1.3.1 Special orthogonal and special Euclidean groups

The Lie groups $\mathrm{SO}(n)$ and $\mathrm{SE}(n)$ are very common in robotics. The special orthogonal group, $\mathrm{SO}(n)$, represents the rotation of rigid bodies in n dimensions and the special Euclidean group, $\mathrm{SE}(n)$, represents the general rigid body motions in n dimensions. The Lie algebra of $\mathrm{SO}(n)$ is $\mathfrak{so}(n)$ which is the group of all skew-symmetric matrices in n dimensions. $\mathfrak{se}(n) = \mathfrak{so}(n) \times \mathbb{R}^n$ is the Lie algebra of $\mathrm{SE}(n)$.

Let the symbol $\circledS$ denote the semi-direct product operator. The special Euclidean groups and special orthogonal groups are related by $\mathrm{SE}(n) = \mathrm{SO}(n)\circledS\mathbb{R}^n$. If $R_1, R_2 \in \mathrm{SO}(n)$ and $p_1, p_2 \in \mathbb{R}^n$, the product operation in the group obtained by the semi-direct product of $\mathrm{SO}(n)$ and $\mathbb{R}^n$ is given by $(R_1, p_1) * (R_2, p_2) = (R_1 R_2, R_1 p_2 + p_1)$. Note that as sets, $\mathrm{SE}(n) = \mathrm{SO}(n) \times \mathbb{R}^n$ but as groups, $\mathrm{SE}(2) = \mathrm{SO}(n)\circledS\mathbb{R}^n$. We identify the tangent

space $T\mathrm{SE}(n)$ of the Lie group $\mathrm{SE}(n)$ with

$$T\mathrm{SE}(n) \simeq \mathrm{SE}(n) \times \mathfrak{se}(n), \tag{1.1}$$

where $\mathfrak{se}(n)$ is the Lie algebra of $\mathrm{SE}(n)$.

We follow the "wedge/hat" notation given in [64]. The "wedge/hat" isomorphism $\mathbb{R} \simeq \mathfrak{so}(2)$ is defined by

$$\widehat{} : \omega \longleftrightarrow \begin{bmatrix} 0 & -\omega \\ \omega & 0 \end{bmatrix} : ^{\vee}$$

where $\mathfrak{so}(2)$ is the Lie algebra of $\mathrm{SO}(2)$. In a standard abuse of notation the "wedge/hat" isomorphism $\mathbb{R} \circledS \mathbb{R}^2 \simeq \mathfrak{se}(2)$ relates angular $\boldsymbol{\omega}$ and translational $\boldsymbol{v}$ velocities to "twists" $\xi \in \mathfrak{se}(3)$ via

$$\xi^{\vee} = \begin{bmatrix} \boldsymbol{v} \\ \omega \end{bmatrix}, \quad \text{and} \quad \begin{bmatrix} \boldsymbol{v} \\ \omega \end{bmatrix}^{\widehat{}} = \begin{bmatrix} \widehat{\omega} & \boldsymbol{v} \\ 0 & 0 \end{bmatrix} = \xi.$$

The "wedge/hat" isomorphism $\mathbb{R}^3 \simeq \mathfrak{so}(3)$ is defined by

$$\widehat{} : \begin{bmatrix} \boldsymbol{\omega}_1 \\ \boldsymbol{\omega}_2 \\ \boldsymbol{\omega}_3 \end{bmatrix} \longleftrightarrow \begin{bmatrix} 0 & -\boldsymbol{\omega}_3 & \boldsymbol{\omega}_2 \\ \boldsymbol{\omega}_3 & 0 & -\boldsymbol{\omega}_1 \\ -\boldsymbol{\omega}_2 & \boldsymbol{\omega}_1 & 0 \end{bmatrix} : ^{\vee}$$

where $\mathfrak{so}(3)$ is the Lie algebra of $\mathrm{SO}(3)$. In a standard abuse of notation the "wedge/hat" isomorphism $\mathbb{R}^3 \circledS \mathbb{R}^3 \simeq \mathfrak{se}(3)$ relates angular $\boldsymbol{\omega}$ and translational $\boldsymbol{v}$ velocities to "twists"

$\xi \in \mathfrak{se}(3)$ via

$$\xi^{\vee} = \begin{bmatrix} \boldsymbol{v} \\ \boldsymbol{\omega} \end{bmatrix}, \quad \text{and} \quad \begin{bmatrix} \boldsymbol{v} \\ \boldsymbol{\omega} \end{bmatrix}^{\widehat{}} = \begin{bmatrix} \widehat{\boldsymbol{\omega}} & \boldsymbol{v} \\ 0 & 0 \end{bmatrix} = \xi.$$

More details may be found in [60], for example. The identification (1.1) occurs via left translation, i.e.

$$(g, \dot{g}) \mapsto (g, g^{-1}\dot{g}) = (g, \xi) \tag{1.2}$$

where $\xi = g^{-1}\dot{g}$, $\boldsymbol{\omega} = \left(R^{-1}\dot{R} \right)^{\vee}$, and $\boldsymbol{v} = R^{-1}\dot{p}$.

1.3.2 Fiber bundles and principal bundles

Fiber bundles are used in the symmetry and the reduction literature to decouple systems into smaller systems. Here we give a brief overview on them. Please refer to [40, 49] for more detail.

Definition 1.3.1 *Let Q be a differentiable manifold and G be a Lie group. The **left (group)** **action** of G on Q is the function $\phi : G \times Q \mapsto Q$ defined such that*

(i) if e is the identity element of G then $\phi(e, q) = q$, $\forall q \in Q$

(ii) $\phi(g, \phi(h, q)) = \phi(gh, q)$, $\forall g, h \in G$ and $q \in Q$.

In general, group actions can be defined on any set X, but for our purposes, it is enough to define them for differentiable manifolds. The action is said to be *free* if for any two distinct $g, h \in G$, $\phi(g, q) \neq \phi(h, q)$ for all $q \in Q$.

Definition 1.3.2 *Let Q be a differentiable manifold and G be a Lie group that acts on Q.*

*Let q be an element of Q. The G **orbit of** q is defined as the set*

$$Orb(q) = \{gq \in Q : g \in G\}.$$

Definition 1.3.3 *Let Q be a differentiable manifold and G be a Lie group. The G-**orbit***

space** is defined as the set containing all the orbits of Q. It is also called as the **quotient

***space** and denoted as Q/G.*

Definition 1.3.4 *[40] A **bundle** is a triple (E, π, M) where E and M are differentiable*

*manifolds and the map $\pi : E \to M$ is a continuous map. The space E is called the **entire***

***space**, M is called the base space and the map π is called the **projection map**. The inverse*

*image $\pi^{-1}(x)$ is called the **fiber** over $x \in M$.*

If all the individual fibers $\pi^{-1}(x)$, for $x \in M$ are diffeomorphic to a common space, F,

*then F is called the fiber of the bundle and such a bundle is referred to as a **fiber bundle**.*

Locally, $E = M \times F$; if this relationship is global, the fiber bundle is *trivial*.

Definition 1.3.5 *Let H be a Lie group. If the fiber bundle (E, π, M) is such that*

(i) H acts freely on E, and

(ii) individual fibers $\pi^{-1}(x)$, for $x \in M$, are diffeomorphic to H.

*then it is called a H-**principal bundle**. Also, in this case M is homeomorphic to E/H.*

All the bundles we deal with in this thesis are *principal bundles.*

Definition 1.3.6 *Let (E, π, M) be a H-principal bundle. The map $s : U \subset M \to E$ is called a* local section *if it is defined such that*

$$\pi \circ s = \mathrm{Id}_U$$

i.e.

$$\pi(s(r)) = r, \forall r \in U.$$

This section, s, induces the **local trivialization** which is a local diffeomorphism defined by the map:

$$\psi : H \times U \to \pi^{-1}(U), \quad (h, r) \to h\, s(r).$$

Its inverse $\psi^{-1} : \pi^{-1}(U) \to H \times U$ is given by

$$\psi^{-1}(q) = (q/s(\pi(q)), \pi(q)),$$

where the operation q_2/q_1, for $q_1, q_2 \in E$ is defined as $q_2/q_1 = \{h \in H : hq_1 = q_2\}$.

If further the entire space is itself a Lie group (as in the cases discussed in this work), the inverse map $\psi^{-1} : \pi^{-1}(U) \to H \times U$ is given simply by

$$\psi^{-1}(q) = (q\, (s(\pi(q)))^{-1}, \pi(q)).$$

Part I

Task-Induced Symmetry and Reduction

with Application to Needle Steering

Chapter 2

Introduction to Needle Steering

Many diagnostic and therapeutic procedures require accurate needle targeting. In interventional brachytherapy for cancer treatment, a physician repeatedly inserts a long thin needle into human tissue, guides it to a target, and then delivers a radioactive seed. In fine needle aspiration biopsy and needle core biopsy, the physician guides a needle to a designated area to remove a small amount of tissue to test whether a tumor is malignant or benign. In thermal ablation, a needle delivers localized thermal energy to destroy malignant tissue. Needles are also used to anesthetize a part of the tissue locally. Successful outcomes for needle-based interventions such as these critically depend on accurate targeting [65,90].

Improving needle targeting accuracy and expanding the applicability of needle interventions in general, involves actively steering a needle as it is inserted into tissue. Physicians often rely on pre- or intra-operative medical imaging to guide a needle to its target. Several factors limit performance, including the amount of steering that a needle affords after it is

inserted, noisy sensors, imperfect actuators, and tissue deformations. Furthermore, navigation in 3D under image guidance by manipulating the needle at its base (from outside the patient) requires profound spatial reasoning skills and extensive training. In the past few years, there has been a large effort to overcome these limitations by improving insertion accuracy through usage of robotic systems or needle placement devices for needle insertions and better imaging modalities, needle tissue interaction models, and planning algorithms. Concurrently, new needle design research, such as on curved or pre-bent needles, flexible needles, needles with a stylus at the tip and flexible bevel-tip needles has aimed to expand the applicability of needle insertions in medical procedures by their ability to reach locations in the tissue that cannot be reached by traditional rigid symmetric-tip needles.

2.1 Previous work

Research on placement devices has focused on designing robots/devices that can be used in tandem with existing imaging systems like fluoroscopy and MRI. Stoianovici *et al.* [76] build a 2 degree-of-freedom (DOF) Remote Center of Motion (RCM) mechanism designed to conjointly work with a radiolucent needle driver for percutaneous renal access. The RCM robot is a 7-DOF robot that can be controlled by a joystick or directly through a computer. The radiolucent needle driver does not impose any obstructions to X-ray or CT scanning, and thus the RCM robot can be used along with existing X-ray and CT scanners in the operating rooms. Wei *et al.* [88] build a 6-DOF robot for prostate

needle interventions. Using real-time transrectal ultrasound (TRUS) guidance, the authors

obtained a target accuracy of less than a 1mm. DiMaio *et al.* [21] design a robot-assisted

needle placement device for prostate interventions to be used along with MRI guidance.

The device is built from non-ferrous materials and therefore the high resolution, high soft

tissue contrast of MRI images can be utilized to improve the needle targeting accuracy.

Sze [77] studies the use of curved needles to perform biopsies and drainages. The

needles are curved so that the radius of curvature is approximately that needed to avoid

sensitive organs or bones to reach a percutaneous target. Procedures were successively

conducted using real-time CT or MR images and required more than one pass, with the

needle curvature altered in between them. Carrasco *et al.* [12] also use manually curved

needles to perform liver biopsies.

Okazawa *et al.* [66] build a hand held steerable needle device; they modify an existing

needle by placing a stylet at the needle tip which can be to deployed to change the curvature

of the needle insertion path. Webster *et al.* [86] design and develop flexible bevel-tip nee-

dles that follow circular arcs when inserted into tissue. Engh *et al.* [27] add airfoil shaped

tips to these flexible needles and obtained tighter curvature of needle insertion paths. Di-

Maio and Salcudean [22] show that needles that are stiff relative to the surrounding tissue

can be steered by moving the base of the needle to deflect the tissue as the needle is inserted;

using finite elements methods, they model this effect as a kinematic control system with a

numerically determined Jacobian matrix that relates base motions to needle-tip motions.

Glozman and Shoham [31] model the interaction between a flexible symmetric-tip nee-

dle and surrounding tissue using virtual springs to compute local deformations and use it for planning purposes. Webster *et al.* [85] consider flexible bevel-tip needles that follow a curved path due to asymmetric cutting forces at the needle tip; they model this effect using a nonholonomic, kinematic system, and the present work builds directly on this model. In both the rigid and flexible cases described above, the inputs to the needle base are treated as inputs to a kinematic control system.

In an operating room image-guided needle insertions are performed using ultrasound, CT scans or MR imaging or a combination of them. Segmenting or tracking the needle in these images is non-trivial as these images are very noisy. Wei *et al.* [87] find the 3D direction of the needle trajectory inside the tissue relative to the transducer using 3D TRUS imaging. The needle segmentation is done using standard image differencing and intensity thresholding then the voxels in the difference image which are not connected are thrown away and a straight line is fit for the rest. This straight line is approximately the trajectory the rigid needle followed in the tissue. Using this technique, the authors obtained an accuracy of about $0.12°$. Ding *et al.* [24] also find the 3D direction of the needle trajectory inside the tissue using 3D TRUS imaging. They present a computationally efficient method that first finds a plane in which the trajectory of the rigid symmetric-tip needle lies in three dimension and then performs a 1D search in this plane to find the needle trajectory direction and obtain an accuracy of $1°$. While these methods are good for needle insertions done with rigid-symmetric tip needles, they will not work for flexible bevel-tip needles since needles do not follow straight lines. To find the 3D needle tip position, Abolhassani *et al.* [2] use a

Aurora magnetic tracking system from Northern Digital Inc. by place its sensor coil inside the needle. If the coil is small enough, this method is amenable to many kinds of hollow needles.

When a rigid needle is inserted into tissue, the tissue deforms and this sometimes results in the needle not reaching the target. Using linear 2D finite element models, Alterovitz *et al.* [5] compute the necessary offset to the needle entry point into the tissue to pre-compensate for these tissue deformations. DiMaio and Salcudean [22] define artificial potential fields with repulsive potentials around obstacles and couple it with the numerical Jacobian to find a path that drives a rigid needle to a desired goal location from an initial position.

Alterovitz *et al.* [3] use soft tissue modeling and local nonconvex optimization to plan paths for flexible bevel-tip needle interventions in 2D while avoiding polygon shaped obstacles. They avoid obstacles by penalizing the needle paths that enter into or pass through the obstacles. In this work, the authors assume that the bevel direction of the needle can be chosen only once and is set at the beginning of the insertion. In [4], Alterovitz *et al.* extend their planning algorithm to multiple bevel direction flips. In this work too, they limit their planning to 2D and use dynamic programming for planning algorithms that generate desired needle trajectories within a 2D plan, for flexible bevel-tip needle insertions. The output of these 2D planners is a path that can be followed by alternating between forward insertion (without rotation) of the needle into the tissue and $180°$ rotation (without insertion) of the needle base. For workspaces with no obstacles, Park *et al.* [68] formulate the

planning problem as diffusion-based motion planning on $\mathrm{SE}(3)$ to obtain feasible paths to reach the target.

Given a set of polygon-shaped obstacles, there are multiple paths that the needle can take to reach the target. If the models for tissue deformation and needle friction were perfect, the ideal path would be the shortest path. But since there are modeling errors, there could be another possibly longer path that should be taken to reach the target. Alterovitz *et al.* [6] develop a planning algorithm (based on a stochastic road map) for flexible bevel-tip needles that maximizes the success of reaching the percutaneous target while avoiding obstacles. Duindam *et al.* [25] propose a 3D motion planning algorithm for flexible bevel-tip needle insertions with spherical obstacles in the workspace. They introduce two different discretizations ((i) insert, stop and turn, and (ii) helical motions where the rotation speed of the needle is proportional to the insertion speed) and nonlinear optimization to plan for feasible paths.

Romano *et al.* [73] use the needle steering robot as a teleoperator slave, with the Freedom 6S haptic device as the master controller operated by a human under image guidance. Experiments on the teleoperation of steerable needles show that subjects employ varied strategies to perform the same task and with varying success; the success increased when the human controlled both the insertion and rotation inputs.

When a flexible bevel-tip needle is inserted in to tissue, there could be differences in the roll orientation of the base of the needle and that of the needle tip due to the friction between the tissue and needle. That is, if the needle base is rotated, the needle tip might

rotate less than the needle base; this creates torsional built up along the needle shaft. Reed *et al.* [71] conducted torsional experiments on different kinds of artificial and biological tissues. They found that plastisol, which we use as our tissue phantom, has some torsional built up but is significantly lower than that observed in porcine gelatin and ballistics gel. They have also noted that real tissue like chicken breast do not exhibit much torsional built up, likely due to the presence of lubricating liquids like blood in the tissue. Reed [70] used finite element methods to model the torsional effects, assuming that the tissue is rigid. The author uses these models to reduce the roll orientation discrepancies between the base and the tip of the needle.

Refer to Abolhassani *et al.* [1] for a recent survey on needle insertions, modeling and simulations. Building on these recent improvements in needle placement, imaging, and planning, we propose to use model-based feedback control theory for the first time, to the best of our knowledge, for real-time image-based needle steering. This approach relies on models of needle steering amenable to systems theory (as opposed to, say, finite element models). Recent efforts make progress towards such "plant models" for manipulating a needle from outside the patient. Among these needle insertion models, the one by Webster *et al.* seems to be the most amenable to a systems-theoretic approach for control.

Of particular note in the present context is the work by Glozman and Shoham [32], who develop a novel image-guidance strategy for flexible needles without a bevel tip. First they plan a needle path that avoids obstacles in the workspace. Then at every time step they invert a virtual spring model to obtain the translation and orientation of the needle

base (the inputs) in order to drive the needle back to the planned path in one step. We build on this notion by taking a systems-theoretic perspective and apply observer-based feedback control to achieve a desired task. This enables us to articulate analytical performance limits of our controller, such as the domain of attraction. Moreover, our approach only requires the extraction of the needle tip position from images, rather than the entire needle curve, thereby simplifying image segmentation.

2.2   Needle steering system overview

A flexible bevel-tip needle can be steered by rotation and insertion at the base of the needle outside the patient [28, 85]. As the flexible bevel-tip needle (Figure 2.1) is inserted, the asymmetry of the bevel creates a moment at the needle tip, deflecting the needle and causing it to follow a circular arc. As the needle base is rotated, the bevel tip is reoriented in space, so that subsequent insertion follows another circular arc in 3D space. If the rotation amount is not a multiple of $180°$ then the new circular arc lies in a different plane from the first one. A sequence of these arcs can be used to reach a point in the tissue while avoiding sensitive organs and bones.

We use the setup shown in Figure 2.2, which is similar to that described in [85], for image-guided needle steering experiments. In the setup, transparent tissue phantoms made from plastisol, which is a mixture of liquid PVC plastics and the plasticizer adipate (M-F Manufacturing Co., Inc., Fort Worth, TX), simulate human tissue. An overhead stereo pair
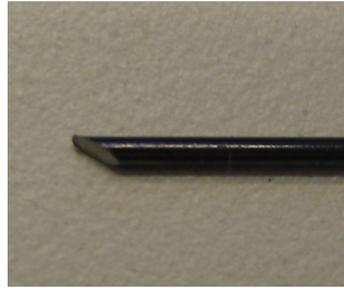
Figure 2.1: Flexible bevel-tip needle. As the needle is inserted into tissue, the asymmetry of the bevel creates a moment at the needle tip, deflecting the needle and causing it to follow a circular arc. As the needle base is rotated, the bevel tip is reoriented in space, so that subsequent insertion follows another circular arc.
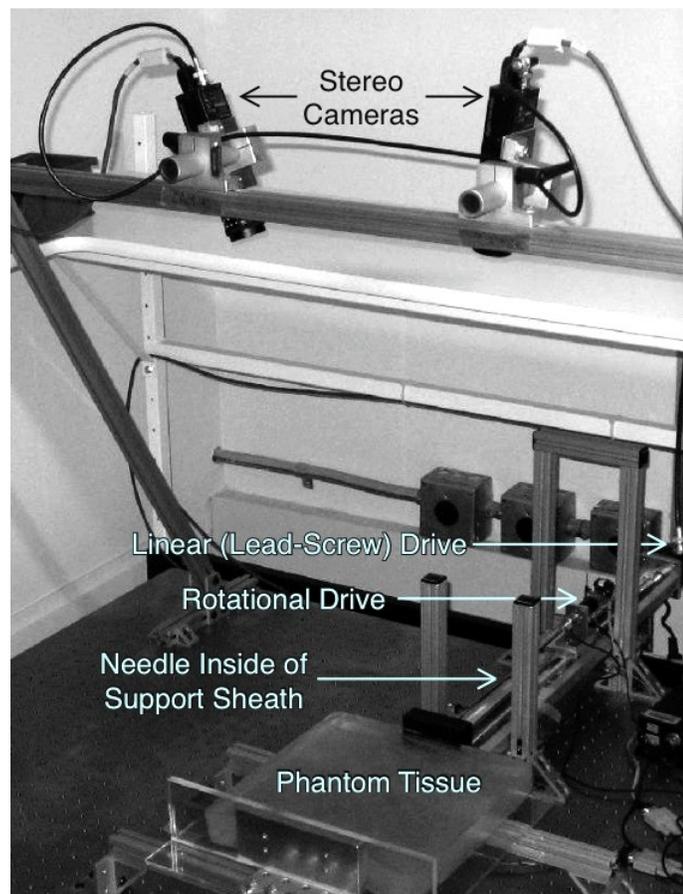
Figure 2.2: The needle steering device inserts the needle into the tissue phantom while the needle tip position is tracked using two overhead stereo cameras.

of XCD-X710 firewire cameras (Sony Corporation, Tokyo, Japan) capture images of the needle as it is inserted into the phantom by a 2-DOF needle insertion device. The insertion device is comprised of a stepper-motor-driven linear stage that drives the insertion degree of freedom and a DC servo motor that axially rotates the needle shaft. The rotary stage is attached to the base of the needle shaft, and as the linear stage drives the rotary stage forward, the needle advances into the tissue. A telescoping support sheath around the needle shaft prevents the needle from buckling outside of the tissue. The needle itself is a 0.7mm Nitinol wire (Nitinol Devices and Components, Fremont, CA), cut with an approximately $45°$ bevel tip, and pre-bent by $10°$ at 9mm from the needle tip to enhance steerability.

2.3 Review of bevel-tip needle kinematics

The insertion and rotation speeds comprise two inputs to the kinematic model for bevel-tip flexible needle steering developed by Webster *et al.* [85]. The model is a generalization of the nonholonomic bicycle model, and neglects torsional compliance of the needle shaft. This model, depicted in Figure 2.3, is reproduced here for reader convenience.

In the model, ℓ_1, ℓ_2 determine the location of bicycle wheels with respect to the needle tip. Parameter ϕ is the fixed front wheel angle relative to the rear wheel. Frame A is the inertial world reference frame and frames B and C are attached to the two wheels of the bicycle. In homogeneous coordinates, the rigid body transformation between frames A and
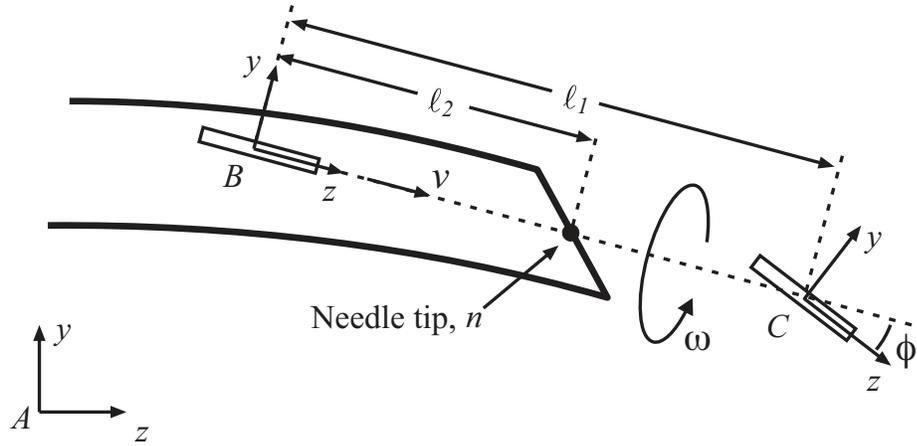
Figure 2.3: Kinematic generalized bicycle model: Frame A is the inertial world reference frame. Frames B and C are attached to the two wheels of the bicycle. This figure is reproduced from [85] with permission from the authors.
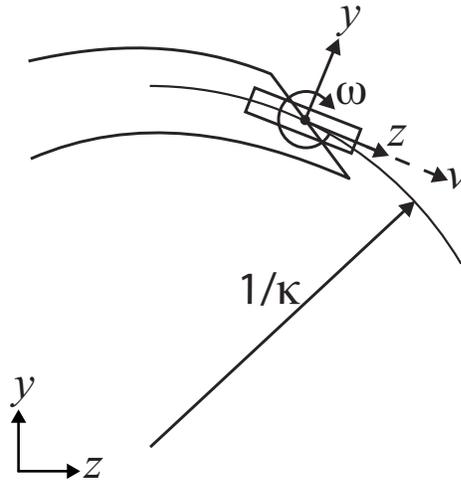
Figure 2.4: Kinematic generalized unicycle model of needle steering. This is same as the generalized bicycle model in Figure 2.3 with $\ell_2 = 0$ and keeping ℓ_1 and ϕ. The curvature of the path the needle follows inside the tissue is given by $\kappa = \tan\phi/\ell_1$. This figure is reproduced from [85] with permission from the authors.

B is given by the rigid body transformation matrix

$$g = \begin{bmatrix} R & \boldsymbol{p} \\ 0^T & 1 \end{bmatrix} \in \mathrm{SE}(3), \tag{2.1}$$

where

$$R \in \mathrm{SO}(3), \text{ and } \boldsymbol{p} \in \mathbb{R}^3.$$

We assume that the imaging system measures $\boldsymbol{p}$, the 3D location of frame B. For ease of exposition, we call $\boldsymbol{p}(t)$ the needle tip throughout this thesis; in the non-generic case that $\ell_2 = 0$ (the unicycle model discussed in [85], Figure 2.4), $\boldsymbol{p}$ indeed coincides with the needle tip.

Let $\boldsymbol{v}, \boldsymbol{\omega} \in \mathbb{R}^3$ denote, respectively, the linear and angular velocities of the needle tip written relative to frame A. Let $\mathbf{V} \in \mathbb{R}^6$ be defined as

$$\mathbf{V} = \begin{bmatrix} \boldsymbol{v} \\ \boldsymbol{\omega} \end{bmatrix}.$$

Webster *et al.* use Lie-group theory to find a "coordinate-free" differential kinematic model on the special Euclidean group, $\mathrm{SE}(3)$:

$$\mathbf{V} = (g^{-1}\dot{g})^\vee = \mathbf{V}_1 v + \mathbf{V}_2 \omega, \tag{2.2}$$

where $^\vee$ and $\hat{}$ denote the usual isomorphism between $\mathfrak{se}(3)$ and $\mathbb{R}^6$, v is the insertion speed, ω is the rotation speed of the needle, and the control vector fields are given by

$$\mathbf{V}_1 = \begin{bmatrix} 0 \\ 0 \\ 1 \\ \kappa \\ 0 \\ 0 \end{bmatrix} \quad \text{and} \quad \mathbf{V}_2 = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 1 \end{bmatrix}.$$

The vector field $\mathbf{V}_1$ corresponds to insertion of the needle into the tissue and $\mathbf{V}_2$ corresponds to rotation of the needle outside the tissue (or human). Here, $\kappa = \tan\phi/\ell_1$ is the curvature that the needle follows. Insertion of the needle, v, causes the needle to move in the body-frame z-axis direction, but also to rotate (due to the bevel tip) about the body-frame x-axis. Rotation of the needle shaft, ω, causes pure rotation of the needle tip about the body-frame z-axis. This nonholonomic system has degree of nonholonomy of four. Note that this model is only valid for forward insertions of the needle into the tissue; during the removal of the needle from the tissue, there are no cutting forces on the needle tip and hence the needle follows the path (in reverse) it followed during the forward insertion into the tissue.

2.4   Discussion

The needle steering model proposed by Webster *et al.* [85] is amenable to using control theory to automatically steer the needle to reach a target in the tissue using sensory information obtained through imaging systems. We believe feedback controls is one the main ingredients necessary for improving the accuracy of needle insertions and for enabling more medical procedures using percutaneous needle insertions. Further, our feedback control approach will also keep the physician in the loop to increase the safety of the procedure.

In the following chapters (Chapters 3-5), we present image-guided control of flexible bevel-tip needles into tissue. We discuss the challenges in automatic control of needle steering and propose an integration of high-level path planning and low-level control to reach a desired target in the tissue. We design and implement a nonlinear observer-controller that drives the needle to a desired 2D plane. This controller is designed to work together with a high-level path planner [6] to drive the needle to a target in the nominal 2D plane. We show that to design the controller we only need to consider three of the six degrees-of-freedom of the needle steering system, thus resulting in a "reduced" system. The reduced system is not nonholonomic thus allowing us to use Lyapunov theory to develop smooth controller and show its stability. Using this task-induced reduction, we develop a full 6-DOF pose estimator from just tip position measurements in Chapter 4. In Chapter 5 we extend the reduction concept to other systems on Lie groups.

Chapter 3

Image-Guided Control of Flexible

Bevel-Tip Needles

As discussed in the previous chapter, needle steering model is a nonholonomic system of degree with the degree of nonholonomy four. Brockett [10] showed that nonholonomic systems cannot be stabilized by a time invariant, continuously differentiable, state feedback controller. This problem can be solved either by using time varying controllers, or by using a hybrid time invariant feedback controller [39, 46, 53, 63]. Murray *et al.* [63] show that if a nonholonomic system can be transformed to a chained form, then it can be steered using sinusoidal inputs. However, the result holds was designed for local coordinates or when the configuration space of the system is in $\mathbb{R}^n$ and hence will need further modifications for Lie group systems when there are other constraints, such as when there are obstacles, field of view constraints, etc.

In addition to the nonholonomy in the system, there is also an unilateral constraint owing to the fact that the needle only curves when pushed forward in to the tissue (when the needle is pulled back, no cutting forces are generated and thus the needle follows the same path it went it during forward insertion into the tissue). The high degree of nonholonomy along with the unilateral constraint in the insertion speed for needle steering renders global feedback control to a point in $\mathrm{SE}(3)$ challenging, and likely impossible in practice. One possible solution is to plan a path in $\mathrm{SE}(3)$ and execute feedback control to follow the path. However, feasible paths cannot necessarily be tracked asymptotically. In fact, the natural trajectory of the system, a circle of radius $1/\kappa$, is not small-time locally "trackable", *i.e.* if there is an arbitrarily small perturbation, the needle cannot be driven back to the desired trajectory in an arbitrarily small amount of time. To see this, refer to Figure 3.1. Let the red dashed circle be of radius $1/\kappa$, which is the natural radius of curvature of the path the needle would follow in the tissue and let there be a small initial perturbation in the heading of the needle (black, solid line in the figure). Since the radius of curvature of the actual needle trajectory can never be less than $1/\kappa$, and thus even for the smallest possible perturbation of the needle heading, the needle must travel at least π/κ before re-intersecting ($*$) the desired path. Pitching out of the plane of the circle only exacerbates this issue, and the problem cannot be overcome with control or replanning.

There are several potential ways to address this problem. One could be to plan paths whose radius of curvature is always greater than $1/\kappa$. We suspect such curves can be fol-lowed arbitrarily closely (if not exactly), *e.g.* using "duty-cycling" [27, 62]. An alternative
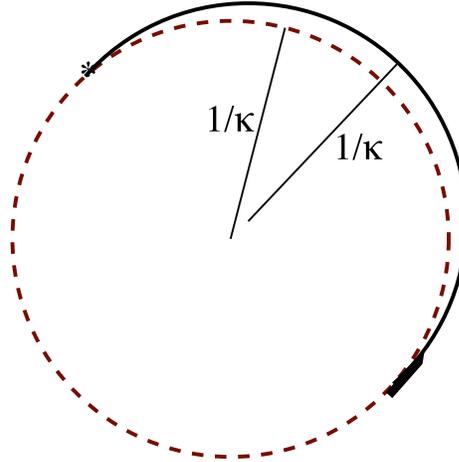
Figure 3.1: Small-time local controllability fails even for tracking a natural trajectory of the needle, a circle (red, dashed) of radius $1/\kappa$. Here, a small initial perturbation in the heading of the needle (black, solid) is depicted. The radius of curvature of the actual needle trajectory can never be less than $1/\kappa$, and thus even for the smallest possible perturbation of the needle heading, the needle must travel at least π/κ before re-intersecting ($*$) the desired path. Pitching out of the plane of the circle only exacerbates this issue, and the problem cannot be overcome with control or replanning.

approach would be to develop low-level controllers that cooperate with the 2D planners by

Alterovitz *et al.* [4, 6]. Their planners, which rely on the needle staying within a specified

2D plane, construct a sequence of circular arcs of radius $1/\kappa$ that can be achieved via alternating insertions and $180°$ rotations of the needle shaft. A real-time replanner corrects for

within plane deviations from the desired piecewise circular path. In effect, the low-level

2D plane-following controller ensures that the needle remains close to a desired 2D plane,

on top of which Alterovitz *et al.*'s planner can operate.

Real-time high-level re-planning and low-level control architecture can be generalized

to three dimensions, as depicted in Figure 3.2. We envision a "subspace planner" that

chooses feasible subspaces, upon which an Alterovitz-style planner [3] builds paths within
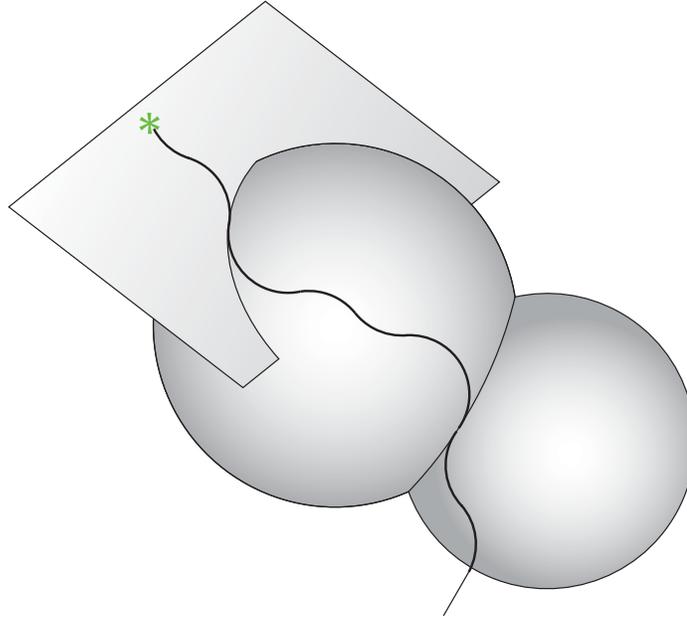
Figure 3.2: A stylized depiction of a needle trajectory that remains within a sequence of 2D patches. The plane-following controller was reported in Section 3.1, and the sphere-following problem is reported in Section 5.4.2.

each subspace. As a first step to the control of flexible bevel-tip needles, in the present chapter, we develop controllers that drive the needle tip a desired 2D plane.

3.1 Driving the needle to a desired plane

As the needle is pushed through tissue, there is a small amount of tissue deformation and the needle must be steered to avoid bones, delicate structures and sensitive organs through which it cannot or should not pass. To address this problem, Alterovitz *et al.* propose planning algorithms to generate desired needle trajectories within a 2D plane [3], for the same type of needles used in the present study. The output of these 2D planners is a path that can be followed by alternating between forward insertion (without rotation)
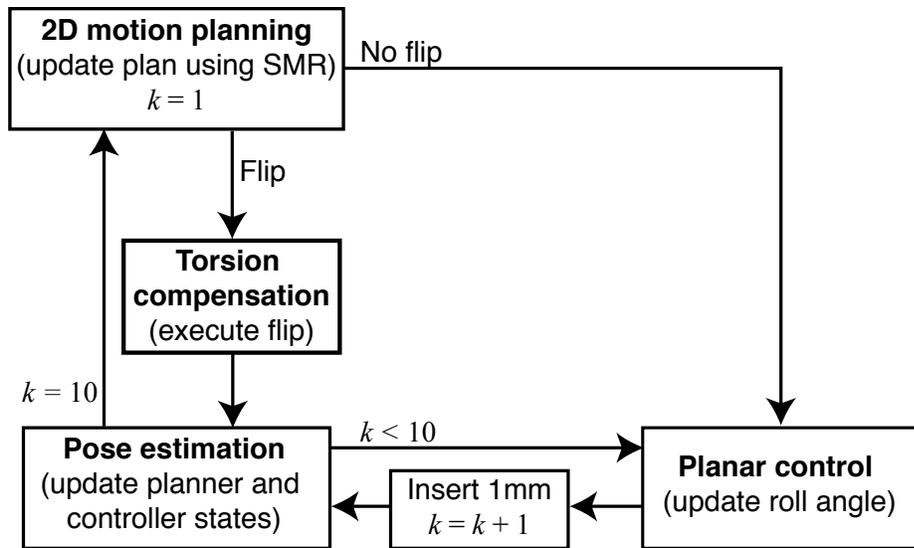
Figure 3.3: State-flow diagram: The estimator–planar controller pair operates at 1mm intervals; the estimator estimates the needle tip orientation from the tip position measurements and the planar controller maintains the needle tip near the desired plane. The planner operates at 1cm intervals and decides the bevel direction. If the planner commands a direction change, then the torsion compensator ensures that the entire needle is rotated by $180°$. k is a counter that corresponds to insertion distance [72].

of the needle into the tissue and $180°$ rotation (without insertion) of the needle base. The
planners assume that during the process the needle stays in a known (nominal) 2D plane.
However, our numerical tests indicate that small errors of only a few degrees in needle tip
orientation cause the needle to deviate rapidly from the nominal 2D plane. The goal of this
chapter is to ensure that the needle tip is stabilized to the desired 2D plane. This integration
of high-level planning and low-level control in two dimensions is shown in Figure 3.3.

3.2   Reduced-order plant model

We use Z-Y-X fixed angles as generalized coordinates to parameterize R, the rotation
matrix between frames A and B. Let γ be the roll of the needle, β be the pitch of the needle
out of the plane and α be the yaw of the needle in the plane. Let the position of the origin
of frame B be $\boldsymbol{p} = [x \ \ y \ \ z]^T \in \mathbb{R}^3$ relative to the inertial frame A. We assume that an
imaging system measures the location of the origin of frame B. Note that by driving the
origin of frame B to the y-z plane, the needle tip will also be stabilized to the y-z plane.

Using this notation, $\boldsymbol{q} = [x \ y \ z \ \alpha \ \beta \ \gamma]^T \in \mathcal{U} \subset \mathbb{R}^6$ forms a (local) set of generalized
coordinates for the configuration of the needle tip. The coordinates are well defined on

$$\mathcal{U} = \left\{ [x \ \ y \ \ z \ \ \alpha \ \ \beta \ \ \gamma]^T \in \mathbb{R}^6 : \alpha, \gamma \in \mathbb{R} \bmod 2\pi, \beta \in (-\pi/2, \pi/2) \right\}. \quad (3.1)$$

One can show that the body frame velocity is given by

$$\mathbf{V} = J\dot{\mathbf{q}}, \text{ where}$$

$$J = \begin{bmatrix} R^T & 0_{3\times3} \\ 0_{3\times3} & J_{22} \end{bmatrix}, \text{ and } J_{22} = \begin{bmatrix} \cos\beta\cos\gamma & \sin\gamma & 0 \\ -\cos\beta\sin\gamma & \cos\gamma & 0 \\ \sin\beta & 0 & 1 \end{bmatrix}.$$

The kinematic model (2.2) of the bevel tip flexible needle reduces to

$$\dot{\mathbf{q}} = J^{-1}\mathbf{V}_1 v + J^{-1}\mathbf{V}_2 \omega = \begin{bmatrix} \sin\beta & 0 \\ -\cos\beta\sin\alpha & 0 \\ \cos\alpha\cos\beta & 0 \\ \kappa\cos\gamma\sec\beta & 0 \\ \kappa\sin\gamma & 0 \\ -\kappa\cos\gamma\tan\beta & 1 \end{bmatrix} \begin{bmatrix} v \\ \omega \end{bmatrix}. \tag{3.2}$$

Due to the introduction of generalized coordinates, there are singularities at $\beta = \pm\pi/2$ that cause $\det J = \cos\beta = 0$.

To stabilize the needle to the y-z plane, the states y, z, and α need not be controlled. Also, these states do not affect the dynamics of the remaining states, x, β, and γ. Let $\mathbf{r} = [r_1 \ r_2 \ r_3]^T = [x \ \beta \ \gamma]^T$ denote the state vector of the "reduced" order system. Note that the configurations of this reduced space lie on $S^2 \times \mathbb{R}$ manifold and the "left-over" states lie on $\mathrm{SE}(2)$.

As we will see later in Chapter 5, this reduced system is the base space of the principal fiber bundle $\big(\mathrm{SE}(3), \pi, S^2 \times \mathbb{R}\big)$. The fiber space of this bundle is $\mathrm{SE}(2)$ and the projection map of the bundle is given by

$$\pi(g) = \pi(R, p) = (R^{-1}e_1, pe_1),$$

where $g \in \mathrm{SE}(3)$, and e_i's for $i = 1, 2, 3$ are principal unit vectors in three dimensions. Observe that $S^2 \times \mathbb{R} = \mathrm{SE}(3)/\mathrm{SE}(2)$. The relevant space for the purposes of the present controller is the reduced base space of the bundle and the fiber space can be ignored.

Tracking the needle tip with an imaging system typically enables us to measure only the position of the needle and not its orientation (without performing any differentiation), which in reduced coordinates is just the distance from the y-z plane, namely x. This system can be represented in state space form:

$$\dot{\boldsymbol{r}} = f_1(\boldsymbol{r})v + f_2(\boldsymbol{r})\omega = \begin{bmatrix} \sin r_2 \\ \kappa \sin r_3 \\ -\kappa \cos r_3 \tan r_2 \end{bmatrix} v + \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} \omega, \tag{3.3}$$

$$w = f_3(\boldsymbol{r}) = r_1.$$

Note that $\boldsymbol{r} = 0$ corresponds to the desired equilibrium state of remaining within the y-z plane to which we wish to stabilize the needle.

3.2.1 Reparameterization based on insertion distance

We reparameterize the system in terms of insertion distance, l, enabling the physician to control the insertion speed. By reparameterizing the kinematic models as functions of arc length, rather than time, we allow for human control of insertion speed, and our controller rotates the needle accordingly. This paradigm enhances safety by keeping the physician in the loop in a manner that enables him or her to regulate the insertion speed while monitoring the progress of corrective steering actions. A related paradigm is the notion of shared control between humans and robots in teleoperation [33].

In a slight abuse of notation, we write $\dot{r}$ where we mean dr/dl, and interpret the insertion distance as "time" for convenience of exposition.[1] This results in

$$\dot{\boldsymbol{r}} = \begin{bmatrix} \sin r_2 \\ \kappa \sin r_3 \\ -\kappa \cos r_3 \tan r_2 \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} u \tag{3.4}$$

$$w = r_1, \text{ where } u = \omega/v.$$

3.2.2 Feedback linearization

Using judiciously chosen generalized coordinates, we reduce the plant model to a third order nonlinear system (3.4). This system can be feedback linearized (see, e.g. [47]) via a

---

[1]This is equivalent to setting $v = 1$ in (3.3).

transformation of state and input coordinates:

$$\boldsymbol{s} = \left[f_3, \mathcal{L}_{f_1} f_3, \mathcal{L}_{f_1}^2 f_3 \right]^T$$

$$= \left[r_1, \quad \sin r_2, \quad \kappa \cos r_2 \sin r_3 \right]^T \tag{3.5}$$

and

$$\nu = \mathcal{L}_{f_1}^3 f_3 + \mathcal{L}_{f_2} \mathcal{L}_{f_1}^2 f_3 u = -\kappa^2 \sin r_2 + \kappa \cos r_2 \cos r_3 u, \tag{3.6}$$

where $\mathcal{L}_{f_i} f_3$ denotes the Lie derivative of f_3 along f_i for $i = 1, 2$. [2] The state equations in the feedback linearized form are:

$$\dot{\boldsymbol{s}} = A\boldsymbol{s} + B\nu = \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ 0 & 0 & 0 \end{bmatrix} \boldsymbol{s} + \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} \nu$$

$$w = C\boldsymbol{s} = \begin{bmatrix} 1 & 0 & 0 \end{bmatrix} \boldsymbol{s}. \tag{3.7}$$

The system (A, B, C) is completely controllable and observable.

---

[2] If f is a scalar function on $\mathbb{R}^n$ and X a smooth vector field on $\mathbb{R}^n$ then Lie derivative of f at $q = [q_1 \cdots q_n]^T \in \mathbb{R}^n$ is the derivative of f at q along $X(q) = [X_1(q) \cdots X_n(q)]^T$ and is defined as $\mathcal{L}_X(q)f = \sum_{i=1}^{n} \frac{\partial f}{\partial q_i} X_i(q)$.

3.3 Observer-based feedback control

In principle, the dead-beat observer (Section 4.2) and feedback-linearized plant model (Section 3.2) "solves" needle guidance to a planar patch, assuming there is no noise: in finite time, the dead-beat observer converges to the true state, enabling exact full-state feedback in the linearized system. In practice, however, there is noise in the measurements (see Section 3.4). Given the sensitivity of our dead-beat observer to noise, we consider an alternative based on an asymptotic observer. The advantage of this is that since there are only three states to estimate, and we expect to have reasonable estimates of sensor noise, the observer can be quickly and effectively tuned using the Linear Quadratic Gaussian framework, leading to successful simulations and laboratory experiments (Section 3.4). The expense, as we will see, is theoretical: we must assume that the observer error is negligible so that the feedback-linearizing change of coordinates described above can be implemented.

3.3.1 Observer and controller design

Note that even though the change of coordinates from the nonlinear system (3.4) to the feedback linearized system (3.7) is nonlinear, the first state—and, importantly, the output—is identical for both systems. In other words the system is completely observable in both coordinate systems based on the sensory measurement $w = s_1 = r_1$.

Hence, simple control system design techniques from linear system theory can be used

to control this system. A full state Luenberger observer with the following dynamics estimates all the states from the output:

$$\dot{\tilde{s}} = A\tilde{s} + Bv + L(w - \tilde{w})$$
$$\tilde{w} = C\tilde{s}. \tag{3.8}$$

The control input to the system is then given by full-state feedback, using the state estimate:

$$\nu = -K\tilde{s}. \tag{3.9}$$

Because the system is linear and time-invariant, the separability principle allows us to select the observer gain matrix, L, and proportional gain matrix, K, independently as we do in our experiments.

Note that one theoretical difficulty arises because we must compute u from (3.6), which requires exact knowledge of r. However, we do not know s nor r exactly, so we must use $\tilde{s}$ to compute an *estimate* of r by plugging $\tilde{s}$ into the inverse of (3.5). This implies that the estimator dynamics will have an input error. Here, we assume that the error computing u is negligible. Another approach may be to enhance our sensory measurement: note that it may be possible to measure the pitch, r_2, if needle shaft orientation can be segmented in both images in a neighborhood of the needle tip. In such a case, we can use a reduced-state Luenberger observer instead of the full state observer simply to estimate the roll, r_3. Measuring the roll itself may be more challenging to extract depending on the imaging

modality due to the very small size of the bevel-tip; it is infeasible with our experimental setup using stereo imaging or for other imaging modalities used in the operating rooms.

3.3.2 Stability analysis

In the present framework, there are singularities at $\beta = \pm\pi/2$ due to the introduction of generalized coordinates. In addition, the nonlinear transformation from r to s also introduces singularities at $\gamma = \pm\pi/2$. This limitation seems inescapable: global linearization is mathematically impossible for dynamical systems on the space of rigid transformations. Fortunately, our feedback linearization scheme works for all needle positions and orientations except when the needle is orthogonal to the plane to which we are trying to stabilize. We believe that this scenario is not of clinical significance; such large errors in orientation should be addressed at the level of planning, not with low-level servo control.

That said, it is important for the above described controller never to take the system – or even the state estimate! – to these singularities. In this section, we perform Lyapunov stability analysis to find the region of attraction of the controller.

Let $d \in \mathbb{R}^+$ be a positive scalar such that $d < \frac{1}{2}\min(1, 1/\kappa)$. For $s \in \mathcal{D} = \{s \in \mathbb{R}^3 : \|s\| \leq 2d\}$, the coordinate transformation mapping r to s is well-defined and invertible. This implies that β and γ never reach the singularities at $\pm\pi/2$. By defining $e = s - \tilde{s}$ as

the error in estimation, the closed-loop feedback system is now given by

$$
\begin{bmatrix} \dot{\boldsymbol{s}} \\ \dot{\boldsymbol{e}} \end{bmatrix} = \begin{bmatrix} A - BK & BK \\ 0 & A - LC \end{bmatrix} \begin{bmatrix} \boldsymbol{s} \\ \boldsymbol{e} \end{bmatrix}.
$$

The matrices K and L are chosen to make $(A - BK)$ and $(A - LC)$ Hurwitz (eigenvalues in the open left-half plane). Hence for every such K and L, there exist real symmetric positive definite matrices P and R such that $(A - BK)^T P + P(A - BK) = -I$ and $(A - LC)^T R + R(A - LC) = -I$, respectively. Consider the sets $D_e = \{\boldsymbol{e} \in \mathbb{R}^3 : \|\boldsymbol{e}\| \leq d\}$ and $D_s = \{\boldsymbol{s} \in \mathbb{R}^3 : \|\boldsymbol{s}\| \leq d\}$. We define a positive definite function $V : D_s \times D_e \rightarrow \mathbb{R}$ as:

$$
V(\boldsymbol{s}, \boldsymbol{e}) = a\boldsymbol{s}^T P\boldsymbol{s} + b\boldsymbol{e}^T R\boldsymbol{e} \text{ where } a, b \in \mathbb{R}^+.
$$

Taking the time derivative of the function V, we obtain

$$
\dot{V}(\boldsymbol{s}, \boldsymbol{e}) = -a\boldsymbol{s}^T\boldsymbol{s} - b\boldsymbol{e}^T\boldsymbol{e} + 2a\boldsymbol{s}^T PBK\boldsymbol{e}
$$

$$
= -a\left\|\boldsymbol{s} - PBK\boldsymbol{e}\right\|^2 - b\left\|\boldsymbol{e}\right\|^2
$$

$$
+ a\boldsymbol{e}^T(K^T B^T P^2 BK)\boldsymbol{e}
$$

Note that $Q = K^T B^T P^2 BK$ is a real symmetric positive semi-definite matrix. Hence we can always choose $a, b \in \mathbb{R}^+$ with $b > a\lambda_{max}(Q)$. With this choice of a and b, we observe that V is a Lyapunov function for the complete closed-loop feedback system.

Our goal is to ensure that neither the states, nor their estimates, encounter the singular-

ities introduced by feedback linearization. Note that

$$V \geq \lambda_{min}(P) \left\| s \right\|^2 + \lambda_{min}(R) \left\| e \right\|^2 \geq \epsilon \left\| s \right\|^2 + \epsilon \left\| e \right\|^2,$$

where $\epsilon = \min(\lambda_{min}(P), \lambda_{min}(R))$. If $c > 0 \in \mathbb{R}$ is chosen such that $c \leq \epsilon d^2$, then for all $(s, e) \in S = \{(s, e) \in D_s \times D_e : V(s, e) \leq c\}$, $s, \tilde{s}$ belong to the set D. If the initial deviation of the system from the desired plane is such that the initial states are in S, then the proposed controller will stabilize the needle to the desired plane without reaching any singularities.

3.4 Results

3.4.1 Numerical simulations

Extensive simulations were conducted in MATLAB (The MathWorks Inc., Natick, MA) to test our proposed controller. We used a discrete-time implementation of the system and the controller-observer pair, to reflect our physical implementation as closely as possible. The plant model was discretized assuming constant insertion by 1mm of the needle into the tissue between samples. We assumed measurement noise of up to ± 1mm with a uniform distribution; this seems clinically reasonable given that 3D ultrasound imaging can be accurate to within 0.8mm [23], and is approximately the same or slightly higher than the noise of our tracking system. The parameter value for the model was taken to be $1/\kappa = 12.2$cm,

which is the radius of curvature of the needle used in laboratory trials.

In our simulations, we observed that if the entry point was too far away from the desired plane, the estimator states (which are in the feedback-linearized coordinates) left the region in which the inverse of the change of coordinates in (3.5) is well-defined. To avoid such singularities, we performed estimator saturation, namely if the estimator states left this region, they were projected to the closest point in that region. For example, if $[\tilde{s}_1, \tilde{s}_2, \tilde{s}_3]^T = [0, 1.5, 0]^T$, then it is projected to $[\tilde{s}_1, \tilde{s}_2, \tilde{s}_3]^T = [0, 1, 0]^T$. Since we used state feedback control in the feedback-linearized space, this pull-back affects only the magnitude of the input and not the sign of the input. Our numerical tests suggested that this saturated nonlinear observer worked quite well.

Two characteristic simulations are presented in Figure 3.4, with the same initial conditions. In the first case, we tested the system without any feedback control, and it rapidly diverged from the desired plane despite relatively small errors in roll, pitch and depth. In the second simulation, our observer-based controller drove the needle to the desired plane within about 5 cm of needle insertion.

We tested our controller over a uniform grid ($10 \times 10 \times 10$) of 1000 initial conditions of up to ± 3mm error in depth from the plane, and up to $\pm 10°$ initial error in "pitch" (r_2) and up to $\pm 30°$ initial error in "roll" (r_3). In all cases, we seeded the initial condition of the observer to $\tilde{s}_2 = \tilde{s}_3 = 0°$, and for the first state, $\tilde{s}_1 = z_1 + \text{noise}$ of up to 1mm. Each initial condition was simulated 10 times with noise, for a total of 10,000 simulations. Each insertion was to a length of 12cm. We found that 98.56% of initial conditions converged to
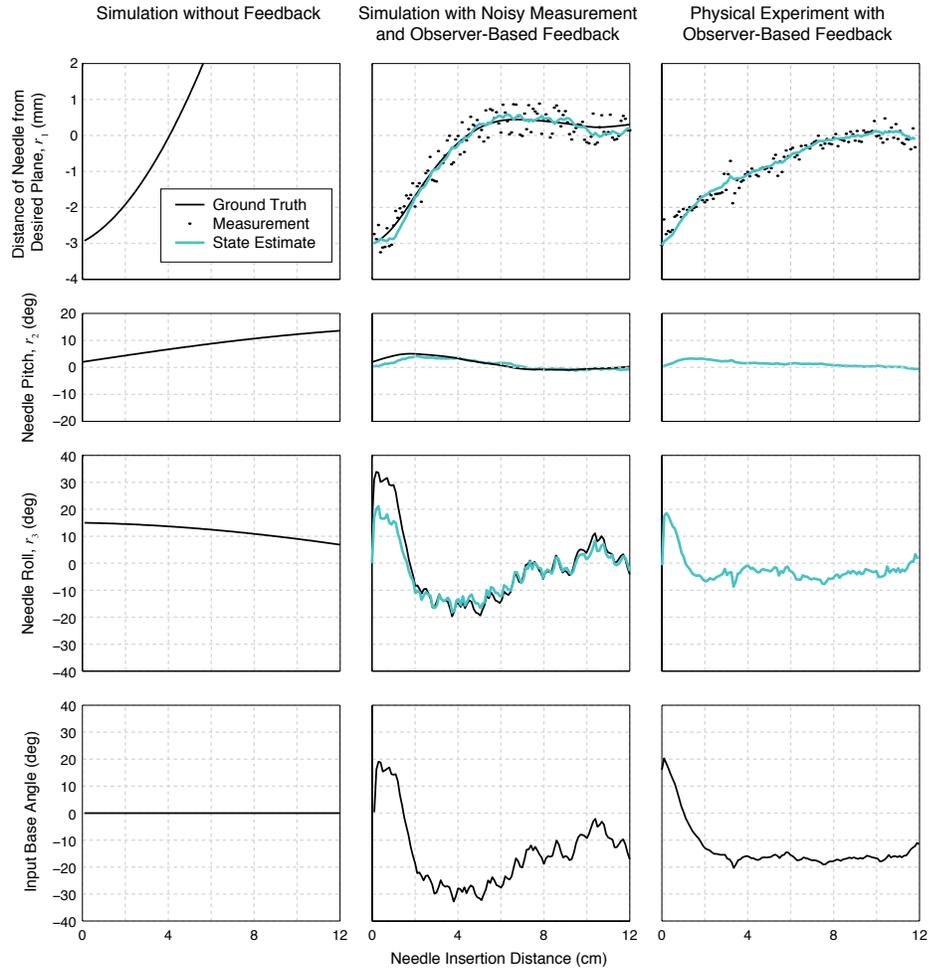
Figure 3.4: Comparison amongst a simulation with no feedback control (first column), a simulation with feedback control (second column), and an experimental trial (third column). The first three rows of plots show the three states (r_1, r_2, r_3), respectively. In the two simulations, the simulated ground truth state is known (solid black line), whereas in the physical experiment, only its estimate (solid teal line) is known. In the feedback control simulation and physical experiment, the first state is measured at each time step (small black circles). The fourth row is the cumulative rotational input given to the system. *First column:* Open-loop simulation with initial conditions of $r = [-3\text{mm}, 2°, 15°]$. With no control, the needle tip diverges the needle from the desired plane. *Second column:* Closed-loop simulation with the same initial conditions. Noise in the needle tip position is modeled as a random variable with a uniform distribution between ± 0.5mm. With the feedback control, the needle converges to the desired 2D plane within the noise levels. *Third column:* One of the nine experimental trials, with approximately the same initial conditions as the simulations (ground truth is not known). With feedback control, the needle tip converges to the desired 2D plane.

within ± 1mm (the sensor noise floor); upon closer inspection of the remaining 144 runs in which the states did not converge to within this tight tolerance in the finite needle insertion distance, we found that they did not diverge.

We tested the controller using an incorrect value of κ (up to 20% error) and found that the controller always converged, albeit slower than it would have if the correct κ was given. Thus the system appeared is robust to parametric uncertainty; to select controller gains for robustness to the parameter uncertainty see Section 3.5.

3.4.2   Experimental validation

Experiments were conducted on the needle steering device described in Section 2.3. The tissue used in the experiments was approximately 35mm thick, and it was sufficiently transparent for visual tracking purposes. We captured the images of the needle inside the tissue using XVision [36]. This tissue phantom had a refractive index of 1.3. Refraction was accounted for in our calculations by assuming that tissue's top surface was horizontal. The needle used for the experiments had a radius of curvature of 12.2cm when inserted into the tissue.

In the experiments the goal was to reach the y-z plane that was 3mm above from the initial x-position. The pitch was approximately zero, but neither the pitch nor the roll of the needle tip were precisely known. The needle was inserted into the tissue for 12cm, which is about the radius of curvature of the needle inside the tissue. Nine trials were conducted on this experimental setup with varying pitch and roll initial conditions. Figure 3.4 shows a
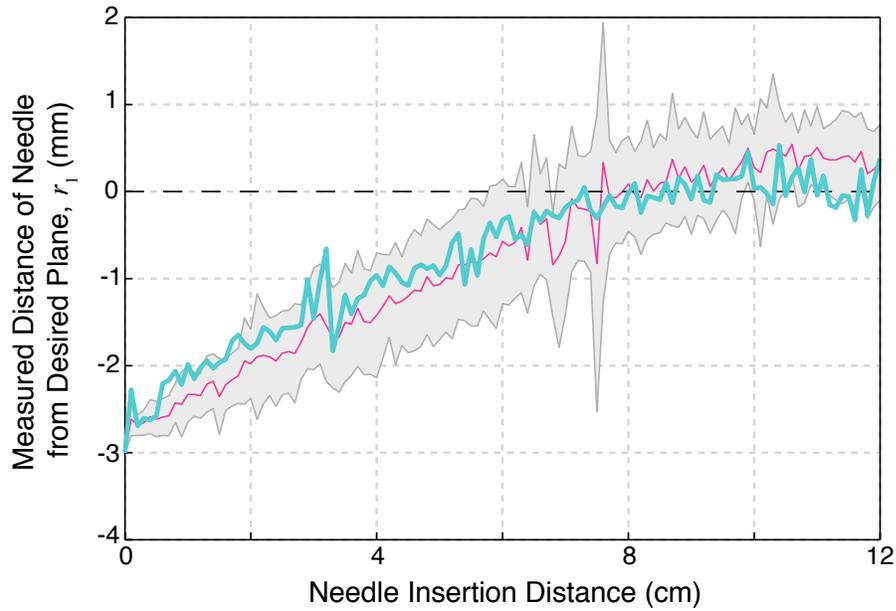
Figure 3.5: Nine experimental trials validate the controller. The mean value of r_1 of the nine trials is plotted against the insertion distance of the needle into the tissue (solid magenta line; gray region indicates mean $\pm$ standard deviation). The specific trial shown in Figure 3.4 is reproduced here (solid teal line). In all trials, the needle tip approaches the desired 2D plane ($r_1 = 0$, dashed black line) and stay within the noise levels of the position measurements of approximately 1mm.

comparison of a typical trial with our simulation results, and Figure 3.5 summarizes all nine experimental trials. In each of the trials, the needle tip converged to the desired plane within the noise levels of the position measurement. As with the simulations, these experiments validate the efficacy of our controller–observer pair and the experiments further support the nonholonomic model for flexible bevel-tip needle insertion developed by Webster *et al.* [85]. It was interesting to note, however, that while the physical and numerical results were qualitatively quite similar, the physical system exhibited a consistently more sluggish response, which we suspect was due to neglected torsional damping due to friction between the tissue and the needle shaft.

3.5   Robust control

In this section, we present a robust controller to drive the needle to the y-z plane in the face of parameter uncertainty in κ. As before, we assume that we have full state access (though in reality, we need to use an observer to estimate pitch and roll of the needle). Consider the following change of coordinates:

$$\bar{s} = \begin{bmatrix} r_1, & \sin r_2, & \cos r_2 \sin r_3 \end{bmatrix}^T$$

and

$$\bar{\nu} = \cos r_2 \cos r_3 u. \tag{3.10}$$

This change of coordinates is very similar to that in the feedback linearization of Section 3.2.2. The state equations in these coordinates are

$$\dot{\bar{s}} = A_{\bar{s}}\bar{s} + B_{\bar{s}}\bar{\nu} = \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & \kappa \\ 0 & -\kappa & 0 \end{bmatrix} \bar{s} + \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} \bar{\nu},$$

$$\bar{w} = C_{\bar{s}}\bar{s} = \begin{bmatrix} 1 & 0 & 0 \end{bmatrix} \bar{s}. \tag{3.11}$$

The system $(A_{\bar{s}}, B_{\bar{s}}, C_{\bar{s}})$ is completely controllable and observable. We want to design a feedback control $\bar{\nu} = -\bar{K}\bar{s}$ such that it is robust to parameter uncertainty. Let $\bar{K} = [\bar{K}_1, \bar{K}_2, \bar{K}_3]$. The characteristic equation of $(A_{\bar{s}} - B_{\bar{s}}\bar{K})$ is given by

$$\lambda^3 + \bar{K}_3\lambda^2 + (\kappa^2 + \bar{K}_2\kappa)\lambda + \bar{K}_1\kappa = 0. \tag{3.12}$$

Using the Routh-Hurwitz criteria, one can show that if

$$\bar{K}_1 > 0,$$

$$\bar{K}_3 > 0, \quad \text{and}$$

$$\bar{K}_2 > \frac{\bar{K}_1}{\bar{K}_3} - \kappa, \tag{3.13}$$

then the system is stable. If the bounds on the uncertainty of parameter κ are known, then the gain matrix $\bar{K}$ can be chosen such that it satisfies (3.13) with $\kappa \leftarrow \kappa_{max}$. Such a controller will be robust to any parameter uncertainties in κ, as long as $\kappa < \kappa_{max}$.

In this chapter, we have seen the controller–observer pair designed to drive the needle to a desired 2D plane. The observer is designed in the reduced space estimates the pitch and the roll of the needle from the tip position measurements. The fiber variables especially the yaw of the needle need to be estimated too if the whole 6-DOF pose is necessary for any purpose. For instance, this information is required for the implementation of 2D planners of Alterovitz *et al.* [6]. In the next chapter, we discuss deadbeat and asymptotic observers that estimate the full 6-DOF pose of the needle.

Chapter 4

Estimation of 6-DOF Needle Pose from Position Measurements

Using images captured by the stereo cameras in our experimental setup, we track the needle tip as the needle is being inserted into the tissue phantom and obtain the 3D position of the needle tip through triangulation. Since the needle is thin, there is no direct method to measure the orientation of the bevel using the stereo cameras. To our knowledge, the needle orientation cannot yet be measured using any other imaging modality, be it ultrasound, X-ray, fluoroscopy or MRI. However, this rotation information is necessary for control purposes, as discussed in Chapter 3.

Traditionally, pose estimation from images such as in the structure from motion (SFM) literature is performed based on sequential video images of a moving cloud of points; this observation typically over-constrains the underlying rigid transformation. The problem is

conceived as a nonlinear optimization problem, where the objective is to find the best-fit rigid body transformation to the image data collected in two views, or in an image sequence [7, 75, 80–83]. When the dynamical model of a system is known and only a part of the rigid body transformation can be directly measured (as in the needle steering system), the problem becomes one of designing an observer that converges—asymptotically or in finite time—to the actual transformation. In this chapter, we show that the rotation of the needle tip may be inferred from the measurements of the needle tip position over time, and we then present dead-beat and asymptotic observers that exploit the task-induced reduction to estimate the full needle pose.

4.1 Observability of orientation using position measurements

Assume that we have a needle tip position over time, denoted by $\boldsymbol{p}(t)$. Assume also that the forward insertion of the needle is held constant at $v(t) = v$. This assumption is equivalent to parameterizing the system based on insertion distance rather than time (see Section 3.2). Let $\boldsymbol{e}_i's$ for $i = 1, 2, 3$ denote the principal unit vectors in three dimensions.[1] Two successive time derivatives of $\boldsymbol{p}(t)$ and using the kinematic model (2.2) yield

$$\dot{\boldsymbol{p}}(t) = R(t)\boldsymbol{e}_3 v$$

---

[1]Here we use $\boldsymbol{e}_i's$ for $i = 1, 2, 3$ and for $i = 1, 2, 3, 4$ to denote the principal unit vectors in three and four dimensions respectively; the distinction will be clear from context.

and

$$\ddot{\boldsymbol{p}}(t) = -\kappa R(t)\boldsymbol{e}_2 v,$$

and their cross product is given by

$$\dot{\boldsymbol{p}}(t) \times \ddot{\boldsymbol{p}}(t) = \kappa R(t)\boldsymbol{e}_1 v^2.$$

Provided that $v > 0$, the rotation of the needle tip, $R(t)$, can be determined:

$$R(t) = \frac{1}{v}\left[\frac{1}{kv}\dot{\boldsymbol{p}}(t) \times \ddot{\boldsymbol{p}}(t) \quad -\frac{1}{k}\ddot{\boldsymbol{p}}(t) \quad \dot{\boldsymbol{p}}(t)\right]. \tag{4.1}$$

This analysis shows that the orientation is indeed observable from position measurements. Equation (4.1) can directly be used as an observer, but in practice this would imply finite difference approximations to first and second time derivatives of the needle tip position because of the availability of only discrete samples of $\boldsymbol{p}(t)$.

4.2 Dead-beat observer and its convergence properties

In this section we propose a dead-beat (or "batch") observer to estimate the rotation of the needle tip from position measurements. For notational convenience, we construct the estimator for the initial state $R(0)$ using the initial measurement and N future position

measurements; it is straightforward to reformulate this estimator for any particular rotation, $R(j)$, at time step j, and the observer can based on future or past (or both) samples.

Consider the discrete version of the system (2.2), given by

$$g(k) = g(k-1)U(k),$$

where $U(k) = \exp(\hat{V}_1 v(k)\Delta t + \hat{V}_2 \omega(k)\Delta t)$. The inputs $v(k)$ and $\omega(k)$ are assumed constant over each time step. The above equation may be rewritten as

$$g(k) = g(0)\Pi_{i=1}^{k}U(i) = g(0)\bar{U}(k). \tag{4.2}$$

Substituting the expression for g in (2.1) into (4.2), and comparing the fourth column on both sides (corresponding to the 3D position), we have

$$\boldsymbol{p}(k) - \boldsymbol{p}(0) = R(0)\begin{bmatrix} \boldsymbol{e}_1^T \bar{U}(k)\boldsymbol{e}_4 \\ \boldsymbol{e}_2^T \bar{U}(k)\boldsymbol{e}_4 \\ \boldsymbol{e}_3^T \bar{U}(k)\boldsymbol{e}_4 \end{bmatrix}. \tag{4.3}$$

Collecting the left-hand side of the above equation for all the samples into the matrix P,

$$P = \begin{bmatrix} \boldsymbol{p}(1) - \boldsymbol{p}(0) & \cdots & \boldsymbol{p}(N) - \boldsymbol{p}(0) \end{bmatrix},$$

and similarly the right-hand side into the matrix Q,

$$
Q = \begin{bmatrix} e_1^T \bar{U}(1)e_4 & & e_1^T \bar{U}(N)e_4 \\ e_2^T \bar{U}(1)e_4 & \cdots & e_2^T \bar{U}(N)e_4 \\ e_3^T \bar{U}(1)e_4 & & e_3^T \bar{U}(N)e_4 \end{bmatrix},
$$

where N is the number of discrete samples, we obtain a linear equation in $R(0)$:

$$
P = R(0)Q. \tag{4.4}
$$

In this equation the unknown, $R(0)$, can be estimated using constrained least-squares optimization. Alternatively, $R(0)$ may be estimated using the linear least-squares method and then projecting the resulting matrix onto the $SO(3)$ manifold [7]. If the rank of the matrix Q is rank three, then the linear-least squares estimation is performed by taking the right pseudo-inverse of Q. If the rank of the matrix Q is two, the third independent column can be obtained by stacking an extra column on both sides of the equation; on the right-hand side the extra column is the cross product of the two independent columns of Q and on the left-hand side, the extra column is the cross product of the corresponding columns of P.[2] In general, any three successive samples do not lie on a straight line. [3] Hence, if we

---

[2]This is possible since rotation preserves the cross-product between two vectors *i.e. if $R \in SO(3)$ and $p, q \in \mathbb{R}^3$ then $R(p \times q) = Rp \times Rq$.*

[3]The are two exceptions to this rule: (1) if there is no rotation of the needle base during these samples, then the three successive points will be collinear if $v_1 \Delta t = m_1 \frac{\pi}{\kappa}$ and $v_2 \Delta t = m_2 \frac{\pi}{\kappa}$, for some $m_1, m_2 \in \{1, 2, 3, \cdots\}$—in this case, the three points p_0, p_1 and p_2 will be such that either they all are the same p_0 or they on the diameter of a circle of radius $\frac{1}{\kappa}$ with p_0 as one end point; (2) if there is rotation during the sampling and if the rotations are such that $(\omega_2 \Delta t - \omega_1 \Delta t) = m\pi$, for some $m \in \{\cdots, -2, -1, 0, 1, 2, \cdots\}$ and if $v_1 \Delta t = v_2 \Delta t$—in this case, the three points p_0, p_1 and p_2 will also be collinear. Both these degenerate

have at least three samples, then the rank of the matrix Q is at least two. If there is noise in

the position measurements, the resultant matrix from the linear-least square step may not

lie on the $\mathrm{SO}(3)$ manifold and hence it may need to be projected onto the $\mathrm{SO}(3)$ manifold

using SVD methods. Once $R(0)$ is estimated, $R(k)$ can be calculated using (4.2).

We conducted simulations in MATLAB to study the impact of noise on the sample

size required to get a good estimate of $R(0)$. In the simulations, a needle with radius of

curvature 12.2cm was inserted into the tissue by 1mm and rotated by $2°$ every sample. The

error metric used in these simulations was

$$\arccos\left(\frac{1 - \mathrm{trace}(R^*(0)R(0)^{-1})}{2}\right), \tag{4.5}$$

where $R^*(0)$ represents the actual rotation at time zero and $R(0)$ is its estimation using

the dead-beat observer. Figure 4.1 shows the error in the rotation estimate plotted against

the noise in position measurements, with each curve corresponding to distance the needle

was inserted. The distance inserted corresponds to the number of samples used in the

estimation, e.g. for $\frac{1}{\kappa} = 12.2$cm, there were 122 samples (spaced at 1mm).

From Figure 4.1, we can see samples collected over a large insertion distance lead to

a significantly more accurate estimate than fewer samples over a shorter distance. The

number of sample points needed for a good estimate increases with an increase in the mea-

surement noise. Another drawback of a dead-beat observer is the lack of tuning ability to

achieve faster convergence of the controller-observer pair to a desired goal. This makes

cases are easily avoided in the experimental setup.

the dead-beat observer approach (in its current form) impractical for feedback control purposes. But the dead-beat observer can be very useful when data is analyzed in a batch mode, as well as for fitting the model parameters in (2.2), since the data set then will be large enough for a good estimate of rotation of the needle tip.
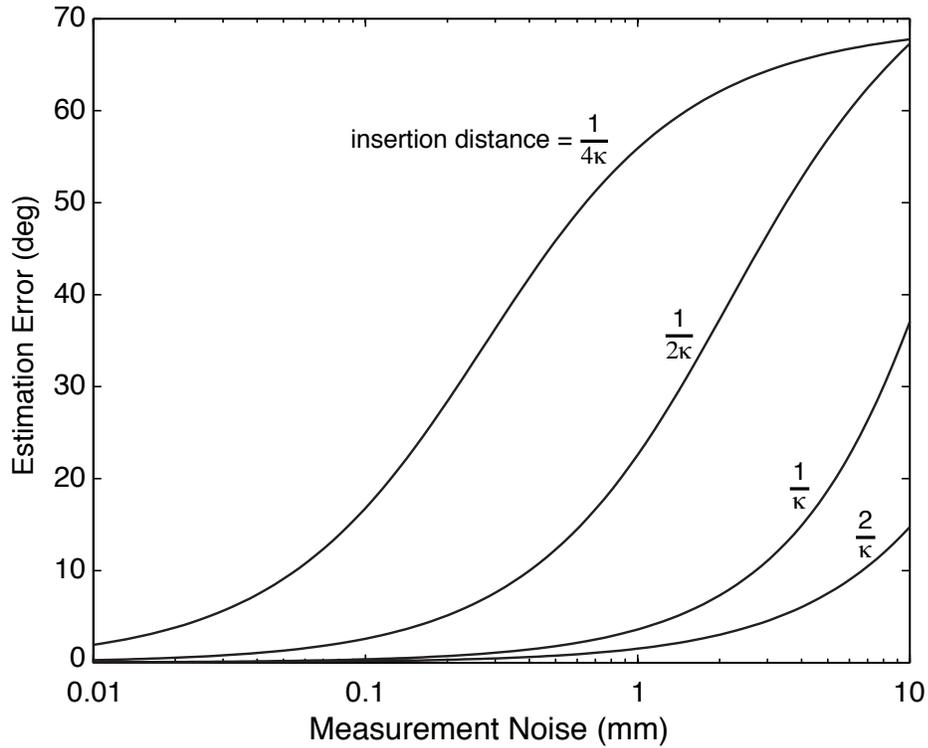
Figure 4.1: A dead-beat observer estimates the initial rotation of the needle tip, $R(0)$, in MATLAB simulations. Noisy needle tip position data was collected every 1mm of insertion, with uniform noise ranging from ± 0.01mm to ± 10mm (x-axis). The estimation error (4.5) increases with measurement noise, as shown (y-axis). Each curve corresponds to a different insertion distance (and therefore, a different number of samples collected), from $\frac{1}{4\kappa}$ to $\frac{2}{\kappa}$. Samples collected over a large insertion distance leads to a significantly more accurate estimate than fewer samples over a shorter distance. As shown, measuring needle tip positions over an insertion distance of twice the radius of curvature provides accurate estimation up to about 1mm of measurement noise, but performance degrades with shorter insertion distances.

4.3 Asymptotic observer using reduction and state immersion

In this section, we present an asymptotic observer that uses the task-induced reduction developed in Chapter 3. Previously, we have seen that to drive the needle to a desired 2D plane (y-z plane without any loss of generality), only three of the six degrees of freedom need to be considered. Using the task-induced reduction, and a change of coordinates through feedback linearization, we obtain a linear system in these three states. Using this linearized system, in Section 3.3, we developed an observer to estimate the x position, the pitch of the needle tip (β) and the roll of the needle (γ) from just x position measurements. In this section, we develop an observer for the the other three states (y, z and yaw (α)) by embedding $\mathrm{SE}(2)$ into a higher dimensional manifold, $\mathbb{R}^4$.

4.3.1 Previous work on state immersion

Linearizing nonlinear systems has tremendous advantages in systems analysis, and in designing controllers and observers. *Jacobian linearization* is a first order Taylor series expansion. Though this is a good approximation of a nonlinear system near an equilibrium point and is useful in analyzing stability properties, it is not always helpful. For example, when the eigenvalues of the linear system lie on the imaginary axis, the stability properties of the nonlinear system cannot be deduced from the linearized system. *Feedback lineariza-*

tion is another technique through which a nonlinear system can be represented by a linear system. Unlike Jacobian linearization, this technique is not an approximation. Here, an exact change of coordinates of the state space variables (both the states and inputs) is obtained by taking successive Lie derivatives of the output. Under some conditions, this technique can result in a global change of coordinates and sometimes the system structure may only result in a local (perhaps large) change of coordinates as in the case we presented in our planar control; see Section 3.2.

While in feedback linearization there is a diffeomorphism from one state space to the other, in Carleman linearization the configuration space is immersed in a higher dimensional space to obtain a linear system. In Carleman linearization, an infinite Taylor series expansion is performed on the system around an equilibrium point, and all the monomials of the states variables are assigned as the new state variables. This technique results in a bilinear system that has an infinite number of states in general. The Taylor expansion in the Carleman linearization can be truncated to obtain a finite order bilinear system, but this makes the linearization approximate. By increasing the order of the Taylor expansion, better approximations can be obtained. We refer the reader to [50, 74] for more details on this method.

In the next subsection, we present a linear model to represent three-state fiber dynamics by state immersion into a finite higher dimensional manifold. We use the output and its derivatives, in a similar manner as that in feedback linearization, and embed the $\mathrm{SE}(2)$ manifold into $\mathbb{R}^4$ to obtain a linear system. Importantly, the technique we present is not an

approximation and further the configuration space of our system is a Lie group.

4.3.2 Fiber space observer though state immersion

Assume that the needle tip is driven the needle a desired plane (the y-z plane). At this configuration, the needle tip position and orientation is such that $x = 0$, $\beta = 0$ and $\gamma = 0°$ or $\gamma = 180°$. For the current analysis, let us assume that $\gamma = 0°$ and let $v = 1$. The other three states lie on the $\mathrm{SE}(2)$ manifold and evolve with the following dynamics:

$$
\begin{bmatrix} \dot{y} \\ \dot{z} \\ \dot{\alpha} \end{bmatrix} = \begin{bmatrix} -\sin\alpha \\ \cos\alpha \\ \kappa \end{bmatrix}.
\tag{4.6}
$$

Consider a change of coordinates, $\varphi : \mathrm{SE}(2) \leftarrow \mathbb{R}^4$ defined by

$$
\varphi\left(\begin{bmatrix} y \\ z \\ \alpha \end{bmatrix} \right) = \begin{bmatrix} y \\ z \\ -\sin\alpha \\ \cos\alpha \end{bmatrix}.
\tag{4.7}
$$

Let $h = [y, z, \alpha]^T$. The Jacobian of the map φ is given by

$$
\frac{\partial \varphi}{\partial h} =
\begin{bmatrix}
1 & 0 & 0 \\
0 & 1 & 0 \\
0 & 0 & -\cos\alpha \\
0 & 0 & -\sin\alpha
\end{bmatrix}
$$

Note that the Jacobian, $\frac{\partial \varphi}{\partial h}$ is full rank every where. Also φ is a smooth mapping and there is

a one-to-one mapping between $\mathrm{SE}(2)$ and $\varphi(\mathrm{SE}(2))$. Hence, the map φ is a diffeomorphism

between $\mathrm{SE}(2)$ and $\varphi(\mathrm{SE}(2)) \subset \mathbb{R}^4$. In these coordinates, $\hbar \in \varphi(\mathrm{SE}(2))$, the dynamics are

$$
\dot{\hbar} = A_h \hbar =
\begin{bmatrix}
0 & 0 & 1 & 0 \\
0 & 0 & 0 & 1 \\
0 & 0 & 0 & -\kappa \\
0 & 0 & \kappa & 0
\end{bmatrix}
\hbar. \tag{4.8}
$$

Thus, by embedding $\mathrm{SE}(2)$ into a higher dimensional $\mathbb{R}^4$, we obtain a linear system. Also,

since we can measure the needle tip position, the output states of this system are given by

$$
\boldsymbol{y}_h = C_h \hbar =
\begin{bmatrix}
1 & 0 & 0 & 0 \\
0 & 1 & 0 & 0
\end{bmatrix}
\hbar. \tag{4.9}
$$

The system (A_h, C_h) is observable and the following Luenberger observer will estimate the

the state vector $\hbar$:

$$\dot{\tilde{\hbar}} = A_h \tilde{\hbar} + L_h(\boldsymbol{y}_h - \tilde{\boldsymbol{y}}_h)$$

$$\tilde{\boldsymbol{y}}_h = C_h \tilde{\hbar},$$

(4.10)

where $(A_h - L_h C_h)$ is Hurowitz. Note that in general $\tilde{h}(t)$ does not lie on $\varphi(\text{SE}(2))$, but

as $t \to \infty$, it lies on $\varphi(\text{SE}(2))$. The state vector h can be estimated by

$$\tilde{h} = \begin{bmatrix} \tilde{\hbar}_1 \\ \tilde{\hbar}_2 \\ \arctan 2(-\tilde{\hbar}_3, \tilde{\hbar}_4) \end{bmatrix}.$$

(4.11)

4.3.3   Coupled reduced space and fiber space observers

Now, when the needle is being driven towards the desired plane (y-z plane), the reduced

space observer can feed into the fiber space observer to estimate the full 6-DOF pose of

the needle from just position measurements; see Figure 4.2. In this case, the fiber space

dynamics is given by

$$\begin{bmatrix} \dot{y} \\ \dot{z} \\ \dot{\alpha} \end{bmatrix} = \begin{bmatrix} -\sin\alpha\cos\beta \\ \cos\alpha\cos\beta \\ \kappa\sec\beta\cos\gamma \end{bmatrix}$$

(4.12)
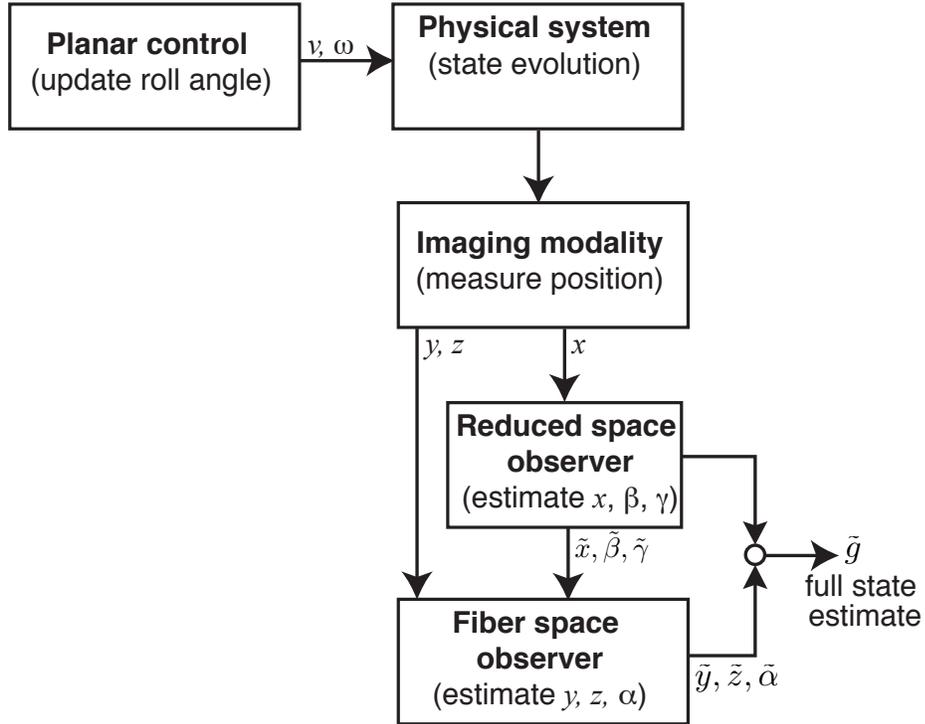
Figure 4.2: Coupled reduced space and fiber space asymptotic observers. The reduced space observer estimates x-position, pitch and roll of the needle tip from (noisy) x measurements. The fiber space observer uses these estimates along with the (noisy) y and z measurements and estimates the rest of the states of the needle.

and the fiber space observer is given by

$$\dot{\tilde{h}} = A_h T \tilde{h} + L_h(\boldsymbol{y}_h - \tilde{\boldsymbol{y}}_h)$$

$$\tilde{\boldsymbol{y}}_h = C_h \tilde{h},$$

(4.13)

where

$$
T = \begin{bmatrix}
\cos\tilde{\beta} & 0 & 0 & 0 \\
0 & \cos\tilde{\beta} & 0 & 0 \\
0 & 0 & \cos\tilde{\gamma}\sec\tilde{\beta} & 0 \\
0 & 0 & 0 & \cos\tilde{\gamma}\sec\tilde{\beta}
\end{bmatrix}.
$$

Note that when the needle is stabilized to the desired plane ($x = 0$, $\beta = 0°$ and $\gamma = 0°$), $T = I$ and the observer converges to the one given in (4.10).

4.4   Simulation results

We perform MATLAB simulations to illustrate the working of the proposed observer. The radius of curvature is assumed to be 12.2cm. Figure 4.3 is one such simulation with no noise in the position measurements. Needle position data was collected every 1mm of insertion. The figure shows the estimation errors in all the six degrees of freedom. The left column shows the estimation errors of the reduced space variables and the right column shows estimation errors in the fiber variables. The estimation error decreases from an initial error and goes to zero asymptotically.

Figure 4.4 is an example simulation run, in which the measured position data is noisy. Here, noisy needle tip position data was collected every 1mm of insertion, with measurement noise that is uniformly distributed within ±1mm in each of three coordinates of the position measurement. In this example, we can see that that the estimate errors decrease
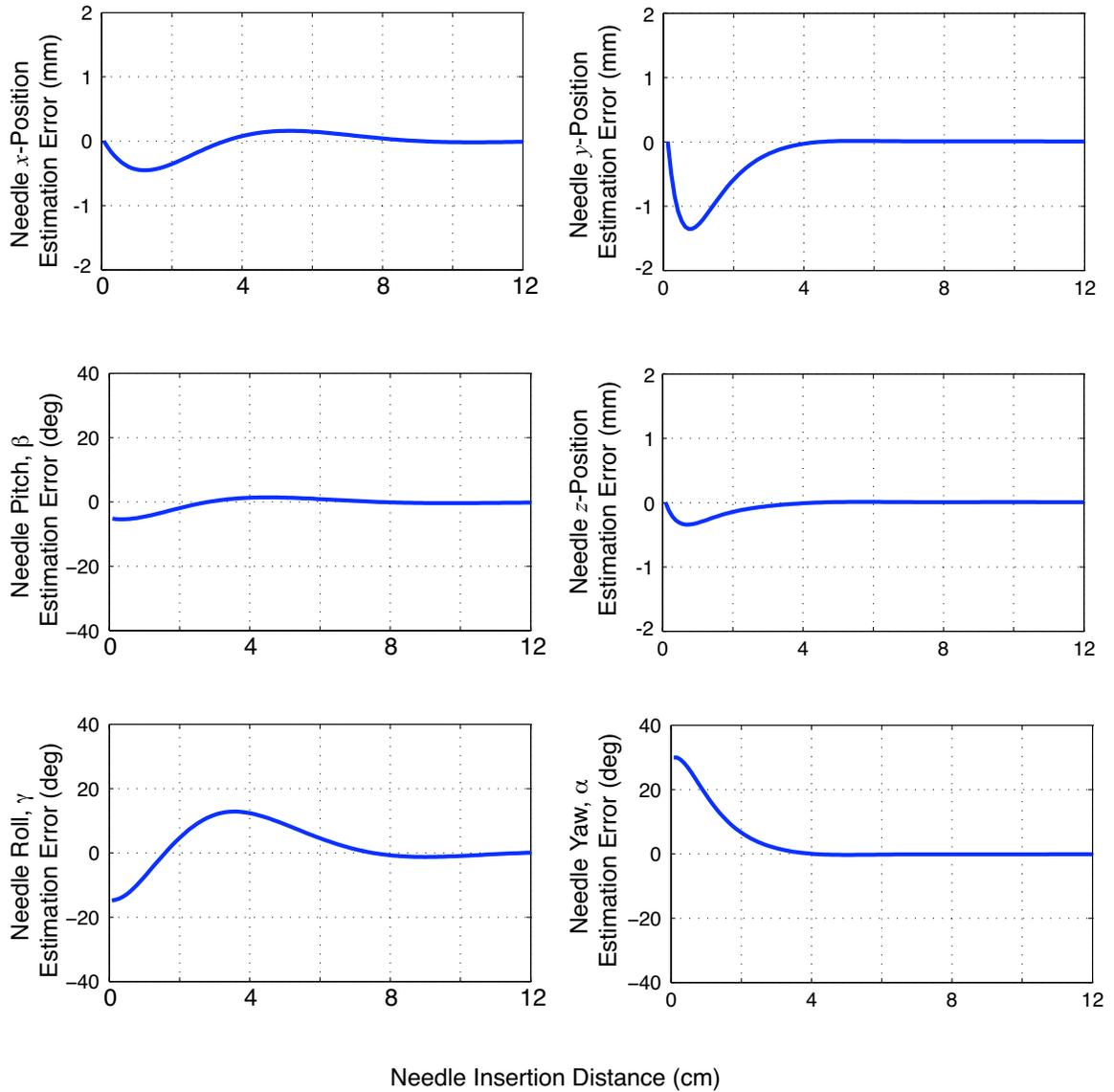
Figure 4.3: Asymptotic observer to estimate the full 6-DOF pose of needle steering from just the position measurements in MATLAB simulations. The radius of curvature of the needle in the tissue is taken as 12.2cm, tip position data is collected every 1mm of insertion with no measurement noise. The estimation errors converge to zero in all the six degrees of freedom.
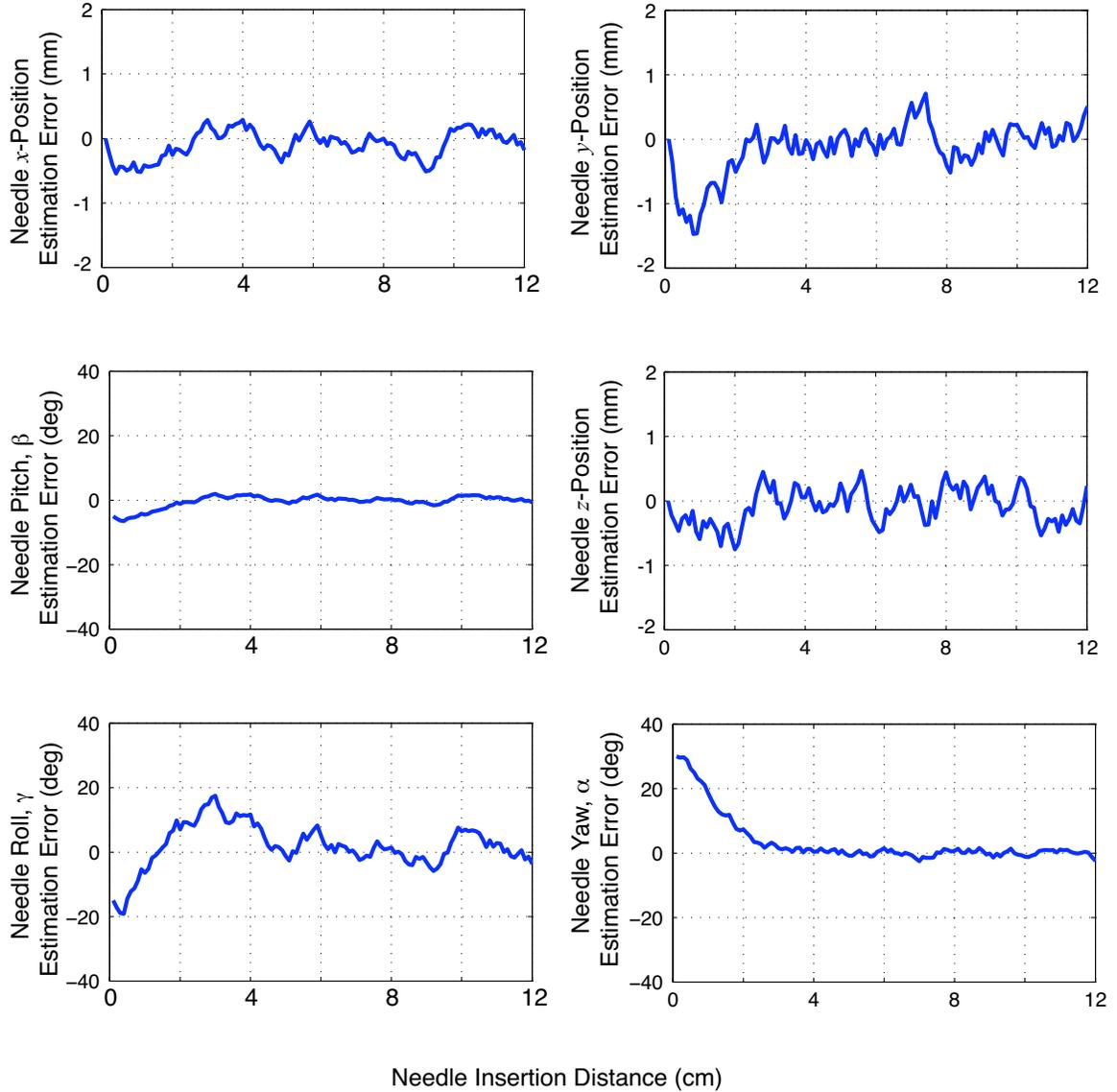
Figure 4.4: Asymptotic observer to estimate the full 6-DOF pose of needle steering from just the position measurements in MATLAB simulations. The radius of curvature of the needle in the tissue is taken as 12.2cm. Noisy needle tip position data is collected every 1mm of insertion, with measurement noise that is uniformly distributed within ±1mm. The estimation errors converge to zero in all the six degrees of freedom with a low steady-state variance.
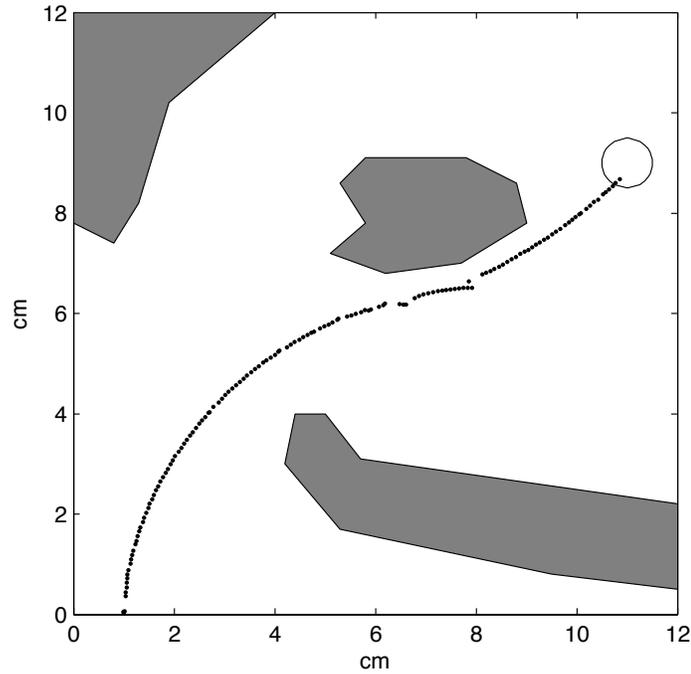
Figure 4.5: Integration of planning and control in needle steering. The goal is to reach the circular target shown in the figure. The workspace has polygon obstacles shown in the figure as dark regions. The sampled trajectory of the needle is denoted by black dots. The planar controllers acts every 1mm of needle insertion into the tissue to drive the needle to the desired 2D plane and the the planner acts is implemented at 1cm insertion intervals. With the integrated planning and control, the target is successfully reached by the needle. [72]

but oscillate around 0 due to the presence of noise. It should be noted that high gains are

used in the planar controller and observer to obtain fast convergence of the needle to the

y-z plane. If less aggressive gains are used, then the steady state oscillations are lower, thus

reducing the steady state variance in estimation.

64

4.5   Integration of planning and control

The full 6-DOF asymptotic observer and the planar controller from Chapter 3 has been integrated with the 2D planner of Alterovitz *et al.* [6] and torsional compensator of Reed [70]. Figure 4.5 shows how modules are integrated to reach a target inside the tissue. Let the nominal plane be y-z plane. Stereo cameras track the needle tip position (x, y, z) as the needle is inserted into the tissue. The observer uses these measurements to estimate the orientation of the needle tip while the low-level planar controller drives the needle to the plane. The high level planner uses the information of (y, z, α) and plans a path in the nominal plane to reach a target by planning for when the needle base needs to be rotated by $180°$ during needle insertion. When the planner acts to rotate the needle base by $180°$, torsional compensator that orients the needle tip to the same orientation of the needle base.

In the experiments, the low level estimation and control act at 1mm intervals of needle insertion while the high level planning is implemented at 1cm insertion intervals. Experiments were done using plastisol tissue phantom with a needle that has a radius of curvature of 6.1cm in the tissue. Given in Figure 4.5 is one of the integration trials [72]. The goal is to reach the circular target shown in the figure. The workspace has polygon obstacles shown in the figure as dark regions. With the integrated planning and control, the target is successfully reached by the needle.

Chapter 5

Task-Induced Symmetry and Reduction

In the previous chapters, we have seen how the task-induced reduction simplified the control and estimation of needle steering. In this chapter, we generalize the notion of task-induced reduction to other kinematic systems. We show that task-induced reduction can be performed on kinematic systems on Lie groups whose vector field is left-invariant. We then extend it to a class of mechanical systems and apply the reduction to specific examples in planar cart, needle steering and differential drive robots.

Consider a task of following a desired line in a planar cart. The motion of a simple cart in a plane, described by the usual kinematic equations of motion,

$$\dot{x} = v\cos\theta,$$

$$\dot{y} = v\sin\theta,$$

$$\dot{\theta} = \omega,$$

where (x, y, θ) is the position and orientation of the cart in the plane and the pair (v, w) is the forward and rotational velocity, treated as the control input. Suppose we are interested in making the cart follow a line, for example the x-axis. There is a natural symmetry induced by the task: any motion along the x-axis does not affect the control task. Hence to follow the x-axis, we only need to consider two of the three states and their dynamics, namely

$$\dot{y} = v \sin \theta,$$

$$\dot{\theta} = \omega.$$

In this example, reduction results from an extrinsic, "user-defined" control task, rather than an intrinsic property of the system itself.

From the trivial system above, it is not clear if a similar reduction would be possible if the goal were to follow some other shape, such as a circle, or if the equations of motion were more complicated. In the following sections we present a framework to address these questions and then demonstrate the approach with several examples on $\mathrm{SE}(2)$ and $\mathrm{SE}(3)$.

5.1 Previous work on Lie group symmetry and reduction

Previous work on reduction methods for mechanical systems deals with intrinsic symmetry in a system. If the Lagrangian of a mechanical system is invariant under the action of a Lie group, then such a system is said to exhibit symmetry and the Lie group is known as the symmetry group. In unconstrained systems with symmetry, mechanical connections as defined in [59] can be used for reduction. Bloch *et al.* [9] define a nonholonomic connection as a combination of mechanical and Ehresmann connections and use it to perform Lagrangian reduction in the presence of nonholonomic constraints. Ostrowski [67] uses these connections to reduce the systems whose configuration spaces are a direct product of the symmetry group (also called the fiber) and a manifold called a shape space (also called the base space). In the context of the present work, the symmetry group (or fiber) for a mechanical reduction is conceived as a *principal bundle* in its own right. To elucidate this connection, we compute a reduced system of equations for a differential drive mobile robot, in which we first apply Lagrangian reduction and then apply task-induced reduction.

Hanssmann *et al.* [37] perform reduction to align coordinated rigid bodies relative to one another. This alignment task induces symmetry, affording a *task-induced* reduction of the coupled system dynamics. This reduction is harnessed to developed control laws that align two rigid bodies based on the relative configuration between the bodies. While the authors take a Hamiltonian approach, similar results can be obtained using the two-stage

approach—Lagrangian reduction plus task-induced reduction—described in Section 5.5.

The work we present here relates conceptually to the literature on relative equilibria [8,11,41,57]. Roughly speaking, given a Hamiltonian system with symmetry, a relative equilibrium is a trajectory that is an equilibrium point of the reduced system. However, our primary system of interest in the present work, needle steering, is inherently highly symmetric; in fact, the needle evolves according to a left-invariant vector field on $\mathrm{SE}(3)$, rendering $\mathrm{SE}(3)$ itself the intrinsic symmetry group. That observation itself does not prove particularly useful from a design point of view; the tasks we introduce *break* some of the symmetry. Then, the "left over" symmetry enables task-specific reduction and leads us, at least in the specific examples presented, to an essentially global control strategy to achieve the task. Moreover, task-induced reduction applies to both kinematic and mechanical systems, whereas relative equilibria typically arise in mechanical systems with intrinsic symmetries.

In Section 5.2, we present a general framework to perform task-induced reduction on kinematic systems. In Sections 5.3 and 5.4 we apply this technique to a planar cart and to needle steering. We show how task-induced reduction simplifies the control design to achieve the tasks at hand. We then extend this technique in Section 5.5 to a class of Lagrangian systems. We use circle following of a differential drive robot as an example to illustrate the task-induced reduction in mechanical systems.

5.2 General framework

Let G be a finite dimensional Lie group. Consider the left-invariant [1] kinematic system

$$g^{-1}\dot{g} = \xi_0 + \sum_{i=1}^{k} \xi_i u_i \qquad (5.1)$$

where $g = g(t) \in G$ is the configuration to be controlled, $\xi_i \in \mathfrak{g}$ are constant vectors in the Lie algebra of G, k is the total number of scalar inputs to the system and u_i's for $i = 1, 2, \ldots, k$ are inputs to the system. We consider cases where the control task is encoded as the zero value of a functional $\varphi : G \to \mathbb{R}$ defined on the configuration space. One way to achieve the task is to design a (feedback) control policy such that $\varphi(g(t)) \to 0$ asymptotically.

Let H be the subgroup [2] of G invariant to the task, namely

$$H = \{h \in G : \forall g \in G, \varphi(hg) = \varphi(g)\}. \qquad (5.2)$$

Intuitively, we ignore motions in the space H, since such motions get us no closer to or further from our goal of bringing $\varphi(g)$ to zero. In effect, we perform control only in the space B that is "left over" after ignoring H. Since B is of lower dimension than G ($\dim B = \dim G - \dim H$), the kinematics (5.1) and task functional φ likely have a sim-

---

[1] A vector field X on a Lie group, G, is called left-invariant if for every $g \in G$, $(T_h L_g)X(h) = X(gh), \forall h \in G$, where L_g is a left translation of G.

[2] Note that H is a subgroup, since if $h_1 \in H$ and $h_2 \in H$, then for all $g \in G$, we have $\varphi(h_1 h_2 g) = \varphi(h_2 g) = \varphi(g)$ (closure). Moreover, the identity $e \in G$ is obviously in H. The inverse of every element in $h \in H$ is also in H, since $\varphi(g) = \varphi(eg) = \varphi(hh^{-1}g) = \varphi(h^{-1}g)$.

pler form on B than on G. In fact, in the examples that follow, the task functional reduces to a candidate Lyapunov function.

Formally, we use H and G to construct a *principal bundle*. Since H is a subgroup of G there exists a bundle projection $\pi : G \rightarrow B := G/H$. Furthermore, since H does not affect the control task, we consider the state evolution only in the base space. We use r to represent coordinates for the base space B and h for those in the fiber H. Since the vector field in (5.1) is G invariant and H is a subgroup of G, the vector field is also H invariant. This results in a well defined vector field on B that does not depend on H.

Proposition 5.2.1 *The vector field on the reduced space, B, can be calculated as*

$$\dot{r} = f_0(r) + \sum_{i=1}^{k} f_i(r)u_i \tag{5.3}$$

where $f_i(r) = T\pi \cdot g\xi_i$, and the fiber dynamics are given by

$$h^{-1}\dot{h} = \mathrm{Ad}_{s(r)} \left(\xi_0 + \sum_{i=1}^{k} \xi_i u_i - s(r)^{-1}Ts \cdot \dot{r} \right). \tag{5.4}$$

Proof: Consider a local section $s : U \subset B \rightarrow G$. By definition, a local section s satisfies $\pi \circ s = \mathrm{Id}_U$. This section, s, induces the local trivialization which is a local diffeomorphism defined by the map:

$$\psi : H \times U \rightarrow \pi^{-1}(U), \quad (h, r) \rightarrow h\, s(r).$$

Its inverse $\psi^{-1} : \pi^{-1}(U) \to H \times U$ is given by $\psi^{-1}(g) = (g \, (s(\pi(g)))^{-1}, \pi(g))$. To obtain

an expression for $\dot{r}$, differentiate $r = \pi(g)$ to obtain

$$\dot{r} = T\pi \cdot g\xi_0 + \sum_{i=1}^{k} T\pi \cdot g\xi_i u_i.$$

In order to show that the $\dot{r}$ is a function of only r and u_i, we exploit the fact that the

projection map is invariant to the left action of H. Using this fact, and choosing $h(t) \in H$

so that $g = h \, s(r)$, we calculate $\dot{r}$ as

$$\dot{r} = \frac{d}{dt}\pi(g) = \frac{d}{dt}\pi(h^{-1}g)$$

$$= T\pi(h^{-1}g) \cdot h^{-1}g\xi_i$$

$$= T\pi(s(r)) \cdot s(r)\xi_i.$$

From this equation it is clear that $\dot{r} = f_0(r) + \sum_{i=1}^{k} f_i(r)u_i$. This is the reduced kine-

matic equation on the base space B. For completeness, we derive the kinematic equation

for the fiber variable $h \in H$. Though one can employ the concept of connections for a

global derivation, we content ourselves with the derivation in local coordinates.[3] From the

definition of ψ we can rewrite $g^{-1}\dot{g}$ as

$$g^{-1}\dot{g} = (h \, s(r))^{-1}\frac{d}{dt}(h \, s(r))$$

$$= \mathrm{Ad}_{s(r)^{-1}} \, h^{-1}\dot{h} + s(r)^{-1}Ts \cdot \dot{r}.$$

---

[3]In the examples that follow, the local chart covers all but a set of measure zero of the base space.

This results in

$$h^{-1}\dot{h} = \mathrm{Ad}_{s(r)} \left(\xi_0 + \sum_{i=1}^{k} \xi_i u_i - s(r)^{-1} T s \cdot \dot{r} \right)$$

which also depends only on r and u. The equation for the fiber variable h in (5.4) is

sometimes called the reconstruction equation [58]. ∎

The above construction can be summarized in a commutative diagram:

$$
\begin{array}{ccc}
G & \supset & \pi^{-1}(U) \xrightleftharpoons[\psi]{\psi^{-1}} H \times U \\
\pi \downarrow /H & & s \uparrow \downarrow \pi \\
B = G/H & \supset & U
\end{array}
$$

We illustrate this reduction method by applying it to the tasks of following lines and

circles with a planar cart. Then, we tackle the problem of steering flexible bevel-tip needles

to follow spheres and planes in $\mathbb{R}^3$. Finally, we describe how the task-induced reduction

presented here dovetails with Lagrangian reduction for mechanical systems with intrinsic

symmetries [9, 67].

5.3 Following curves with a planar cart

In this section we revisit the planar cart example (Figure 5.1(A)) from the beginning

of this chapter and apply the above reduction framework. We then show how this can be

applied to a more interesting problem of following a circle. These toy problems are useful

because they have similar (but simpler) structure to that of the needle steering problems
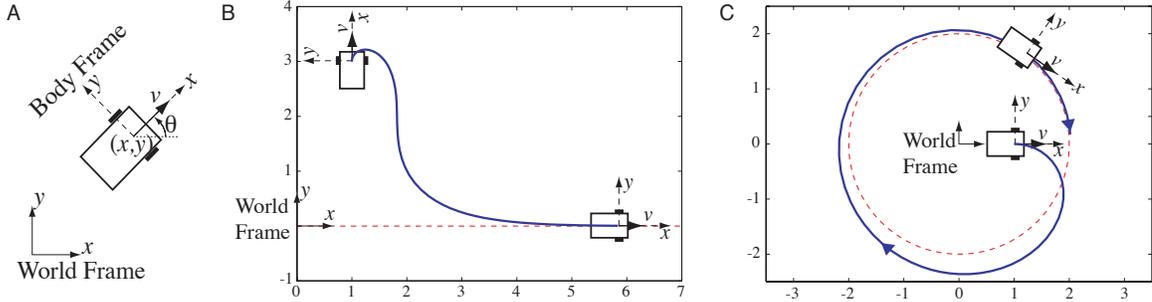
Figure 5.1: Planar cart. (A) Planar cart moving on a $2D$ horizontal plane. The x-axis of the body frame points along the forward velocity of the cart. (B) Cart following a line. Dashed line is the desired line to be followed by the cart. Initially the cart starts away from the desired line but with control, it follows the line. (C) Cart following a circle: Dashed circle is the desired circle to be followed by the cart. Initially the cart is away from the desired circle but with the application of the controller, it follows the circle. Axis scales are dimensionless for purposes of demonstration.

discussed in Section 5.4.

Consider a body reference frame attached to the cart at (x, y) with the frame x-axis

pointing along the forward translational velocity of the cart. The configuration space G

is the group of transformations in 2D, namely $G = \mathrm{SE}(2) \cong \mathrm{SO}(2)\circledS\mathbb{R}^2 \cong S^1\circledS\mathbb{R}^2$. To

denote an element $g \in \mathrm{SE}(2)$, we use both $g = (R, p) \in \mathrm{SO}(2)\circledS\mathbb{R}^2$ and its homogeneous

representation

$$g = \begin{bmatrix} R & p \\ 0^T & 1 \end{bmatrix}, \text{ where } R = \begin{bmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{bmatrix}, p = \begin{bmatrix} x \\ y \end{bmatrix}. \tag{5.5}$$

The kinematic equation of the unicycle can be written as

$$g^{-1}\dot{g} = \begin{bmatrix} \hat{\Omega} & \begin{bmatrix} v \\ 0 \end{bmatrix} \\ 0 & 0 \end{bmatrix}, \text{ where } \hat{\Omega} = \begin{bmatrix} 0 & -\omega \\ \omega & 0 \end{bmatrix}. \tag{5.6}$$

We now consider two control tasks, following a line and following a circle.

5.3.1 Straight line following

To follow the x-axis, the cart's position should be such that it is oriented parallel to the x-axis and its y-coordinate should be zero. This is precisely the example discussed in the introduction, and as shown, is trivial in local coordinates on $\mathrm{SE}(2)$; because of its simplicity, it may seem that the machinery uses in this section is more complicated than needed, but the example serves to illustrate the general framework in detail.

For the case when the cart is following the x-axis in the positive direction, this task can be defined as a zero of the function φ defined as

$$\varphi(g) = \varphi(R,p) = (1 - e_1^T R^{-1} e_1) + \frac{1}{2}(e_2^T p)^2,$$

where e_i's for $i = 1,2$ are principal unit vectors in two dimensions. When the cart is following the x-axis in the negative direction, this task can be defined as a zero of the function, $\varphi(R,p) = (1 + e_1^T R^{-1} e_1) + \frac{1}{2}(e_2^T p)^2$.

The function φ is invariant to translations along the x-axis of the world frame. Hence given this control task, the symmetry group is

$$H = \{h \in \mathrm{SE}(2) : \forall g \in \mathrm{SE}(2), \varphi(hg) = \varphi(g)\}$$

$$= \left\{ h \in \mathrm{SE}(2) \mid h = \begin{bmatrix} 1 & 0 & x \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}, x \in \mathbb{R} \right\} \cong \mathbb{R}.$$

Let $B := G/H \cong S^1 \times \mathbb{R}$. Define the projection map, $\pi : G \to B$ by

$$\pi(g) = (R^{-1}e_1, e_2^T p).$$

For all $h \in H$, $\pi(hg) = \pi(R\ ,\ p + xe_1) = (R^{-1}e_1, e_2^T p) = \pi(g)$. Therefore, π is H invariant. Also, if $g_1 \in \pi^{-1}(r)$ then $g_2 = hg_1 \in \pi^{-1}(r), \forall h \in H$. Therefore $\pi^{-1}(r) \cong H$ for each $r \in G/H$. Let $r = [r_1\ \ r_2\ \ r_3]^T \in \mathbb{R}^3$ denote an element in B with $r_1^2 + r_2^2 = 1$. Define a *global* section $s : B = S^1 \times \mathbb{R} \to \mathrm{SE}(2)$ by

$$s(r) = \begin{bmatrix} r_1 & r_2 & 0 \\ -r_2 & r_1 & r_3 \\ 0 & 0 & 1 \end{bmatrix}.$$

Note that $\pi \circ s = \mathrm{Id}_B$. Hence, the projection $\pi : \mathrm{SE}(2) \to S^1 \times \mathbb{R}$ defines a trivial bundle. With this section, a global diffeomorphism $\psi : H \times B \cong \mathbb{R} \times (S^1 \times \mathbb{R}) \to \mathrm{SE}(2)$ can be

constructed by $\psi(h, r) = hs(r)$. This principal bundle can be represented by the following diagram:

$$
\begin{array}{c}
\mathrm{SE}(2) \\
{\scriptstyle s} \uparrow\!\!\!\downarrow {\scriptstyle /\mathbb{R}} \\
B = S^1 \times \mathbb{R}
\end{array}
$$

Assuming that the forward velocity of the cart is held constant at $v = 1$, the dynamics in the reduced space can be calculated as

$$
\dot{r} = \begin{bmatrix} 0 \\ 0 \\ -r_2 \end{bmatrix} + \begin{bmatrix} r_2 \\ -r_1 \\ 0 \end{bmatrix} \omega \tag{5.7}
$$

with a holonomic constraint given by $r_1^2 + r_2^2 = 1$. The reconstruction equation for the fiber variable $h = x$ is given by

$$
\dot{x} = r_1. \tag{5.8}
$$

Observe that $R^{-1}e_1$ is the unit vector along the x-axis of the world reference frame as seen in the body reference frame. If the cart is following the x-axis, $r = \begin{bmatrix} \pm 1 & 0 & 0 \end{bmatrix}^T$ depending on whether the cart is following the line along the positive or negative x-axis. We claim that choosing the following control input will result in the cart following the desired line:

$$
\omega = kr_2 - r_3 \text{ for some } k \in \mathbb{R}^+. \tag{5.9}
$$

We use the task function reduced to the base space as a candidate Lyapunov function, $\varphi(r) = 1 - r_1 + \frac{1}{2}r_3^2$. The time derivative of φ, upon plugging (5.9), is $\dot{\varphi}(r) = -kr_2^2$ which is negative semi-definite. Each level set $\Omega_c = \{r \in B : \varphi(r) \leq c\}$, $c > 0$ is positive invariant. The subset where $\dot{\varphi} = 0$ is given by $E = \{r \in S^1 \times \mathbb{R} : r_2 = 0\} \subset \Omega_c$. The largest invariant subset in E contains only the points $r = [\pm 1 \quad 0 \quad 0]^T$. Therefore from LaSalle's invariance principle, we conclude that the cart follows the desired line. Further by considering the local chart (r_2, r_3) and by eigenvalue analysis we can see that the equilibrium point $r = [1 \quad 0 \quad 0]^T$ is stable and the other one is a saddle. The additional (unstable) critical point is a topological obstruction: our essentially global controller is the most that can be expected of a smooth feedback on $S^1 \times \mathbb{R}$. To follow the line in the other direction, simply let $\omega = r_3 - kr_2$ for some $k \in \mathbb{R}^+$.

In Figure 5.1(B) we show an anecdotal trajectory (in simulation) the cart follows using the above developed controller. As we can see from the figure, the cart is initially far away from the x-axis and it asymptotically follows the desired line.

5.3.2 Circle following

Suppose we are interested in making the cart follow a circle given by $X = \{(x, y) \in \mathbb{R}^2 \mid x^2 + y^2 - \rho^2 = 0\}$. This task can also be described as the zero of the function, φ defined by

$$\varphi(g) = \varphi(R, p) = \frac{1}{2} \left\| \rho e_2 - R^{-1}p \right\|^2.$$

One can show that $H = \{g = (R, p) : p = 0\} \simeq \mathrm{SO}(2) < \mathrm{SE}(2)$ is the largest subgroup

that leaves φ invariant. Construct a fiber-bundle $\pi : \mathrm{SE}(2) \to \mathrm{SE}(2)/\mathrm{SO}(2)$ with

$$\pi(g) = \pi(R, p) = R^{-1}p.$$

Notice that $B := \mathrm{SE}(2)/\mathrm{SO}(2) \cong \mathbb{R}^2$. If we introduce a global section $s : \mathbb{R}^2 \to \mathrm{SE}(2)$

given by

$$s(r) = \begin{bmatrix} I_{2\times 2} & r \\ 0 & 1 \end{bmatrix}$$

for $r \in B$, then we can see that π defines a trivial bundle. This $\mathrm{SO}(2)$-bundle can be

summarized by the following diagram:

$$
\begin{array}{c}
\mathrm{SE}(2) \\
s \Big\Updownarrow /\mathrm{SO}(2) \\
B = \mathbb{R}^2
\end{array}
$$

Let $r = [r_1 \ \ r_2]^T$. Assuming that the forward velocity of the cart is constant (say, $v = 1$),

the reduced space dynamics are given by

$$\dot{r} = R^{-1}\dot{p} - R^{-1}\dot{R}r = \begin{bmatrix} 1 \\ 0 \end{bmatrix} + \begin{bmatrix} 0 & 1 \\ -1 & 0 \end{bmatrix} r\omega \tag{5.10}$$

and the fiber variable θ satisfies

$$\dot{\theta} = \omega. \tag{5.11}$$

Observe that $r = R^{-1}p$ is the vector from the center of the desired circle to the origin of the body frame, written in the body frame. This implies that if the cart is following the desired circle clockwise then $r = [0 \ \ \rho]^T$, and if it is following the desired circle counter-clockwise then $r = [0 \ \ -\rho]^T$. We show that choosing the following control input will result in the cart following the desired circle clockwise:

$$\omega = -\frac{1 + r_1 k}{\rho}, \ \text{ for some } k \in \mathbb{R}^+. \tag{5.12}$$

Again we use the task function (restricted to the base space) as a candidate Lyapunov function, namely $\varphi(r) = \frac{1}{2}r_1^2 + \frac{1}{2}(r_2 - \rho)^2$. Then the time derivative of the Lyapunov function is $\dot{\varphi}(r) = r_1(1 + r_2\omega) + (r_2 - \rho)(-r_1\omega) = -kr_1^2$. We note that $\dot{\varphi}(r)$ is negative semi-definite and that the invariant subset of the set where $\dot{\varphi}(r) = 0$ consists of only $r = [0 \ \ \rho]^T$. As before, one can show using LaSalle's invariance principle that the cart follows the desired circle clockwise. To follow the circle counter-clockwise, set $\omega = \frac{1+rk}{\rho}$. Note that the convergence is global.

Figure 5.1(C) shows an example simulation of the circle following by a cart. The cart starts away from the desired circle and asymptotically follows the desired circle.

5.4 Applications to needle steering

In Chapter 3 we use local coordinates to parameterize the configuration space, $\mathrm{SE}(3)$, and perform a coordinate-based reduction "by hand" by noting that the three configuration

variables central to the control task evolved independently of the others. Here, we apply the general framework developed in Section 5.2 for two additional tasks.

It is important to understand how the planning and control architecture relates to the methodology outlined in Section 5.2. In needle steering, the entire space G is all of $\mathrm{SE}(3)$—positions and orientations of the needle tip in the tissue. The space in which the planner acts—that is, motions within a surface—is the symmetry group, or fiber, that defines the projection from the entire space into the base space. The base space is where the low-level controller operates in an effort to drive the needle back to the desired surface.

5.4.1 Stabilizing the needle tip parallel to a plane

Without any loss of generality, let the desired plane be the y-z plane. The configuration space G is $\mathrm{SE}(3)$. The control task is to stabilize the needle tip to a plane parallel to the y-z plane. In other words, it is desired that the circle the needle tip traces at steady state, during insertion without any rotation at the base, be parallel to the y-z plane.

The task can be described as the zero of the function, φ given by

$$\varphi(g) = \varphi(R, p) = 1 - e_1^T R^{-1} e_1$$

where $g \in \mathrm{SE}(3)$ is represented as in (2.1) and e_i's for $i = 1, 2, 3$ are the principal unit vectors in three dimensions. This form of the function φ is for the cases when the needle tip in steady state undergoes rotation about the positive x-axis. In the case when the needle

tip undergoes rotation about the negative x-axis, the function, φ, can be defined as $\varphi = \varphi(R, p) = 1 + e_1^T R^{-1} e_1$.

Here the symmetry group is

$$H = \{h \in \mathrm{SE}(3) : \forall g \in \mathrm{SE}(3), \varphi(hg) = \varphi(g)\}$$

$$= \left\{x \in \mathrm{SE}(3) \mid x = (e^{\hat{e}_1 \theta}, p) \text{ where } \theta \in \mathbb{R} \text{ and } p \in \mathbb{R}^3\right\}$$

$$= X \cong \mathrm{SO}(2) \circledS \mathbb{R}^3 \cong \mathrm{SE}(2) \times \mathbb{R}.$$

The symmetry group, H, can be thought of as a combination of 2D transformations in the y-z plane and translations along the x-axis relative to the world frame. With these group definitions, $B := G/H = S^2$. Define a projection map from G to B as

$$\pi(g) = \pi(R, p) = R^{-1} e_1.$$

Choose an open subset U of B as the unit sphere excluding the $(-1, 0, 0)$ point. That is $U = S^2 - \{(-1, 0, 0)\}$. Define the bundle section on U, $s : U \to \pi^{-1}(U) \subset G$ as

$$s(r) = (e^{\hat{\varpi}(r)\pi}, 0),$$

where $\varpi(r) = \frac{e_1 + r}{\|e_1 + r\|}$. This section can be seen in Figure 5.2. With this section, locally $g \in G$ can be expressed as $g = hs(r) \simeq (h, r)$, where $h \in H$ and $r \in B$. To completely
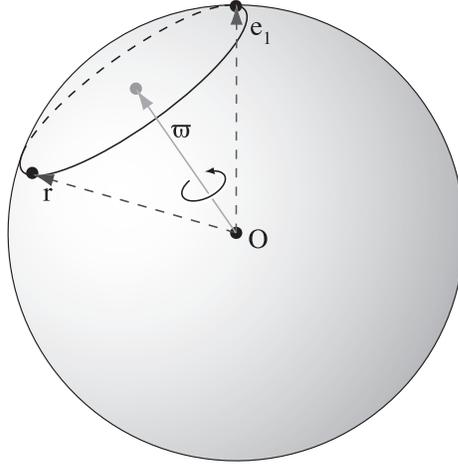
Figure 5.2: Illustration of the bundle section. Unit vector ϖ bisects r and e_1. If the r is rotated about ϖ as axis by π then it coincides with e_1. The circle through the points r and $(1,0,0)$ is a latitude of S^2 drawn perpendicular to ϖ. Note that $\pi \cdot s(r) = (e^{\hat{\varpi}(r)\pi})^{-1}e_1 = r$.

cover S^2, consider $U' = (S^2 - \{(1,0,0)\})$ and a similar section can be constructed.[4] These make the map π a principal bundle which can be summarized by the following commutative diagram:

$$
\begin{array}{ccc}
\text{SE}(3) & \supset & \pi^{-1}(U) \xrightleftharpoons[\psi]{\psi^{-1}} H \times U \\
{\scriptstyle \pi}\Big\downarrow {\scriptstyle /SO(2)\times\mathbb{R}^3} & & {\scriptstyle s}\Big\uparrow\Big\downarrow{\scriptstyle \pi} \\
B = S^2 & \supset & U = S^2 - \{(-1,0,0)\}
\end{array}
$$

We assume that the insertion speed is held constant at $v = 1$. Base space elements can be written in coordinates as $r = [r_1 \ \ r_2 \ \ r_3]^T \in \mathbb{R}^3$ with $r_1^2 + r_2^2 + r_3^2 = 1$. The base space

---

[4]One can choose $s'(r) = (e^{\hat{e}_3 \pi} e^{\hat{\varpi}' \pi}, 0)$, where $\varpi' = \frac{-e_1 + r}{\|-e_1 + r\|}$.

dynamics are

$$
\dot{r} = \begin{bmatrix} 0 \\ \kappa r_3 \\ -\kappa r_2 \end{bmatrix} + \begin{bmatrix} r_2 \\ -r_1 \\ 0 \end{bmatrix} \omega.
\tag{5.13}
$$

The fiber variable $h = (e^{\hat{e}_1 \theta}, p) \in H$ satisfies

$$
\dot{\theta}\hat{e}_1 = \mathrm{Ad}_{\bar{s}(r)}\left(\kappa\hat{e}_1 + \omega\hat{e}_3 - \bar{s}(r)^{-1}\frac{\partial \bar{s}(r)}{\partial r}\dot{r} \right),
$$

$$
\dot{p} = e^{\hat{e}_1 \theta}\bar{s}(r)e_3,
\tag{5.14}
$$

where $\bar{s}(r) = e^{\hat{\varpi}(r)\pi}$.

For the task to be completed, the state in the reduced system should be $r = [\pm 1 \ \ 0 \ \ 0]^T$.

Choosing the following control input we show that the task is achieved:

$$
\omega = kr_2, \text{ for some } k \in \mathbb{R}^+.
\tag{5.15}
$$

Once again, we use the task function in base coordinates as a candidate Lyapunov function,

namely $\varphi(r) = 1 - r_1$. The time derivative of the Lyapunov function is $\dot{\varphi}(r) = -kr_2^2$ which

is negative semi-definite in r. The set where $\dot{\varphi} = 0$ is given by $E = \{r = [r_1 \ \ r_2 \ \ r_3]^T \in$

$S^2 \mid r_2 = 0\}$. The largest invariant set in E contains only the points $r = [\pm 1 \ \ 0 \ \ 0]^T$.

Hence, from LaSalle's invariance principle, we can conclude that the needle tip can be

stabilized to a plane parallel to the desired plane. Notice that *asymptotically* the fiber

dynamics become

$$\dot{\theta} = \kappa,$$

$$\dot{p} = (0, \sin\theta, -\cos\theta)^T.$$

This shows us that the fiber variables trace a circle of radius $1/\kappa$, which we knew from the needle steering model. Further by considering the local chart (r_2, r_3) and by eigen-value analysis one can show that the equilibrium point $r = \begin{bmatrix} 1 & 0 & 0 \end{bmatrix}^T$ (corresponding to the needle tip tracing a circle of radius $1/\kappa$ parallel to the y-z plane counter clockwise) is stable, and the other equilibrium point unstable. For the needle tip to trace the circle clockwise, let $\omega = -kr_2$ for some $k \in \mathbb{R}^+$.

Figure 5.3(A) shows an example simulation run. The initial condition is such that without any control the needle tip would trace circle not parallel to the y-z plane. With the use of the above developed controller, the needle tip in steady state traces a circle parallel to the desired y-z plane.

5.4.2   Controlling the needle tip to a sphere

In this section we use the reduction technique developed in section 5.2 to control the needle tip to the surface of a sphere with radius $\rho > 1/\kappa$. If the needle tip stays on the surface of the desired sphere at steady state it has to follow a circle of radius $\varrho = 1/\kappa$. Let $d = \sqrt{\rho^2 - \varrho^2}$. If the needle is staying on such a circle on the surface of the sphere, the
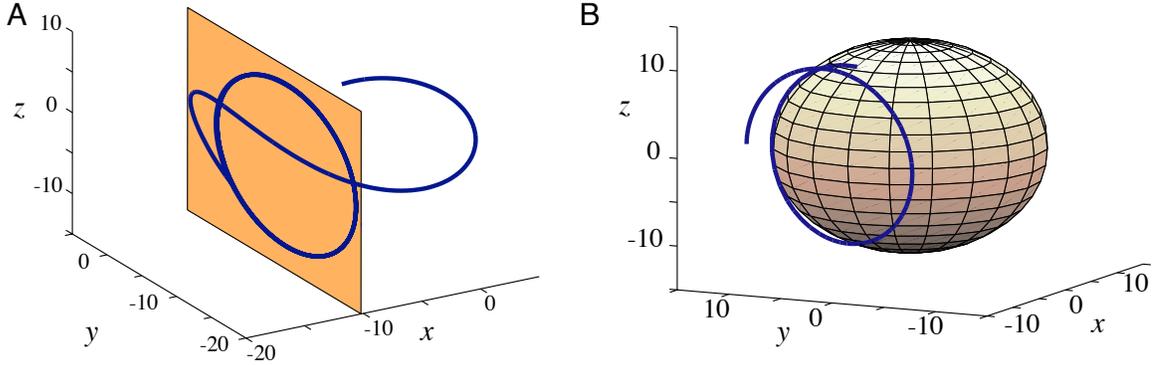
Figure 5.3: (A) Needle tip driven so that at needle tip trajectory becomes parallel to the y-z plane. (B) Needle tip driven to the desired sphere. Initially the needle tip is away from the desired sphere (drawn in the figure) but with the application of the controller, it converges to the surface of the sphere. Axis scales are in cm.

position vector of the center of the sphere written in the body frame is given by $R^{-1}p$. This must be $[-d \ \varrho \ 0]^T$ or $[d \ \varrho \ 0]^T$ depending on which direction the needle tip is following the circle of radius ϱ.

In this example, the control task can also be described as the zero of the function

$$\varphi(g) = \varphi(R, p) = \left\| R^{-1}p - \left[-d, \varrho, 0 \right]^T \right\|^2$$

or

$$\varphi(g) = \varphi(R, p) = \left\| R^{-1}p - \left[d, \varrho, 0 \right]^T \right\|^2$$

depending on the direction the needle tip follows the circle. It is easy to see that $H = \mathrm{SO}(3)$ leaves the function, φ, invariant. Hence, the symmetry group, H, for this control task is $\mathrm{SO}(3)$. The base space, $B := G/H = \mathbb{R}^3$. The bundle projection $\pi : \mathrm{SE}(3) \to \mathbb{R}^3$ is given

by

$$\pi(g) = R^{-1}p$$

where $g = (R, p) \in \mathrm{SO}(3)\mathbb{S}\mathbb{R}^3 \cong \mathrm{SE}(3)$. This bundle is trivial because we can construct

a global section $s : \mathbb{R}^3 \to \mathrm{SE}(3)$ as

$$s(r) = \begin{bmatrix} I_{3\times3} & r \\ 0^T & 1 \end{bmatrix}.$$

Given $(R, r) \in H \times B = \mathrm{SO}(3) \times \mathbb{R}^3$, the corresponding $g \in \mathrm{SE}(3)$ can be calculated

as $g = \begin{bmatrix} R & Rr \\ 0 & 1 \end{bmatrix}$. This principal bundle can be represented by the following commutative

diagram:

$$\mathrm{SE}(3)$$
$$s \Big\Uparrow\Big\downarrow {}_{/\mathrm{SO}(3)}$$
$$B = \mathbb{R}^3$$

Assuming $v = 1$, the dynamics in the base space can be written as

$$\dot{r} = \begin{bmatrix} 0 \\ \kappa r_3 \\ 1 - \kappa r_2 \end{bmatrix} + \begin{bmatrix} r_2 \\ -r_1 \\ 0 \end{bmatrix} \omega \tag{5.16}$$

and the fiber variable $R \in H = \mathrm{SO}(3)$ satisfies

$$R^{-1}\dot{R} = \kappa \hat{e}_1 + \hat{e}_3 \omega. \tag{5.17}$$

The control input

$$\omega = -k(dr_2 + \varrho r_1) \text{ for some } k \in \mathbb{R}^+ \tag{5.18}$$

makes the base variable, r, converge to $[-d \ \ \varrho \ \ 0]^T$. Consider the reduced task functional

as the candidate Lyapunov function: $\varphi(r) = \frac{1}{2}(r_1+d)^2 + \frac{1}{2}(r_2-\varrho)^2 + \frac{1}{2}r_3^2$. Its time derivate,

$\dot{\varphi}(r) = \omega(dr_2 + \varrho r_1) = -(dr_2 + \varrho r_1)^2$, is negative semi-definite. The invariant subset of

the set where $\dot{\varphi}(r) = 0$ contains only $r = [-d \ \ \varrho \ \ 0]^T$. Again using LaSalle's invariance

principle we can conclude that the needle tip stays on the surface of the desired sphere. To

follow the circle in the desired sphere in the other direction, choose $\omega = dr_2 - \varrho r_1$.

Figure 5.3(B) is an anecdotal simulation of the needle staying on the surface of a desired

sphere. The needle starts away from the desired sphere and asymptotically converges to the

surface of the sphere.

5.5 Task-induced reduction on mechanical systems

In this chapter we present how the above developed task-induced reduction extends to the Lagrangian systems. In the first stage, we perform a previously developed mechanics-based "intrinsic" reduction to separate the configuration space into a shape space and a symmetry (Lie) group. We then perform task-induced reduction to further reduce the symmetry group into a base space and a fiber as in Section 5.2.

5.5.1 Review of Lagrangian reduction on mechanical systems

Consider the mechanical system whose configuration space is Q, Lagrangian is $L(q, \dot{q})$, $(q, \dot{q}) \in TQ$ and whose constraints, if they exist, are nonholonomic and written as $w(q)\dot{q} = 0$. Assume that the the Lagrangian and the constraints are invariant to a left action of a Lie group, G. It has been previously shown that such a system can be decoupled into the symmetry group, G and the shape space, $S = Q/G$,[5] namely $Q = G \times S$ (locally). The reduced equations are given in Proposition 5.5.1.

Proposition 5.5.1 *(Ostrowski [67]) For the above described mechanical system, assume that the input forcing occurs only in the shape space and that no constraint lies completely*

---

[5]In the literature the space $S = Q/G$ is referred as both shape space and as base space. In the current work, we only refer to it only as the shape space and $B = G/H$ (defined in Section 5.2) as the base space.

in the shape space. The reduced system equations can be written as

$$M(\sigma)\ddot{\sigma} = -C(\sigma, \dot{\sigma})\dot{\sigma} - N(\sigma, \dot{\sigma}, \Pi) + \tau \tag{5.19}$$

$$\dot{\Pi} = f(\sigma, \dot{\sigma}, \Pi), \tag{5.20}$$

where $\sigma \in S$ denotes the shape variable, $\Pi \in \mathfrak{g}^*$ denotes the generalized momenta and τ the generalized input force on the systems. The symmetry group variable, $g \in G$, can be reconstructed as

$$g^{-1}\dot{g} = -\mathbb{A}(\sigma)\dot{\sigma} + \bar{I}^{-1}(\sigma)\Pi. \tag{5.21}$$

If there are m nonholonomic constraints, then there are $n - m$ generalized momenta, where n is the dimension of the Lie group, G. In particular, for principally kinematic systems $m = n$ thus resulting in no generalized momenta. In other words, for these systems the group variable can be reconstructed from the nonholonomic constraint equations and this simplifies the reduced system equations further. More details on the structure and computations can be found in propositions 4.3 and 5.1 in [67] and also in [8, 9].

5.5.2   General framework

Observe that the symmetry group dynamics given in (5.21) is a left-invariant vector field on the Lie group, G. Assume that there is a task defined completely on the group space. If this task induces a symmetry as described in previous sections, we can further

decompose G. As before, we consider cases where the control task is encoded as the zero value of a functional $\varphi : G \to \mathbb{R}$ defined on the configuration space. Let H be the subgroup of G invariant to the task, namely

$$H = \{h \in G : \forall g \in G, \varphi(hg) = \varphi(g)\}.$$

Since H is a subgroup of G, there exists a bundle projection $\pi : G \to B := G/H$. Also since H does not affect the control task, we consider the state evolution only in the base space.

Corollary 5.5.2 *The reduced equations in the base space, $B = G/H$, are*

$$\dot{r} = f_1(r, \sigma, \Pi) + f_2(r, \sigma)\dot{\sigma} \tag{5.22}$$

where $f_1(r, \sigma, p) = T\pi \cdot g\bar{I}^{-1}p$ and $f_2(r, \sigma) = -T\pi \cdot g\mathbb{A}(\sigma)$ and those in the fiber, H, are

$$h^{-1}\dot{h} = \mathrm{Ad}_{s(r)}\left(-\mathbb{A}(\sigma)\dot{\sigma} + \bar{I}^{-1}(\sigma)\Pi - s(r)^{-1}Ts \cdot \dot{r}\right). \tag{5.23}$$

Proof: Following the proof for the proposition 5.2.1, the base dynamics (5.22) result from the observation that the vector field on G is left-invariant to G and the projection π is invariant to H, which is a subgroup of G. The fiber dynamics (5.23) is exactly the same computation as (5.4). ∎

Locally, Proposition 5.5.1 decomposes an entire configuration space as the cross prod-

uct of the symmetry group and shape space, namely $Q = G \times S \equiv G \times (Q/G)$. Corollary

5.5.2 can be conceived as further decomposing G into a base and fiber in its own right, due

to the task-induced symmetry fiber H, so that we have

$$
Q = H \times \overbrace{\underbrace{(G/H)}_{B} \times \overbrace{(Q/G)}^{S}}^{G} \tag{5.24}
$$

at least locally, or globally in the case that $G = H \times B$ and $Q = G \times S$ are both trivial

bundles. Here, Q is decomposed via the mechanical reduction of Proposition 5.5.1, and G

is further decomposed via the task reduction of Corollary 5.5.2.

5.5.3   Circle-following of a differential drive robot

We illustrate task-induced reduction on mechanical systems via an example: circle-

following of a differential drive robot moving on a 2D horizontal plane as shown in Fig-

ure 5.4(A). Let $g \in \mathrm{SE}(2)$ be the position and orientation of the differential drive robot and

(σ_1, σ_2) be the wheel angles. Let ρ_w be the wheel radius and w be the half-distance between

the wheel axes, and m, J and J_w be the mass of the robot, inertia of the robot and inertia of

the wheels respectively, and $\tau = [\tau_1, \tau_2]^T$ be the motor torques at the wheels. The configu-

ration space of the robot is $Q = \mathrm{SE}(2) \times S^1 \times S^1$. Ostrowski [67] showed the Lagrangian

reduction can be performed on this robot, decoupling the configuration space into the fiber
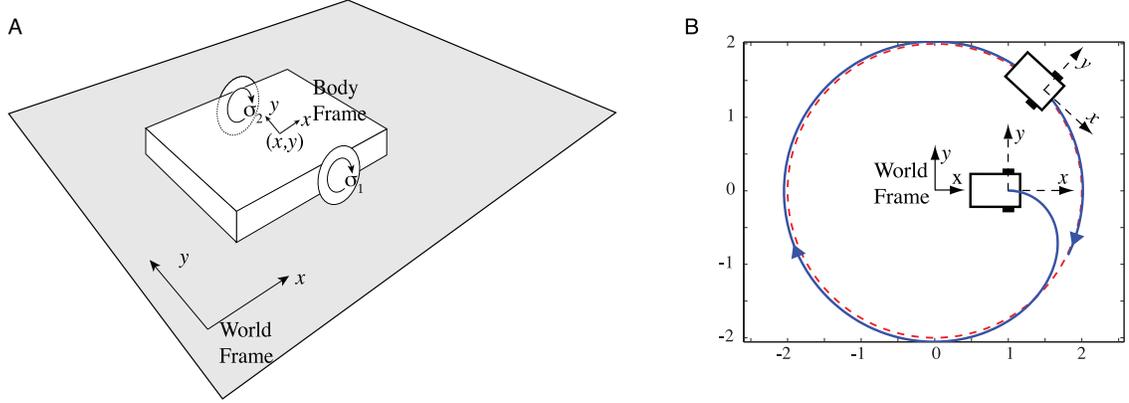
Figure 5.4: (A) Differential drive robot. World and body reference frames are shown. The body frame's x-axis is along the forward translational velocity of the robot. (B) Differential drive robot following a circle. Dashed circle is the desired circle to be followed by the differential drive robot. Initially the robot is away from the desired circle but with the application of the controller, it follows the circle. Axis scales are dimensionless for purposes of demonstration.

$SE(2)$ and the base space $S^1 \times S^1 = T^2$. This system has three nonholonomic constraints describing the no-slip condition making this a principally kinematic system. The dynamics in the reduced space are given by

$$M \begin{bmatrix} \ddot{\sigma}_1 \\ \ddot{\sigma}_2 \end{bmatrix} = \begin{bmatrix} \tau_1 \\ \tau_2 \end{bmatrix}, \tag{5.25}$$

where

$$M = \begin{bmatrix} J_w + \frac{m\rho_w^2}{4} + \frac{J\rho_w^2}{4w^2} & \frac{m\rho_w^2}{4} - \frac{J\rho_w^2}{4w^2} \\ \frac{m\rho_w^2}{4} - \frac{J\rho_w^2}{4w^2} & J_w + \frac{m\rho_w^2}{4} + \frac{J\rho_w^2}{4w^2} \end{bmatrix}.$$

The symmetry group dynamics are

$$g^{-1}\dot{g} = \begin{bmatrix} 0 & -\dot{\varsigma}_2 & \dot{\varsigma}_1 \\ \dot{\varsigma}_2 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix},$$

(5.26)

where

$$\varsigma_1 = \frac{\rho_w}{2}(\sigma_1 + \sigma_2) \text{ and } \varsigma_2 = \frac{\rho_w}{2w}(\sigma_1 - \sigma_2)$$

is a reparameterization of the shape space, $\dot{\varsigma}_1$ is the forward velocity and $\dot{\varsigma}_2$ is the angular velocity of the robot. Also, let $u_1 = \frac{\rho_w}{2}(e_1 + e_2)^T M^{-1}\tau$ and $u_2 = \frac{\rho_w}{2w}(e_1 - e_2)^T M^{-1}\tau$, where $e_1 = [1, 0]^T$ and $e_2 = [0, 1]^T$. In these new coordinates, the shape space dynamics are given by

$$\begin{bmatrix} \ddot{\varsigma}_1 \\ \ddot{\varsigma}_2 \end{bmatrix} = \begin{bmatrix} u_1 \\ u_2 \end{bmatrix}.$$

(5.27)

Consider the task of making the differential drive robot follow a circle of radius ρ whose center is at the origin of the world frame. Observe that the symmetry group dynamics resemble the equations of motion of a planar cart with the forward and angular velocities of the cart given as functions of the wheel velocities. As in Section 5.3.2, we perform a task-induced reduction with $\mathrm{SO}(2)$ as the fiber. The task-induced reduced space is given

by $r = R^{-1}p$, where $g = (R, p)$. The dynamics of the reduced space is given by

$$\dot{r} = \begin{bmatrix} 1 \\ 0 \end{bmatrix} \dot{\varsigma}_1 + \begin{bmatrix} 0 & 1 \\ -1 & 0 \end{bmatrix} r \dot{\varsigma}_2 \tag{5.28}$$

and the fiber variable, θ, satisfies

$$\dot{\theta} = \dot{\varsigma}_2. \tag{5.29}$$

We now use the control scheme developed in Section 5.3.2 along with an integrator back-stepping algorithm [47] to make the differential drive robot follow the desired circle. One can show that the following control law makes the robot follow the desired circle

$$u = \frac{\partial \phi}{\partial r} A \dot{\varsigma} - \left(\frac{\partial \varphi}{\partial r} A \right)^T - k(\dot{\varsigma} - \phi), \quad k \in \mathbb{R}^+,$$

where

$$A = \begin{bmatrix} 1 & r_2 \\ 0 & -r_1 \end{bmatrix}, \quad \phi = \begin{bmatrix} 1 \\ -(1 + r_1)/\rho \end{bmatrix},$$

and

$$\varphi = \frac{1}{2} r_1^2 + \frac{1}{2} (r_2 - \rho)^2.$$

Global convergence of this controller can be shown using the Lyapunov function, $V =$

$\varphi + \frac{1}{2} \left\| \acute{\varsigma} - \phi \right\|^2$. Figure 5.4(B) is an example simulation of the circular wall-following in a differential drive robot. The robot is initially away from the desired circle but, it asymptotically follows the desired circle.

Part II

Kernel-Based Visual Servoing

Chapter 6

Kernel-Based Visual Servoing (KBVS)

Visual servoing (VS) entails moving either a camera or the camera's visual target such that the image of the target converges to a known desired image. In this work, we consider the eye-in-hand configuration for which the camera is mounted on the robot's end effector. In this case, the visual servoing controller aims to control the end effector such that its position relative to certain features in the scene converges to a desired position. Traditionally, a visual servoing system consists of two sub-problems or processes: (i) feature tracking and (ii) feedback control. An image processing unit processes the images of the scene captured by the camera and tracks the feature points frame to frame. The feedback controller then uses these feature locations in the images as feedback signals either directly or indirectly and sends control commands to the robot. This separation, though convenient, does not allow for tuning the whole system together; the sub-problems are usually tuned independently assuming the other sub-problem is solved perfectly.

Here we propose a novel method to perform visual servoing without separating the tracking and control tasks. We build on spatial kernel-based tracking algorithms [15, 20, 29, 35] to design vision-based feedback controllers to drive the robot to a position where the image of the target scene converges to a known image. In this method a weighted sum of the image with the weights given by a smooth kernel function is the feedback signal to the controller. We use Lyapunov theory to prove the stability of the controllers and obtain formal regions of convergence. Further, we present experimental results on the implementation of this KBVS approach along four of the six degrees of motion, namely all the three translational degrees and roll about the camera's optical axis.

6.1 Previous work

6.1.1 Visual servoing

There are three main visual servoing techniques classified based on the error signals to the feedback controller:

3D visual servoing: Also called *position-based visual servoing*, in this method the Cartesian coordinates of the robot end-effector are calculated from the feature positions in the image from a known camera model and the geometry of the target [38, 54, 89]. The end-effector position error from the desired position is used as the feedback signal to the controllers. This allows the control design to be simple and not sensor modality dependent

(other than sensing errors).

2D visual servoing: Also called *image-based visual servoing*, here the features' error from their desired locations is used directly as the feedback signal for the controller [16–18, 34, 38, 61, 92]. This error along with the image Jacobian relating the motions of the features in the image to the motions of the end-effector is calculated or estimated to determine the control inputs. Even though the feature position error is completely determined in the image space, the image Jacobian calculation in general needs the 3D position and orientation of the end effector. Since the feedback signal is completely in the sensor space, this method is thought to be more robust to camera calibration errors. The main problem in this method is that sometimes the image Jacobian can lose rank and the feedback signal can lie in its null space, thus making the control ineffective. Also, for nonholonomic systems, this method cannot be directly applied.

2-$\frac{1}{2}$D visual servoing: In 2-$\frac{1}{2}$D visual servoing [55, 56, 79, 91] the feedback error signal is a combination of feature locations and 3D end effector location (up to a scale) and orientation. For this method, only the estimates of 3D information suffice and hence, it is robust to camera modeling errors while keeping the control in the task space.

Kragic and Christensen [52] use a combination of 2D and 3D visual servoing methods to perform tasks motivated by service robotics problems where the robots are given a task to fetch a particular object. They use object recognition to recognize the object and use 2D visual servoing methods for the gross motions of the robots towards the object. In the final stage, the authors use 3D visual servoing techniques for precise motions of the robots and

object manipulations. For an experimental comparison of 2D, 3D and 2-$\frac{1}{2}$D visual servoing techniques see Gans *et al.* [30]. We refer to [38] for a tutorial on visual servoing and to [51] for a newer survey on visual servoing algorithms.

While most of these visual servoing algorithms involve tracking feature locations frame to frame, there has been some recent work on vision-based control without feature tracking. Deguchi [19] defines a high dimensional space whose dimensions encode all the image pixels and the camera/target motions. Since motion of the camera/target and the resulting image are related, they lie on a lower dimensional surface within this high dimensional space. The authors use this relationship and use numerical methods to find the equation defining the surface and its Jacobian to perform visual servoing. Tahri and Chaumette [78] perform visual servoing via moments which is related to the KBVS approach presented here. Polynomial functions defined on the image pixel location can be used as spatial kernels (see 6.2.1 for kernel definition) and kernel measurement can be thought of as a collection of generalized moments of feature space around the kernel center.

6.1.2 Kernel-based tracking

In kernel-based tracking, a spatial weighted average of the image is taken to obtain a kernel measurement (see 6.2.1). Tracking is then cast as finding the optimal kernel placement in each frame of the image sequence to minimize the difference in the kernel measurements. Comaniciu *et al.* [15] bin each pixel location into one of the finite number of bins (clusters) with the binning function defined on the image features at that location. Then,

a weighted spatial average is taken to obtain a histogram vector, which is the kernel measurement. Tracking the region of interest then becomes equivalent to moving the location of the kernel center in the second frame so that a cost function based on the Bhattacharyya coefficient is minimized.

Hager *et al.* [35] extend the tracking algorithm to multiple kernels for tracking complex motions by defining kernels that depend on only degree of motion and invariant other motions. Fan *et al.* also use multiple kernels to track articulated objects like human hands. Dewan and Hager [20] develop optimal kernels for tracking purposes to improve the tracking performance in terms of lesser iterations needed for tracking convergence and lower computational cost.

6.2   Kernel-based visual servoing controllers

In this section we develop KBVS controllers for a subset of generic camera motions in $\mathrm{SE}(3)$. We first describe our overall method for a simplified simple 1D translation parallel to the image, and then extend it to 2D parallel translation (x-y), translation along the optical axis (z), and roll about the camera optical axis (θ).

In this work, we make some simplifying assumptions. First, we consider the "eye-in-hand" configuration in which the camera is mounted on the robot end effector, and the target is stationary. Further, we assume a kinematic motion model for the robot, whose control inputs are its joint velocities. We treat image pixels as continuous variables over

all of $\mathbb{R}^2$ measured in continuous time, rather than discrete variables over a finite image measured in discrete time. The visual targets are assumed to be planar and parallel to the image plane of the camera, thus having a uniform scaling of the target. In practice these assumptions are clearly violated; see experimental results Section 6.4. In all the analysis below the image or its transformations are treated as signals that are directly measured.

Definition 6.2.1 *Let $K : \mathbb{R}^2 \to \mathbb{R}$ be a real-valued piecewise continuous function defined on the location space of the image. Given a signal, $s(\boldsymbol{w}, t)$ (such as the image pixel intensities) the kernel-projected value of the signal at time t, called the* **kernel-projected measurement** *or simply* **kernel measurement** *is defined as the scalar*

$$\xi(t) = \int_{\mathcal{I}} K(\boldsymbol{w}) s(\boldsymbol{w}, t) d\boldsymbol{w}, \tag{6.1}$$

where $\boldsymbol{w} \in \mathcal{I} = \mathbb{R}^2$ is the image spatial indexing variable. The function K is known as the **kernel**.

As the camera moves relative to the target, the signal, $s(\cdot, t)$, changes thus affecting the kernel-projected value. At the goal, let the signal be $s_0 = s(\cdot, 0)$ and the kernel-projected measurement at the goal be denoted by $\xi_0 = \xi(0)$. The aim of KBVS is to drive the robot/camera to goal configuration by driving $\xi(t) \to \xi_0$.

Figure 6.1: Experimental configuration. Camera is mounted on a 6-DOF industrial robot. The camera is looking at a planar scene in front of it.

6.2.1 1D Translation

Consider a robot with a camera mounted on it as shown in Figure 6.1. Image intensity at each pixel is taken as the signal, ignoring illumination changes. Since, we are interested in translation only along one direction, we assume that the image is one-dimensional, that is $w \in \mathbb{R}$. Let the kernel projection of the image at the goal be ξ_0 at the position $x = 0$ (without loss of generality). Our goal is to determine a control input u that will drive the kernel-projected measurement to ξ_0, thus driving $x(t) \to 0$.

Assume that the camera moves parallel to the scene according to the simplified dynamics

$$\dot{x}(t) = u(t), \tag{6.2}$$

where $u(t)$ is the robot control input. For the remainder of the chapter, we assume the

signal only depends on time via the camera motion, which in the 1D case here implies, in an abuse of notation, that $s(w, t) = s(w, x(t))$. Since the scene is a unit distance away from the image plane, $s(w, x(t)) = s_0(w - x(t))$. Through a change of variables, $\bar{w} = w + x$, and recalling that $\mathcal{I} = \mathbb{R}^2$, the kernel-projected measurement ξ can be rewritten as

$$\xi = \int_{\mathcal{I}} K(w) s_0(w - x) dw = \int_{\mathcal{I}} K(\bar{w} + x) s_0(\bar{w}) d\bar{w}. \tag{6.3}$$

From (6.3), observe that even when the images or the signal are discontinuous and hence not differentiable, the kernel-projected measurement is analytically time differentiable as long as the kernel is smooth. As we show below, we exploit the differentiability of $\xi(t)$ in the design of KBVS controllers.

Consider a candidate Lyapunov function $V = \frac{1}{2}(\xi - \xi_0)^2$. Applying the chain rule, we have

$$\begin{aligned}
\dot{V} &= (\xi - \xi_0) \frac{\partial \xi}{\partial x} \dot{x} \\
&= (\xi - \xi_0) \left[\int_{\mathcal{I}} \frac{\partial K(\bar{w} + x)}{\partial x} s_0(\bar{w}) d\bar{w} \right] \dot{x} \\
&= (\xi - \xi_0) \left[\int_{\mathcal{I}} K'(\bar{w} + x) s_0(\bar{w}) d\bar{w} \right] \dot{x} \\
&= (\xi - \xi_0) \left[\int_{\mathcal{I}} K'(w) s_0(w - x) dw \right] u
\end{aligned}$$

where $K'(w) = \frac{\partial K(w)}{\partial w}$. Note that in the last step, we revert the coordinates back to w. Now,

choose the input, u, as

$$u = -(\xi - \xi_0) \int_{\mathcal{I}} K'(w)s(w, x)dw \qquad (6.4)$$

which only requires the current signal projection, ξ, the signal projection at the goal, ξ_0, the kernel function derivative, K', and the current signal, $s(w, t)$, which depends on t only through $x(t)$ (see above). With this choice, $\dot{V}$ becomes

$$\dot{V} = -(\xi - \xi_0)^2 \left(\int_{\mathcal{I}} K'(w)s(w, x)dw \right)^2.$$

Assuming the candidate Lyapunov function, V, is positive definite in the configuration variable, then $\dot{V}$ is negative semi-definite. The assumption that $V > 0$ admittedly depends on the signal and kernel properties, although it appears from our experiments (Section 6.4) to be a valid (locally) generic assumption and, in any case, can be numerically tested and optimized [20]. This choice of input guarantees stability (in the Lyapunov sense) of the controller with mild assumptions on the image and kernel.

For practical applications, it is crucial to obtain at least local *asymptotic* stability. If the kernel-image pair is such that in a neighborhood around the goal, $\int_{\mathcal{I}} \frac{\partial K(w)}{\partial w} s_0(w - x)dw \neq 0$ (again, which appears to be true generically), local asymptotic stability is guaranteed. $\frac{\partial K(w)}{\partial w}$ becoming zero is analogous to the image error lying in the null space of the Jacobian in tracking literature [13].

For good practical performance of the controller, the Lyapunov function in the configuration space of the robot should be quadratic near the goal, the Hessian at the goal

should be positive (for the 2D KBVS algorithms developed below, the Hessian have positive eigenvalues with condition number as close to one as possible). This provides us with an objective function that likely can be optimized for larger regions of attraction and better performance.

6.2.2   2D Translation

Let the motion of the camera relative to the target be 2D translation parallel to the image plane of the camera. Let the configuration of the robot be denoted by $q = [x, y]^T \in \mathbb{R}^2$. Assume that the camera moves parallel to the optical axis according to

$$\dot{q} = u, \tag{6.5}$$

where $u \in \mathbb{R}^2$ is the robot control input. Since the target scene is a unit distance away from the image plane $s(w, q(t)) = s_0(w - q(t))$. Through a change of variables, $\bar{w} = w + q$, and recalling that $\mathcal{I} = \mathbb{R}^2$, the kernel-projected measurement ξ can be rewritten as

$$\xi = \int_{\mathcal{I}} K(w) s_0(w - q) dw = \int_{\mathcal{I}} K(\bar{w} + q) s_0(\bar{w}) d\bar{w}. \tag{6.6}$$

Consider a Lyapunov function candidate $V = \frac{1}{2}(\xi - \xi_0)^2$. Applying the chain rule, we have

$$\dot{V} = (\xi - \xi_0)\frac{\partial \xi}{\partial \boldsymbol{q}}\dot{\boldsymbol{q}}$$
$$= (\xi - \xi_0)\left[\int_{\mathcal{I}} K'(\bar{\boldsymbol{w}} + \boldsymbol{q})s_0(\bar{\boldsymbol{w}})d\bar{\boldsymbol{w}}\right]\dot{\boldsymbol{q}}$$
$$= (\xi - \xi_0)\left[\int_{\mathcal{I}} K'(\boldsymbol{w})s_0(\boldsymbol{w} - \boldsymbol{q})d\boldsymbol{w}\right]\boldsymbol{u},$$

where $K'(\boldsymbol{w}) = \frac{\partial K(\boldsymbol{w})}{\partial \boldsymbol{w}} \in \mathbb{R}^{1 \times 2}$. Note that in the last step, we revert the coordinates back to $\boldsymbol{w}$. Now, choose the input, u, as

$$\boldsymbol{u} = -(\xi - \xi_0)\int_{\mathcal{I}} \nabla K(\boldsymbol{w})s(\boldsymbol{w}, \boldsymbol{q})d\boldsymbol{w}, \tag{6.7}$$

where $\nabla K = (\frac{\partial K}{\partial \boldsymbol{w}})^T \in \mathbb{R}^2$. With this choice $\dot{V}$ becomes

$$\dot{V} = -(\xi - \xi_0)^2 \left\|\int_{\mathcal{I}} \nabla K(\boldsymbol{w})s(\boldsymbol{w}, \boldsymbol{q})d\boldsymbol{w}\right\|^2.$$

Similar to the ideas presented in [20], an alternate way of doing 2D translation is to decouple it into two 1D translations using two independent x and y directional kernels. Each kernel is invariant to the motion in the other direction thereby providing independent controllers. For example, a kernel oriented in the x direction can be formed by stacking a Gaussian kernel along every pixel in the y direction. As discussed in [20], we also found that using the two independent kernels provides better results than using a single kernel. The experiments presented for the 2D translation case in section 6.4 use the two kernel

approach.

6.2.3 Translation along optical axis

Cideciyan [14] uses a spatial Fourier transform (FT) of images for tracking and registration to decouple translation and scaling. We seek to capitalize on this invariance of the magnitude of the FT to translation to develop controllers for depth and rotation that can integrated with the previously developed 2D controllers in the x-y plane. As a first step, we consider motions in depth only.

Here, we consider motions of a camera along its optical axis. Even though this corresponds to a translation as in the previous two cases, there is a fundamental difference between the two: 2D x-y translations simply translate the image, while motions in depth scale the image. Thus, we seek an appropriately transformed signal and control strategy. Specifically, we use the magnitude of the FT of the image as the signal.

Let I_0 denote the image at the goal, and F_0 the magnitude of its spatial FT. We assume that the goal corresponds to unity depth (without any loss of generality). Let the inertial world reference frame be such that the z-axis is parallel to the camera's optical axis. In this frame, the camera is moving along the z-axis according to

$$\dot{z}(t) = u, \tag{6.8}$$

with goal $z_0 = 1$. At any generic position of the camera, the image I is a scaled version of

I_0, *i.e.*

$$I(\boldsymbol{w}, z) = I_0(\boldsymbol{w}/z).$$

One can show that the magnitudes of the spatial FT of these images (F and F_0 respectively) are related by

$$F(\boldsymbol{v}, z) = z^2 F_0(z\boldsymbol{v}), \ \boldsymbol{v} \in \mathbb{R}^2.$$

We define the kernel-projected measurement as

$$\xi = \int_{\mathcal{I}} K(\boldsymbol{v})F(\boldsymbol{v}, z)d\boldsymbol{v} = \int_{\mathcal{I}} K(\bar{\boldsymbol{v}}/z)F_0(\bar{\boldsymbol{v}})d\bar{\boldsymbol{v}}, \tag{6.9}$$

where $\bar{\boldsymbol{v}} = z\boldsymbol{v}$. At the goal we have $\xi_0 = \int_{\mathcal{I}} K(\boldsymbol{v})F_0(\boldsymbol{v})d\boldsymbol{v}$. Our aim is to drive the robot to $z = 1$ by driving $\xi(t) \to \xi_0$. Consider a Lyapunov function candidate: $V = \frac{1}{2}(\xi - \xi_0)^2$ and choosing the input as

$$u = (\xi - \xi_0) \int_{\mathcal{I}} K'(\boldsymbol{v})\boldsymbol{v}F(\boldsymbol{v}, z)d\boldsymbol{v}, \tag{6.10}$$

then $\dot{V} = -\frac{1}{z}(\xi - \xi_0)^2 \left\| \int_{\mathcal{I}} K'(\boldsymbol{v})\boldsymbol{v}F(\boldsymbol{v}, z)d\boldsymbol{v} \right\|^2$. If $z > 0$, $\dot{V}$ is negative semi-definite, which is a realistic assumption for objects seen by the camera.

6.2.4   Rotation about the optical axis

In this section we develop KBVS for rotation of the camera relative to the target about its optical axis. Let the robot dynamics be

$$\dot{\theta} = u, \tag{6.11}$$

where u is the control input. As in the case of scaling, we use the magnitude of the spatial FT of the image as the signal. Let I_0 and F_0 denote the image and signal at the goal, where $\theta = 0$ (without any loss of generality). At any generic roll position of the camera, the image I is a rotated version of I_0:

$$I(\boldsymbol{w}, \theta) = I_0(R_\theta \boldsymbol{w}), \text{ where } R_\theta = \begin{bmatrix} \cos\theta & \sin\theta \\ -\sin\theta & \cos\theta \end{bmatrix} \in \mathrm{SO}(2).$$

The magnitudes of the spatial FT of these images are related by

$$F(\boldsymbol{v}, \theta) = F_0(R_\theta \boldsymbol{v}), \ \boldsymbol{v} \in \mathbb{R}^2. \tag{6.12}$$

We define the kernel-projected measurement as

$$\xi = \int_{\mathcal{I}} K(\boldsymbol{v}) F(\boldsymbol{v}, \theta) d\boldsymbol{v} = \int_{\mathcal{I}} K(R_\theta^T \bar{\boldsymbol{v}}) F_0(\bar{\boldsymbol{v}}) d\bar{\boldsymbol{v}}, \tag{6.13}$$

where $\bar{\boldsymbol{v}} = R_\theta \boldsymbol{v}$. At the goal, the kernel-projected measurement is $\xi_0 = \int_{\mathcal{I}} K(\boldsymbol{v}) F_0(\boldsymbol{v}) d\boldsymbol{v}$. As before, our aim is to drive the robot to $\theta = 0$ by driving $\xi(t) \to \xi_0$. Consider a Lyapunov function candidate: $V = \frac{1}{2}(\xi - \xi_0)^2$. Choose the control input as

$$u = -(\xi - \xi_0) \int_{\mathcal{I}} K'(\boldsymbol{v}) J\boldsymbol{v} F(\boldsymbol{v}, \theta) d\boldsymbol{v}, \qquad (6.14)$$

where $J = R_{-\frac{\pi}{2}}$. With this choice of u, $\dot{V} = -(\xi - \xi_0)^2 \left\| \int_{\mathcal{I}} K'(\boldsymbol{v}) J\boldsymbol{v} F(\boldsymbol{v}, \theta) d\boldsymbol{v} \right\|^2$, which is negative semi-definite.

6.2.5   Extensions to $SE(2)$ + depth motions

In the above controllers, we used the image as the signal for the x-y translations and the magnitude of the FT of the image as the signal for depth and roll. As discussed before, the FT of the image removes any translation effects while controlling depth and roll. For 3D translational control, one can execute the depth controller first, since it is invariant to translation, and then run the 2D x-y controller. Similarly, to control all of $SE(2)$ (identified with x, y, and roll), one can control for roll first, since it is again invariant to translation, and then control in the 2D plane. Furthermore, all four degrees of freedom (x, y, z, and roll) can be controlled in a similar manner. In the experiments discussed below, we control all the four degrees of freedom ($SE(2)$ and depth) simultaneously.

6.3 Assumptions and their practicality

In the design of the above KBVS controllers, we made several simplifying assumptions which might not hold in real experiments. We assume that the spatial indexing of the image, w or $\boldsymbol{w}$ is a continuous variable while in practice it is discrete (pixels locations are discrete). For modern hight resolution cameras (480×640 or higher) this assumption seems mild and is handled in practice by treating the image intensities as a piecewise constant function with intensities being constant along a each pixel length. To improve computation speed, this can also handled by evaluating the kernel function and its derivatives at discrete pixel locations and storing them ahead of the experiment. The kernel measurements can now be calculated as the dot product of two matrices—image intensities and kernel matrix. Similarly the control inputs can be calculated as dot products of relevant matrices.

The other major assumption we make is that we assume that the image captured by the camera is infinite. This assumption in the 1D and 2D cases is handled by using finite support kernels (e.g. a truncated Gaussian with small standard deviation). In the depth and the roll cases, we handle this by using scenes that have black regions in their boundaries, thus making the fast FT exact. In future, this class of images used for depth and roll cases need to be expanded. One way to expand the applicability is to find an important feature region in the image/scene and make the rest of the image be made to have an intensity of zero. This approach though will need some kind of coarse feature region tracking frame to frame. Binning functions similar to the ones used in Kernel-based tracking methods [15, 35] can be used for these purposes.

We also assume that the target scene of the camera is planar. This assumption made the controller design simpler but we have seen in our laboratory experiments that this assumption can be violated to a certain extent. It has been observed that the controllers work when the scene largely consists of of planar patches even though the scene is not completely planar. The formal bounds on how much this assumption can be violated is left as future research.

6.4   Experiments

In this section we present the experimental results of the KBVS controllers developed in this chapter for an eye-in-hand configuration. Experiments are performed on an American Robot Merlin 6200 series robot arm. This robot has six degrees of freedom located at the waist, shoulder, elbow and 3-DOF wrist. The camera attached to the robot's end effector is a grayscale Basler 602fc firewire camera. The resolution of the camera is 480×640 pixels. The robot is controlled via a dedicated workstation running Linux with real-time extensions. In order to facilitate algorithm development and implementation, the software infrastructure allows for direct control of the robot and the capture of images directly from the GNU Octave mathematical software [26].

Below are five sets of experiments that show the convergence of the KBVS controllers. In the first set, the robot moves in the 2D plane parallel the image plane of the camera. In the second the robot moves along the optical axis of the camera and in the third set the

robot rotates about the camera's optical axis. Then we present experiments that combine these degrees of freedom. In the fourth set, the robot is allowed in move in three degrees of freedom: two translations parallel to the camera's image plane and one more translational motion along the camera's optical axis. In the fifth set of experiments, the robot is allowed to move in all the three translational degrees of freedom and also in the roll DOF about the camera's optical axis. We have conducted 10 sets of experiments with random initial conditions in each of these sets.

6.4.1 2D Translational motion
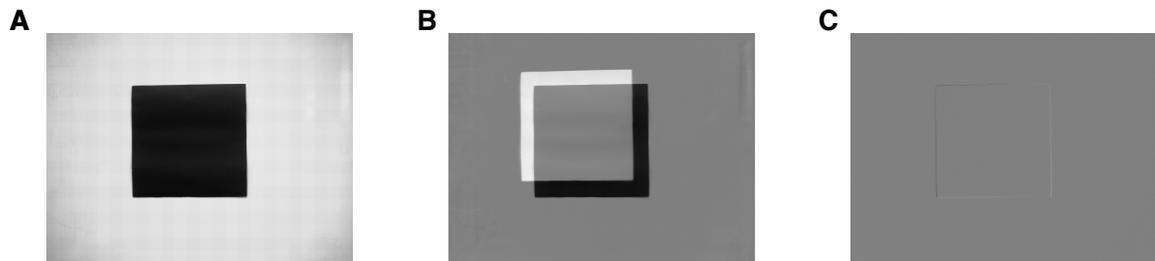
Figure 6.2: (A) 2D goal image. (B) Difference between the goal and the initial displacement images. (C) Difference between the goal and the final images.

In these experiments the robot is allowed to translate parallel to the image plane of a camera mounted on it. Figure 6.2(A) shows the image of the scene at the goal location. From the goal location, the camera is moved away in the two translational directions (ap-
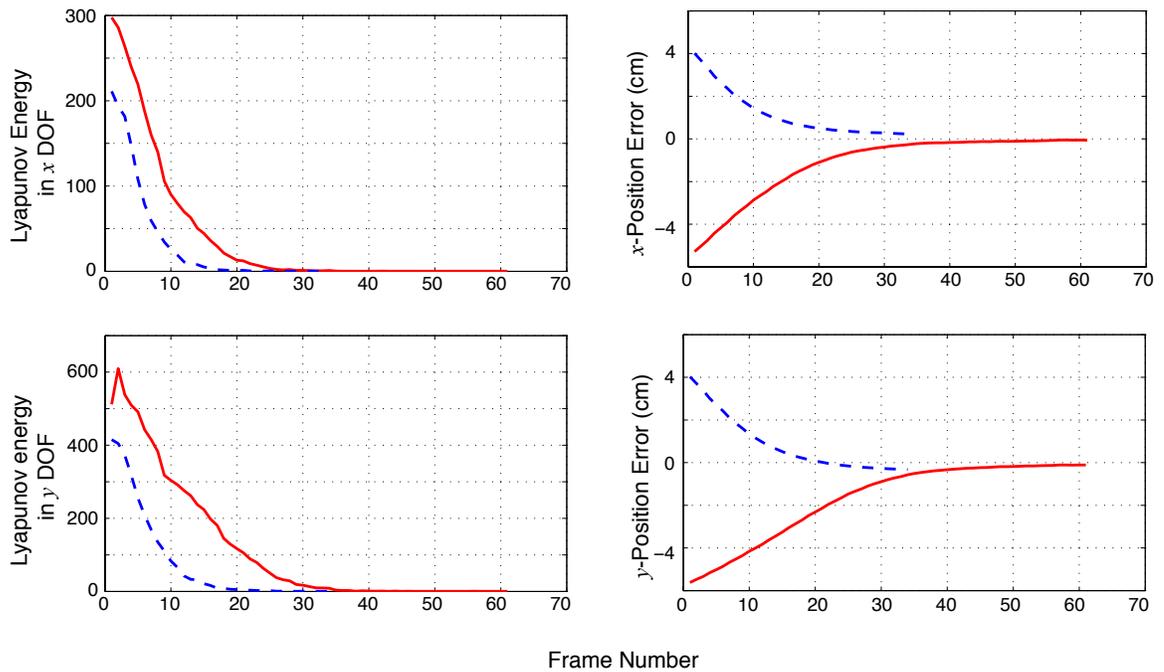
Figure 6.3: Convergence in the 2D translational DOFs. Two experimental trials are plotted; solid line represents one trial and dashed line represents the other. The first row shows the convergence in x-position and the second in y-position. *Left column:* Convergence in Lyapunov functions defined for each of the DOFs to zero. *Right column:* Convergence of position errors relative to the goal location to zero.

proximately) parallel to the image plane. The kernels used for the experiments are:

$$K_x(\boldsymbol{w}) = \frac{1}{sqrt(2\pi)\sigma_x} e^{-\frac{(\boldsymbol{w}_1 - \mu_x).^2}{2\sigma_x^2}},$$

$$K_y(\boldsymbol{w}) = \frac{1}{sqrt(2\pi)\sigma_y} e^{-\frac{(\boldsymbol{w}_2 - \mu_y).^2}{2\sigma_y^2}},$$

where $\mu_x = \mu_y = -100$ and $\sigma_x = \sigma_y = 70$ and K_x and K_y are kernels in the x and y directions. $\boldsymbol{w}$ represents the pixel indexing of the image.

We conducted 10 trials with random initial positions. In these experiments, the mean initial position error was 6.98 cm with a standard deviation of 0.72cm. The trial was stopped when the difference of between the current kernel measurement and that at the goal location was below a certain threshold; in these experiments this was chosen as 0.1 which is $0.02 - 0.03\%$ of the goal kernel measurement in each direction. In all the trials, the position converged to the goal position with a mean position error of 0.31cm and standard deviation of 0.15cm. In Figure 6.9 we show two of the 10 trials; these trials represent the maximum and minimum of the inital displacement of the robot from its goal location. The left column is show the convergence of Lyapunov functions defined on each of the DOFs and the right column shows convergence in the error of respective DOFs relative to the goal location.
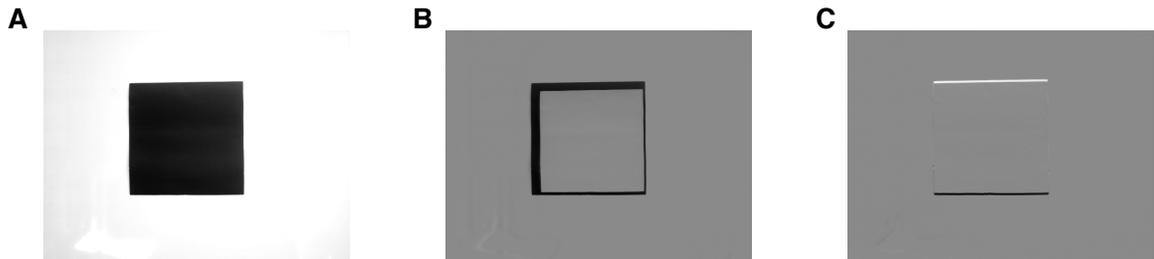
Figure 6.4: (A) Depth goal image. (B) Difference between the goal and the initial displacement images. (C) Difference between the goal and the final images.
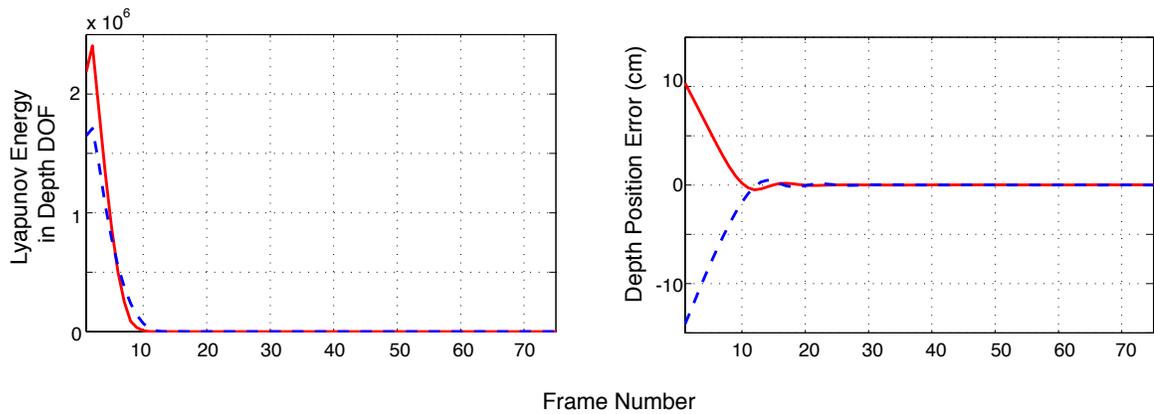
Figure 6.5: Convergence in depth DOF. Two experimental trials are plotted; solid line represents one trial and dashed line represents the other. *Left:* Convergence in Lyapunov functions to zero. *Right:* Convergence of depth error relative to the goal location to zero.

6.4.2   Depth motion

In these experiments the robot is allowed to translate along the camera's optical axis. Recall that the signal for the depth DOF controller is the magnitude of the FT of the image. As discussed in Section 6.3, since the field of view is finite, the depth controller needs the coarse image segmentation to make the intensity of the image background zero. In our experiments, we perform a simple color segmentation and assign a value of zero to the background and one to the rest of the image. The kernel function used is:

$$K_z(\boldsymbol{v}) = e^{-\frac{1}{8}||\boldsymbol{v}||^2},$$

where $\boldsymbol{v}$ represents the spatial indexing for the FT.

We conducted 10 trials with random initial positions. In these experiments, the mean initial position error was 8.02cm with a standard deviation of 4.52cm. The trial was stopped when the difference of between the current kernel measurement and that at the goal location was below a threshold; in these experiments this was chosen as 5 which is 0.02% of the goal kernel measurement. In all the trials, the position converged to the goal position with a mean position error of 0.06cm and standard deviation of 0.065cm. In Figure 6.7 we show two of the 10 trials; these trials represent the two extremas of the inital depth displacement of the robot from its goal location. The left column shows the convergence in Lyapunov function and the right column shows convergence in depth position error.

6.4.3 Roll motion

Figure 6.6: (A) Roll goal image. (B) Difference between the goal and the initial displacement images. (C) Difference between the goal and the final images.
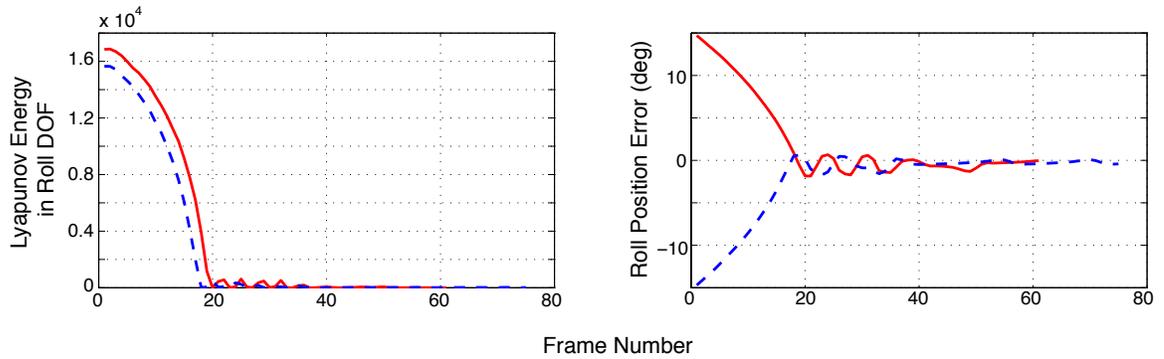
Figure 6.7: Convergence in roll DOF. Two experimental trials are plotted; solid line represents one trial and dashed line represents the other. *Left:* Convergence in Lyapunov functions to zero. *Right:* Convergence of roll error relative to the goal location to zero.

In these experiments the robot is allowed to rotate about the camera's optical axis. As in the depth case, the signal to the controller is the FT of the background segmented case. The kernel function used is:

$$K_\theta(\boldsymbol{v}) = e^{-\frac{1}{8}\boldsymbol{v}_1^2} + e^{-\frac{1}{8}\boldsymbol{v}_2^2},$$

where v represents the spatial indexing for the FT.

We conducted 10 trials with random initial positions. In these experiments, the mean initial position error was $12.67°$ with a standard deviation of $1.51°$. The trial was stopped when the difference of between the current kernel measurement and that at the goal location was below 5 which is 0.02% of the goal kernel measurement. In all the trials, the position converged to the goal position with a mean position error of $0.52°$ and standard deviation of $0.53°$. In Figure 6.7 we show two of the 10 trials; these trials represent the two extremas of the inital roll displacement of the robot from its goal location. The left column shows the convergence in Lyapunov function and the right column shows convergence in roll error relative to the goal location.

6.4.4 3D Translational motion
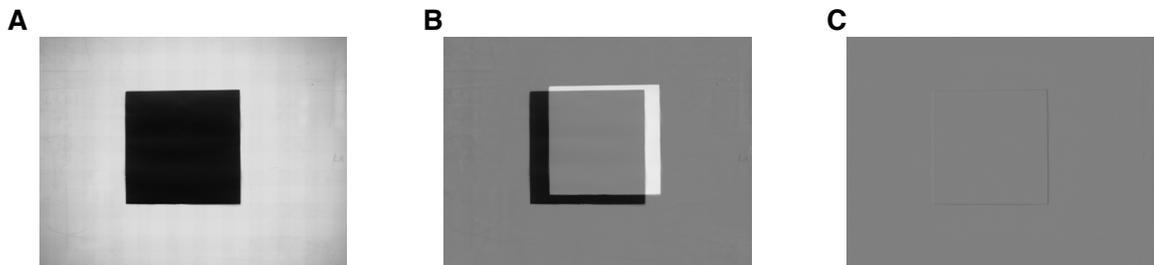
Figure 6.8: (A) Three translational DOFs goal image. (B) Difference between the goal and the initial displacement images. (C) Difference between the goal and the final images.

In these experiments the robot is allowed in move in all the three translational degrees of motion. Recall that the signal for the depth DOF controller is the magnitude of the FT of the image and that for the other two translational DOFs is the image itself. Since the magnitude
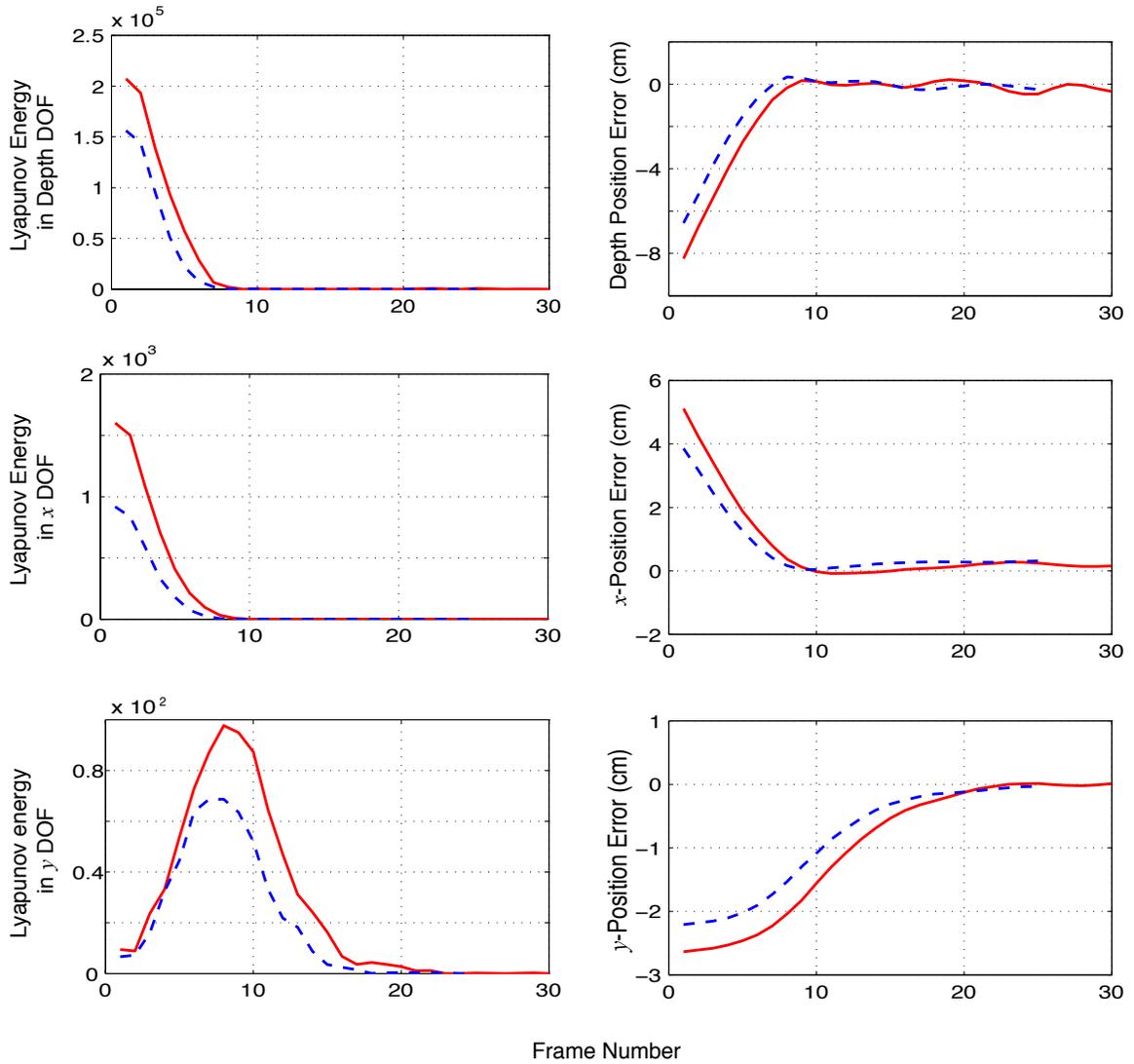
Figure 6.9: Convergence in three translational DOFs. Two experimental trials are plotted; solid line represents one trial and dashed line represents the other. *Left:* Convergence in Lyapunov functions defined for each of the DOFs to zero. *Right:* Convergence of position errors relative to the goal location to zero.

of the FT does not depend on the translations motions parallel to the image plane, the vector field given by (6.10) is independent of the other two translational directions. Hence, as discussed in Section 6.2.5, we can first control the depth independently and once this controller has converged, we can turn on the 2D controller. But in laboratory settings, one might have to do more than one iteration to obtain the full convergence in all the three DOFs as the target scene is not exactly parallel to the image plane of the camera (in our experimental trials we have observed that two iterations are usually enough). To steer clear of deciding when to switch from one controller to the other, we added the perpendicular vector fields given by (6.10) and (6.7). We have made sure that depth convergence is faster by choosing higher gains for this controller compared to the other two controllers.

Figure 6.8(A) shows the image of the scene at the goal location. From the goal location, the camera is moved away in the three translational directions. The controller acts to bring the robot/camera to the goal. The kernels used for the experiments are:

$$K_x(\boldsymbol{w}) = \frac{1}{sqrt(2\pi)\sigma_x}e^{-\frac{(\boldsymbol{w}_1-\mu_x).2}{2\sigma_x^2}},$$

$$K_y(\boldsymbol{w}) = \frac{1}{sqrt(2\pi)\sigma_y}e^{-\frac{(\boldsymbol{w}_2-\mu_y).2}{2\sigma_y^2}},$$

$$K_z(\boldsymbol{v}) = e^{-\frac{1}{8}||\boldsymbol{v}||^2},$$

where $\mu_x = \mu_y = -100$ and $\sigma_x = \sigma_y = 70$ and K_x, K_y and K_z are kernels in the x, y and z directions. $\boldsymbol{w}$ represents the pixel indexing of the image and $\boldsymbol{v}$ is the spatial indexing for the FT.

We conducted 10 trials and in all of them the robot converged to the goal with a mean position error of 0.46cm and standard deviation of 0.2cm. The initial position difference from the goal location in these 10 trials has a mean of 8.74cm with a standard deviation of 0.75cm. In Figure 6.9 we show two of the 10 trials; these trials represent the maximum and minimum of the inital displacement of the robot from its goal location. The left column shows the convergence in Lyapunov functions defined on each of the DOFs and the right column shows convergence in the errors in the respective DOFs. From this figure, we can see that the Lyapunov function defined in the y direction goes up and then comes down. This happens because the controller in the y direction is dependent on the depth DOF and once the depth has converged to its value at the goal, the Lyapunov function defined on the y coordinate converges to zero and the goal location is reached in all the three DOFs. In general, the depth DOF converges faster than the other two.

6.4.5   SE(2) and depth motion
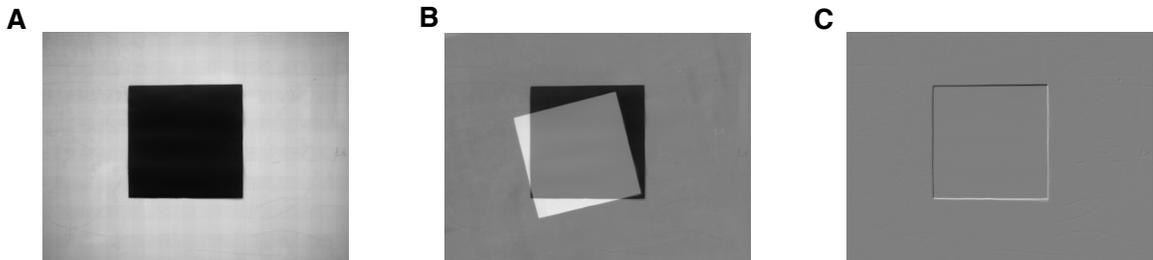
Figure 6.10: (A) $SE(2)$ and depth goal image. (B) Difference between the goal and the initial displacement images. (C) Difference between the goal and the final images.

Here we discuss the experiments that are conducted when the robot moves in all the
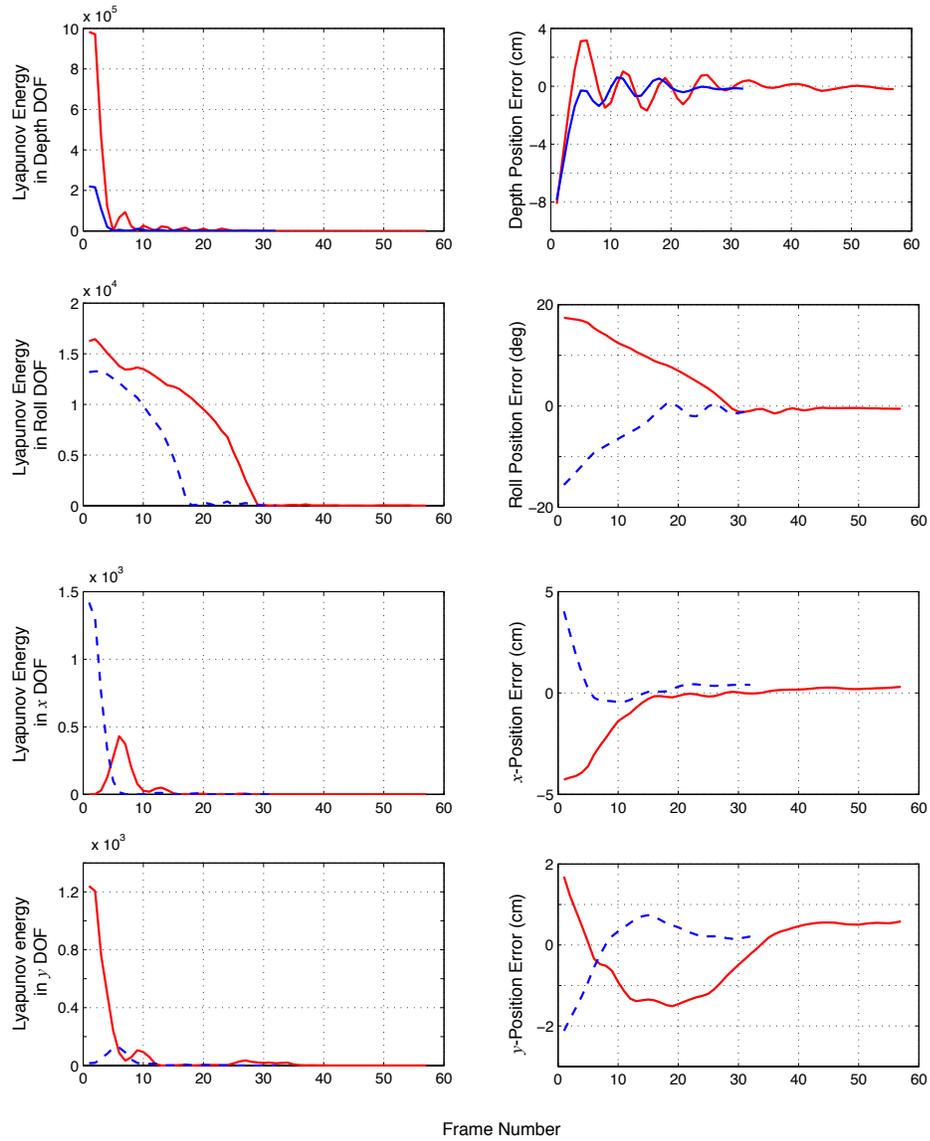
Figure 6.11: Convergence in $\mathrm{SE}(2)$ and depth DOFs. Two experimental trials are plotted; solid line represents one trial and dashed line represents the other. *Left:* Convergence in Lyapunov functions defined for each of the DOFs to zero. *Right:* Convergence of position errors relative to the goal location to zero.

three translational DOFs and roll DOF about the camera's optical axis. As in the previous case, we have conducted the experiments by adding the vector fields in the four DOFs given by (6.7), (6.10) and (6.14). Figure 6.10(A) shows the image of the scene at the goal location. From the goal location, the camera is moved away in the three translational and roll directions. The controller acts to bring the robot/camera to the goal. The kernels used for the experiments are:

$$K_x(\boldsymbol{w}) = \frac{1}{sqrt(2\pi)\sigma_x} e^{-\frac{(\boldsymbol{w}_1 - \mu_x).^2}{2\sigma_x^2}},$$

$$K_y(\boldsymbol{w}) = \frac{1}{sqrt(2\pi)\sigma_y} e^{-\frac{(\boldsymbol{w}_2 - \mu_y).^2}{2\sigma_y^2}},$$

$$K_z(\boldsymbol{v}) = e^{-\frac{1}{8}||\boldsymbol{v}||^2},$$

$$K_\theta(\boldsymbol{v}) = e^{-\frac{1}{8}\boldsymbol{v}_1^2} + e^{-\frac{1}{8}\boldsymbol{v}_2^2},$$

where $\mu_x = \mu_y = -100$ and $\sigma_x = \sigma_y = 70$ and K_x, K_y, K_z and K_θ are kernels in the x, y, z and roll (θ) directions. $\boldsymbol{w}$ represents the pixel indexing of the image and $\boldsymbol{v}$ is the spatial indexing for the FT. Note that with the chosen depth kernel, the depth kernel measurement and control is invariant to not just 2D translations but also to roll. Roll kernel measurement and control is invariant to just 2D translations. Hence, the depth DOF must converge before the roll DOF can converge to the goal.

We conducted 10 trials and in all of the trials the robot converged to the goal with a mean position error of 0.44cm with standard deviation of 0.21cm and mean roll error $0.56°$ with standard deviation of $0.39°$ in roll DOF. The initial position difference from the goal

location in these 10 trials has a mean of 8.94cm with a standard deviation of 0.48cm in the translational DOFs and a mean of $12.01°$ with standard deviation of $3.15°$ in the roll DOF. In Figure 6.11 we show two of the 10 trials; these trials represent the maximum and minimum of the inital roll displacement of the robot from its goal location. As in the last experiment set, the depth DOF converges first and the rest of the DOFs converge later.

Chapter 7

Conclusions

The focus of this thesis is on the applications of vision as a sensor for feedback control of systems on Lie groups. In the first part, we present a feedback controller that stabilizes a flexible bevel-tip needle to a desired 2D plane. We note that this task induces a symmetry on the system, resulting in a reduced order three-state system. This task-induced reduction simplifies the controller and observer design. In practice, only the 3D position (but not orientation) of the needle tip can be measured using an imaging system. First, we show that the full 6-DOF pose is observable given only these position measurements and then design dead-beat and asymptotic observers that estimate the full pose of the needle from just the position measurements. We then extend the notion of task-induced symmetry and reduction to other kinematic systems defined on Lie groups and to other mechanical systems. In the second part, we present kernel-based visual servoing as a novel image-based algorithm that does not need feature tracking to servo a robot with a camera mounted on it to a goal

128

location.

Image-guided control of needle steering

Flexible bevel-tip needles show great promise in extending the applications of percutaneous needle insertions. These needles can be used to reach places in the tissue that cannot be accessed by straight-line trajectories while avoiding sensitive organs. Also, these needles can be controlled for steering after they are inserted into the tissue, thus facilitating the use of run-time feedback control or replanning. We have designed feedback controllers to drive the needle to a subspace of its configuration space using task-induced reduction. These controllers are designed to work in conjunction with subspace planners to reach a target location in the tissue.

We have also implemented a nonlinear observer–controller pair to drive the needle to a desired 2D plane in tissue phantom made from plastisol, using stereo-image guidance. This controller has been successfully integrated with the 2D planners of Alterovitz *et al.* [6] and the torsional compensator of Reed [70] to drive the needle to a desired target location while avoiding polygon-shaped obstacles.

An important next step is to evaluate the performance of this controller by conducting tests on a variety of tissues (phantom, *ex vivo*, and live) using ultrasound or fluoroscopy imaging systems. Due to tissue inhomogeneity, implementing control on real tissue might benefit from an adaptive version of our controller that would "learn" the model parameters while stabilizing the needle to a 2D plane. Alternately, if bounds on the parameter κ are

known, then the robust controller presented in Section 3.5 can be used.

In the present work, we have studied stability properties of the planar controller and empirically studied its convergence properties (in terms of the needle insertion length). When the planar controller is being used along with the path planners, the needle could be rotated by $180°$ every time the planner is being called. Due to the torsional effects present between the tissue and the needle, this could adversely affect the performance of the planar controller and indirectly also the performance of the motion planner. Hence, for the purpose of pure motion planning, the high level planning loop should be fast (neglecting the time needed for re-planning) but for the better performance of the planar controller the planning loop must be run as slow as possible. Also, one can imagine that the faster the low-level estimation and control loop, the better it is. But, since the system is parametrized with respect to insertion distance of the needle into the tissue, this can result a really long time for a given amount of needle insertion distance. An understanding of these trade-offs to would be very helpful in designing better integrated planning and controller system.

If a path has curvature less than κ, then such a path may be followed by using path-following controllers. Such controllers would not ever follow these paths exactly but could follow it very closely. Designing these path-following controllers is an interesting open question.

Observers on the group of rigid-body transformations

In this thesis, we develop dead-beat and asymptotic observers on the group of rigid-body transformations. While this work specifically deals with needle steering, similar computations may apply to other systems on the group of rigid motions. The dead-beat observer on $SO(3)$ that we develop in this thesis is essentially a least-squares solution over a finite number of samples. There has been some research done on (asymptotic) adaptive identification on $SO(3)$, such that given a stream of measurements, the estimator converges to the correct constant parameter which lies on the $SO(3)$ manifold [48]. But, controllers, such as in this thesis, typically require the estimation of a rigid transformation that changes with time.

We present a local asymptotic observer that estimates the full 6-DOF pose of the needle from just the position measurements. This observer uses task-induced reduction and state immersion but is in local coordinates. An asymptotic observer on rigid-body transformations that does not require local coordinates would be ideal. We know of no work to date that addresses this problem of developing model-based asymptotic observers on rigid groups in a statistically sound way; recent work by Wang *et al.* [84] error propagation on $SE(3)$ and $SO(3)$ and Park *et al.* [69] on probability density estimation on $SE(3)$ and $SO(3)$ could be helpful in developing such observers.

Task-induced reduction

We develop a general framework to perform task-specific reduction in kinematic systems on Lie groups. We apply this method to two systems: a planar cart (as an illustrative example), and flexible bevel-tip needle steering with configuration spaces $\mathrm{SE}(2)$ and $\mathrm{SE}(3)$, respectively. In both the examples, task-induced reduction enables us to design essentially global feedback controllers on the reduced systems. We envision that the methodology presented in this thesis will form the basis for a new planning and control framework for needle steering.

We extend this framework to mechanical systems whose Lagrangian and constraints are left-invariant to a Lie group. From previous work on "intrinsic" Lagrangian reduction, it is known that dynamics on the shape space can be obtained from the reduced Lagrangian, while a reconstruction equation generates motions in the symmetry group. This reconstruction equation describes how motions in the shape space produce Lie group motions. As we show in the differential drive robot example, certain control tasks enable us to perform a second "extrinsic" (task-specific) reduction, thus further decomposing the symmetry group into a base and fiber. Such an analysis is also amenable to other mechanical systems like skateboards, bicycles, and snake robots. As in the needle steering case, task-induced symmetry will likely facilitate planning and control for other mechanical systems.

In all the examples in Sections 5.3 and 5.4, task-induced reduction simplifies control design. Further, in the kinematic systems, the task-functional served as a Lyapunov function and the control input was chosen accordingly. In the differential drive robot example,

the task-functional was again used as a part of the Lyapunov function. In all these cases, the task-functional was used as a guiding mechanism for controller design. This suggests that there may be a general control theory for systems exhibiting task-induced symmetry, at least for some class of mechanical systems. Specifically, we expect this to hold for principally kinematic systems whose base variables are completely controllable. For these systems, we speculate that integrator back-stepping is possible, thus reducing the mechanical control problem to that of kinematic control. This direction of future research, we think, will be very useful in controlling complicated higher-dimensional systems.

Kernel-based visual servoing

We present kernel-based visual servoing algorithms to control an eye-in-hand robot to a goal location without any feature tracking. This paradigm combines the traditional tracking and control subproblems in visual servoing by eliminating the need for tracking feature location in a image sequence frame-to-frame. In this approach, a weighted sum of the image or its transformations is used as signal to the feedback controller. We use Lyapunov theory to show the stability of these controllers. This approach provides formal guarantees on the convergence/stability of visual servoing algorithms. We develop KBVS algorithms for a subclass of rigid-body motions—the three translational DOFs and roll about the camera's optical axis—and implement them on a 6-DOF industrial robot. Controllers for the pitch and the yaw motions are not developed and we suspect that using stereo cameras instead of a single camera may simplify this.

CHAPTER 7. CONCLUSIONS

There is a strong correlation between the performance of the KBVS controllers and the choice of a kernel function for a given target scene. In the experiments, we use Gaussian kernel functions for all the target scenes. These functions and their mean and variances are selected in an ad-hoc way. We have observed that having a large variance typically increases the region of convergence but the convergence is slower and if variance is low, then the region of convergence is small but the local performance of the controller increases. The advantages of both kinds of kernels can be harnessed by adaptively changing the variance of the kernel function. This would only require a few additional computations of kernel measurement at the goal location. Also, different classes of kernel functions (apart from Gaussian functions) may be used to improve the performance of the KBVS controllers. Also, we have not verified the effects/advantages of using multiple kernels in the controller performance. Future work on kernel optimization for a given target scene would be highly desirable.

Bibliography

[1] N. Abolhassani, R. Patel, and M. Moallem. Needle insertion into soft tissue: a survey. *Medical Engineering and Physics*, 29:413–431, 2007.

[2] N. Abolhassani and R. V. Patel. Deflection of a flexible needle during needle insertion into soft tiisue. In *Conf. of IEEE EMBS*, pages 3858–3861, August 2006.

[3] R. Alterovitz, K. Goldberg, and A. Okamura. Planning for steerable bevel-tip needle insertion through 2D soft tissue with obstacles. In *Proc. IEEE Int. Conf. Robot. Autom.*, pages 1652–1657, 2005.

[4] R. Alterovitz, A. Lim, K. Goldberg, G. S. Chirikjian, and A. M. Okamura. Steering flexible needles under markov motion uncertainty. In *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, pages 1570–1575. IEEE/RSJ, August 2005.

[5] R. Alterovitz, J. Pouliot, R. Taschereau, I.-C. J. Hsu, and K. Goldberg. Sensorless planning for medical needle insertion procedures. In *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, pages 3337–3343, October 2003.

BIBLIOGRAPHY

[6] R. Alterovitz, T. Simeon, , and K. Goldberg. The stochastic motion roadmap: A sampling framework for planning with markov motion uncertainty. In *Robotics: Science and Systems*, June 2007.

[7] K. S. Arun, T. S. Huang, and S. D. Blostein. Least-squares fitting of two 3-d point sets. *IEEE Trans. Pattern Anal. Mach. Intell.*, 9:698–700, 1987.

[8] A. Bloch. *Nonholonomic Mechanics and Control*, volume 24. Springer, NY, 2003.

[9] A. Bloch, P. Krishnaprasad, J. Marsden, and R. Murray. Nonholonomic mechanical systems with symmetry. *Archive for Rational Mechanics and Analysis*, 136:21–99, 1996.

[10] R. Brockett. Asymptotic stability and feedback stabilization. In R. W. Brockett, R. S. Milliman, and H. J. Sussman, editors, *Differential Geometric Control Theory*, pages 181–191. Birkhauser Publishers, 1983.

[11] F. Bullo and A. D. Lewis. Reduction, linearization, and stability of relative equilibria for mechanical systems on Riemannian manifolds. *Acta Applicandae Mathematicae*, 1:53–95, 2007.

[12] C. Carrasco, S. Wallace, and C. Charnsangavej. Aspiration biopsy: use of a curved needle. *Radiology*, pages 155–254, 1985.

[13] F. Chaumette. Potential problems of stability and convergence in image-based and position-based visual servoing. In D. Kriegman, G. . Hager, and A. Morse, editors,

BIBLIOGRAPHY

The Confluence of Vision and Control, pages 66–78. LNCIS Series, No 237, Springer-Verlag, 1998.

[14] A. V. Cideciyan. Registration of ocular fundus images: an algorithm usingcross-correlation of triple invariant image descriptors. *Engineering in Medicine and Biology Magazine, IEEE*, 14:52–58, 1995.

[15] D. Comaniciu, V. Ramesh, and P. Meer. Kernel-based object tracking. *IEEE Trans. Pattern Anal. Mach. Intell.*, 25(5):564–575, 2003.

[16] P. I. Corke and S. A. Hutchinson. A new partitioned approach to image-based visual servo control. *IEEE Trans. Robot. Autom.*, 17(4):507–515, 2001.

[17] N. J. Cowan and D. E. Chang. Geometric visual servoing. *IEEE Trans. Robot.*, 21(6):1128–1138, December 2005.

[18] N. J. Cowan, J. D. Weingarten, and D. E. Koditschek. Visual servoing via navigation functions. *IEEE Trans. Robot. Autom.*, 18(4):521–533, 2002.

[19] K. Deguchi. A direct interpretation of dynamic images with camera and object motions for vision guided robot control. *Int. J. Computer Vision*, 37(1):7–20, 2000.

[20] M. Dewan and G. Hager. Towards optimal kernel-based tracking. In *Proc. Computer Vision and Pattern Recognition*, volume 1, pages 618–625, June 2006.

[21] S. DiMaio, G. Fischer, S. Haker, N. Hata, I. Iordachita, C. Tempany, R. Kikinis, and

BIBLIOGRAPHY

G. Fichtinger. A system for MRI-guided prostate interventions. In *Proc. International Conference on Biomedical Robotics and Biomechatronics*, pages 68–73, 2006.

[22] S. DiMaio and S. Salcudean. Needle insertion modeling and simulation. *IEEE Trans. Robot. Autom.*, 19:864–875, 2003.

[23] M. Ding and H. N. Cardinal. Automatic needle segmentation in three-dimensional ultrasound image using two orthogonal two-dimensional image projections. *Medical Physics*, 30(2):222–34, 2003.

[24] M. Ding, H. N. Cardinal, W. Guan, and A. Fenster. Automatic needle segementation in 3D ultrasound images. In *SPIE Med. Imaging*, 2002.

[25] V. Duindam, R. Alterovitz, S. Sastry, and K. Goldberg. Screw-based motion planning for bevel-tip flexible needles in 3d environments with obstacles. In *Proc. IEEE Int. Conf. Robot. Autom.*, May 2008.

[26] J. W. Eaton. *GNU Octave Manual*. Network Theory Limited, 2002.

[27] J. Engh, G. Podnar, S. Khoo, and C. Riviere. Flexible needle steering system for percutaneous access to deep zones of the brain. In *Proc. IEEE Northeast Bioengineering Conf.*, pages 103–104, April 2006.

[28] J. Engh, G. Podnar, D. Kondziolka, and C. Riviere. Toward effective needle steering in brain tissue. In *Proc. Int. Conf. of the IEEE Engineering in Medicine and Biology Society*, pages 559–562, 2006.

BIBLIOGRAPHY

[29] Z. Fan, Y. Wu, and M. Yang. Multiple collaborative kernel tracking. In *Proc. Computer Vision and Pattern Recognition*, 2005.

[30] N. R. Gans, S. A. Hutchinson, and P. I. Corke. Performance tests for visual servo control systems, with application to partitioned approaches to visual servo control. *Int. J. Robot. Res.*, 22(10):955–981, 2003.

[31] D. Glozman and M. Shoham. Flexible needle steering and optimal trajectory planning for percutaneous therapies. In *Proc. Medical Image Computing and Computer-Assisted Intervention - MICCAI*, pages 137–144, 2004.

[32] D. Glozman and M. Shoham. Image-guided robotic flexible needle steering. *IEEE Trans. Robot.*, 23:459–467, 2007.

[33] C. Guot, T.-J. Tarnt, N. Xit, and A. K. Bejcz. Fusion of human and machine intelligence for telerobotic systems. In *Proc. IEEE Int. Conf. Robot. Autom.*, volume 3, pages 3110–3115, May 1995.

[34] G. D. Hager. A modular system for robust hand-eye coordination. *IEEE Trans. Robot. Autom.*, 13(4):582–595, 1997.

[35] G. D. Hager, M. Dewan, and C. V.Stewart. Multiple kernel tracking with SSD. In *Proc. Computer Vision and Pattern Recognition*, volume 1, pages 790–97, 2004.

[36] G. D. Hager and K. Toyama. The "XVision" system: A general purpose substrate for

real-time vision applications. *Computer Vision and Image Understanding*, 69(1):23–27, Jan. 1998.

[37] H. Hanssmann, N. E. Leonard, and T. R. Smith. Symmetry and reduction for coordinated rigid bodies. *European Journal of Control*, 2:176–194, 2006.

[38] S. Hutchinson, G. D. Hager, and P. I. Corke. A tutorial on visual servo control. *IEEE Trans. Robot. Autom.*, 12(5):651–670, 1996.

[39] T. Ikeda, T. Mita, and B. Anderson. Variable constraint control for a class of nonholonomic systems. Technical report, Department of Mechanical Control and Systems, Tokyo Institute of Technology, 2003.

[40] C. J. Isham. *Modern Differential Geometry for Physicists*. World Scientific, 1999.

[41] S. Jalnapurkar and J. Marsden. Stabilization of relative equilibria. *IEEE Trans. Autom. Control*, 45(8):1483–1491, 2000.

[42] V. Kallem, D. E. Chang, and N. J. Cowan. Task-induced symmetry and reduction in kinematic systems with application to needle steering. In *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, pages 3302–3308, San Diego, CA, USA, October 2007.

[43] V. Kallem and N. J. Cowan. Image guidance of flexible bevel-tip needles. *IEEE Trans. Robot.* Accepted.

[44] V. Kallem and N. J. Cowan. Image-guided control of flexible bevel-tip needles. In *Proc. IEEE Int. Conf. Robot. Autom.*, pages 3015–3020, Rome, Italy, April 2007.

BIBLIOGRAPHY

[45] V. Kallem, M. Dewan, J. P. Swensen, G. D. Hager, and N. J. Cowan. Kernel-based visual servoing. In *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, pages 1975–1980, San Diego, CA, USA, October 2007.

[46] G. Kantor and A. Rizzi. Feedback control of underactuated systems via sequential composition: Visually guided control of a unicycle. In *International Symposium of Robotics Research*, pages 281–290. Interntional Foundation for Robotics Research, Siena, Italy, 2003.

[47] H. K. Khalil. *Nonlinear Systems*. Prentice Hall, 2 edition, 2002.

[48] J. C. Kinsey and L. L. Whitcomb. Adaptive identification on the group of rigid-body rotations and its application to underwater vehicle navigation. *IEEE Trans. Robot.*, 23:124–136, 2007.

[49] S. Kobayashi and K. Nomizu. *Foundations of Differential Geometry, Vol. I.* Wiley-Interscience, 1996.

[50] K. Kowalski and W.-H. Steeb. *Nonlinear Dynamical Systems and Carleman Linearization*. World Scientific Publishing Company, 1991.

[51] D. Kragic and H. I. Christensen. Survey on visual servoing for manipulation. Technical Report, January 2002.

[52] D. Kragic and H. I. Christensen. Robust visual servoing. *Int. J. Robot. Res.*, 22(10-11):923–939, 2003.

BIBLIOGRAPHY

[53] G. A. D. Lopes and D. E. Koditschek. Level sets and stable manifold approxima-
tions for perceptually driven non-holonomically constrained navigation. *Advanced
Robotics*, 19(10):1081–1095, 2005.

[54] C.-P. Lu, G. D. Hager, and E. Mjolsness. Fast and globally convergent pose estima-
tion from video images. *Transactions on Pattern Analysis and Machine Intelligence*,
22(6):610–622, 2000.

[55] E. Malis and F. Chaumette. Theoretical improvements in the stability analysis of
a new class of model-free visual servoing methods. *IEEE Trans. Robot. Autom.*,
18(2):176–186, 2002.

[56] E. Malis, F. Chaumette, and S. Boudet. 2-1/2-d visual servoing. *IEEE Trans. Robot.
Autom.*, 15(2):238–250, Apr 1999.

[57] J. Marsden. *Lectures on Mechanics*. Cambridge University Press, 1992.

[58] J. Marsden, R. Montgomery, and T. Ratiu. *Reduction, Symmetry and Phases in Me-
chanics*. American Mathematical Society, 1990.

[59] J. E. Marsden and J. P. Ostrowski. Symmetries in motion: Geometric foundations of
motion control. *Nonlinear Science Today*, 1998.

[60] J. E. Marsden and T. S. Ratiu. *Introduction to Mechanics and Symmetry*. Springer,
1999.

BIBLIOGRAPHY

[61] Y. Mezouar and F. Chaumette. Path planning for robust image-based control. *IEEE Trans. Robot. Autom.*, 18(4):534–549, 2002.

[62] D. S. Minhas, J. A. Engh, M. M. Fenske, and C. N. Riviere. Modeling of needle steering via duty-cycled spinning. In *Conf. of the IEEE EMBS*, 2007.

[63] R. Murray and S. Sastry. Nonholonomic motion planning: Steering using sinusoids. *IEEE Trans. Autom. Control*, 1993.

[64] R. M. Murray, Z. Li, and S. S. Sastry. *A Mathematical Introduction to Robotic Manipulation*. CRC Press, Boca Raton, FL, 1994.

[65] S. Nath, Z. Chen, N. Yue, S. Trumpore, and R. Peschel. Dosimetric effects of needle divergence in prostate seed implant using I and Pd radioactive seeds. *Medical Physics*, 27:1058–1066, May 2000.

[66] S. Okazawa, R. Ebrahimi, J. Chuang, S. E. Salcudean, and R. Rohling. Hand-held steerable needle device. *IEEE/ASME Trans. Mech.*, 10(3):285–296, 2005.

[67] J. Ostrowski. Computing reduced equations for robotic systems with constraints and symmetries. *IEEE Trans. Robot. Autom.*, 15(1):111–123, 1999.

[68] W. Park, J. S. Kim, Y. Zhou, N. J. Cowan, A. M. Okamura, and G. S. Chirikjian. Diffusion-based motion planning for a nonholonomic flexible needle model. In *Proc. IEEE Int. Conf. Robot. Autom.*, pages 4600–4605, Barcelona, Spain, April 2005. IEEE.

BIBLIOGRAPHY

[69] W. Park, Y. Liu, Y. Zhou, M. Moses, and G. S. Chirikjian. Kinematic state estimation and motion planning for stochastic nonholonomic systems using the exponential map. *Robotica*, 2008. Accepted.

[70] K. B. Reed. Compensating for torsion windup in steerable needles. In *BioRob*, 2008. Submitted.

[71] K. B. Reed, N. J. Cowan, and A. M. Okamura. Torsion windup in steerable needles, 2008. In prep.

[72] K. B. Reed, R. A. Vinutha Kallem, K. Goldberg, A. M. Okamura, and N. J. Cowan. Integrated planning and image-guided control for plan needle-steering. In *BioRob*, 2008. Submitted.

[73] J. M. Romano, R. J. Webster III, and A. M. Okamura. Teleoperation of steerable needles. In *Proc. IEEE Int. Conf. Robot. Autom.*, pages 934–939, April 2007.

[74] S. Sastry. *Nonlinear Systems*. Springer, 1999.

[75] S. Soatto, R. Frezza, and P. Perona. Motion estimation via dynamic vision. *IEEE Trans. Autom. Control*, 41:393–413, 1996.

[76] D. Stoianovici, L. L. Whitcomb, J. H. Anderson, R. H. Taylor, and L. R. Kavoussi. A modular surgical robotic system for image guided percutaneous procedures. In *Proc. Medical Image Computing and Computer-Assisted Intervention - MICCAI*, volume 1496, pages 404–410, Octobar 1998.

BIBLIOGRAPHY

[77] D. Y. Sze. Use of curved needles to perform biopsies and drainages of inaccessible targets. *Vascular Interventional Radiology*, 12:1441–1444, 2001.

[78] O. Tahri and F. Chaumette. Point-based and region-based image moments for visual servoing of planar objects. *IEEE Trans. Robot.*, 21(6):1116–1127, December 2005.

[79] C. J. Taylor and J. P. Ostrowski. Robust vision-based pose control. In *Proc. IEEE Int. Conf. Robot. Autom.*, volume 3, pages 2734–2740, San Francisco, CA, 2000.

[80] C. Tomasi and T. Kanade. Shape and motion from from imgae streams under orthography: a factorization method. *Int. J. Computer Vision*, 9:137–154, 1992.

[81] S. Umeyama. Least-squares estimation of transformation parameters between two point patterns. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 13:376–380, 1991.

[82] R. Vidal and R. Hartley. Motion segmentation with missing data using powerfactorization and gpca. In *Proc. Computer Vision and Pattern Recognition*, volume 2, pages 310–316, 2004.

[83] R. Vidal, Y. Ma, S. Soatto, and S. Sastry. Two-view multi body structure from motion. *Int. J. Computer Vision*, 68:7–25, June 2004.

[84] Y. Wang, Y. Zhou, D. K. Maslen, and G. S. Chirikjian. Solving phase-noise fokker-planck equations using the motion-group fourier transform. *IEEE Trans. Commun.*, 54:868–877, 2006.

BIBLIOGRAPHY

[85] R. J. Webster III, J. S. Kim, N. J. Cowan, G. S. Chirikjian, and A. M. Okamura. Nonholonomic modeling of needle steering. *Int. J. Robot. Res.*, 25(5/6):509–526, May/June 2006.

[86] R. J. Webster III, J. Memisevic, and A. M. Okamura. Design considerations for robotic needle steering. In *Proc. IEEE Int. Conf. Robot. Autom.*, pages 3588–3594, April 2005.

[87] Z. Wei, L. Gardi, D. Downey, and A. Fenster. Oblique needle segmentation for 3D TRUS-guided rovot-aided transperineal prostate brachytherapy. In *IEEE Biomedical Imaging: Nano to Macro*, volume 1, pages 960–963, April 2004.

[88] Z. Wei, G. Wan, L. Gardi, G. Mills, D. Downey, and A. Fenster. Robot-assisted 3d-trus guided prostate brachytherapy: system integration and validation. *Med. Phys.*, 31 (3):539–48, March 2004.

[89] W. J. Wilson. Visual servo control of robots using kalman filter estimates of robot pose relative to work-pieces. *In K. Hashimoto, editor, Visual Servoing, World Scientific*, pages 71–104, 1994.

[90] J. H. Youk, E. K. Kim, M. J. Kim, J. Y. Lee, and K. K. Oh. Missed breast cancers at US-guided core ceedle biopsy: How to reduce them. In *Radiographics*, 2007.

[91] P. Zanne, G. Morel, and F. Plestan. Robust 3d vision based control and planning. In *Proc. IEEE Int. Conf. Robot. Autom.*, volume 5, pages 4423–4428, April 2004.

[92] H. Zhang and J. P. Ostrowski. Visual motion planning for mobile robots. *IEEE Trans. Robot. Autom.*, 18(2):199–208, 2002.

Vita

Vinutha Kallem received her B. Tech. degree from Indian Institute of Technology, Madras in 2002 and M. S. degree from Stanford University in 2004; both in Mechanical Engineering. She enrolled in the Mechanical Engineering Ph.D. program at Johns Hopkins University in 2004. She was a part of the Locomotion of Mechanical and Biological Systems (LIMBS) laboratory and Laboratory of Sensing and Robotics (LCSR) and was advised by Prof. Noah J. Cowan. Her research interests lie at the intersection of control theory, geometric mechanics, computer vision and their applications to mobile and medical robotics. During her stay at Johns Hopkins she also served on the Whiting School of Diversity Council, Women of Whiting and Mechanical Engineering Graduate Association.