BodySLAM: Localization and Mapping for Surgical Guidance

Submitted in partial fulfillment of the requirements for
the degree of
Doctor of Philosophy
in
Electrical and Computer Engineering

Stephen T. Tully
B.S., Electrical Engineering and Computer Science, Yale University
M.S., Electrical and Computer Engineering,
Carnegie Mellon University

Carnegie Mellon University
Pittsburgh, PA
May, 2012

For my parents, John and Mary Tully.

Abstract

Minimally invasive surgery (MIS), as opposed to open surgery, reduces patient trauma and post-operative discomfort. Unfortunately, when a surgeon adopts MIS for a procedure, he or she loses the direct view of the operation that is expected with open surgery. To overcome this loss of situational awareness, one technique is to display a 3D rendered visualization of the operation on a computer screen in the operating room (OR), which is a method referred to as *image-guidance*. Unfortunately, existing image-guidance systems can be inaccurate due to registration error and tool tracking error.

In this thesis, we are introducing novel filtering algorithms that are aimed at advancing the state-of-the-art in image-guided surgery. Our approach is to use stochastic models and nonlinear filtering to recursively estimate the configuration of a surgical robot during MIS. Then, using a novel inequality constrained Kalman filtering algorithm, we recursively adjust shape and registration parameters during an operation to improve the accuracy of surgical guidance. To account for situations in which the robot is interacting forcefully with a flexible surface, we then apply an equality constrained filtering algorithm that corrects the state and accounts for surface deformation. Lastly, to improve the representation of organs during image-guidance, we use a novel feature mapping algorithm based on Fourier series parameters to estimate the motion of periodically moving visually detected features.

Our overall approach relates to the mobile robot problem called *simultaneous localization and mapping* (SLAM), which investigates the problem of incrementally constructing an environment map with a mobile robot while simultaneously localizing the robot in the map. Due to the fact that our algorithm specifically applies to surgical estimation within the body, we have named our overall approach *BodySLAM*. We have performed bench-top and live animal experiments to validate our algorithms and to demonstrate that this work has the potential to improve the accuracy and efficacy of image-guided robotic surgery.

Acknowledgments

To begin, I would like the thank my parents and family. You instilled an intellectual curiosity in me, without which I would not have reached this point in my life. Your love and support has carried me through all of life's challenges since the beginning and I have you to thank for my successes.

I would like to especially thank my advisors Howie Choset and George Kantor. Your support, both personally and professionally, has helped me reach this goal and has helped shape my future. I appreciate all of the guidance you have given me over the years. Also, I would like to thank the other two members of my doctoral committee, Bruno Sinopoli and Dr. Marco Zenati, for their support and constructive feedback. You added a valuable perspective on the filtering and surgical aspects of my work.

For her patience and loving support during my time in graduate school, I would like to thank my fiancé, Kacy. You picked up my slack when I had difficult deadlines and helped carry me through the final stages of my thesis preparation. I will always love you and am very much looking forward to our wedding this summer. I want to also thank all of my friends from Guilford, Yale, and CMU for adding balance to my life. Thanks to you, my time in graduate school has not only been a lot of work, but also a lot of fun.

Additionally, I would like to thank all past and present members of the Biorobotics lab at CMU for their help and support: Cornell Wright, Brett Zubiate, Michael Schwerin, Amir Degani, Ross Hatton, Matthew Tesch, Chaohui Gong, David Rollinson, Glenn Wagner, Matthew Travers, Hyungpil Moon, David Conner, Sarjoun Skaff, Benjamin Morse, Austin Buchan, Deryck Morales, Florian Enner, Jason Geist, Peggy Martin, and others. I would also like to thank other past and present members of the CMU community with whom I have collaborated and/or interacted: Nate Wood, Joe Djugash, Alli Kator, Arianna Golden, and Hyun Soo Park. Also, I owe a debt of gratitude to my external collaborators outside of the CMU community: Andrea Bajo, Nabil Simaan, Tao Tong, Yu Fu, Felix Werner, Dr. Shyam Thakkar, Dr. Michael Awad, and Dr. David Schwartzman.

Finally, the completion of this thesis and my doctoral degree would not have been possible without the financial support provided by Army Research Office (ARO) grant #W911NF1010343 and by National Institutes of Health (NIH) grant #1R01HL079940-01A2.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

Minimally invasive surgery (MIS) has become increasingly popular in medicine due to its ability to reduce trauma and hospitalization cost when compared to conventional open surgery [23]. With MIS, one or more small incisions are made through which a surgical device can enter a patient for diagnostic or interventional procedures, as seen in Fig. 1.1-(b). This is instead of creating a large opening to access the organs, as seen in Fig. 1.1-(a). While the majority of MIS procedures are performed using hand-held laparoscopic and endoscopic tools, the use of robotic devices for surgery is increasing due to the potential benefits that robots can add to surgical accuracy and maneuverability.

Unfortunately, when adopting MIS for a procedure, a physician loses the direct view of the operation that is expected with open surgery, thereby sacrificing direct visual and tactile feedback of the surgical tool's position. To overcome this loss of situational awareness, one technique is to display a 3D rendered visualization of the operation on a computer screen in the operating room (OR). This form of visual feedback during surgery is usually referred to as *image-guidance*.

Existing commercial image-guidance systems include Ensite NavX (St Jude Medical, St Paul, MN, USA) and Carto (Bio-Sense Webster, Diamond Bar, CA, USA), which are both

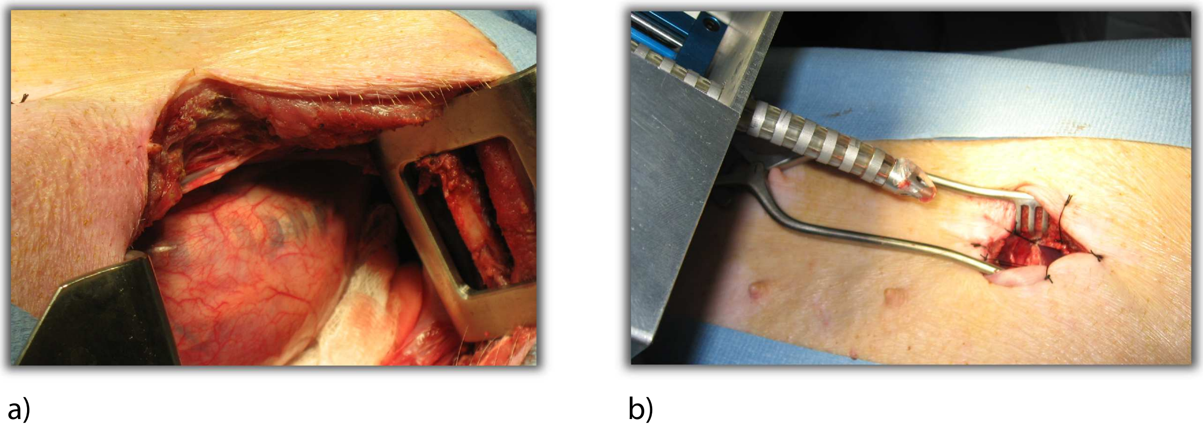


Figure 1.1: In (a), a large surgical opening is shown which would result in significant patient trauma. In (b), a small incision is shown that can be used for surgical access during a minimally invasive procedure.

used for electro-anatomical mapping for endocardial ablation. But, unfortunately, these existing solutions are often inaccurate due to organ movement and registration error [51].

In this thesis, we are introducing novel filtering algorithms aimed at advancing the state-of-the-art in image-guided surgery and overcoming the shortfalls of existing solutions. An example of our resulting image-guidance system is shown in Fig. 1.2. Our approach can be described in four parts: 1) using stochastic models and nonlinear state estimation to compute the shape and pose of a surgical robot during MIS, 2) using inequality constrained Kalman filtering to incrementally adjust shape and registration parameters during an operation for improving the accuracy of surgical guidance, 3) using equality constrained Kalman filtering to constrain the estimation problem when there is forceful contact between the robot and the environment, and 4) using a mapping procedure based on Fourier series parameters to estimate the motion of periodically moving organs.

Our approach has been largely influenced by a popular research topic in robotics called *simultaneous localization and mapping* (SLAM), which is a topic most commonly associated with wheeled mobile robots. SLAM is the task of having a robot incrementally construct a map of its surrounding environment while simultaneously localizing itself in the map. Due

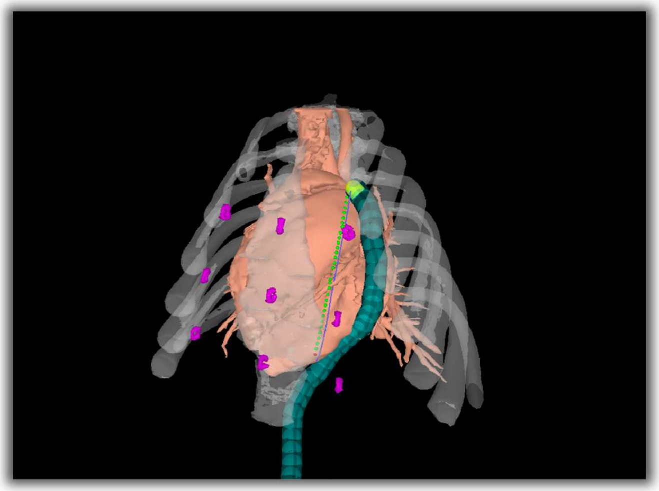


Figure 1.2: This visualization is an example of our surgical guidance software which displays a surgical snake robot along with preoperative rendered surface models.

to the fact that our algorithms specifically apply to surgical estimation within the body and due to its relation to SLAM, we have named our overall approach *BodySLAM*.

In this thesis, we will discuss several bench-top and *in vivo* experiments that we have completed in order to validate our BodySLAM approach. We believe that these experiments successfully demonstrate the benefit that can be obtained when adopting our filtering techniques for surgical estimation, and, thus, we believe that this work has the potential to improve the accuracy and efficacy of image-guided robotic surgery.

1.1 Motivation

For guiding surgical tools during MIS, surgeons often rely on the use of indirect medical imaging, such as fluoroscopy [28], ultrasound [31], real-time MRI [47] and CT [35], to view a surgical tool's position relative to a patient's anatomy. While these imaging technologies are more commonly used for preoperative diagnostic purposes, they have had notable success in a real-time OR setting to guide surgical devices [28, 31, 35, 47]. Unfortunately, each of these imaging modalities, when applied to MIS, has a significant drawback: e.g., ultrasound is inherently noisy [73], real-time CT is typically slow and cumbersome [22], and fluoroscopy is limited due to radiation concerns [56].

1.1.1 Motivation for Image-Guided Surgery

Instead of using medical imaging for guidance, we believe that online *image-guidance* is an approach with more potential for long-term success. With image-guidance, the surgeon typically views a detailed 3D visualization that acts as a rendered representation of the operation site with patient specific models, as seen in Fig. 1.2. The visualization is most commonly computed by tracking a surgical tool and registering its position to preoperative surface models. The main goal of image-guidance is to have the surgeon regain the situational awareness that he or she is accustomed to with open surgery.

We believe that there are several key advantages to using image-guidance during an MIS procedure: 1) the surgeon can observe an operation from different viewpoints by rotating the 3D rendering on a computer screen, which allows the surgeon to visually consider different points of access and different tool trajectories; 2) the visualization can display historical points to mark where a tracked tool has been, which can help distinguish between paths that have succeeded at reaching particular anatomical targets and those that have not succeeded; lastly, 3) an image-guidance system can potentially serve as a higher level user interface where surgeons can virtually input semi-autonomous paths for a surgical robot to follow during an operation.

1.1.2 Motivation for using SLAM to Perform Surgical Estimation

While there are advantages to having a detailed image-guidance system, existing solutions are often inaccurate and misrepresent the true state of the operation [51]. Possible sources of error include organ movement, registration error, and inaccurate tool tracking. In some reported cases, users have observed positioning errors for image-guided ablation procedures that have been as high as 10-15 mm [51].

We believe that SLAM, appropriately applied to surgery, can improve the accuracy and performance of image-guided surgery. As discussed before, SLAM is a technique in which a robot uses onboard sensing to incrementally construct a map of its surrounding environment while simultaneously localizing its position in the map. In general, SLAM is a useful tool for robotic systems because it enables accurate navigation due to the fact that the robot is continuously aware of its location relative to its surroundings. The common approaches to SLAM in robotics research all rely on some form of probabilistic filtering to estimate a probability distribution over a joint state space that parameterizes the environment map and the robot pose. By extending SLAM to surgery, we believe the following improvements can be made to image-guidance:

1. **Improved Accuracy:** By adopting a probabilistic filtering approach, registration parameters, surface deformation, and the configuration of the robot are recursively adjusted at each time step to be the statistically most likely solution. Additionally, the individual aspects of SLAM (localization and mapping) compound to produce a more accurate and representative model of the operation.

2. **Correcting for Infeasible States:** When inaccuracies, such as registration error, place the estimate of the surgical robot in a pose that is infeasible given preoperative models, the estimate can be moved to a feasible region by performing a constraint update step. This process can shrink the uncertainty of the system and thereby produce a more accurate representation of the operation for guidance.

3. **Sensor-fusion:** We believe that, at a fundamental level, SLAM is essentially a sensor-fusion algorithm that is continuously collecting and combining different sources of information to produce the most likely state of a system. This is an advantage for surgery, when it is desirable to have an efficient way to combine information from various imaging modalities, tracking sensors, preoperative data, and live measurements.

4. **Annotating Surface Models with Motion Data:** While the majority of current image-guidance systems rely on static preoperative models, the mapping aspect of SLAM can be used to estimate the periodic motion of nearby surfaces. We believe that the addition of surface motion to image-guidance models can provide a more informative graphical interface with which surgeons can plan paths to anatomical targets.

5. **Computing Error-Bounds:** By using a probabilistic filtering method, a probability distribution is estimated over the state space of the system. Thus, a measure of uncertainty is computed related to the models that are displayed within the image-guidance system. This is beneficial as it can provide for the surgeon a source of feedback that can determine which aspects of the visualization system can be trusted for accurate guidance.

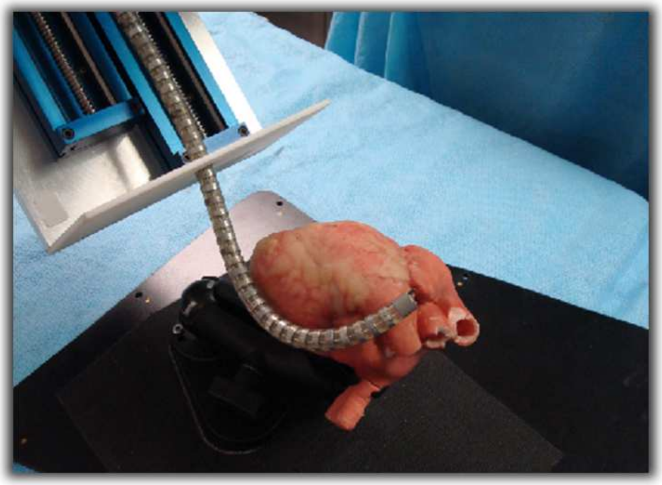


Figure 1.3: The HARP surgical snake robot, which is designed for MIS procedures.

1.2 Approach

Our BodySLAM approach seeks to extend the theoretical concepts that have been developed for mobile robot SLAM to the application of minimally invasive surgery (MIS). Unfortunately, when extending SLAM to surgery, many additional challenges present themselves: the anatomical structures surrounding the robot may be moving due to physiological motion, the workspace of the robot may be constrained due to nearby organs, and there may be instances when the robot is interacting forcefully with the environment.

In this thesis, we present four novel filtering algorithms that we have designed to overcome these challenges and to improve the accuracy of image-guided surgery: 1) a shape estimation algorithm for a high degree of freedom snake robot, 2) an inequality constrained filtering algorithm for localization, 3) an equality constrained filtering algorithm for localization with forceful contact, and 4) a periodic feature mapping algorithm that encodes motion with Fourier series parameters.

1.2.1 Snake Shape Estimation

For the majority of our experiments, we are using a surgical robot called HARP [15, 16], which stands for the *highly articulated robotic probe*. HARP is a high degree-of-freedom snake robot designed for minimally invasive surgical procedures, as seen in Fig. 1.3. In order to display a rendering of the robot's relative position during image-guided surgery, it is important to accurately estimate the shape and pose of the robot throughout a surgical procedure. We have designed an extended Kalman filter (EKF) to recursively estimate the shape of the HARP during an operation. For our implementation, the probabilistic models that we use within our filtering algorithm depend upon the mode of the robot, e.g. advancing, steering, or retracting.

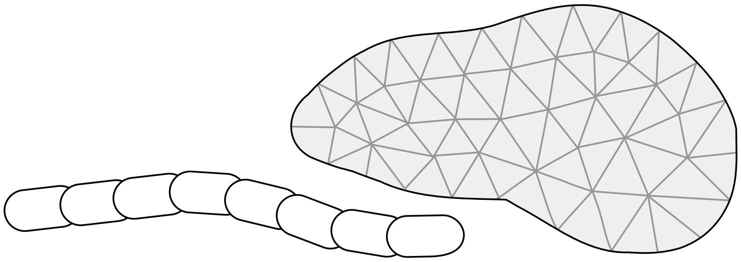


Figure 1.4: The first component of our approach involves estimating the shape and pose of a snake robot relative to preoperative surface models.

By design, the HARP robot is a follow-the-leader mechanism, meaning that points along the backbone of the robot will coincide with historical positions of the distal tip of the robot. For this reason, and because of well designed actuation models, we have been able to define appropriate process models for the robot's motion for use within the prediction step of the EKF that is estimating the shape of the robot. For the measurement correction step of the EKF, we are incorporating measurements from a 5-DOF magnetic tracking sensor that is inserted into the distal tip of the HARP robot. With some analysis, we have been able to show that this filtering problem is fully observable, meaning that the shape and pose can be sufficiently recovered given our stochastic models.

An image-guidance experiment begins by performing a preoperative scan of the patient with CT or MRI. These scans can then be used to reconstruct a set of preoperative surface models that represent the patient-specific organs. For our estimation approach, we can register the coordinate frame of the robot to the coordinate frame of these preoperative surface models using fiducial markers with known locations in both coordinate frames. With this filtering and registration approach, we have shown that we can produce an informative visualization suitable for image-guided surgery. Fig. 1.4 shows a conceptual depiction of this component of our approach. We will discuss our snake shape estimation method in more detail in Chapter 3.

1.2.2 Inequality Constrained Filtering for Localization

Due to possible registration and shape estimation error, there may be situations in which the rendered depiction of the robot in the image-guidance system is piercing through one of the preoperative surface models, as in Fig. 1.5. This is an infeasible state for the robot and such an occurrence should be corrected. We have devised a way to use the information provided by the detection of an infeasible state to our advantage with a novel update procedure that incorporates geometric constraints into the Kalman filter. The idea behind our constrained approach relies on the understanding that when it is possible to determine where the robot *cannot* be, it indirectly informs the estimation process about where the robot *can* be. Our novel implementation of inequality constrained Kalman filtering for surgical estimation effectively moves the estimate of the robot to a feasible state and reduces the overall uncertainty of the system.

The constrained update algorithm that we have developed is a two step process: our method first projects the Gaussian ellipsoid that represents the uncertainty in our filtering estimate to the robot workspace in order to truncate the Gaussian according to the

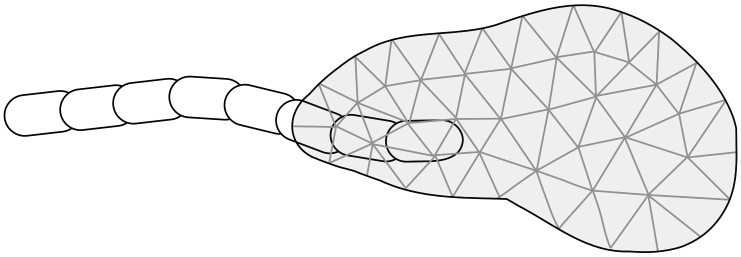


Figure 1.5: The second component of our approach deals with correcting the state estimate when an infeasible state is detected.

geometric constraint. Then, we project the Gaussian back to the state space, much in the same way that the measurement update in a conventional Kalman filter projects a measurement back to the state space. This moves the filter mean into a feasible region. The theoretical contribution of this work is a new approach to constrained filtering that is particularly beneficial to the problem of surgical estimation for MIS. We will discuss our inequality constrained filtering algorithm in more detail in Chapter 4.

1.2.3 Equality Constrained Filtering

Our inequality constrained filtering approach is primarily beneficial when it can be determined that the robot is on one side of a rigid geometric constraint. But when the surface models that are loaded for image-guidance are known to be deformable, an entirely different algorithm is required. We have additionally developed an equality constrained filtering algorithm that performs a constraint-based update when forceful contact is detected between the robot and a flexible surface.

The first step of our equality constrained update algorithm is to infer the force that is acting upon the end effector of the robot via the use of deflection and compliance models. Then, an equality constraint equation can be defined that balances the force acting upon the robot with the force acting upon the flexible surface (which we assume is a function of the surface stiffness). By incorporating this constraint equation into a pseudo-

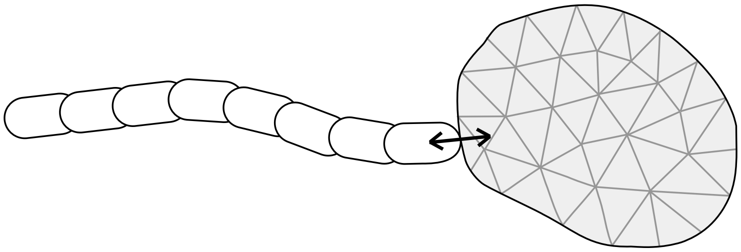


Figure 1.6: The third component of our approach uses equality constraints to handle forceful contact with the environment.

measurement update formulation of the Kalman filter, we simultaneously shift the robot pose to a more feasible state while computing an estimate for the local surface deformation. The theoretical contribution of this approach is a new method to estimate deformation and surface interaction for a robot pushing against tissue during an MIS procedure. In Fig. 1.6, a conceptual depiction of this component of the thesis is shown, in which a robot is forcefully pushing on an organ model which would trigger an equality constrained update. We will discuss our equality constrained filtering technique in more detail in Chapter 5.

1.2.4 Periodic Feature Mapping

Preoperative surface models used for image-guided surgery are typically static models that are fixed throughout an operation. But for cardiac and abdominal applications, for which there is significant physiological motion, these models will most likely not be accurate representations of the patient's organs at any given point in time. To provide the surgeon with a more informative visualization for surgical feedback, it would be desirable for an image-guidance system to render 4D motion-enabled surface models. Towards this goal, we are presenting a filtering algorithm, related to SLAM, that can map the motion of periodically moving visual features that are detected in the robot's onboard camera images.

The mapping of 3D periodic motion with 2D camera images is a challenging task. For our filtering approach, we rely on a bearing-only measurement model that maps a

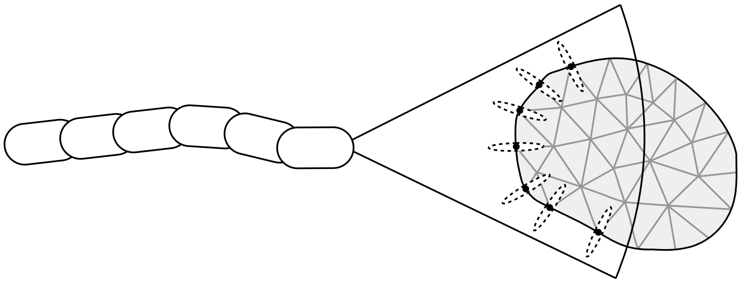


Figure 1.7: This fourth component of our approach uses Fourier series parameters to estimate the periodic motion of visually detected features.

3D point in the global coordinate frame to a 2D feature position in the camera image frame. This bearing-only measurement model is highly nonlinear, which can be problematic for conventional filtering algorithms due to linearization error and filter divergence. Our solution is to use a novel iterative approach for the measurement update step of the Kalman filter. This idea is motivated by the fact that the measurement update task of the Kalman filter, at a fundamental level, can be interpreted as an optimization problem whose solution is obtained through numerical iteration. For our experiments, the pose of the camera is tracked by a magnetic tracking system and the motion of each feature is encoded by Fourier series parameters (specifically the elements of a Fast Fourier Transform (FFT)) within the Kalman filter. We have been able to demonstrate experimentally that our approach is able to successfully map the periodic motion of visually detected features. The theoretical contribution of this work is a stable filtering approach for feature-based periodic motion estimation for surgical applications. In Fig. 1.7, a conceptual depiction of this component of the thesis is shown, in which a robot is mapping moving features. We will discuss our periodic feature mapping algorithm in more detail in Chapter 6.

Chapter 2

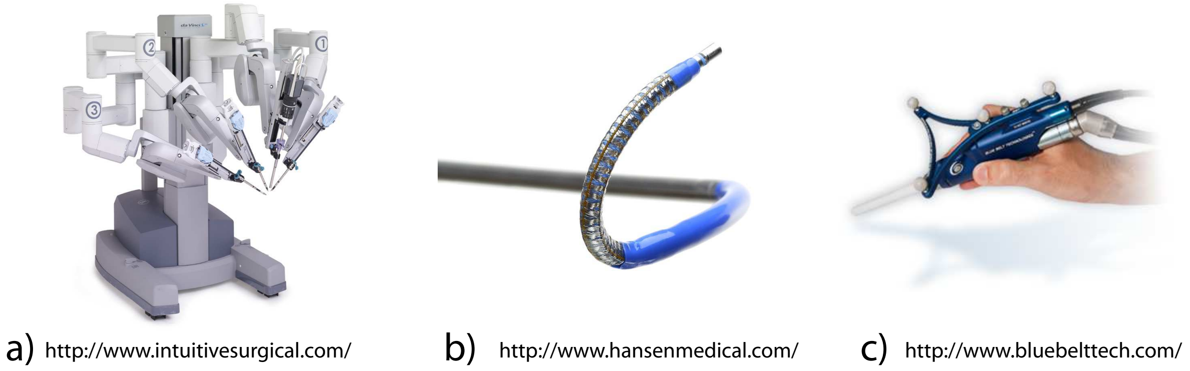
Background

The related work for this thesis topic spans many research areas including robotics, medical imaging, nonlinear filtering theory, mechanism design, image-guided surgery, and SLAM. In this chapter, we will introduce some of the research efforts that have led to our work on BodySLAM.

2.1 Surgical Robots

One of the most widely used commercial surgical systems designed specifically for robotic MIS is the Da Vinci ® surgical robot (Intuitive Surgical Inc., Sunnyvale, CA, USA). The Da Vinci surgical system consists of a multi-arm surgical slave, shown in Fig. 2.1-(a), and a surgical master. The surgical slave can manipulate 2-3 laparoscopic tools while the surgical master acts as a controller and visual feedback console for the primary surgeon. The robot was originally targeted for minimally invasive cardiac applications, but has recently gained traction for surgeries relating to the prostate. A stereo endoscopic camera provides a 3D view that can be used to guide the robot and help manipulate the robot's tools.

Another example of a commercially produced robot-like surgical system is the Sensei



a) http://www.intuitivesurgical.com/ b) http://www.hansenmedical.com/ c) http://www.bluebelttech.com/

Figure 2.1: Commercial surgical robots, including (a) the Da Vinci® surgical robot (Intuitive Surgical Inc., Sunnyvale, CA, USA), (b) the Sensei X® robotic catheter system (Hansen Medical Inc., Mountain View, CA, USA), and (c) the Navio PFS™ hand-held device (Blue Belt Technologies Inc., Pittsburgh, PA, USA).

X® robotic catheter system (Hansen Medical Inc., Mountain View, CA, USA), which is shown in Fig. 2.1-(b). The Sensei is a catheter-like robot that has been designed for cases related to the diagnosis or treatment of cardiac arrhythmias. The benefit of the Sensei system versus conventional catheter tools is the ability of the Sensei to add stability to the delivery of a tool to a location within a patient's heart. By adding improved control and actuation, the Sensei can potentially aid in the accurate placement of complex ablation lesions within the heart.

Simaan et. al. have developed a multi-backbone continuum robot [4, 5] that is designed for MIS procedures. The robot is a small snake-like robot with three independently actuated segments. Each segment is constructed using three circumferentially located super elastic NiTi secondary backbones and one centrally located super elastic NiTi primary backbone. Actuation, for each segment, is achieved by changing the lengths of the secondary backbones in order to actively bend the segments. In [4, 5], kinematics derivations are presented that can be used to infer surface contact and wrenches on the robot backbone.

The Navio PFS™ hand-held device (Blue Belt Technologies Inc., Pittsburgh, PA, USA) is another commercial robotic tool for surgical applications, shown in Fig. 2.1-(c).

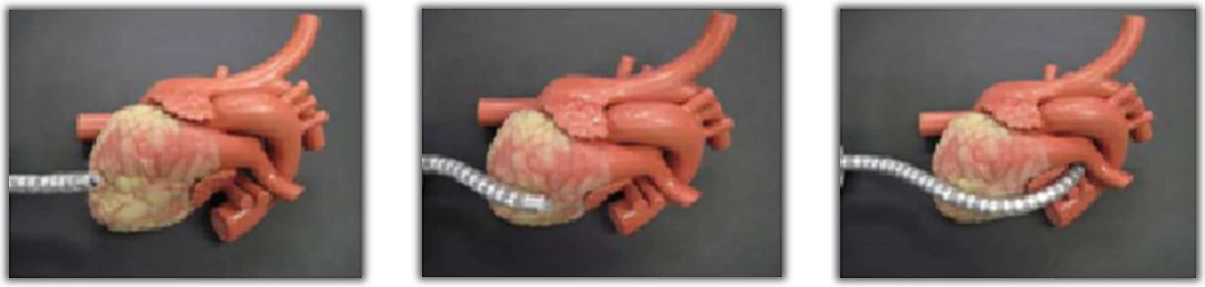


Figure 2.2: The HARP robot is shown advancing and steering around a bench-top heart phantom.

The original technology for this device was developed through a project called HipNAV [33], that developed an image-guided approach for implanting hip replacements. The Blue Belt Navio PFS device is tracked using optical tracking methods and its position is displayed on a computer monitor relative to renderings of the patient's bones. One advantage of this device is the system safely limits the sculpting of bone to only the targeted area.

2.2 The HARP Surgical Robot

The robot that we have adopted for the majority of our experimental evaluation is the HARP surgical snake robot [15, 16], which stands for the *highly articulated robotic probe*. The HARP is shown in Fig. 1.3 and Fig. 2.2. The main advantage of the HARP is that it has the stability of a rigid device as well as the maneuverability of a flexible tool. The current iteration of the HARP robot has a probe with a 12mm diameter.

The robot is somewhat unique, in that it can navigate any curve (up to a curvature limit) in a three-dimensional space with only six actuators (see Fig. 2.2). This is possible because of its unique mechanical design, which consists of a feeding mechanism (that utilizes two conventional motors for advancing) and a 30cm snake-like probe (that is actuated by cables connected to four additional motors). The probe is mechanically constrained to actuate in a follow-the-leader fashion because it is comprised of an inner mechanism (IM) and an outer mechanism (OM), each of which can either be rigidified or made limp

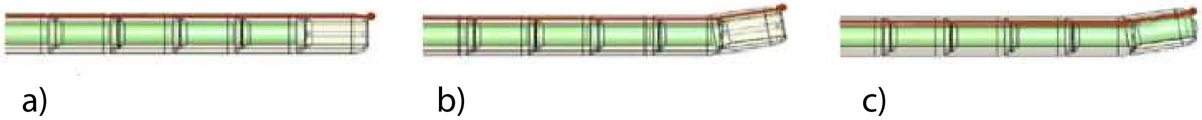


Figure 2.3: The HARP robot advances in a follow-the-leader fashion by alternating the rigidity of an inner mechanism and an outer mechanism.

depending on the tension of the actuation cables.

In Fig. 2.3, we show how the robot steers and advances. First the OM is limp while the IM is rigid and the OM is advanced past the IM, as shown in Fig. 2.3-(a). The distal link can then be steered according to joystick control, as shown in Fig. 2.3-(b). After steering, the OM is rigidified and the IM is made limp and advanced within the OM (shown in Fig. 2.3-(c)). During this stage, the OM is maintaining the robot shape. This process is repeated as desired to steer and advance the robot into an arbitrary curve in 3D space.

One of the target applications for the HARP robot upon its inception was epicardial ablation, which was experimentally evaluated in [48]. It was shown that the robot can enter the intrapericardial space through a subxiphoid incision and navigate to remote locations without adversely affecting hemodynamics [48]. Unlike the approach that Hansen Medical is taking with their Sensei system (delivering a tool to an endocardial location inside the heart), the HARP robot is designed for epicardial procedures (outside of the heart). The goal is to treat cardiac arrhythmias that cannot be treated via an endocardial approach and to possibly extend the treatment of arrhythmias to patients that do not qualify for the more conventional catheter ablation procedure.

Recently, we have been investigating the development of a longer version of the HARP robot that can be used for natural orifice transluminal endoscopic surgery (NOTES). For recent animal trials, we have extended the length of the robot to 80cm for the purpose of enabling access to the pancreas from either a transrectal or transvaginal approach. An image showing successful access to the pancreas is shown in Fig. 2.4-(a). For this particular experiment, we were able to successfully resect the distal tip of the pancreas using a needle

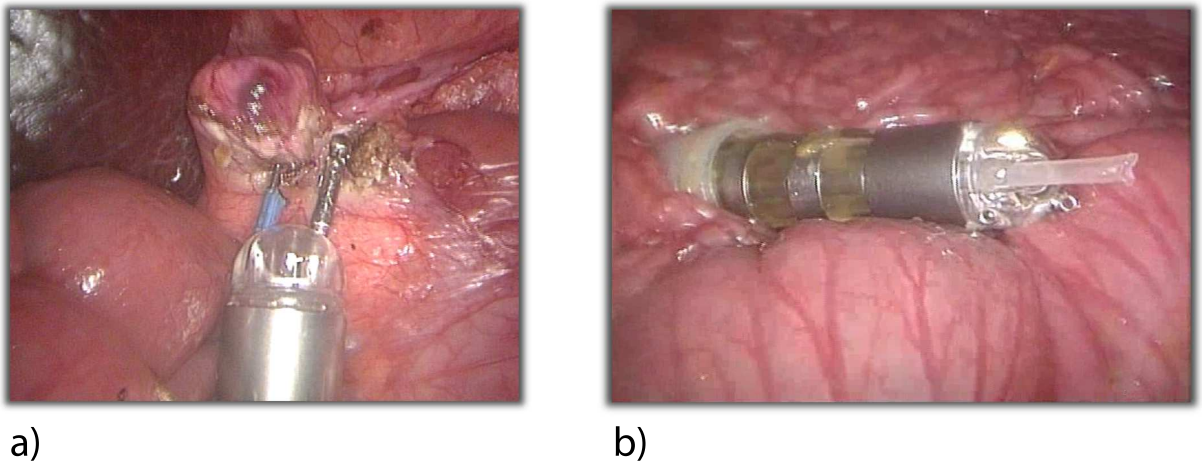
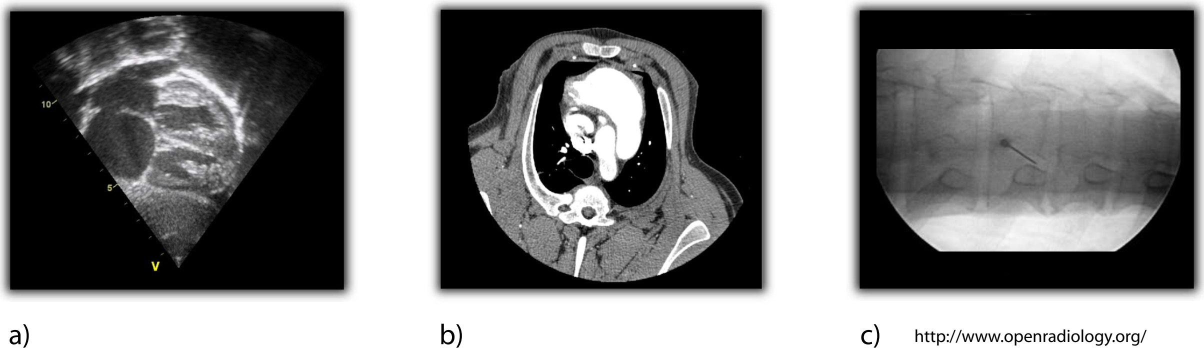


Figure 2.4: In (a), an image is shown depicting the HARP during a NOTES experiment that investigated accessing and resecting the pancreas via a transrectal approach. In (b), the robot is shown access the abdominal cavity through an overtube.

knife cautery tool inserted into one of the working channels of the HARP snake robot. Accessing the abdominal cavity was achieved through the use of an overtube, as seen in Fig. 2.4-(b), that acted as both a cannula for access and an assistive structure that supported the increased length of the robot. While this experiment is a promising result, we note that the guidance of the robot was performed through supplemental laparoscopic visualization and assistance.

2.3 Medical Imaging for Guidance

Fluoroscopy is a widely used technique for tracking and guiding surgical tools and robotic systems during surgery [28]. The technology used for fluoroscopy is similar to X-ray except for the fact that it is used primarily for real-time imaging. By viewing a robot or tool in a fluoroscopic image and seeing its position relative to an anatomical target, the tool can be guided to a new location. Unfortunately, fluoroscopy only offers a 2D projective view of the operation and can only be used in short bursts of time due to concerns about the added radiation. In fact, it has been shown recently that guidance using fluoroscopy can result



http://www.openradiology.org/

Figure 2.5: Medical imaging technology includes (a) ultrasound, (b) CT, and (c) fluoroscopy.

in radiation-induced skin damage after extended periods of exposure [56]. An example of a fluoroscopic guidance image is shown in Fig. 2.5-(c).

Another imaging modality that is often used for surgical guidance is ultrasound [31]. Ultrasound uses ultrasonic sound waves to image subcutaneous anatomical structures. The resulting image is computed by processing the ultrasonic echo that is received by the transducer. An example ultrasound image is shown in Fig. 2.5-(a). For surgical guidance, a robot can be detected in the ultrasound images [73] and then guided to a desired anatomical location. Unfortunately, ultrasound can be noisy and often difficult to interpret without significant training. Similarly, while ultrasound has been extended to real-time 3D imaging, this technology can have a limited field of view and can be difficult to segment [8, 34, 46, 80].

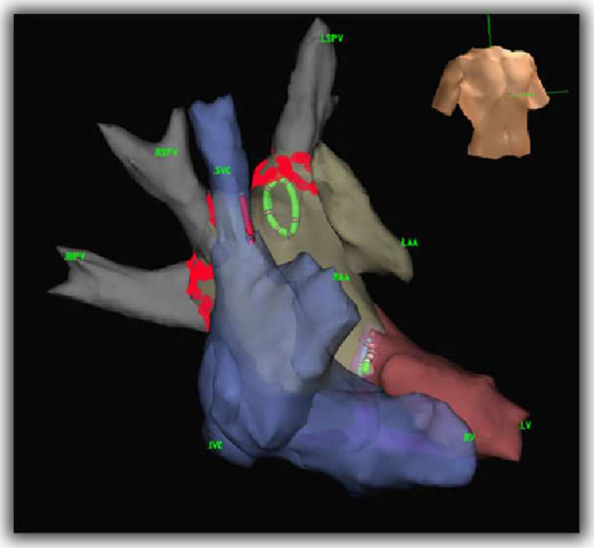
CT and MRI are other possible imaging modalities that can be used for surgical guidance. CT and MRI [36, 47] are accurate medical imaging modalities that can provide 3D volumetric information via the processing of many 2D image slices. Typically, CT and MRI are used preoperatively for diagnostic purposes and do not, by themselves, offer live image-guidance. Recently, though, real-time MRI has been investigated and targeted to small areas of interest [27, 44, 47]. Unfortunately, this methodology is particularly affected by the existence of metal in the operating room [44] and is limited by robot compatibility, high cost, and training time. CT has recently been extended to real-time guidance using a rotational X-ray alongside preoperative images [36, 37, 66]. The shortcoming of this approach is the significant radiation that is generated by the real-time X-ray [36]. Both CT and MRI systems are costly and can be slow and cumbersome [22]. An example of a CT scan image is shown in Fig. 2.5-(b).

2.4 Image-Guided Surgery

Image-guided surgery is a term that is often used to describe a procedure that uses patient-specific medical images in conjunction with tool-tracking as a form of visual feedback during surgery. In many cases, this equates to using preoperative CT or MRI data to construct a 3D surface model of anatomical structures, as in [11, 75]. With image-guided surgery, a tracking device is integrated with a surgical tool and registered to the preoperative images so that the position of the tool can be overlayed on the rendered models. This provides to the surgeon a spatial understanding of where the tool or robot is located in relation to different anatomical targets within the body. Thus, image-guidance is a visual tool that can tell the surgeon where to insert tools or where to drive a robot.

A commercial example of an image-guidance system is Ensite NavX (St Jude Medical, St Paul, MN, USA), which is used for cardiac mapping and ablation. NavX tracks the positions of mapping and ablation catheters through the use of electrical field generation patches placed on the patient. This is often referred to as impedance-based localization. The mapping catheter is swept along the endocardial surface of the heart and activation points are stored. After sufficiently covering the desired surfaces of the heart, a full activation map is produced that can dictate the regions of the heart that need to be ablated. During an ablation procedure, the locations of the ablation lesions can be displayed on the image-guidance visualization in their respective positions relative to the mapped surface. An example of a NavX surface model is shown in Fig. 2.6.

Carto (Bio-Sense Webster, Diamond Bar, CA, USA) is a similar commercial system that is designed for electroanatomical mapping and the guidance of EP catheters. The



http://www.sjmprofessional.com/

Figure 2.6: An example image demonstrating the result of using EnSite NavX ™ (St. Jude Medical Inc., St. Paul, Minnesota, USA).

localization task to determine the position of a catheter is somewhat different than the method used by NavX. Instead, Carto relies on the use of a magnetic field generator and magnetic sensing coils within proprietary catheters to localize tools. A supplemental software system named CartoMerge is capable of merging a live surface map generated by the swept volume of a mapping catheter with a preoperative surface reconstructed from CT scans of a patient's heart. Examples of Carto's uses are to diagnose atrial flutter [43] and to ablate activation sites for ventricular tachycardia [62]. It has been shown clinically that these image-guidance and electro-anatomical mapping systems (both NavX and Carto) can reduce a doctor's reliance on radiating fluoroscopy [21].

Other examples of image-guidance include [75], in which Wein et. al. use an EM tracker to track the tip of a catheter that is registered to preoperative CT images. The authors also fuse ultrasound data for improving surgical feedback. Another example of image-guidance is [11], in which Cleary et. al. use an electromagnetic (EM) tracker registered with preoperative images. Also, in [81], an automatic registration method is introduced for image-guidance in order to align EM tracker measurements with preoperative images using an iterative closest point (ICP) method. Unfortunately, this is a tedious task that can take on the order of 40 minutes to complete [18]. Recently, a software package has been created for researchers to design custom image-guidance solutions [24]. Lastly, an image-guidance system that combines multiple imaging modalities and tool tracking for brain surgery was introduced in [50].

2.5 Filtering for Surgical Applications

In [40], the authors estimate the pose of a stereoscope while simultaneously estimating the 3D positions of visually detected features during a minimally invasive surgical procedure. This work is, to our knowledge, the first to adapt a mobile robot simultaneous localization and mapping (SLAM) algorithm to the medical domain. The authors use a conventional extended Kalman filter (EKF) approach to jointly estimate the location of the fiberscope and the positions of features on the surface of an in-vivo porcine tissue sample.

Grasa et. al. [25] use an endoscope to perform SLAM for minimally invasive surgery. The authors also use an EKF estimator, as in [40], but instead perform monocular SLAM with undelayed initialization of visual landmarks. For their approach, an inverse-depth parameterization is used to transform the nonlinear estimation problem into a state space that behaves more linearly. Similarly, in [45], the authors use a fiberscope with a proximally mounted CCD camera to recover camera motion and 3D scene information during minimally invasive surgery.

Another example of the use of filtering for surgical estimation is the work presented in [31]. In this paper, the authors use an unscented particle filter (UPF) to localize an intracardiac echo (ICE) ultrasound catheter with measurements obtained from the ultrasound itself. The algorithm performs global localization, meaning that a prior estimate of registration is not required. The update step of this localization procedure compares the live ultrasound images with the expected image (based on a preoperative CT scan) for each particle/hypothesis in the filter. The authors demonstrate that their technique converges to an accurate localization estimate within 30 seconds for an experiment involving the left atrium.

2.6 Kalman Filtering

Filtering is the foundation of our BodySLAM approach. We are specifically applying filtering theory to perform nonlinear state estimation for systems that have noisy motion and output models. Filtering algorithms, for state estimation, typically rely on a recursive Bayesian formulation with stochastic models. For robotics problems, the state that is estimated could potentially be any parameters that are important to the estimation problem, such as the robot's position, orientation, shape, velocity, and/or almost any numerical quantity related to the task of the robot.

Typically, Bayesian filtering algorithms seek to estimate a posterior probability distribution over an unknown state vector $\mathbf{x}_k$ at time step k given state process inputs $\mathbf{u}^k$ and system outputs (measurements) $\mathbf{z}^k$. For this formulation, we use the notation that $\mathbf{z}^k$ is the entire sequence of measurements including $\{\mathbf{z}_0, \mathbf{z}_1, \mathbf{z}_2, \ \dots \ \mathbf{z}_k\}$ and $\mathbf{u}^k$ is the entire sequence of process inputs including $\{\mathbf{u}_0, \mathbf{u}_1, \mathbf{u}_2, \ \dots \ \mathbf{u}_k\}$. The probability density function (PDF) can be factored using Bayes law,

$$p(\mathbf{x}_k|\mathbf{z}^k, \mathbf{u}^k) = \eta \ p(\mathbf{z}_k|\mathbf{x}_k, \mathbf{z}^{k-1}, \mathbf{u}^k) \ p(\mathbf{x}_k|\mathbf{z}^{k-1}, \mathbf{u}^k),$$

where η is a normalization factor. This can be expanded using the law of total probability,

$$p(\mathbf{x}_k|\mathbf{z}^k, \mathbf{u}^k) = \eta \ p(\mathbf{z}_k|\mathbf{x}_k, \mathbf{z}^{k-1}, \mathbf{u}^k) \int_{\mathbf{x}_{k-1}} p(\mathbf{x}_k|\mathbf{x}_{k-1}, \mathbf{z}^{k-1}, \mathbf{u}^k) p(\mathbf{x}_{k-1}|\mathbf{z}^{k-1}, \mathbf{u}^k) \partial \mathbf{x}_{k-1},$$

which can then be simplified using independence and Markov assumptions,

$$p(\mathbf{x}_k|\mathbf{z}^k, \mathbf{u}^k) = \eta \underbrace{p(\mathbf{z}_k|\mathbf{x}_k)}_{\substack{\text{Measurement}\\ \text{Model}}} \int_{\mathbf{x}_{k-1}} \underbrace{p(\mathbf{x}_k|\mathbf{x}_{k-1}, \mathbf{u}_k)}_{\substack{\text{Process}\\ \text{Model}}} p(\mathbf{x}_{k-1}|\mathbf{z}^{k-1}, \mathbf{u}^{k-1}) \partial \mathbf{x}_{k-1}. \tag{2.1}$$

For a system with a linear measurement model and a linear process model, the Kalman filter is the optimal stochastic estimator for the state $\mathbf{x}_k$ [29]. The Kalman filter, introduced in [29, 30], parameterizes the posterior distribution with a mean vector $\hat{\mathbf{x}}_{k|k}$ and its associated covariance matrix $\mathbf{P}_{k|k}$, thus,

$$\mathbf{x}_k \sim \mathcal{N}(\hat{\mathbf{x}}_{k|k}, \mathbf{P}_{k|k}).$$

The Kalman filter has two alternating recursive steps: the prediction step and the measurement correction step, as seen in Fig. 2.7. The prediction step evolves the Gaussian distribution $(\hat{\mathbf{x}}_{k-1|k-1}, \mathbf{P}_{k-1|k-1})$ at time step k by taking the previous estimate and applying the process model,

$$\begin{aligned} \hat{\mathbf{x}}_{k|k-1} &= \mathbf{F}_k \hat{\mathbf{x}}_{k-1|k-1} + \mathbf{G}_k \mathbf{u}_k \\ \mathbf{P}_{k|k-1} &= \mathbf{F}_k \mathbf{P}_{k-1|k-1} \mathbf{F}_k^T + \mathbf{G}_k \mathbf{U}_k \mathbf{G}_k^T, \end{aligned}$$

where $\mathbf{u}_k$ is the control input, $\mathbf{F}_k$ is the state transition matrix, $\mathbf{G}_k$ is the control input matrix, and $\mathbf{U}_k$ is the covariance of the control input noise. The measurement correction step computes the Kalman gain matrix $\mathbf{K}$ that optimally corrects the predicted estimate with the information provided by the newly acquired sensor measurement $\mathbf{z}_k$,

$$\begin{aligned} \mathbf{K} &= \mathbf{P}_{k|k-1} \mathbf{H}_k^T \left(\mathbf{H}_k \mathbf{P}_{k|k-1} \mathbf{H}_k^T + \mathbf{R} \right)^{-1} \\ \hat{\mathbf{x}}_{k|k} &= \hat{\mathbf{x}}_{k|k-1} + \mathbf{K} \left(\mathbf{z}_k - \mathbf{H}_k \hat{\mathbf{x}}_{k|k-1} \right) \\ \mathbf{P}_{k|k} &= \mathbf{P}_{k|k-1} - \mathbf{K} \mathbf{H}_k \mathbf{P}_{k|k-1}, \end{aligned}$$

where $\mathbf{H}_k$ is the state measurement observation model, as follows,

$$\mathbf{z}_k \sim \mathcal{N}(\mathbf{H}_k \mathbf{x}_k, \mathbf{R}).$$

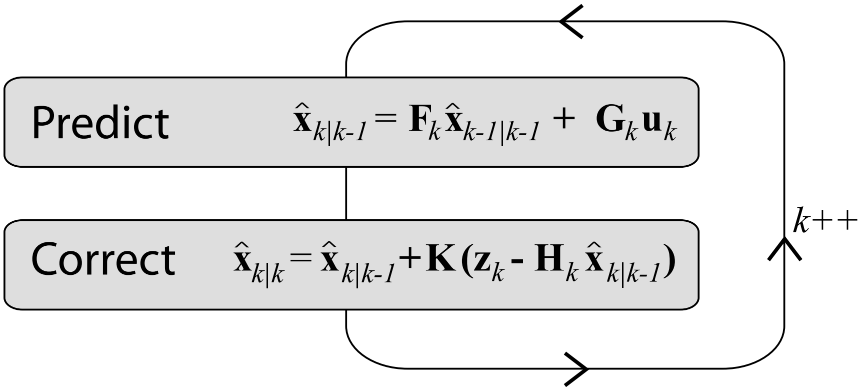

Predict
$\hat{\mathbf{x}}_{k|k-1} = \mathbf{F}_k \hat{\mathbf{x}}_{k-1|k-1} + \mathbf{G}_k \mathbf{u}_k$
Correct
$\hat{\mathbf{x}}_{k|k} = \hat{\mathbf{x}}_{k|k-1} + \mathbf{K}(\mathbf{z}_k - \mathbf{H}_k \hat{\mathbf{x}}_{k|k-1})$
k++

Figure 2.7: The Kalman filter is comprised of two stages: a prediction step and a measurement correction step.

The square matrix $\mathbf{R}$ is the modeled covariance of the measurement noise. We emphasize again that the Kalman filter is a Bayesian filter: the equations presented here can be directly derived from the Bayesian filter formulation in Eq. 2.1 (assuming Gaussian models).

Unfortunately, the Kalman filter is only optimal for linear systems. For systems with nonlinear process models and nonlinear measurement models, the popular method is to linearize the models about the current estimate and then perform a similar prediction and measurement correction step. This is called the extended Kalman filter (EKF) namely because it extends the Kalman filter algorithm to nonlinear systems. The prediction step for the EKF is written as follows,

$$\begin{aligned}
\hat{\mathbf{x}}_{k|k-1} &= \mathbf{f}(\hat{\mathbf{x}}_{k-1|k-1}, \mathbf{u}_k) \\
\mathbf{F}_k &= \left.\frac{\partial \mathbf{f}(\mathbf{x}_k, \mathbf{u}_k)}{\partial \mathbf{x}_k}\right|_{\hat{\mathbf{x}}_{k|k-1}, \mathbf{u}_k} \\
\mathbf{G}_k &= \left.\frac{\partial \mathbf{f}(\mathbf{x}_k, \mathbf{u}_k)}{\partial \mathbf{u}_k}\right|_{\hat{\mathbf{x}}_{k|k-1}, \mathbf{u}_k} \\
\mathbf{P}_{k|k-1} &= \mathbf{F}_k \mathbf{P}_{k-1|k-1} \mathbf{F}_k^T + \mathbf{G}_k \mathbf{U}_k \mathbf{G}_k^T,
\end{aligned}$$

where $\mathbf{f}(\mathbf{x}_k, \mathbf{u}_k)$ is the nonlinear state transition function. The measurement correction step for the EKF is written as follows,

$$\begin{aligned}
\mathbf{H}_k &= \left.\frac{\partial \mathbf{h}(\mathbf{x}_k)}{\partial \mathbf{x}_k}\right|_{\hat{\mathbf{x}}_{k|k-1}} \\
\mathbf{K} &= \mathbf{P}_{k|k-1}\mathbf{H}_k^T\left(\mathbf{H}_k\mathbf{P}_{k|k-1}\mathbf{H}_k^T + \mathbf{R}\right)^{-1} \\
\hat{\mathbf{x}}_{k|k} &= \hat{\mathbf{x}}_{k|k-1} + \mathbf{K}\left(\mathbf{z}_k - \mathbf{h}(\hat{\mathbf{x}}_{k|k-1})\right) \\
\mathbf{P}_{k|k} &= \mathbf{P}_{k|k-1} - \mathbf{K}\mathbf{H}_k\mathbf{P}_{k|k-1}, \qquad (2.2)
\end{aligned}$$

where $\mathbf{h}(\mathbf{x}_k)$ is the nonlinear observation model defining the mapping of a system state vector $\mathbf{x}_k$ to the system output.

In Fig. 2.8, we depict the effect of a Kalman filter update step. The ellipsoids in the figure represent the uncertainty associated with the Gaussian random variables. The ellipse for the measurement depicts the region of the state space that is probabilistically feasible given the measurement. The ellipse for the original state estimate depicts the region of the state space that is probabilistically feasible given all previous information. The Kalman filter essentially merges the two ellipsoids to produce a new updated state estimate by performing the measurement correction step of the Kalman filter. It can be seen that the size of the ellipsoid, and therefore the uncertainty of the system, is reduced upon the inclusion of the information provided by the measurement.

2.7 SLAM

To accomplish any task, a robot must be able to navigate from one point to another in its work space. And whether it be a wheeled robot operating in a factory or a surgical robot operating in and among soft tissue, the accuracy of this navigation will affect the success and outcome of the corresponding task. The idea of simultaneous localization and mapping (SLAM) is to enable accurate and efficient robot navigation via the online construction of detailed environment maps that can be used to better localize the robot.

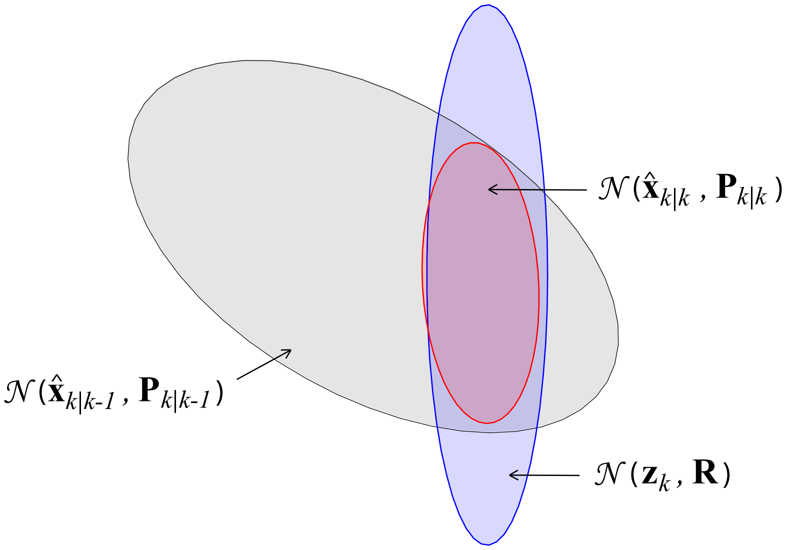

$\mathcal{N}(\hat{\mathbf{x}}_{k|k}, \mathbf{P}_{k|k})$
$\mathcal{N}(\hat{\mathbf{x}}_{k|k\text{-}1}, \mathbf{P}_{k|k\text{-}1})$
$\mathcal{N}(\mathbf{z}_k, \mathbf{R})$

Figure 2.8: In this figure, the measurement correction step of the Kalman filter produces an updated state estimate ($\hat{\mathbf{x}}_{k|k}$,$\mathbf{P}_{k|k}$) when a measurement is applied. The original ellipse is labeled ($\hat{\mathbf{x}}_{k|k-1}$,$\mathbf{P}_{k|k-1}$) and the measurement is labeled ($\mathbf{z}_k$,$\mathbf{R}$).

The seminal work by Smith and Cheeseman [60] introduced the stochastic map to the robotics community, which is thought to be the original filtering solution to SLAM. Soon afterwards, the first experimental implementation of SLAM was presented by Moutarlier in [42] and the first provably convergent solution was introduced by Dissanayake in [17]. SLAM, today, is most commonly used for wheeled mobile robot applications that combine exteroceptive sensing such as vision, sonar, or lidar with proprioceptive wheel odometry to map the positions of features or obstacles in the environment for navigational purposes [3, 17, 20, 38]. In Fig. 2.9, we show a conceptual depiction of the SLAM problem, in which a mobile robot is observing a pair of environment features from different vantage points as it travels through the environment. The observations of the features allow the robot to map their locations.

The conventional approach to SLAM is to estimate, with a Bayesian filter, a probability distribution over a joint state vector that contains both the positions of measured features/landmarks in the robot's surrounding environment as well as parameters defining the pose of the robot [17, 60]. Typically, a prediction step is used to compute the most

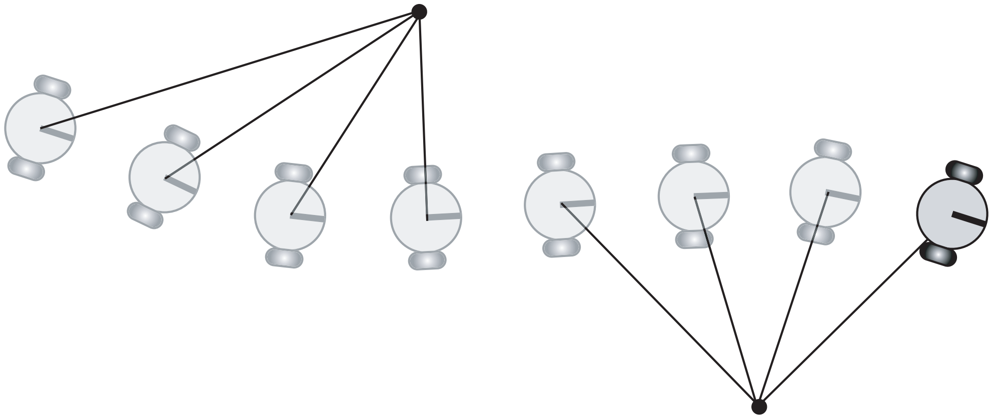


Figure 2.9: This is a conceptual SLAM depiction in which a robot is mapping two landmarks via observations from different vantage points.

likely estimate of the state based on a motion model of the robot's wheel odometry and a measurement update step is used to correct the estimate of the state based on acquired range/bearing measurements to features in the environment.

Bearing-only SLAM (monocular SLAM) is a specific variant of SLAM in which a robot is only equipped with a bearing sensor, such as a camera. In general, bearing-only SLAM can be a more challenging estimation problem due to the nonlinearity of the bearing-only measurement model and the effect of linearization. For this reason, bearing-only SLAM is susceptible to a diverging state estimate, as shown in [71]. There have been many solutions that have been introduced to counteract the divergence of bearing-only SLAM. Delaying the initialization of features is a popular method. One example is an approach that uses a batch update that stores feature observations until the landmarks can be accurately placed in the map [14]. In [2], initialization is postponed until a pair of measurements are distinguishable enough and the probability density of the corresponding landmark becomes sufficiently Gaussian. In [12], a set of particles is maintained along the viewing ray of a landmark and initialization is delayed until the range distribution is roughly Gaussian. Other methods to solve bearing-only SLAM involve multi-hypothesis Gaussian filtering [32, 61], and a modified parameterization based on inverse-depth to transform the nonlinear measurement function so that the problem behaves "more linearly" [10].

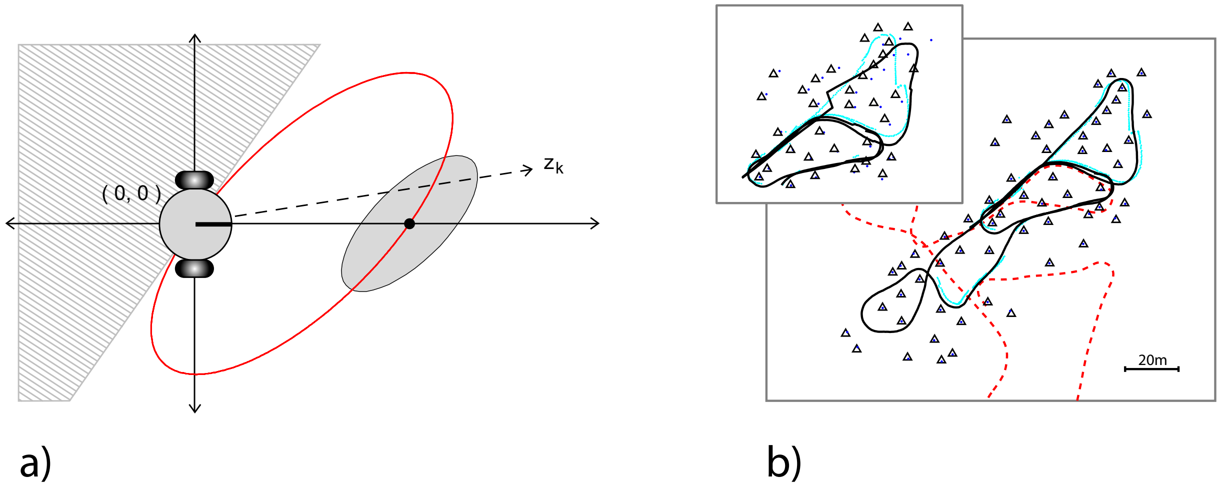


Figure 2.10: In (a), we show the manifold of possible solutions when performing a bearing-only SLAM measurement correction step. In (b), we show the result of our bearing-only SLAM approach that uses an iterative measurement correction.

In some of our previous work, we introduced novel algorithms for bearing-only SLAM [68, 71]. Our approach recognizes that the goal of the measurement correction step of the Kalman filter, at a fundamental level, is actually to find the optimum of a nonlinear cost function involving the measurement likelihood. We realized that a better approach, that can handle highly nonlinear measurement functions, is to implement an iterative approach to the Kalman measurement correction step, as in [7], which can prevent divergence of the state estimate. Furthermore, we developed a method to reduce the dimensionality of the underlying cost function for bearing-only SLAM so that the optimization problem requires only searching along a manifold in the workspace of the robot, as seen in Fig 2.10-(a). An example of the maps that can be created using bearing-only sensing is shown in Fig. 2.10-(b). While this initial work only concerned the mapping of static features rather than dynamic surface motion, it was a stepping stone that led to our dynamic feature estimation algorithm for BodySLAM.

Chapter 3

Snake Shape Estimation for Image-Guidance

For a 3D rendered image-guidance technique to be truly beneficial for surgical guidance, it must accurately portray the true state of an MIS procedure. We believe that this requires estimating the full configuration of the surgical device in order to display its position and orientation relative to anatomical structures. For snake robots, such as the highly articulated robotic probe (HARP) [15, 16], this equates to performing *shape estimation* to determine the 3D curve in which the device is configured at any given time-step. In this chapter, we will present a nonlinear filtering scheme that uses an EKF to estimate the shape of the HARP snake robot. Our filtering approach uses kinematic models and a 5-DOF pose measurement at the distal tip of the robot from an electro-magnetic (EM) tracking sensor to estimate the robot shape. The work presented in this chapter was also presented in our paper [70].

Besides the goal of achieving a representative 3D visualization for image-guidance, there are several additional reasons why it is important to perform shape estimation during an MIS procedure with a surgical snake robot. First, while it may seem sufficient to simply track the end-effector of the robot, we believe that estimating the full shape would provide more informative feedback to the surgeon: e.g., if a trajectory to a target fails due to an anatomical obstruction, viewing the full shape of the robot in relation to the anatomy would tell the surgeon how and why the intended path failed. Second, as we will discuss in detail in Chapters 4 and 5, knowing the full state of the surgical tool and its geometry can benefit registration and localization through the use of constrained filtering.

Existing methods for shape estimation include the use of Fiber Bragg Grating (FBG) sensors to determine the shape of a flexible tool. For example, in [78], the authors use an optical fiber with FBG sensing to determine, in real-time, the shape of a colonoscope. Likewise, in [79], a novel slim FBG wire is inserted into the biopsy channel of a colonoscope to determine its shape. In [49], the authors use optical FBG strain-sensors to measure the shape of a flexible needle in the field of an MRI. While this is one of the more popular methods for computing the shape of a flexible tool, there are several issues: the first is that the sensor is temperature dependent. The second issue is that the end-effector would likely be the point along the estimated shape of the tool that would have the largest error. This is undesirable due to the fact that the end-effector is the part of the tool that we care about most when localizing the robot. Our shape estimation algorithm that we present in this chapter avoids these two drawbacks.

The specific contributions of the work presented in this chapter are: 1) the novel use of an EKF to estimate the shape of a surgical snake robot with a 5-DOF pose measurement at the distal tip, 2) the introduction of process models specific to the HARP surgical snake robot, 3) an analysis of the observability of our shape estimation approach, and 4) a feasibility study of the resulting image-guidance system via an experiment involving a robot navigating semi-autonomously on the epicardial surface of a porcine heart.

3.1 Filter Formulation

To estimate the shape of the HARP during MIS procedures, we propose the use of an extended Kalman filter to recursively estimate shape parameters given kinematic models of the motion of the robot and EM tracker pose measurements at the distal tip of the probe. In this section, we will describe the filter implementation.

3.1.1 State Vector Definition

When defining the parameters that encode the shape of the snake robot, we are only concerned with the links that extend past the distal end of the robot's feeding mechanism. For this purpose, the links that are retracted within the feeding mechanism do not have any influence on the shape and configuration of the HARP. This means that not all of the links that make up the snake robot are tracked by the filter at any given point in time, as seen in Fig. 3.1. To encode the shape of the robot, we define the Kalman state vector as follows,

$$\mathbf{x}_k = \left[\ x_k^0,\ y_k^0,\ z_k^0,\ \alpha_k^0,\ \beta_k^0,\ \gamma_k^0,\ \phi_k^1,\ \theta_k^1,\ \ldots\ \phi_k^{N-1},\ \theta_k^{N-1}\ \right]^T. \tag{3.1}$$

The parameters (x_k^0, y_k^0, z_k^0) specifically define the 3D position of the first (most proximally located) tracked link of the robot. Also, $(\alpha_k^0, \beta_k^0, \gamma_k^0)$ are the yaw, pitch, and roll, respectively, of that first tracked link. Together, these 6 parameters define the 6-DOF pose of the first tracked link. In Fig. 3.1, this link is labeled $\mathbf{T}_k^0$. The terms ϕ_k^i and θ_k^i for each link i, for $i = 1toN-1$, are angle offsets that, along with the pose of the initial link (indexed $i = 0$), sufficiently encode the full shape and pose of the robot. We will discuss the mathematical definitions for ϕ_k^i and θ_k^i shortly.

The following three rotation matrices are useful when discussing the mapping of shape configuration parameters to transformation matrices that can be used to represent the pose

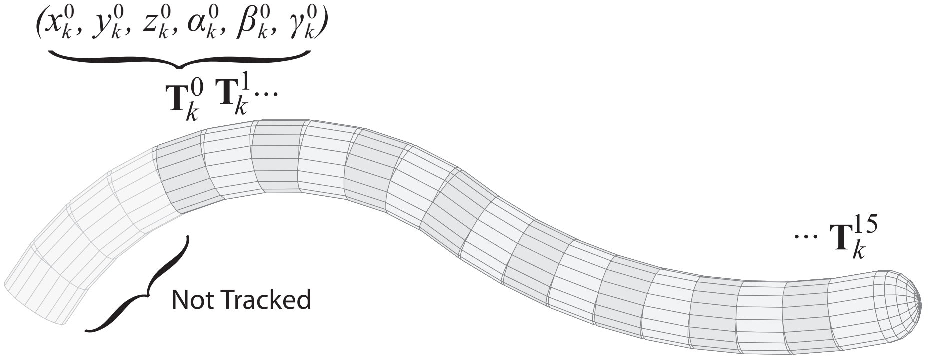

$(x_k^0, y_k^0, z_k^0, \alpha_k^0, \beta_k^0, \gamma_k^0)$
$\mathbf{T}_k^0\ \mathbf{T}_k^1\cdots$
$\cdots\ \mathbf{T}_k^{15}$
Not Tracked

Figure 3.1: A depiction of the state parameterization that we use for defining the configuration of the HARP snake robot. Transformation matrices derived from the state describe the pose of each link.

of each of the links,

$$\mathbf{R}_z(\alpha) = \begin{bmatrix} c_\alpha & -s_\alpha & 0 \\ s_\alpha & c_\alpha & 0 \\ 0 & 0 & 1 \end{bmatrix}, \quad \mathbf{R}_y(\beta) = \begin{bmatrix} c_\beta & 0 & s_\beta \\ 0 & 1 & 0 \\ -s_\beta & 0 & c_\beta \end{bmatrix}, \quad \mathbf{R}_x(\gamma) = \begin{bmatrix} 1 & 0 & 0 \\ 0 & c_\gamma & -s_\gamma \\ 0 & s_\gamma & c_\gamma \end{bmatrix}.$$

Matrix $R_z(\alpha)$ is a rotation matrix that rotates a 3-dimensional vector about the z-axis, $R_y(\beta)$ rotates a vector about the y-axis, and $R_x(\gamma)$ rotates a vector about the x-axis. For this formulation, the trigonometric notation has been simplified for convenience (i.e., $c_\alpha = \cos(\alpha)$, $s_\gamma = \sin(\gamma)$). With these rotation matrices, we can define the pose of the most proximally referenced link as a function of the Kalman state with the following transformation matrix,

$$\mathbf{T}_k^0(\mathbf{x}_k) \quad = \quad \begin{bmatrix} \mathbf{R}_z(\alpha_k^0)\mathbf{R}_y(\beta_k^0)\mathbf{R}_x(\gamma_k^0) & \mathbf{p}_k^0 \\ \mathbf{0}_{1\times 3} & 1 \end{bmatrix}, \tag{3.2}$$

where $\mathbf{p}_k^0 = (x_k^0, y_k^0, z_k^0)$ is the point vector defining the 3D position of the first proximal link.

The pose of the more distally located links are also defined by the state vector as follows: the elements ϕ_k^i and θ_k^i in the Kalman filter state definition from Eq. 3.1 are offset angles associated with link i that define link i's orientation relative to the link preceding it. More specifically, the angle θ_k^i defines the angle at which link i is oriented away from its nominal orientation and the angle ϕ_i defines the magnitude of this angle offset change. A visual interpretation of ϕ_k^i and θ_k^i can be seen in Fig. 3.2.

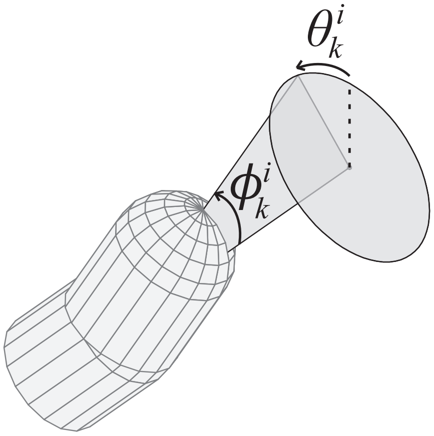


Figure 3.2: The effect of the offset angles ϕ_k^i and θ_k^i on the pose of a robot link relative to the preceding link.

To compute the transformation matrix $\mathbf{T}_k^i(\mathbf{x}_k)$ that represents the pose of link i, for $i = 1$ to $N-1$, we define the following recursive process that is initialized with the pose of the starting link,

$$\begin{aligned}
\mathbf{T}_k^{i,ang}(\mathbf{x}_k) &= \begin{bmatrix} \mathbf{R}_x(\theta_k^i)\mathbf{R}_y(\phi_k^i)\mathbf{R}_x(-\theta_k^i) & \mathbf{0}_{3\times1} \\ \mathbf{0}_{1\times3} & 1 \end{bmatrix} \\
\mathbf{T}_{adv} &= \begin{bmatrix} 1 & 0 & 0 & L \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \\
\mathbf{T}_k^i(\mathbf{x}_k) &= \mathbf{T}_k^{i-1}(\mathbf{x}_k)\mathbf{T}_k^{i,ang}(\mathbf{x}_k)\mathbf{T}_{adv},
\end{aligned}$$

where L is the length of a link. Also, the matrix T_{adv} is a transformation matrix that translates a point by a distance L along the x-axis. As seen in Fig. 3.1, we have associated a transformation matrix with each link i and each transformation matrix can be computed from the preceding link via the offsets ϕ_k^i and θ_k^i. Thus, the state vector from Eq. 3.1 sufficiently defines the pose of all links and thus the full shape and configuration of the robot.

3.1.2 State Process Models

Advancing Process Model

The HARP is a multi-link robot that is, by design, a follow-the-leader device, as discussed in Chapter 2.2. The robot maintains its shape in three-dimensional space and when commanded, advances one link-length at a time: each link theoretically moves into the corresponding pose of the link in front of it. In this case, a link behind the most proximally referenced link will move into its place and assume the role of the starting link of the Kalman filter state vector with transformation matrix $\mathbf{T}_k^0$. The way that the robot advances can be seen in Fig. 3.3.

When all of the links advance one step ahead, the state space grows by two parameters (there is effectively one extra link in the state), as seen in Fig. 3.3. While it is not physically the case, it is mathematically equivalent to view the advancing of the robot as an addition of a new link at the distal tip of the robot that assumes the same nominal orientation as the link preceding it. The motion model for this advancing step can therefore be defined with the following function,

$$\mathbf{f}_a(\mathbf{x}_k) = \left[\mathbf{x}_k^T,\ 0,\ 0\right]^T . \tag{3.3}$$

For this model, the additional parameters are set to zero because the link is advanced without steering, thus the offset angles for its orientation are zero.

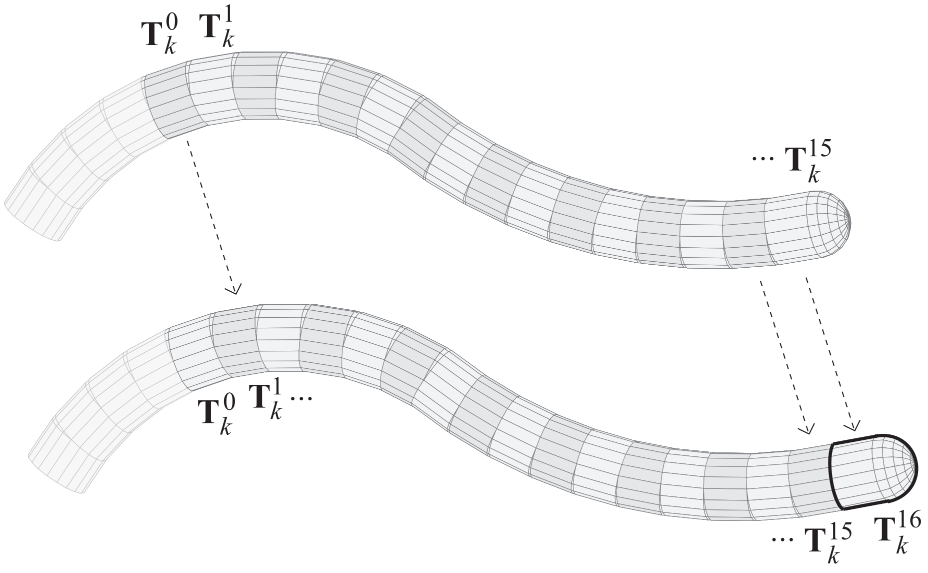


Figure 3.3: A depiction of the way that the HARP advances in a follow-the-leader fashion when commanded.

Retracting Process Model

Like advancing, when commanded to retract, the HARP maintains its shape in three-dimensional space. The most proximally referenced link moves backwards into a position that is no longer tracked by the Kalman state vector while the link that is one position ahead of the most proximally located link moves into its place and assumes the role of the starting link of the Kalman state with an associated transformation matrix $\mathbf{T}_k^0$. The distal link also theoretically moves into the position of the link preceding it thereby decreasing the number of links in the state by one. Assuming M is the length of the state vector $\mathbf{x}_k$ at time-step k, the motion model for retracting is,

$$\mathbf{f}_r(\mathbf{x}_k) = \left[\mathbf{I}_{(M-2)\times(M-2)}\ \mathbf{0}_{(M-2)\times 2}\right]\mathbf{x}_k.$$

The length of the state is reduced by two because the number of links tracked by the Kalman state is reduced by one. A depiction of the way that the HARP robot retracts is shown in Fig. 3.4.

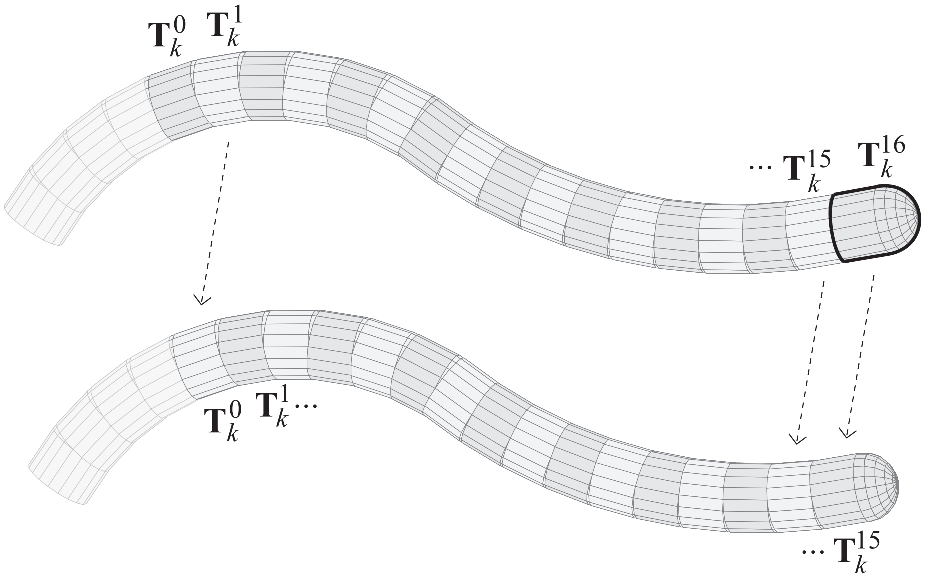


Figure 3.4: A depiction of the way that the HARP retracts in a follow-the-leader fashion when commanded.

Steering Process Model

When the HARP is in *steering mode*, all of the links preceding the distal link in the state space are restricted from moving because the inner mechanism is locked into assuming the current shape. This means that actuating the three cables that run through the entirety of the snake will theoretically control just the orientation of the distal link. Thus, by pulling on these three cables in different amounts via the tensioning of the cables by the proximally mounted motors, the pose of the distal link will change.

We have formulated a steering model that determines the angle offsets of the distal link relative to the link preceding it based on the lengths of the three actuation cables, where N is the total number of links we are tracking in the Kalman state vector at the current time step,

$$\theta_k^{N-1} = \arctan\left(\frac{\sqrt{3}(2c_2 + c_1)}{3c_1}\right) \tag{3.4}$$

$$\phi_k^{N-1} = \left| \arcsin\left(\frac{-c_1}{C_R \cos(\theta_k^{N-1})}\right) \right|. \tag{3.5}$$

For this model, C_R is a radius term that depends on the separation of the cables and (c_1, c_2) are the measured differential lengths of each of two cables running down the robot, relative to the positions that the cables were in after advancing. The value c_3 associated with the third steering cable in the robot does not appear in this model because it is geometrically a function of c_1 and c_2 and is therefore redundant information. A more detailed explanation of the steering model is provided in Appendix A.1.

From the measured cable lengths, which are obtained from encoders on the actuated motors, we can obtain the new angle offsets ϕ_k^{N-1} and θ_k^{N-1} of the distal link of the robot using Eqs. 3.4 and 3.5. We use these updated values to compute the change in angles from the previous time step, stored as $\Delta\phi_k^{N-1}$ and $\Delta\theta_k^{N-1}$, and then formulate the following motion model for steering the HARP,

$$\mathbf{f}_s(\mathbf{x}_k) = \mathbf{x}_k + \left[\mathbf{0}^T_{(M-2)\times 1},\ \Delta\phi_k^{N-1},\ \Delta\theta_k^{N-1}\right]^T, \tag{3.6}$$

where, again, M is the length of the state vector $\mathbf{x}_k$.

3.1.3 Sensor Measurement Model

The sensor that we are using to close the loop for snake shape estimation is a magnetic tracking sensor that we have situated at the distal end of the snake robot. The device we are using is the trakSTAR$^{\text{TM}}$ (Ascension Technologies, Burlington, VT, USA), which can measure the 6-DOF pose of a sensing coil in three-dimensional space. We have inserted the magnetic tracking sensor into the robot via one of the tool channels of the HARP.

While the tracker is theoretically designed for measuring the 6-DOF pose in the workspace of the robot, only 5 degrees of freedom are useful in our application. This is because the tracker must be inserted into the HARP such that it can be removed easily for the purpose of exchanging tools, and thus the sensor is free to rotate within the working channel of the robot. The measurement therefore directly observes five elements of the pose of the distal link of the robot, and we can formulate the measurement model as follows,

$$\mathbf{h}(\mathbf{x}_k) = \left[\mathbf{p}_k^{N-1\,T},\ \alpha_k^{N-1},\ \beta_k^{N-1}\right]^T, \tag{3.7}$$

where $\mathbf{p}_k^{N-1}$ is the position of the distal link which is indexed by $N-1$, and $(\alpha_k^{N-1}, \beta_k^{N-1})$ are the yaw and pitch of the distal link, respectively. All of these parameters can be extracted from the transformation matrix $\mathbf{T}_k^{N-1}(\mathbf{x}_k)$, which is computed from the state vector $\mathbf{x}_k$.

3.1.4 Kalman Filter Formulation

For this component of the thesis, we are introducing a method to estimate the state of the HARP given the measurements obtained at the distal tip by a magnetic tracker. Because we have well defined motion models and a forward kinematic measurement model, it is reasonable to formulate this filtering task in the framework of a Kalman filter (specifically an extended Kalman filter because of nonlinear models). The purpose of using a filter to estimate the state is that the motion and measurements are subject to noise and disturbances. By using a Kalman filter framework, we are making an approximation that these disturbances are Gaussian random noise.

The first step of our EKF formulation is to initialize the estimate of the state. To do this, we begin an experiment with the snake robot completely retracted with the magnetic tracker inserted into the distal link, which upon initialization also happens to be the only link in the Kalman filter state. A depiction of the state of the system is shown in Fig. 3.5-(a). In this situation, a single magnetic tracker measurement directly measures five of the six parameters defining the pose of the first and only link in the Kalman state. We can

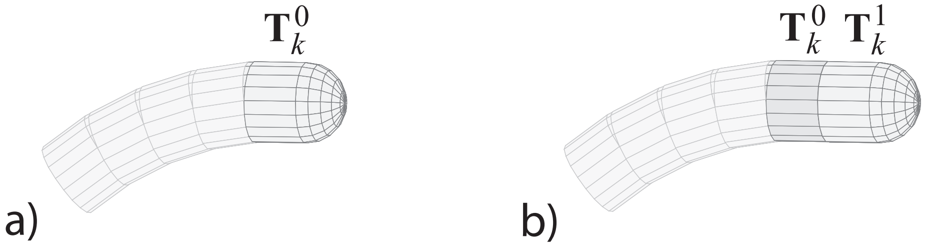


Figure 3.5: This shows the first two steps of our initialization process for the Kalman filter.

therefore initialize the mean and covariance matrix of our EKF implementation as follows,

$$\hat{\mathbf{x}}_{0|0} = \begin{bmatrix} \mathbf{z}_0 \\ 0 \end{bmatrix}, \quad \mathbf{P}_{0|0} = \begin{bmatrix} \mathbf{R} & \mathbf{0}_{5\times1} \\ \mathbf{0}_{1\times5} & \sigma_\gamma^2 \end{bmatrix},$$

where $\mathbf{z}_0$ is the initial sensor measurement which is modeled according to Eq. 3.7. The roll parameter in the initialized mean is set to zero because we do not yet have enough information to set a value for this element and thus we must initialize the roll arbitrarily. For the covariance initialization, $\mathbf{R}$ is the uncertainty in the sensor noise (a 5×5 matrix) and σ_γ^2 is a variance value chosen by the user that models the large initial uncertainty in the roll parameter of the state.

After this first measurement, we advance the robot one step and evolve the mean of the filter based on the motion model in Eq. 3.3. As for the covariance, we add a small amount of noise to represent the fact that the state parameters may be disturbed through the actuation of the cables. The state of the robot after advancing is depicted in Fig. 3.5-(b). The new predicted estimate becomes $(\hat{\mathbf{x}}_{1|0}, \mathbf{P}_{1|0})$. The reason that the robot is advanced a step before any steering takes place is for two reasons: 1) the mechanism has been designed to automatically advance a single step immediately after homing and 2) the steering model for our filtering algorithm, described by Eqs. 3.4 and 3.5, is defined for at least two links. After the robot advances, another measurement is acquired from the magnetic tracking sensor and the standard Kalman measurement update is applied using the measurement model in Eq. 3.7. The new estimate then becomes $(\hat{\mathbf{x}}_{1|1},\mathbf{P}_{1|1})$.

After this initialization procedure, there are two links in the Kalman state vector. We can then subsequently rely on the motion and measurement models defined in this section to predict and update the EKF in real-time using the well known Kalman prediction and update equations. We note that we add prediction noise (to the variances of the link angles) after each steering command in order to capture some of the effect of unmodeled disturbances. We also note that, for our EKF implementation, the prediction step that we perform at any given time-step will depend on the mode that the robot is in (e.g. steering, advancing, or retracting). The overall algorithm for our EKF implementation is described in Alg. 1.

Algorithm 1 Snake Shape Estimation Algorithm

```
$(\hat{\mathbf{x}}_{0|0}, \mathbf{P}_{0|0}) \leftarrow$ initializeStateEstimate($\mathbf{z}_0$)
$(\hat{\mathbf{x}}_{1|0}, \mathbf{P}_{1|0}) \leftarrow$ advancePredictionStep($\hat{\mathbf{x}}_{0|0}, \mathbf{P}_{0|0}$)
$(\hat{\mathbf{x}}_{1|1}, \mathbf{P}_{1|1}) \leftarrow$ measurementCorrectionStep($\hat{\mathbf{x}}_{1|0}, \mathbf{P}_{1|0}, \mathbf{z}_1$)
for $k \leftarrow 2$ to $\infty$ do
    if $mode == steer$ then
        $(\hat{\mathbf{x}}_{k|k-1}, \mathbf{P}_{k|k-1}) \leftarrow$ steerPredictionStep($\hat{\mathbf{x}}_{k-1|k-1}, \mathbf{P}_{k-1|k-1}, \Delta\theta_k^{N-1}, \Delta\phi_k^{N-1}$)
    else if $mode == advance$ then
        $(\hat{\mathbf{x}}_{k|k-1}, \mathbf{P}_{k|k-1}) \leftarrow$ advancePredictionStep($\hat{\mathbf{x}}_{k-1|k-1}, \mathbf{P}_{k-1|k-1}$)
    else
        $(\hat{\mathbf{x}}_{k|k-1}, \mathbf{P}_{k|k-1}) \leftarrow$ retractPredictionStep($\hat{\mathbf{x}}_{k-1|k-1}, \mathbf{P}_{k-1|k-1}$)
    end if
    $(\hat{\mathbf{x}}_{k|k}, \mathbf{P}_{k|k}) \leftarrow$ measurementCorrectionStep($\hat{\mathbf{x}}_{k|k-1}, \mathbf{P}_{k|k-1}, \mathbf{z}_k$)
end for
```

3.2 Observability Analysis

To perform shape estimation, we are estimating the joint angles of a high degree of freedom snake robot with only a magnetic tracker that measures the pose at the distal tip of the robot. To support our claim that this methodology is sufficient for shape estimation, we will introduce here an analysis of the observability of the filtering problem defined in Sec. 3.1.

Observability is a measure of whether the state of a system can be obtained from the system outputs (which in our case are the sensor measurements) [30]. For this analysis, we are using linearized models to perform the observability analysis, which we believe is an acceptable approximation, in this case, as it is sufficient to reveal the conditions under which successful shape estimation for the HARP robot is possible.

As discussed in Sec. 3.1, we initialize the Kalman estimate when the robot is completely retracted. At this point in time, we receive an initial sensor measurement $\mathbf{z}_0$. Based on the measurement model in Eq. 3.7, we can define the initial observability matrix $\mathbf{O}_0$ as follows,

$$\mathbf{O}_0 = \frac{\partial \mathbf{h}}{\partial \mathbf{x}_k}(\hat{\mathbf{x}}_{0|0}) = [\mathbf{I}_{5\times 5},\ \mathbf{0}_{5\times 1}] \leftarrow \text{rank}(\mathbf{O}_0) = 5,$$

which has a rank of only 5 when the length of the state vector is 6. Because the rank of the observability matrix is less than the length of the state, the system is not fully observable upon initialization. This is expected, given that the roll parameter is not observable with the initial 5-DOF measurement from the magnetic tracking sensor.

According to our initialization procedure defined in Sec. 3.1, we then advance the snake one link forward. Due to the inherent design of the snake robot, we know that the values θ_1 and ϕ_1 will be equal to zero when there is yet to be actuation from steering. Thus, for analysis purposes, we can decide to treat our knowledge of these two values as a hypothetical measurement with the following model,

$$\mathbf{h}_{adv}(\mathbf{x}_k) = [\theta_k^{N-1},\ \phi_k^{N-1}]^T. \tag{3.8}$$

A new observability matrix can then be written for time step $k = 1$,

$$\mathbf{O}_1 = \begin{bmatrix} \mathbf{O}_0 & \mathbf{0}_{5\times 2} \\ \multicolumn{2}{c}{\frac{\partial \mathbf{h}_{adv}}{\partial \mathbf{x}_k}(\hat{\mathbf{x}}_{1|1})} \end{bmatrix}$$

$$= \begin{bmatrix} \mathbf{O}_0 & \mathbf{0}_{5\times 2} \\ \mathbf{0}_{2\times 6} & \mathbf{I}_{2\times 2} \end{bmatrix} \leftarrow \text{rank}(\mathbf{O_1}) = 7.$$

At this point, the Kalman filter is estimating a state vector that has 8 parameters, but the rank of the observability matrix is only 7. Thus, the state of the system is still not observable at this point of the filtering process. It turns out that the roll of the initial link is still not observable given the measurements obtained thus far.

Finally, after the robot has been initialized and a second link has been advanced, that second link can be steered by actuating the cables of the robot. The steering motion model in Eq. 3.6 defines how the state changes when there is a steering command $\left[\Delta\phi_k^{N-1}, \Delta\theta_k^{N-1}\right]$ applied to the system. After steering, a third component can be added to the observability matrix,

$$\mathbf{O_2} = \begin{bmatrix} \mathbf{O_0} & \mathbf{0}_{5\times 2} \\ \mathbf{0}_{2\times 6} & \mathbf{I}_{2\times 2} \\ \multicolumn{2}{c}{\mathbf{H}_2\mathbf{F}_2} \end{bmatrix} \leftarrow \text{rank}(\mathbf{O_2}) = 8,$$

where $\mathbf{H}_2$ and $\mathbf{F}_2$ are linearized Jacobians,

$$\begin{aligned} \mathbf{H}_2 &= \frac{\partial \mathbf{h}}{\partial \mathbf{x}_k}(\hat{\mathbf{x}}_{2|1}) \\ \mathbf{F}_2 &= \frac{\partial \mathbf{f}_s}{\partial \mathbf{x}_k}(\hat{\mathbf{x}}_{2|1}). \end{aligned} \tag{3.9}$$

The rank of $\mathbf{O_2}$ is equal to 8 as long as $\Delta\phi_k^{N-1}$ is unequal to zero. The significance of this analysis is that as long as we steer the robot after advancing the link, we can observe the roll of the system. This is expected because by moving the distal link of the robot out of its nominal orientation, so that the link is off-axis, the direction in which the robot steers can be observed, and thus we can determine the orientation of the coordinate frame of the robot.

Lastly, once the 8 parameters that define the state of the first two links are observed, we can say with confidence that the system will continue to be observable as more links are advanced, as the Kalman state grows, and as the robot steers into new shapes. The reason that we can assume full observability in this case is that the motion models, defined above, will precisely define the evolution of the state vector based on motion inputs. In other words, once the initial pose is observed, the remainder of the configuration is all relative to that initial pose given the models, and thus is observable as well.

While this is a significant result for supporting the use of this filtering approach for shape estimation, the observability analysis we present here assumes perfect models in which the robot is a true follow-the-leader device that maintains its shape during actuation. In real life, though, the shape estimate may be affected by noise and external forces. Thus, to evaluate the realistic performance of our filtering scheme, we will discuss two experiments in the next section.

3.3 Experimental Evaluation

The first experiment that we will discuss is a bench-top test in which we drove the snake robot into an S-curve configuration by turning maximally to the right and then maximally to the left. The shape of the curve can be seen in Fig. 3.6. The magnetic tracker (trakSTAR$^{\text{TM}}$ from Ascension Technologies, Burlington, VT, USA) remained at the tip of the robot throughout the experiment and was used to update the shape of the robot with the filtering algorithm that we presented in Sec. 3.1.

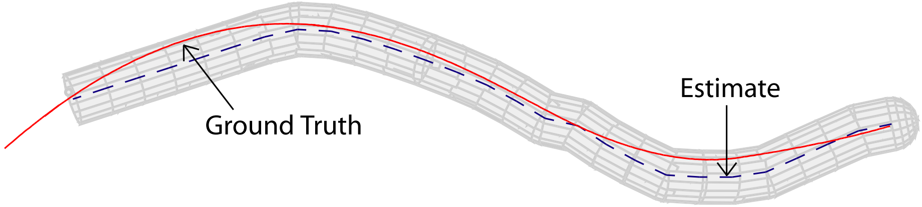


Figure 3.6: Data from an experiment we performed in the lab, for which we recorded ground truth shape data (solid line) to compare with our shape estimation algorithm. The average error for a link location was 2.98mm.

At the end of the experiment, we externally locked the shape of the snake and subsequently pulled the magnetic tracker through the working channel to record a trail of data points that we could post-process as a ground truth path. We compared our shape estimate to the ground truth with an average error of 2.98mm and a maximum error of 6.74mm between each robot link and the nearest corresponding point on the ground truth path. The result is shown in Fig. 3.6. We attribute the accuracy of this result to our estimation algorithm as well as the robot's inherent ability to preserve its shape as a follow-the-leader device. We note that, while this is a promising outcome, we believe the algorithm has the potential for more accurate estimation. With additional tuning of noise parameters the performance could be improved.

The second experiment we performed involved a live animal experiment in which we tested our own image-guidance software while navigating the HARP robot along the epicardial surface of a porcine subject. One of the goals of this experiment was to investigate the feasibility of performing epicardial ablation with the HARP robot with a subxiphoid approach. In Fig. 1.1-(b), we show the HARP next to the single-port incision that is typical for experiments in which the robot accesses the epicardial space. We note that the photo in Fig. 1.1-(b) was actually taken during a different experiment than the one in which we tested our shape estimation algorithm. The surface models for the anatomical structures shown in Fig. 3.7 were obtained by performing image segmentation on a series of CT images obtained preoperatively. The coordinate frame of the robot was registered to the coordinate frame of the preoperative CT images through the use of fiducial markers attached to the skin of the porcine subject. By touching the 9 fiducial markers with the magnetic tracking coil, we were able to acquire 3D points in the coordinate frame of the robot that could be associated with known points in the CT scans marking the locations of the fiducials. These point correspondences were sufficient to compute the transformation for registration.

For this experiment, we tested our shape estimation algorithm and show the qualitative result in Fig. 3.7. Although we unfortunately do not have ground truth data for this experiment, we have reason to believe that the shape estimation filtering algorithm performed well: the shape estimate appears to be accurate given that the result shows the robot correctly configured in just the right shape so as to lie above the surface of the heart while also lying beneath the rib cage of the porcine subject.

During the animal experiment, we also operated the robot semi-autonomously, which meant that the robot steered itself along a prescribed path to a target location. Seeing the robot drive itself along the beating heart demonstrated the efficacy of our approach. The prescribed path that was navigated autonomously was chosen during the live experiment via a user interface that we wrote to accept mouse inputs for defining a projected path on a loaded surface model. While the robot did not follow the path perfectly, it successfully reached the final goal location with minimal error. We believe that this result is significant due to the fact that the feedback controller that we implemented for guiding the robot along a prescribed path is dependent upon the shape estimation algorithm that we have presented in this chapter. Snapshots of the image-guidance visualization during this semi-autonomous trial are shown in Fig. 3.8.

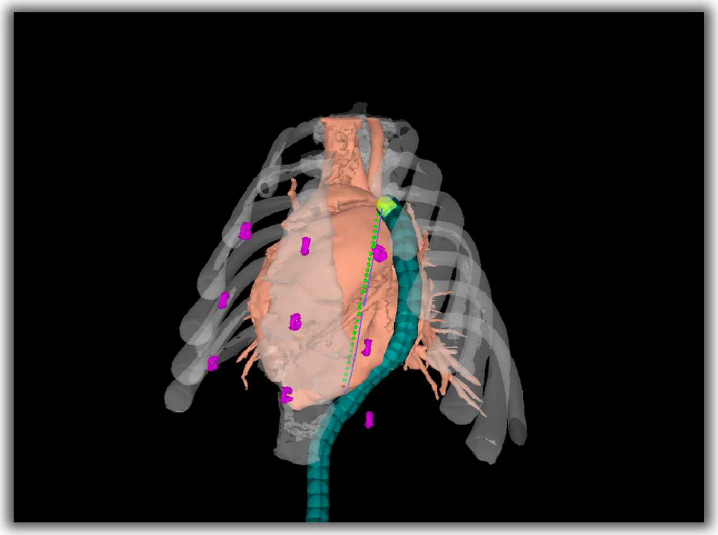


Figure 3.7: This is a result from an experiment in which we tested our shape estimation algorithm on a porcine model.

3.4 Discussion

The contribution of the work presented in this chapter is a novel filtering method for estimating in real-time the shape and pose of a highly articulated surgical snake robot. Our algorithm combines new kinematic models of the motion of the robot with measurements from a magnetic tracking sensor in a custom EKF framework. We have shown that the system is fully observable under appropriate motion, which supports the use of this algorithm in experiments. The advantage of the proposed approach is that we can leverage existing magnetic tracking technology that we would most likely already be using (to track the position of the distal tip) to also determine the full configuration of the snake robot.

We have shown promising results, both with bench-top testing and animal experiments. In one experiment, the HARP was driven semi-autonomously along a predefined path using feedback from our EKF implementation. This is an exciting result, for it demonstrates both the capabilities of the robot as well as the ability of our algorithm to accurately estimate the configuration of the robot in real-time.

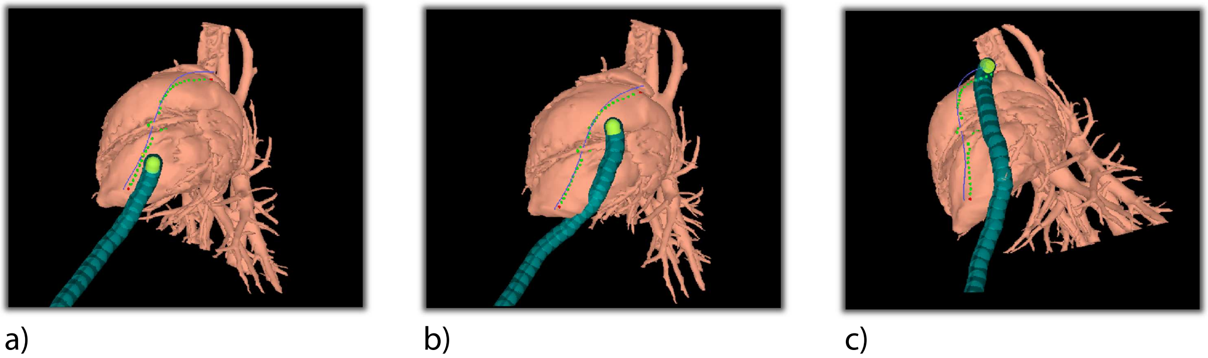


Figure 3.8: This is a result from an experiment in which we semi-autonomously navigated the HARP along the epicardial surface of a porcine heart and used our shape estimation algorithm for visual feedback.

The most significant benefit of shape estimation for image-guidance is the information that it can potentially provide for registering and localizing the robot relative to anatomical structures. In the next two chapters, we will introduce novel constrained filtering algorithms that will use the geometry of the robot shape along with surface models of the surrounding organs to eliminate infeasible robot states from a stochastic estimator.

Chapter 4

Inequality Constrained Surgical Estimation

In Chapter 3, we discussed the use of Kalman filtering to recursively estimate the shape and pose of a highly articulated surgical robot for the purpose of displaying a rendered model of the robot relative to preoperative surface models within an image-guidance system. The result showed the efficacy of our approach and the benefit of filtering for surgical estimation.

Unfortunately, the registration process is not perfect. There might be error in the alignment of the robot to the coordinate frame of the preoperative images. On top of that, the shape estimation task, while it is a fully observable filtering problem, can be susceptible to noise and environment disturbances. Due to the inherent inaccuracies associated with image-guidance and our filtering approach, a surgical tool may appear in a location that is completely unrealistic, for example inside of a cardiac surface model when the intervention is performed outside of the heart. Such errors are not desirable during a live operation. Fig. 4.1 shows an example of exactly this situation from an experiment that we performed with the HARP robot on a porcine model.

In this chapter, we will introduce a novel inequality constrained filtering approach to

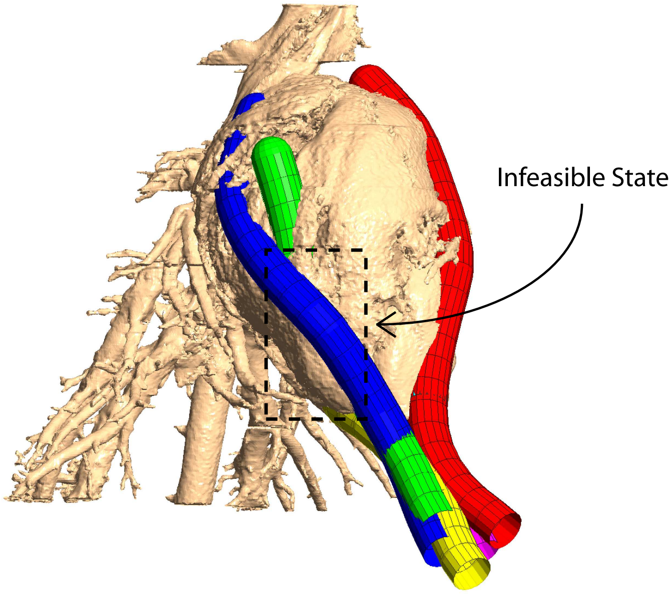


Figure 4.1: Our image-guidance system is shown here with multiple paths drawn for the robot. In this image, the robot is shown intersecting the surface model and violating geometric constraints.

correct the Kalman filter estimate of the robot shape and pose when the robot is shown piercing through a surface model during image-guidance. Our approach recursively corrects registration and localization parameters via the incorporation of geometric constraints within a new update step of the Kalman filter. The new filtering scheme can be seen in Fig. 4.2, in which a constrained filtering step takes as an input the result from the Kalman filter measurement update. This algorithm is applicable to image-guided surgery because anatomical surface models can indirectly provide information about infeasible robot configurations that we can formulate as constraints when estimating the state of the system in a filtering framework. The work presented in this chapter was also presented in our paper [69].

In previous literature, constrained filtering has been defined as the problem of correcting or constraining the Kalman update step or the Kalman prediction step to account for known constraints on the state vector [57]. In [58], Simon and Chia present a method for *equality* constrained Kalman filtering in which linear constraints are incorporated into the Kalman

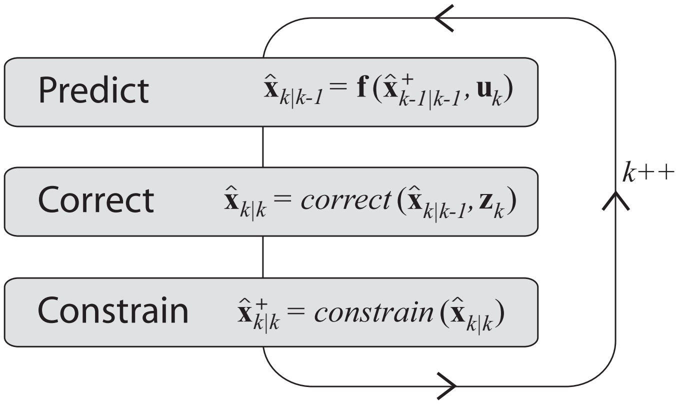

Predict
$\hat{\mathbf{x}}_{k|k\text{-}1} = \mathbf{f}(\hat{\mathbf{x}}^{+}_{k\text{-}1|k\text{-}1}, \mathbf{u}_k)$
Correct
$\hat{\mathbf{x}}_{k|k} = correct(\hat{\mathbf{x}}_{k|k\text{-}1}, \mathbf{z}_k)$
Constrain
$\hat{\mathbf{x}}^{+}_{k|k} = constrain(\hat{\mathbf{x}}_{k|k})$
k++

Figure 4.2: A flow chart for our constrained filtering estimation scheme is depicted.

filter update by projecting the unconstrained estimate onto the constraint hyperplane. Another popular method is to use a pseudo-measurement approach that simulates a perfect measurement to enforce equality constraints on the Kalman filter update step [1, 63, 74]. In [26, 64], it was shown that the pseudo-measurement update method is actually equivalent to the state projection method from [58].

There are also several existing approaches to the inequality constrained Kaman filtering problem [55, 59, 65, 72]. One method for inequality constrained filtering is to simply apply one of the aforementioned equality constrained filtering algorithms in order to shift the estimate back into a feasible state. This process, though, would not necessarily place the mean of the filter near the true mean of the feasible region in the state space. Instead, several researchers suggest the use of a PDF truncation method that updates the filter with the mean and covariance of the region of the PDF that lies outside of the constraint hyperplane [55, 59].

The contributions of the work presented in this chapter are: 1) the novel use of inequality constrained filtering to correct the localization and registration of a surgical robot, 2) a new constrained filtering algorithm called *uncertainty projection*, 3) a pseudo-measurement update method to correct the state estimate of a nonlinear system after performing uncertainty projection, and 4) an experimental result that demonstrates the impact this methodology can have in improving the quality of surgical feedback during MIS.

4.1 Uncertainty Projection for Linear Constraints

4.1.1 Problem Definition

Inequality constrained filtering seeks to eliminate infeasible states from the PDF of a Kalman filter, which can improve the accuracy of the state estimate and reduce uncertainty. In this section, we will introduce a new algorithm that we call *uncertainty projection* that can be used to update the PDF of a Gaussian random vector given a set of *linear* constraints. In Sec. 4.2, we will then extend our solution to nonlinear constraints.

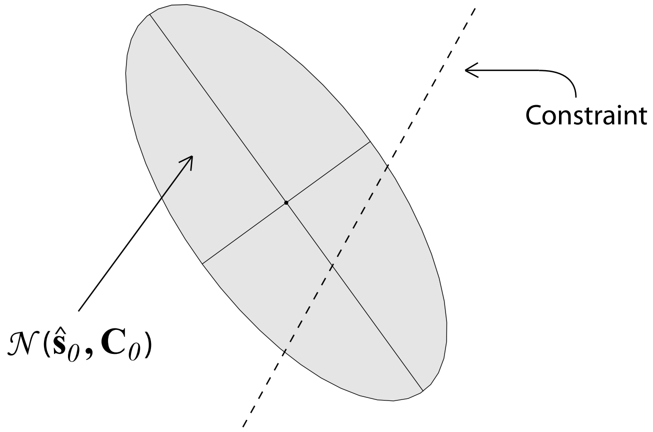


Figure 4.3: A Gaussian random vector, shown with an ellipsoid representing the uncertainty of the estimate, is constrained such that the true state must lie on one side of a hyperplane.

Suppose we have a prior PDF of an $M\times1$ random vector $\mathbf{s}$ parameterized by a Gaussian distribution with mean $\hat{\mathbf{s}}_0$ and covariance $\mathbf{C}_0$, thus,

$$\mathbf{s} \sim \mathcal{N}(\hat{\mathbf{s}}_0, \mathbf{C}_0).$$

Also, as in [59], suppose we have a set of L linear constraints that define feasible states,

$$\boldsymbol{\phi}_i{}^T\mathbf{s} \;\geq\; \alpha_i \qquad i \;=\; 1 \;...\; L, \tag{4.1}$$

where each α_i is a scalar and each $\boldsymbol{\phi}_i$ is an $M\times 1$ vector. We note that, if desired, a "less-than" constraint can be defined in the above formulation by negating the $\boldsymbol{\phi}_i$ and α_i terms. Each constraint equation can be viewed as a hyperplane in the state space that divides the space between feasible states and infeasible states, as seen in Fig. 4.3. This constraint information can either be known *a priori* or determined online during an experiment.

The objective of our uncertainty projection algorithm is to update the state estimate $(\hat{\mathbf{s}}_0, \mathbf{C}_0)$ given the inequality constraints defined in Eq. 4.1. Unfortunately, there is no efficient way to apply all of the constraints together. Thus, as in [59], our formulation will apply each constraint in Eq. 4.1 one-by-one in sequence: thus the estimate $(\hat{\mathbf{s}}_i, \mathbf{C}_i)$ obtained after applying the i-th constraint will be used as the new prior when applying the $(i+1)$-th constraint. The result of our algorithm will be a new estimate $(\hat{\mathbf{s}}_L, \mathbf{C}_L)$ that will better approximate the true PDF given the constraints.

4.1.2 Algorithm

In order to compute $(\hat{\mathbf{s}}_i, \mathbf{C}_i)$ given the estimate $(\hat{\mathbf{s}}_{i-1}, \mathbf{C}_{i-1})$ and the associated i-th constraint defined in Eq. 4.1, the first step of our method is to transform the state space in such a way that the constraint is decoupled from all elements of the transformed state vector except for the first element. This method is similarly used in [55, 59] for PDF truncation.

First we can perform a Jordan canonical decomposition of the matrix $\mathbf{C}_{i-1}$, which essentially decomposes the square matrix as follows,

$$\mathbf{C}_{i-1} = \mathbf{T}\mathbf{W}\mathbf{T}^T. \tag{4.2}$$

Then we can use Gram-Schmidt orthogonalization, as in [39, 59],

$$\mathbf{V}\mathbf{W}^{1/2}\mathbf{T}^T\boldsymbol{\phi}_i = \begin{bmatrix} (\boldsymbol{\phi}_i^T\mathbf{C}_{i-1}\boldsymbol{\phi}_i)^{1/2} & 0 & \dots & 0 \end{bmatrix},$$

to obtain the matrix $\mathbf{V}$ which will serve as a rotation matrix to align the constraint to only one dimension of the transformed state space. To transform the state space to decouple the constraint, we define the following transformed random vector,

$$\begin{aligned} \mathbf{y} &= \mathbf{V}\mathbf{W}^{-1/2}\mathbf{T}^{T}\left(\mathbf{s} - \hat{\mathbf{s}}_{i\text{-}1}\right), \\ \mathbf{y} &\sim \mathcal{N}(\mathbf{0}_{Mx1}, \mathbf{I}_{MxM}). \end{aligned} \tag{4.3}$$

The significance of this transformation is that 1) $\mathbf{y}$ is both zero mean and white and 2) the constraint only applies to the first element of $\mathbf{y}$. This means that all elements of $\mathbf{y}$ are statistically independent of each other and thus we only need to worry about updating the PDF of the first element of $\mathbf{y}$, simply because it is the only element that is constrained. A new constraint equation can thus be written,

$$\begin{aligned} \begin{bmatrix} 1 & 0 & \ldots & 0 \end{bmatrix} \mathbf{y} \geq \beta_i \\ \beta_i = \tfrac{\alpha_i - \boldsymbol{\phi}_i^T \hat{\mathbf{s}}_{i\text{-}1}}{(\boldsymbol{\phi}_i^T \mathbf{C}_{i\text{-}1} \boldsymbol{\phi}_i)^{1/2}}. \end{aligned} \tag{4.4}$$

An example of the transformation performed on the original estimate ($\hat{\mathbf{s}}_{i\text{-}1}$, $\mathbf{C}_{i\text{-}1}$) to decouple the noise is shown in Fig. 4.4. In this figure, the dotted line in Fig. 4.4-(a) represents a geometric constraint that dictates the feasible region of the state space: the state must be below (and to the right) of this line. In Fig. 4.4-(b), the state space has been transformed so that the uncertainty of the state is zero mean and white and the constraint only applies to the x-dimension of the state.

Because the first element of $\mathbf{y}$ is the only constrained element of this transformed problem, we have reduced multivariate constrained filtering to a simpler PDF update problem for a scalar Gaussian random variable (with mean equal to zero and variance equal to one) that is subject to a scalar constraint. The constraint determines which values are feasible for that scalar random variable and which are not (usually one tail of the Gaussian distribution is deemed infeasible).

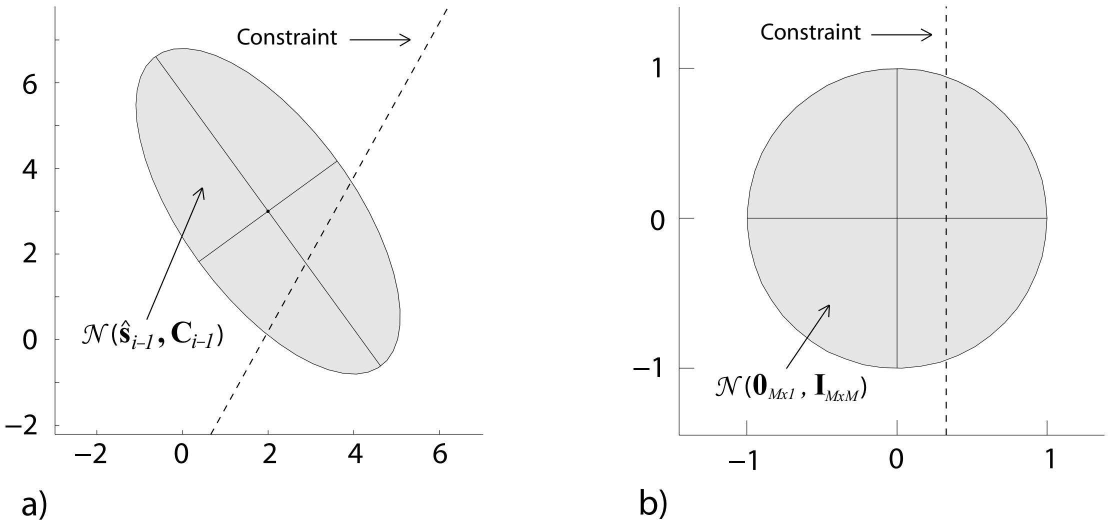

Constraint
6
4
2
0
−2
−2
0
2
4
6
$\mathcal{N}(\hat{\mathbf{s}}_{i\text{-}1}, \mathbf{C}_{i\text{-}1})$
a)
Constraint
1
0
−1
−1
0
1
$\mathcal{N}(\mathbf{0}_{Mx1}, \mathbf{I}_{MxM})$
b)

Figure 4.4: Shown here is an example of the transformation that decouples the constraint so that it only applies to a single dimension of the state space. In (a), the original estimate is shown with its associated covariance matrix and in (b) the new transformed estimate is shown.

To update this one-dimensional PDF, we project the portion of the probability mass that is less than the constraint onto the constraint itself and then compute the new mean and variance, see Fig. 4.5. The motivation for computing the new estimate this way is that it best preserves the contribution of the probability mass that is less than the constraint. The solution that we have derived, that projects the uncertainty and computes a new mean and variance, μ_i and σ_i^2, is as follows,

$$\begin{aligned}
\gamma_i &= -2\pi - 2\beta_i e^{\frac{-\beta_i^2}{2}} \sqrt{2\pi} \\
\mu_i &= \frac{1}{\sqrt{2\pi}} e^{\frac{-\beta_i^2}{2}} + \frac{1}{2}\beta_i \left[1 + \text{Erf}\left(\frac{\beta_i}{\sqrt{2}}\right)\right] \\
\sigma_i^2 &= \frac{1}{4\pi}\left[-2e^{-\beta_i^2} + (2+\beta_i^2)\pi + \gamma_i \text{Erf}\left(\frac{\beta_i}{\sqrt{2}}\right) - \beta_i^2 \pi \text{Erf}\left(\frac{\beta_i}{\sqrt{2}}\right)^2 \right].
\end{aligned} \tag{4.5}$$

An explanation of this derivation is presented in Appendix A.2.1. We can then transform this uncertainty projected mean and variance back into the original state space by reversing the transformation that originally whitened the noise to decouple the elements of the state

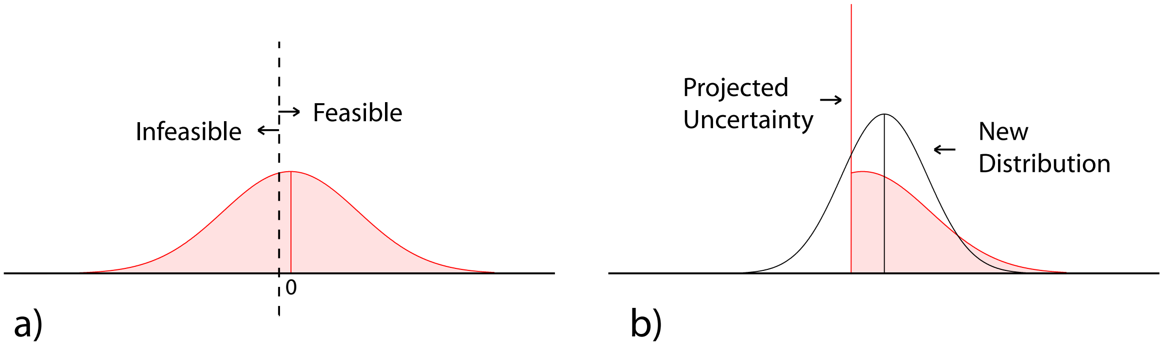


Figure 4.5: In (a), a normally distributed random variable is shown with a scalar constraint that defines the feasible and infeasible regions. In (b), we show our uncertainty projection method which projects the tail of the Gaussian onto the constraint and then computes the new mean and variance.

vector,

$$\begin{aligned}
\hat{\mathbf{y}}_i &= \begin{bmatrix} \mu_i & 0 & ... & 0 \end{bmatrix}^T \\
\mathbf{\Sigma}_i &= \text{diag}(\sigma_i^2, 1, \ ... \ , 1) \\
\hat{\mathbf{s}}_i &= \mathbf{T}\mathbf{W}^{1/2}\mathbf{V}^T\hat{\mathbf{y}}_i + \hat{\mathbf{s}}_{i-1} \\
\mathbf{C}_i &= \mathbf{T}\mathbf{W}^{1/2}\mathbf{V}^T\mathbf{\Sigma}_i\mathbf{V}\mathbf{W}^{1/2}\mathbf{T}^T.
\end{aligned}$$

An example showing the constrained estimate produced by our uncertainty projection approach and the new estimate transformed back into the original state space is shown in Fig. 4.6 (a) and (b). In summary, through the update process above, the mean $\hat{\mathbf{s}}_i$ and covariance $\mathbf{C}_i$ are updated recursively from $(\hat{\mathbf{s}}_{i-1}, \mathbf{C}_{i-1})$ given the i-th constraint equation in Eq. 4.1. It can be shown (see Appendix A.2.2) that the uncertainty in the estimate is guaranteed to be reduced when adopting our inequality constrained uncertainty projection approach.

The significance of the example in Fig. 4.6 is that the resulting uncertainty ellipse qualitatively bounds the region of the PDF that is deemed feasible by the constraint equation while successfully discarding the majority of the infeasible region of the PDF. We note that our efficient algorithm can be used for any set of linear constraints imposed

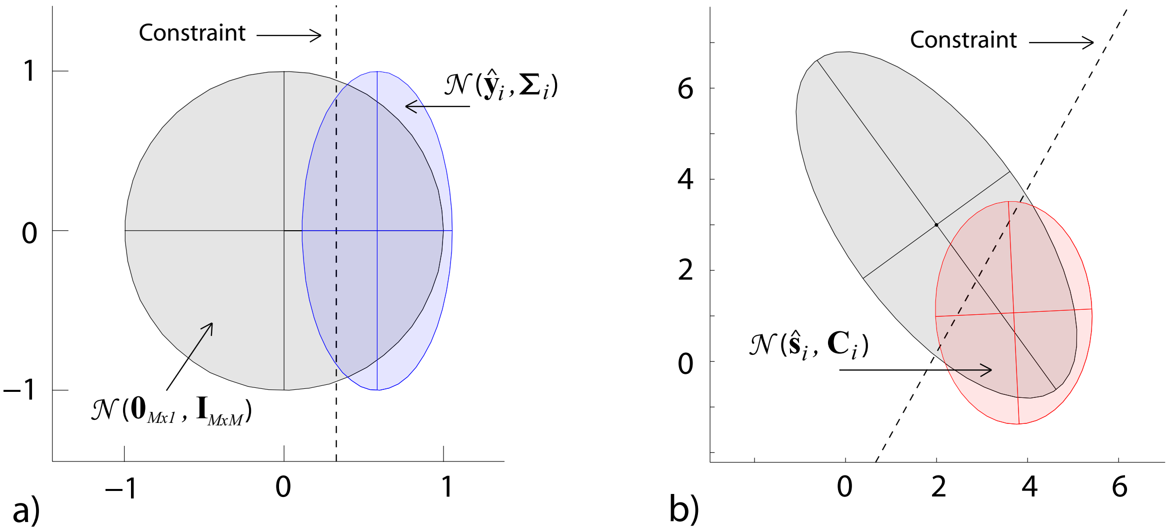


Figure 4.6: Shown here is an example of the uncertainty projection method we use for updating the mean and variance. In (a) we show how a new estimate is generated using uncertainty projection in the transformed state space. Then, in (b), we show the result when the estimate is transformed back into the original state space.

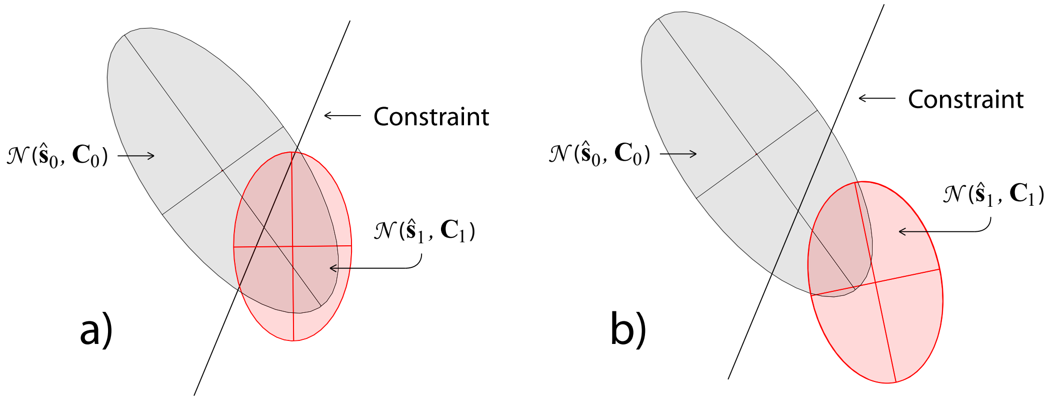
on a multidimensional Gaussian distribution. Our uncertainty projection algorithm is summarized in Alg. 2.

Algorithm 2 Uncertainy Projection for Linear Constraints

```
($\hat{\mathbf{s}}_0$, $\mathbf{C}_0$) ← LoadInitialEstimate()
for $i \leftarrow 1$ to $L$ do
    ($\mathbf{T}$, $\mathbf{W}$, $\mathbf{T}^T$) ← JordanDecomposition($\mathbf{C}_{i-1}$)
    ($\mathbf{V}$) ← GramSchmidtOrthogonalization($\mathbf{W}$,$\mathbf{T}$,$\boldsymbol{\phi}_i$,$\mathbf{C}_{i-1}$)
    $\beta_i \leftarrow \left(\alpha_i - \boldsymbol{\phi}_i^T \hat{\mathbf{s}}_{i-1}\right) / \left(\boldsymbol{\phi}_i^T \mathbf{C}_{i-1} \boldsymbol{\phi}_i\right)^{1/2}$
    ($\mu_i$, $\sigma_i^2$) ← ComputeUncertaintyProjection($\beta_i$)
    ($\hat{\mathbf{s}}_i$, $\mathbf{C}_i$) ← TransformBackToState($\mu_i$,$\sigma_i^2$,$\mathbf{T}$,$\mathbf{W}$,$\mathbf{V}$, $\hat{\mathbf{s}}_{i-1}$)
end for
```

4.1.3 Empirical Evaluation

Before we discuss actual experimental results related to our approach, it is meaningful to look at the theoretical evidence supporting the use of our uncertainty projection method within a constrained filtering update step. In Fig. 4.7, we show the result that is obtained when constraining a 2D Gaussian random vector by a linear constraint. In (a), our solution is provided, which demonstrates that the new estimate qualitatively bounds the feasible region of the original Gaussian ellipsoid. This means that the constrained update is properly fitting a new Gaussian given the constraint. In (b), the same problem is shown but the resulting ellipse for this figure was instead computed by performing the PDF truncation method that was introduced in [59]. It can be seen that this result does not fit as well to the feasible region of the constrained Gaussian ellipsoid. For this comparison, 1-sigma ellipsoids have been generated.



$\mathcal{N}(\hat{\mathbf{s}}_0, \mathbf{C}_0)$
Constraint
$\mathcal{N}(\hat{\mathbf{s}}_1, \mathbf{C}_1)$
a)
$\mathcal{N}(\hat{\mathbf{s}}_0, \mathbf{C}_0)$
Constraint
$\mathcal{N}(\hat{\mathbf{s}}_1, \mathbf{C}_1)$
b)

Figure 4.7: This is a comparison between our uncertainty projection method (a) versus PDF truncation (b) for a single constraint.

In Fig. 4.8, we show another comparison between our approach and the truncation method introduced in [59]. For this empirical evaluation, the original Gaussian was constrained by two different hyperplanes. The feasible region is shown to be between the two constraints. The Gaussian ellipsoids in this figure are shown with 1-sigma uncertainty ellipses. In Fig. 4.8-(a), our method is shown performing well in terms of bounding the feasible region while in (b), the truncation method is shown with a result that is undesirably shifted away from the feasible region.

For constrained filtering, the ordering that the constraints are applied can change the resulting Gaussian that is fit to the feasible region. To compare the performance, we again can refer to Fig. 4.8. In (a), when using our uncertainty projection algorithm, it can be seen that the ordering of the constraints does not significantly affect the outcome of the constrained filtering algorithm, which is desirable. In (b), though, it is shown that the ordering of the constraint can have a larger affect when using the PDF truncation method introduced in [59].

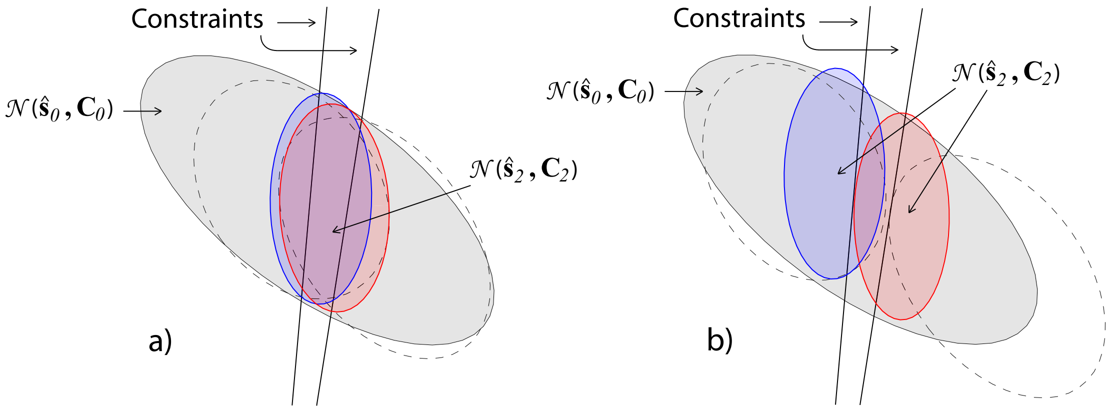

Constraints
$\mathcal{N}(\hat{\mathbf{s}}_0, \mathbf{C}_0)$
$\mathcal{N}(\hat{\mathbf{s}}_2, \mathbf{C}_2)$
a)
Constraints
$\mathcal{N}(\hat{\mathbf{s}}_0, \mathbf{C}_0)$
$\mathcal{N}(\hat{\mathbf{s}}_2, \mathbf{C}_2)$
b)

Figure 4.8: This is a comparison between our uncertainty projection method (a) versus PDF truncation (b) for a sequence of two constraints.

4.1.4 Repeated Application of Constraints

Our approach to inequality constrained filtering, as is the case with other existing solutions, is susceptible to a problem in which the repeated application of the same inequality constraint will undesirably drive the covariance of a system estimate to zero over time. The issue arises because the filter is effectively treating each constraint update as an independent source of information. Thus, applying the same or similar constraint will artificially shrink the covariance resulting in an overconfidence in the state estimate. An example is shown in Fig. 4.9. In (a), a result is shown in which a single constraint is applied to a Gaussian random variable using our uncertainty projection method. In (b), the same constraint is applied 10 times which has significantly reduced the uncertainty of the system. This is undesirable given that no additional information has been gained upon the repeated application of the constraint.

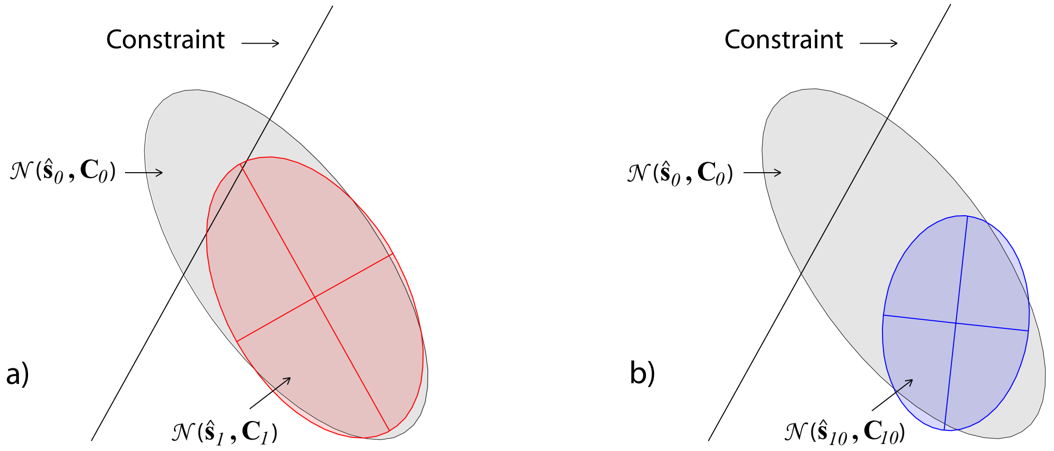

Constraint
$\mathcal{N}(\hat{\mathbf{s}}_0, \mathbf{C}_0)$
a)
$\mathcal{N}(\hat{\mathbf{s}}_1, \mathbf{C}_1)$
Constraint
$\mathcal{N}(\hat{\mathbf{s}}_0, \mathbf{C}_0)$
b)
$\mathcal{N}(\hat{\mathbf{s}}_{10}, \mathbf{C}_{10})$

Figure 4.9: This is the effect of repeating a constraint update over time. In (a), the effect of a single constraint update is shown. In (b), the same constraint is applied 10 times, which demonstrates the problem of overconfidence.

Our solution to this problem is to only apply a constrained update to a state estimate if the mean of that estimate lies in the infeasible region. This is a sufficient solution to the repeated constraint problem because once a single constraint is applied once, it is guaranteed to force the mean of the estimate into the feasible region. This means that there is no inherent risk of applying the same constraint twice. In Appendix A.2.3, we provide a proof that the update procedure, when applied just once, will shift the mean to a feasible state thus avoiding this repeated constraint problem.

4.2 Pseudo-Measurement Update for Nonlinear Constraints

In this section, we will introduce a second algorithm that will extend our uncertainty projection algorithm to systems with *nonlinear* constraints, which is particularly applicable to our surgical application.

When a Kalman filter or nonlinear filter is used to estimate an unknown state $\mathbf{x}_k$

via the recursive computation of a mean vector $\hat{\mathbf{x}}_{k|k}$ and covariance matrix $\mathbf{P}_{k|k}$, *and* the constraints are linear, we can use the algorithm to apply an update for the constraints. When the constraints are nonlinear, though, which will be the case for our desired application, BodySLAM, we must first project the uncertainty of the state into what we call the *constraint-space* (where the constraints can be applied linearly), and then apply the update algorithm.

4.2.1 Problem Definition

Suppose that a Kalman filter is being used to estimate the state of a system with an $N \times 1$ state vector $\mathbf{x}_k$. After each prediction step, the Kalman filter produces an estimate $(\hat{\mathbf{x}}_{k|k-1}, \mathbf{P}_{k|k-1})$ and after each measurement correction step, the Kalman filter produces an estimate $(\hat{\mathbf{x}}_{k|k}, \mathbf{P}_{k|k})$. The constrained filtering problem that we define here is to update the estimate $(\hat{\mathbf{x}}_{k|k}, \mathbf{P}_{k|k})$ given a set of L constraints. This is similar to our formulation in Sec. 4.1, except suppose now that the constraints are nonlinear,

$$\boldsymbol{\phi}_i{}^T \mathbf{a}(\mathbf{x}_k) \;\geq\; \alpha_i \qquad i \;=\; 1\;,\;\ldots\;,\;L. \tag{4.6}$$

In this formulation, we assume that a known function $\mathbf{a}(\mathbf{x}_k)$ exists that will map the state vector to a space in which the constraints are linear, which we call, from here on, the *constraint-space*. For many applications, the constraint-space will be the workspace of the robot, for this is where the system will be limited by geometric constraints imposed by the robot's surrounding environment.

4.2.2 Algorithm

We can find the Jacobian of the function $\mathbf{a}(\mathbf{x}_k)$ by linearizing about the current estimate,

$$\mathbf{A}_k = \left.\frac{\partial \mathbf{a}(\mathbf{x}_k)}{\partial \mathbf{x}_k}\right|_{\hat{\mathbf{x}}_{k|k}}$$

The first step in our constrained filtering algorithm is to project the Kalman filter estimate $(\hat{\mathbf{x}}_{k|k}, \mathbf{P}_{k|k})$ into the constraint-space with the following projection,

$$\hat{\mathbf{s}}_0 = \mathbf{a}(\hat{\mathbf{x}}_{k|k}), \quad \mathbf{C}_0 = \mathbf{A}_k \mathbf{P}_{k|k} \mathbf{A}_k^T,$$

where $\mathbf{A}_k$ is the Jacobian of the nonlinear function $\mathbf{a}(\mathbf{x}_k)$ linearized about the current estimate $\hat{\mathbf{x}}_{k|k}$.

After performing this projection, the problem has been cast into a form that is solvable with our uncertainty projection algorithm for linear constraints. After substituting the estimate $(\hat{\mathbf{s}}_0, \mathbf{C}_0)$ into our uncertainty projection algorithm along with the constraints, the result would be the updated estimate $(\hat{\mathbf{s}}_L, \mathbf{C}_L)$. This result still lies in the constraint-space, but now has reduced uncertainty and is likely a better estimate of the PDF given the constraints.

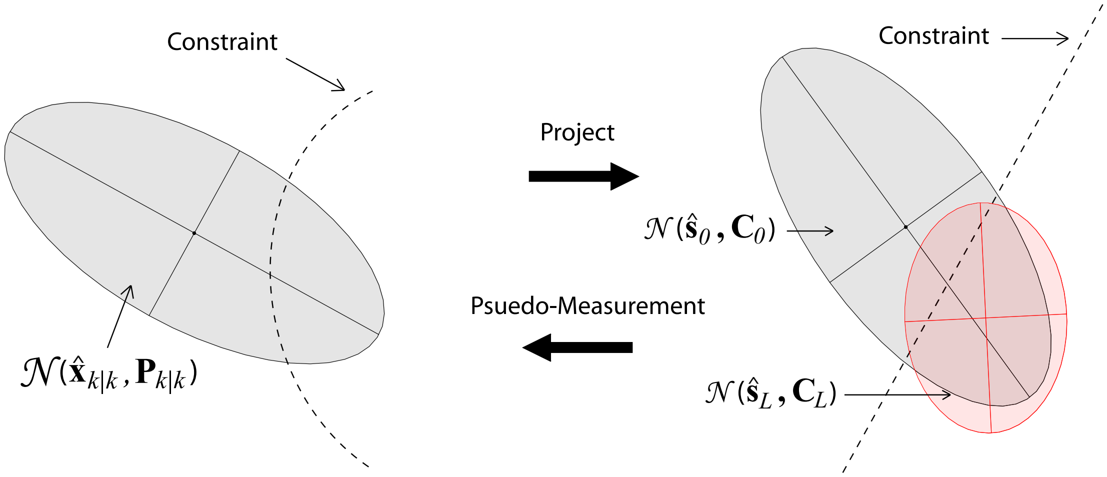

Constraint
Constraint
Project
$\mathcal{N}(\hat{\mathbf{s}}_0, \mathbf{C}_0)$
Psuedo-Measurement
$\mathcal{N}(\hat{\mathbf{x}}_{k|k}, \mathbf{P}_{k|k})$
$\mathcal{N}(\hat{\mathbf{s}}_L, \mathbf{C}_L)$

Figure 4.10: For nonlinear constraints, the Kalman filter estimate is projected into the constraint-space, the constraints are applied as linear constraints, and then the updated estimate is projected back to the original state space.

To update the actual Kalman estimate, we need to project the updated estimate in the constraint-space back into the original state space. A conceptual depiction of the process is shown in Fig. 4.10. The overall algorithm is described in Alg. 3.

Our method of projecting the estimate back to the original state space is to modify the original PDF so that its projection into the constraint-space corresponds to the new estimated uncertainty in the constraint-space. In most applications, unfortunately, the dimension of the constraint-space will be lower than that of the state space. This means that such a projection is not uniquely defined nor easy to compute. To solve this problem, we introduce a covariance update and a mean update based on applying a pseudo-measurement.

Algorithm 3 Nonlinear Inequality Constrained Filtering

```
$(\hat{\mathbf{x}}^+_{0|0},\ \mathbf{P}^+_{0|0}) \leftarrow$ InitializeEstimate()
for $k \leftarrow 1$ to $\infty$ do
    $(\hat{\mathbf{x}}_{k|k-1},\ \mathbf{P}_{k|k-1}) \leftarrow$ Prediction$(\hat{\mathbf{x}}^+_{k-1|k-1}, \mathbf{P}^+_{k-1|k-1}, \mathbf{u}_k)$
    $(\hat{\mathbf{x}}_{k|k},\ \mathbf{P}_{k|k}) \leftarrow$ Correction$(\hat{\mathbf{x}}_{k|k-1},\ \mathbf{P}_{k|k-1},\ \mathbf{z}_k)$
    $(\hat{\mathbf{s}}_0,\ \mathbf{C}_0) \leftarrow$ ProjectToConstraintSpace$(\hat{\mathbf{x}}_{k|k},\ \mathbf{P}_{k|k})$
    for $i \leftarrow$ to $L$ do
        $(\hat{\mathbf{s}}_i,\ \mathbf{C}_i) \leftarrow$ UncertaintyProjection$(\hat{\mathbf{s}}_{i-1}, \mathbf{C}_{i-1}, \boldsymbol{\phi}_i, \alpha_i)$
    end for
    $(\hat{\mathbf{x}}^+_{k|k}, \mathbf{P}^+_{k|k}) \leftarrow$ ProjectBackToState$(\hat{\mathbf{s}}_L, \mathbf{C}_L, \hat{\mathbf{x}}_{k|k}, \mathbf{P}_{k|k})$
end for
```

4.2.3 Covariance Update

To compute the updated covariance matrix of the state, we would like to find the matrix $\mathbf{P}^+_{k|k}$ that, when projected into the constraint-space, is equal to the uncertainty projected covariance $\mathbf{C}_L$, thus,

$$\mathbf{A}_k \mathbf{P}^+_{k|k} \mathbf{A}_k^T = \mathbf{C}_L. \tag{4.7}$$

Unfortunately, there are possibly infinite solutions to this problem when the dimension of the state space is higher than the dimension of the constraint-space.

Our solution is to pretend that there is a pseudo-measurement in the constraint-space applied to the state that we can define in such a way to update the PDF so that its projection into the constraint space is the uncertainty projected solution. The motivation for using a pseudo-measurement for this update process is that it lowers the dimension of the problem that we need to solve to update the covariance.

We first force the resulting covariance of the pseudo-measurement update to satisfy Eq. 4.7. To do this, we take the standard Kalman covariance update and then both left and right multiply by $\mathbf{A}_k$,

$$\mathbf{A}_k\mathbf{P}^{+}_{k|k}\mathbf{A}_k^T = \mathbf{A}_k\mathbf{P}_{k|k}\mathbf{A}_k^T - \mathbf{A}_k\mathbf{P}_{k|k}\mathbf{A}_k^T\left(\mathbf{A}_k\mathbf{P}_{k|k}\mathbf{A}_k^T + \mathbf{R}\right)^{-1}\mathbf{A}_k\mathbf{P}_{k|k}\mathbf{A}_k^T,$$

which can be simplified, as follows,

$$\mathbf{C}_L = \mathbf{C}_0 - \mathbf{C}_0\left(\mathbf{C}_0 + \mathbf{R}\right)^{-1}\mathbf{C}_0. \tag{4.8}$$

The unknown in Eq. 4.8 is the covariance of the pseudo-measurement $\mathbf{R}$. The closed form solution that will determine the matrix $\mathbf{R}$ that satisfies Eq. 4.7 and Eq. 4.8 is,

$$\mathbf{R} = \left(\mathbf{C}_0^{-1} - \mathbf{C}_0^{-1}\mathbf{C}_L\mathbf{C}_0^{-1}\right)^{-1} - \mathbf{C}_0. \tag{4.9}$$

If we plug this measurement covariance into the standard Kalman covariance update equation, we will have an efficient equation for updating the covariance in the state space from the estimated covariance $\mathbf{C}_L$ in the constraint-space that was obtained from uncertainty projection,

$$\begin{aligned}\mathbf{P}^{+}_{k|k} &= \mathbf{P}_{k|k} - \mathbf{P}_{k|k}\mathbf{A}_k^T\left(\mathbf{A}_k\mathbf{P}_{k|k}\mathbf{A}_k^T + \mathbf{R}\right)^{-1}\mathbf{A}_k\mathbf{P}_{k|k} \\ &= \mathbf{P}_{k|k} - \mathbf{P}_{k|k}\mathbf{A}_k^T\left(\mathbf{C}_0^{-1} - \mathbf{C}_0^{-1}\mathbf{C}_L\mathbf{C}_0^{-1}\right)\mathbf{A}_k\mathbf{P}_{k|k}. \qquad (4.10)\end{aligned}$$

It can be shown (see Appendix A.2.4) that this updated uncertainty $\mathbf{P}^{+}_{k|k}$ is guaranteed to be reduced when adopting our approach, which is desirable given that information is gained.

4.2.4 Mean Update

To compute the updated mean vector $\hat{\mathbf{x}}^+_{k|k}$, we would like to find the appropriate state that, when projected into the constraint-space, is equal to the newly computed uncertainty projected estimate $\hat{\mathbf{s}}_L$,

$$\mathbf{a}(\hat{\mathbf{x}}^+_{k|k}) = \hat{\mathbf{s}}_L. \tag{4.11}$$

We will again use a pseudo-measurement method to update the mean of the Kalman filter. Assuming the same pseudo-measurement covariance $\mathbf{R}$ from Eq. 4.9, we can write a Kalman gain matrix, as follows,

$$\begin{aligned}\mathbf{K} &= \mathbf{P}_{k|k}\mathbf{A}_k^T\left(\mathbf{A}_k\mathbf{P}_{k|k}\mathbf{A}_k^T + \mathbf{R}\right)^{-1} \\ &= \mathbf{P}_{k|k}\mathbf{A}_k^T\left(\mathbf{C}_0^{-1} - \mathbf{C}_0^{-1}\mathbf{C}_L\mathbf{C}_0^{-1}\right)\end{aligned}$$

We then force the resulting mean of the pseudo-measurement update to satisfy Eq. 4.11. To do this, we take the standard Kalman mean update equation,

$$\hat{\mathbf{x}}^+_{k|k} = \hat{\mathbf{x}}_{k|k} + \mathbf{K}\left(\mathbf{z} - \hat{\mathbf{s}}_0\right),$$

and then cast both sides into the constraint-space (this assumes linearization),

$$\hat{\mathbf{s}}_L = \hat{\mathbf{s}}_0 + \mathbf{A}_k\mathbf{K}\left(\mathbf{z} - \hat{\mathbf{s}}_0\right). \tag{4.12}$$

The unknown in Eq. 4.12 is the pseudo-measurement vector $\mathbf{z}$. The closed form solution that will compute the vector $\mathbf{z}$ that satisfies Eq. 4.11 and Eq. 4.12 is,

$$\mathbf{z} = \hat{\mathbf{s}}_0 + \left(\mathbf{A}_k\mathbf{K}\right)^{-1}\left(\hat{\mathbf{s}}_L - \hat{\mathbf{s}}_0\right).$$

If we plug this measurement vector into the standard Kalman mean update equation, we will have an efficient equation for updating the mean in the state space from the mean $\hat{\mathbf{s}}_L$ in the constraint-space obtained from uncertainty projection,

$$\hat{\mathbf{x}}^+_{k|k} = \hat{\mathbf{x}}_{k|k} + \mathbf{K}\left(\mathbf{A}_k\mathbf{K}\right)^{-1}\left(\hat{\mathbf{s}}_L - \hat{\mathbf{s}}_0\right).$$

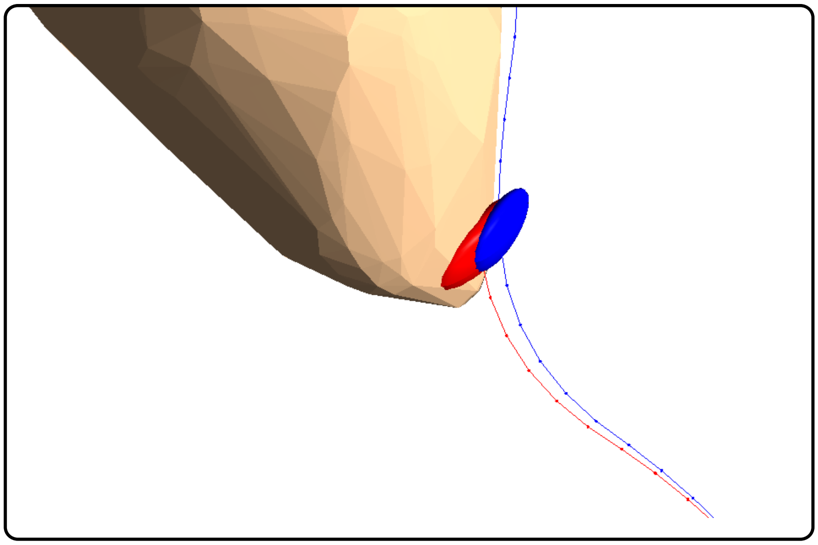


Figure 4.11: This is an example of our constrained filtering method correcting the estimated configuration of the HARP surgical robot based on the constraints imposed by the surface model.

In summary, the mean update presented here uses a pseudo-measurement correction to modify the Kalman estimate with the result from our uncertainty projection algorithm.

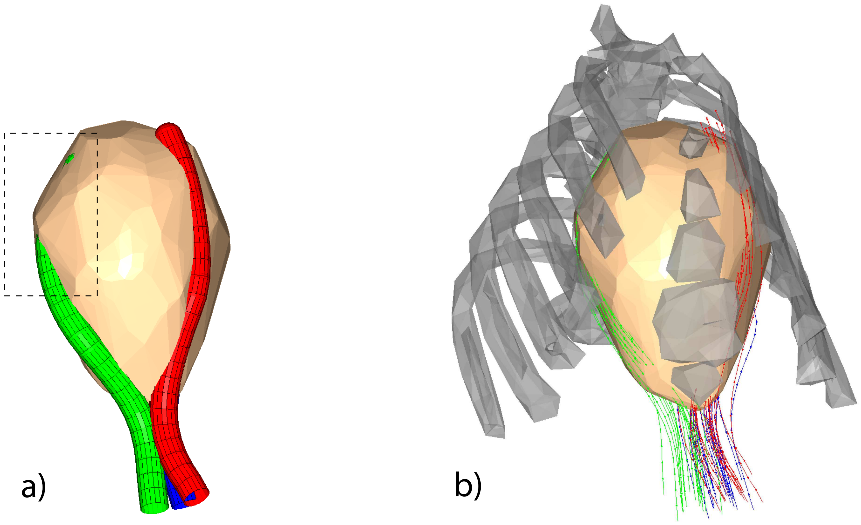


Figure 4.12: Experiment I: (a) shows an initial estimate of three paths performed during an experiment on a porcine subject that violates constraints, (b) depicts the many hypotheses we are testing.

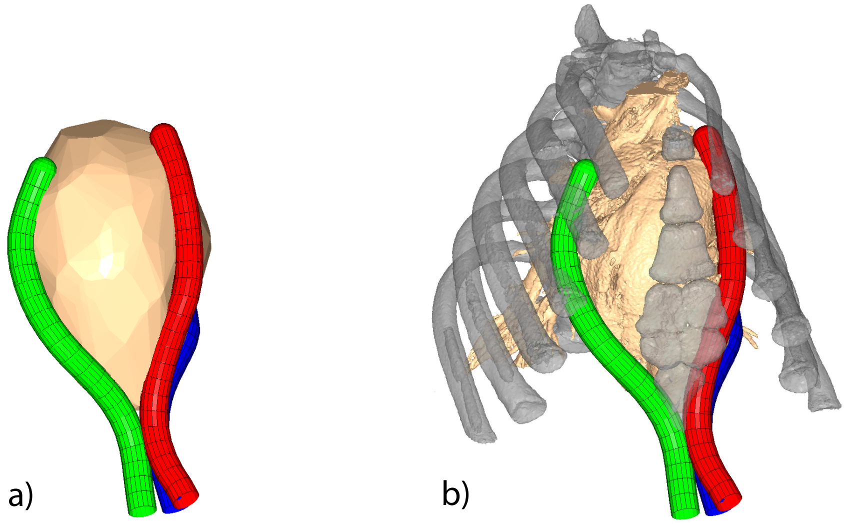


Figure 4.13: Experiment I: (a) shows the final result of the inequality constrained filtering algorithm and (b) shows the same result with more detailed models.

4.3 Experimental Evaluation

The robot we are using for experiments is a highly articulated robotic probe (HARP). The HARP was originally presented in [15, 16, 48]. The advantage of the HARP is that it has the stability of a rigid device and the maneuverability of a flexible tool. The device is shown in Fig. 1.3. Details about the mechanism are discussed in Chapter 2.2.

4.3.1 Experiment I

We have performed several image-guidance experiments on porcine subjects where we obtained preoperative CT images and then processed preoperative surface models. In one experiment, we used a magnetic tracker (NDI Aurora from Northern Digital Inc, Waterloo, Ontario, Canada) to track the distal tip of the robot to collect data for the shape/configuration of the robot when driving along three paths around the epicardial surface of the porcine heart.

In Fig. 4.12-(a), we show the initial estimate of the three paths based on an initial fiducial-based registration procedure. It can be seen that the estimate of the robot in Fig. 4.12-(a) clearly intersects the heart surface, which must be an infeasible state for the robot. This should trigger a constraint update.

To perform constrained filtering for this problem, we first sample over a set of parameters that define the registration of the robot to the surface model (see Fig. 4.12-(b)). For each hypothesis, we perform the inequality constrained filtering algorithm that we have presented in this chapter. The constraint equations we formulate at each time step are based on computing the positions of various points along the robot paths using a forward kinematic model and then intersecting the point's projected uncertainty ellipse in the workspace with the heart surface model. In Fig. 4.11, we show an example of a constraint update changing the estimated path to agree with the constraint. The parameters we estimate for each hypothesis are the joint angles of the links of the robot along with the pose of the initial link, for each of the three paths.

The result, after applying our novel constrained filtering approach and choosing the most likely hypothesis (in this case, the one that deviates the least from the initial estimate), is shown in Fig. 4.13-(a). The final result shown with detailed surface models is shown in Fig. 4.13-(b). We note that while this experimental result was post-processed after the fact, the algorithm we are introducing is defined in such a way that it could be efficiently applied during a live experiment.

4.3.2 Experiment II

In a second experiment, we tested our constrained filtering algorithm with data from a bench-top experiment. For this experiment, we used a trakSTAR$^{\text{TM}}$ EM sensor (Ascension Technologies, Burlington, VT, USA). We configured the robot in 5 different paths around a rubber heart model and collected shape data for each of the paths. We obtained ground

Table 4.1: Results from Inequality Constrained Experiment II.

Path	Error, Initial	Error, Corrected
1	16.60 mm	1.98 mm
2	19.28 mm	2.13 mm
3	16.28 mm	2.90 mm
4	17.78 mm	2.56 mm
5	10.69 mm	1.36 mm
AVG	16.13 mm	2.19 mm

truth alignment by carefully registering the tracker to the surface model with fiducial markers. To simulate the errors that would occur in a surgical trial, we added extra noise to the ground truth data to achieve an erroneous initial estimate (see the initial estimate in Fig. 4.14-(a)).

As before, we sampled over the registration parameters and for each hypothesis, we recursively ran a Kalman filter that estimated the joint angles and the pose of the initial link. We used our constrained filtering algorithm to update an estimate when a point along a robot path intersected the surface model, thus violating a constraint. The result of our constrained filtering algorithm is shown in Fig. 4.14-(b). The initial average error (before constraining the estimate) of the end effector positions for the 5 paths was 16.13mm. After applying our algorithm, the average error of the end effector positions was 2.19mm versus ground truth. This demonstrates the gain in accuracy we can achieve by leveraging known geometric constraints in the environment. In Table 4.1 we provide more details about the results.

4.4 Discussion

The work presented here is a novel method for enforcing nonlinear inequality constraints in the estimation of a high degree of freedom robotic system within a Kalman filter. We have introduced a new algorithm based on uncertainty projection and have developed a

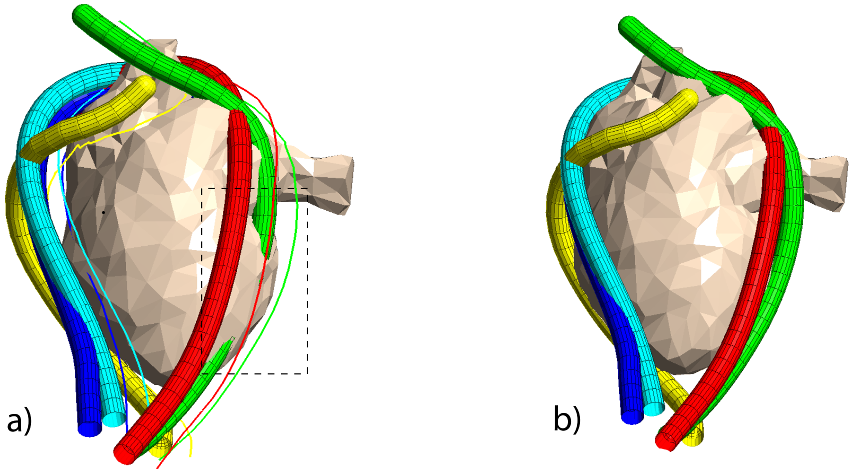


Figure 4.14: Experiment II: in (a) we show the initialized estimate for the registration and configuration of the HARP robot for 5 different paths around the heart model. In (b), we show the corrected estimate that we achieved with our constrained filtering method.

constraint update process based on a pseudo-measurement correction step.

The application we discussed in this chapter is the localization and automatic registration of a robotic surgical probe. We have shown promising results from data collected with bench-top and animal experiments. We have shown that our method is particularly beneficial for correcting registration error in the case of known geometric constraints imposed by preoperative surface models.

The alternative to the constrained Kalman filtering approach which we are presenting here is to sample possible configurations of the robot until a hypothesis is found that aligns well with the preoperatively generated surface models. Unfortunately, for the type of robotic system we use with many degrees of freedom, sampling different robot configurations with a particle filter would be computationally intractable. Our proposed method instead uses an efficient Gaussian parameterization in the form of a Kalman filter to estimate the distribution over possible robot configurations.

It is worth noting that our work has assumed that the tissue models that we are registering to are inherently rigid structures. When the robot is interacting with a deformable surface, a different algorithm must be used. We will discuss a method for estimation in the presence of deformation in the next chapter.

Chapter 5

Equality Constrained Compliant Estimation

In Chapter 4, we discussed inequality constrained filtering and how it is possible to leverage geometric constraints to eliminate infeasible regions from a stochastic estimate of where the robot can be in the state space. We also showed that this is particularly useful for correcting registration and localization parameters for image-guided surgery. By leveraging constraints imposed by the endocardial surface for a cardiac application, we were able to assume that the robot must be outside of the heart surface. But there is a certain rigidness of the environment that is assumed when adopting this approach. When the robot is interacting with a flexible environment, on the other hand, a different approach must be employed.

The approach we have developed for deformable estimation relies on *equality* constrained Kalman filtering. In [58], Simon and Chia present a method for equality constrained Kalman filtering in which linear constraints are incorporated into the Kalman filter update by projecting the unconstrained estimate onto the constraint hyperplane. In that work, the constraints on a state $\mathbf{s}_k$ can be defined the following way,

$$\boldsymbol{\phi}_i^T \mathbf{s}_k = \alpha_i \qquad i = 1 \ldots L. \tag{5.1}$$

In [77], Yang improves upon the paper by Simon and Chia by instead presenting a method that allows for second-order nonlinear state constraints. It is based on an algorithm that iteratively finds the Lagrangian multiplier for the nonlinear constraints. This method provides a better approximation for nonlinear problems.

Another solution to the equality constrained filtering problem is to use a pseudo-measurement approach to enforce constraints on the Kalman filter update step [1, 9, 13, 19, 58, 63, 76]. One way to interpret this method is by viewing the constraint update as a simulated perfect measurement on the constraint hyperplane that will force the state onto the constraint surface. This technique has been used by Chia in [9] to estimate the parameters of electrically stimulated muscles and by Doran in [19] for estimating economic models. In [76], Wen determined that the pseudo-measurement update method is actually equivalent to the state projection method from [58].

Our approach involves the extension of constrained filtering to SLAM, specifically in the body. The only example, that we are aware of thus far, of constrained estimation being used in SLAM is the work by Beevers et. al., in [6]. In this work, Beevers presents a method for SLAM that obeys equality constraints. The algorithm they introduce is applicable to mobile robots exploring an office-like building where the geometric structure of the environment will result in position constraints between observed landmarks. An example is when several nearby landmarks are expected to be co-linear if they are assumed to be on the same wall. The authors present a Rao-Blackwellized constraint filter that infers these constraints and enforces them in a particle filter framework. They show that, by obeying the environmental constraints, the estimation is more efficient and more accurate.

In this chapter, we are introducing a novel equality constrained filtering approach that

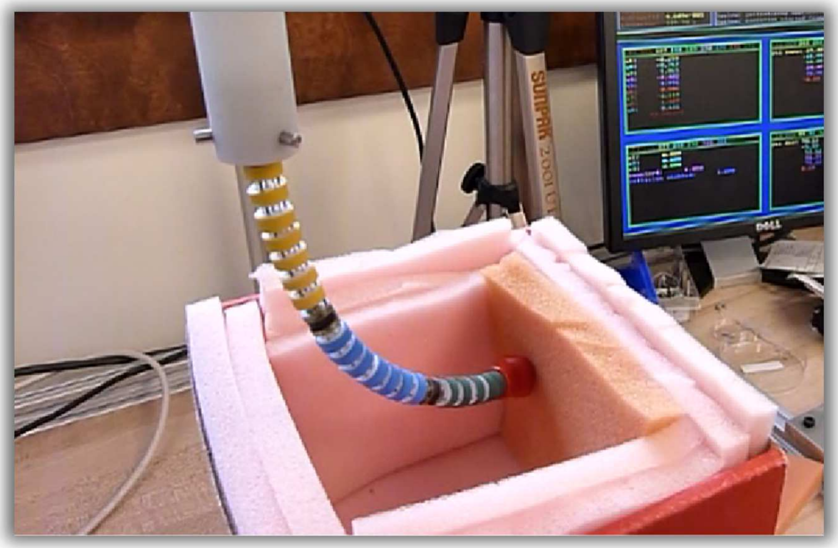


Figure 5.1: The experimental setup used for evaluating our constrained filtering algorithm in the presence of surface deformation.

enables the automatic registration and localization of a compliant surgical robot operating in a deformable environment. The central idea of our method is that when environmental contact is detected between the robot and the surface, such information can be used as a constraint in a Kalman filter to incrementally eliminate infeasible regions of the state space. With a sufficient number of constraints applied to the filter, the true state will emerge. The work presented in this chapter was also presented in our paper [67].

The advantage of this probabilistic approach is its potential accuracy. By constraining the robot to only lower-dimensional subsets of the state-space, the registration of the robot can be improved. Also, there are two favorable by-products of our algorithm: the estimation of surface deformation and the detection of false-positives with respect to contact estimation. The inference of surface deformation can be used within the visualization system to display robot/tissue interaction for improved surgical guidance.

There are several theoretical contributions of this component of the thesis. The first is the novel application of equality constrained Kalman filtering to infer registration and

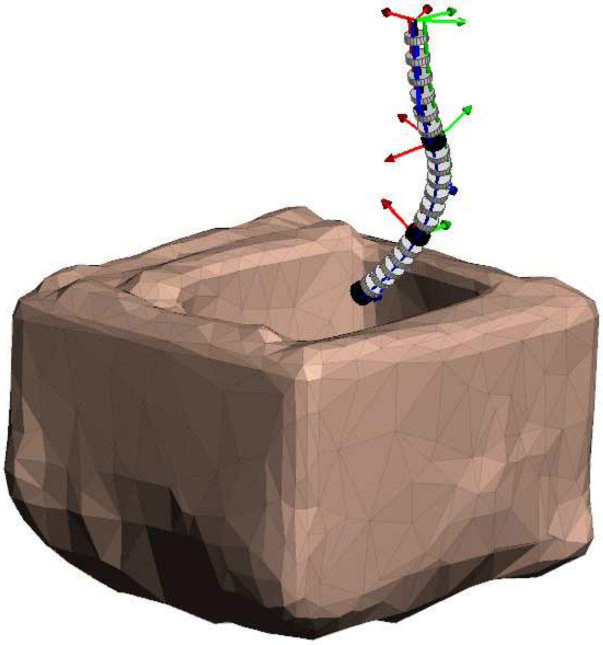


Figure 5.2: A depiction of the visualization result when using equality constrained filtering to estimate the registration of a surgical snake robot to a deformable surface.

configuration parameters for a surgical robot. Another contribution is a method to consider elastic force models within a Kalman filter to constrain an estimation problem. A third contribution is the introduction of an iterative pseudo-measurement update method to correct the state estimate of a nonlinear system given an equality constraint. We have validated these ideas experimentally with a bench-top experiment using a continuum robot designed for MIS [4, 5], see Fig. 5.1. To goal is to use the state estimates from our equality constrained method to produce a detailed visualization of the experiment, as in Fig. 5.2.

5.1 Iterative Pseudo-Measurement Update

Constrained Kalman filtering is the problem of correcting or constraining the Kalman update or Kalman prediction to account for known constraints on the state vector [57]. With constrained filtering, the uncertainty of the state estimate can be significantly reduced. This is particularly useful for our surgical application because preoperative surface models can directly provide geometric constraints. In this section, we will discuss our algorithms to handle equality constraints within a Kalman filter. Then, in the following section, we will discuss the novel application of these algorithms to surgical estimation.

5.1.1 Problem Definition

Instead of linear constraints, as in Eq. 5.1, suppose that there are nonlinear equality constraints imposed on the state estimate of a Kalman filter, as follows,

$$\begin{aligned} \mathbf{x}_k &\sim \mathcal{N}(\hat{\mathbf{x}}_{k|k}, \mathbf{P}_{k|k}) \\ g(\mathbf{x}_k) &= c, \end{aligned} \tag{5.2}$$

where $\mathbf{x}_k$ is the unknown system state, $\hat{\mathbf{x}}_{k|k}$ is the $N \times 1$ mean of the system state estimate and $\mathbf{P}_{k|k}$ is the covariance matrix associated with the mean. The constraint is defined by the function $g(\mathbf{x}_k)$ equating to the scalar value c. For this problem, *equality constrained Kalman filtering* [1, 26, 63, 64, 74] seeks to update the Kalman filter mean and covariance $(\hat{\mathbf{x}}_{k|k}, \mathbf{P}_{k|k})$ to incorporate the information provided by the constraint, thus producing a new estimate $(\hat{\mathbf{x}}^+_{k|k}, \mathbf{P}^+_{k|k})$,

$$\mathbf{x}_k \sim \mathcal{N}(\hat{\mathbf{x}}^+_{k|k}, \mathbf{P}^+_{k|k}).$$

With this formulation, each constraint essentially defines a constraint hypersurface in the space $\mathbb{R}^N$ where the state could potentially be located. Given the intersection of a sufficient number of constraint hypersurfaces, the true state will emerge.

5.1.2 Algorithm

To update the state estimate given nonlinear equality constraints, we have developed an iterative solution that uses numerical optimization. We note that, while our approach to this problem is novel, it is in some ways similar to Wang's maximum likelihood approach [74] due to its use of iteration. To formulate our iterative solution, we can define the problem as follows: at a given time-step k, we seek to update the mean of the filter with the vector $\mathbf{x}$ that minimizes the following cost function,

$$\begin{aligned} \hat{\mathbf{x}}^{+}_{k|k} &= \operatorname*{argmin}_{\mathbf{x}} \ \Big([\hat{\mathbf{x}}_{k|k} - \mathbf{x}]^T \mathbf{P}^{-1}_{k|k} [\hat{\mathbf{x}}_{k|k} - \mathbf{x}]\Big) \\ & \quad \text{subject to} \ \ g(\mathbf{x}) = c. \end{aligned} \tag{5.3}$$

The purpose of this optimization problem is to replace the Kalman estimate $(\hat{\mathbf{x}}_{k|k}, \mathbf{P}_{k|k})$ with a new mean $\mathbf{x}$ that, out of the set of all states that comply with the equality constraint, has the highest likelihood according to the *a priori* distribution.

An equivalent problem to Eq. 5.3 can be defined instead using unconstrained optimization, as follows,

$$\hat{\mathbf{x}}^{+}_{k|k} = \operatorname*{argmin}_{\mathbf{x}} \ \Big([\hat{\mathbf{x}}_{k|k} - \mathbf{x}]^T \mathbf{P}^{-1}_{k|k} [\hat{\mathbf{x}}_{k|k} - \mathbf{x}] + (\sigma_g^2)^{-1} ||g(\mathbf{x}) - c||\Big) .$$

where σ_g^2 is defined to be the measurement noise of a pseudo-measurement applied to the system. In order to make this pseudo-measurement formulation equivalent to Eq. 5.3, this unconstrained optimization problem must be solved for $\sigma_g^2 \to 0^+$. This essentially simulates a perfect measurement in order to strictly enforce the equality constraint. As a nonlinear optimization problem, Gauss-Newton's method can be used to minimize this cost function numerically. The recursive procedure to compute the updated state vector follows,

$$\begin{aligned} \mathbf{x}_0 &= \hat{\mathbf{x}}_{k|k} \\ \mathbf{K}_i &= \mathbf{P}_{k|k} \mathbf{G}_i^T [\mathbf{G}_i \mathbf{P}_{k|k} \mathbf{G}_i^T]^{-1} \\ \mathbf{x}_{i+1} &= \hat{\mathbf{x}}_{k|k} + \mathbf{K}_i [c - g(\mathbf{x}_i) - \mathbf{G}_i(\hat{\mathbf{x}}_{k|k} - \mathbf{x}_i)], \end{aligned} \tag{5.4}$$

where $\mathbf{G}$ is the Jacobian of $g(\mathbf{x})$ linearized about $\mathbf{x}_i$. Finally, we update the mean of the Kalman filter $\hat{\mathbf{x}}^{+}_{k|k}$ with the state $\mathbf{x}_i$ at convergence. The reason that our method

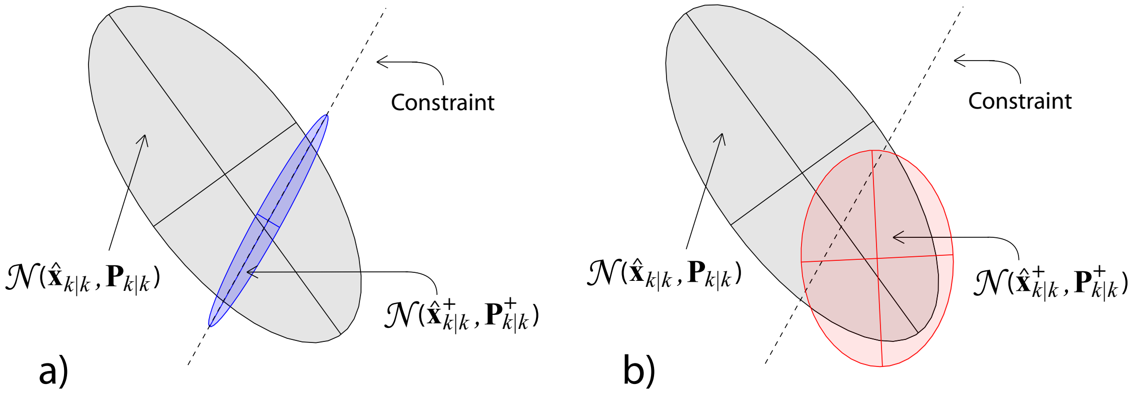

Constraint
$\mathcal{N}(\hat{\mathbf{x}}_{k|k}, \mathbf{P}_{k|k})$
$\mathcal{N}(\hat{\mathbf{x}}^+_{k|k}, \mathbf{P}^+_{k|k})$
a)
Constraint
$\mathcal{N}(\hat{\mathbf{x}}_{k|k}, \mathbf{P}_{k|k})$
$\mathcal{N}(\hat{\mathbf{x}}^+_{k|k}, \mathbf{P}^+_{k|k})$
b)

Figure 5.3: With equality constrained Kalman filtering, seen in (a), the PDF will lie on the constraint. With inequality constrained filtering, seen in (b), the updated PDF will encompass the feasible region.

relates to the pseudo-measurement technique for constrained filtering which was presented in [1, 63, 74] is that we are essentially performing an iterated extended Kalman filter (IEKF) update step [7] with a zero variance pseudo-measurement $g(\mathbf{x})$.

After updating the mean of the filter using the iterative approach, the covariance matrix can then be updated with the conventional Kalman covariance update equation,

$$\begin{aligned} \mathbf{K} &= \mathbf{P}_{k|k}\mathbf{G}^T[\mathbf{G}\mathbf{P}_{k|k}\mathbf{G}^T + \epsilon]^{-1} \\ \mathbf{P}^+_{k|k} &= \mathbf{P}_{k|k} - \mathbf{K}\mathbf{G}\mathbf{P}_{k|k}, \end{aligned} \tag{5.5}$$

where $\mathbf{G}$ is again the Jacobian of $g(\mathbf{x})$ linearized about the updated state mean $\hat{\mathbf{x}}^+_{k|k}$. Additionally, as shown in Eq. 5.5, we choose to adjust the Kalman gain by inflating the simulated measurement noise by a small amount ϵ. This can help prevent numerical issues that would occur with a singular covariance matrix. The most significant benefit of adding the inflated noise ϵ is that it prevents over-confidence in the Kalman state estimate due to the linearization of the state vector when updating the covariance matrix.

The advantage for using an iterative optimization approach for incorporating a nonlinear constraint into the Kalman filter is that the new state estimate is guaranteed to comply with the equality constraint. This is particularly important for our surgical application when the robot is detected to be in contact with a surface, thus it is important for the state to accurately reflect this information. This compares to the use of an EKF for the pseudo-measurement update, which would not necessarily place the new estimate on the constraint due to linearization issues.

The example in Fig. 5.3-(a) illustrates our equality constrained filtering method for a two-dimensional problem. In Fig. 5.3-(a), a constraint tells us that the true state must lie on the hyperplane shown with a dotted line. The algorithm updates the Gaussian PDF associated with the Kalman estimate to lie along that dimension according to the constraint. The uncertainty ellipse does not shrink to a thin line, in this case, because of the inflated pseudo-measurement noise ϵ. It is worth noting the difference between the effect that an equality constraint (shown in Fig. 5.3-(a)) can have on the state estimate versus an inequality constraint (shown in Fig. 5.3-(b)).

5.2 Equality Constrained Surgical Estimation

In the previous section, we introduced our implementation of equality constrained filtering, which uses a novel iterative optimization technique to force the state estimate onto the constraint hyper-surface. In this section, we will present our solution for registration and surgical estimation in the context of a surgical robot interacting with a flexible environment.

The first assumption that we make is that we have a preoperative model of the surrounding environment as well as *a priori* models of the environment stiffness. Also, for this problem formulation, the robot is assumed to be a compliant mechanism for which we have a well-defined compliance model for the robot that relates the deflection of the robot to a force on the distal end of the device.

The state of the system, which we would like to estimate using equality constrained filtering, is defined as follows,

$$\mathbf{x}_k = \left[\mathbf{p}^T, \boldsymbol{\psi}_k^T\right]^T,$$

where $\mathbf{p}$ is the location of the robot in the coordinate frame of the known environment and $\boldsymbol{\psi}$ are additional state parameters that define the shape or full configuration of the robot in the body frame. The overall problem is that we assume no *a priori* knowledge of the state vector, which means that the solution to the registration and localization problem is unknown. The idea, then, is to gain this information through palpating the compliant environment to gain information about the robot's surroundings and to apply this information as equality constraints in the filter.

5.2.1 Force-Balanced Equality Constraints

When a robot is deflected beyond a predetermined threshold, then the robot is thus declared to be in contact with the flexible environment. This equates to a constraint on where the robot can be (the robot must be touching the environment). This constraint can be used to eliminate feasible states from the PDF of the Kalman filter. At a first glance, it may seem appropriate to apply a form of *soft* constraint to the filter estimate when contact is detected due to the possibility of the robot being positioned along a range of states depending on the amount of surface deformation that is expected. If such a constraint could be applied, it would reduce the uncertainty in the filter but will not exactly constrain the robot onto the surface nor will it allow for the inference of the surface deformation.

An arguably more impactful solution, which we adopt for our experiment, is to apply a strict equality constraint to the filter estimate to more informatively constrain the filtering problem using our equality constraint update algorithm from Sec. 5.1. To do this, we leverage stiffness and compliance models to express the following equality relationship,

$$\mathbf{F}_s(\mathbf{x}_k) = \mathbf{F}_r(\mathbf{x}_k),$$

where $\mathbf{F}_s(\mathbf{x}_k)$ represents the elastic force caused by the deformation of the surface and $\mathbf{F}_r(\mathbf{x}_k)$ represents the elastic force caused by the compliance of the robot.

In the context of our equality constraint update algorithm, the appropriate pseudo-measurement function is defined as follows,

$$g(\mathbf{x}_k) = ||\mathbf{F}_s(\mathbf{x}_k) - \mathbf{F}_r(\mathbf{x}_k)|| = 0.$$

Experimentally, we use a simplified model for $\mathbf{F}_r(\mathbf{x}_k)$ by assuming that the force is proportional to the magnitude of the difference between the configuration of the robot, according to the current estimate, and the desired configuration of the robot according to the configuration that is commanded by the onboard controller that drives the continuum robot. The parameters for this simplified compliance model of the robot was determined empirically.

5.2.2 False-Positive Detection

It is possible for the contact detection procedure, due to modeling uncertainty and actuator noise, to report false-positives. This means that the equality constraint update procedure might be invoked when in fact the robot is not actually contacting the environment. This could have drastic consequences on the convergence of the filtering algorithm.

To avoid a diverging state estimate, it is important to detect these false-positives, and when one occurs, temporarily ignore the equality constraint that is imposed by the preoperative surface model. To detect false-positives, we can apply the equality constraint update algorithm assuming the contact detection is correct, and then evaluate the likelihood of the result with a Mahalanobis test,

$$\left[\hat{\mathbf{x}}^{+}_{k|k} - \hat{\mathbf{x}}_{k|k}\right]^T \mathbf{P}^{-1}_{k|k} \left[\hat{\mathbf{x}}^{+}_{k|k} - \hat{\mathbf{x}}_{k|k}\right] > \zeta. \tag{5.6}$$

In Eq. 5.6, $\hat{\mathbf{x}}^{+}_{k|k}$ is the resulting state mean after performing the equality constraint update, $\hat{\mathbf{x}}_{k|k}$ is the prior mean before applying the update, $\mathbf{P}_{k|k}$ is the prior covariance, and ζ is a threshold. When the Mahalanobis distance exceeds some threshold, we report a false-positive for contact detection and revert to the previous estimate (before having applied the constraint). Incorporating this test allows for a more robust estimation process.

5.2.3 Algorithm

Alg. 4 is a high-level description of the the entire filtering process. As with a conventional Kalman filter, our algorithm performs a prediction step and a correction step. Then, using the geometric constraints provided by the robot's interaction with the surface model, we perform our novel constraint update procedure to reduce the uncertainty in the Kalman estimate and to resolve the true registration parameters.

Algorithm 4 Constrained Filtering for Surgical Estimation

```
($\hat{\mathbf{x}}^{+}_{0|0}$, $\mathbf{P}^{+}_{0|0}$) $\leftarrow$ InitializeEstimate()
for $k \leftarrow 1$ to $\infty$ do
    ($\hat{\mathbf{x}}_{k|k-1}$, $\mathbf{P}_{k|k-1}$) $\leftarrow$ Prediction($\hat{\mathbf{x}}^{+}_{k-1|k-1}$, $\mathbf{P}^{+}_{k-1|k-1}$, $u_k$)
    ($\hat{\mathbf{x}}_{k|k}$, $\mathbf{P}_{k|k}$) $\leftarrow$ Correction($\hat{\mathbf{x}}_{k|k-1}$, $\mathbf{P}_{k|k-1}$, $\mathbf{z}_k$)
    if DetectContact() then
        ($\hat{\mathbf{x}}^{+}_{k|k}$, $\mathbf{P}^{+}_{k|k}$) $\leftarrow$ EqualityConstrain($\hat{\mathbf{x}}_{k|k}$, $\mathbf{P}_{k|k}$)
        if $(\hat{\mathbf{x}}^{+}_{k|k} - \hat{\mathbf{x}}_{k|k})^T \mathbf{P}^{-1}_{k|k} (\hat{\mathbf{x}}^{+}_{k|k} - \hat{\mathbf{x}}_{k|k}) > \zeta$ then
            ($\hat{\mathbf{x}}^{+}_{k|k}$, $\mathbf{P}^{+}_{k|k}$) $\leftarrow$ FalsePositiveLoadPrevious($\hat{\mathbf{x}}_{k|k}$, $\mathbf{P}_{k|k}$)
        end if
    end if
end for
```

5.2.4 Conceptual Example

A conceptual example generalizing our equality constrained filtering approach is shown in Fig. 5.4. In this example, a single arm robot is shown that has a movable base and an end effector (shown as a small circle). The true state of the robot is shown in red and the estimated state is shown in black. The initial registration of the robot is erroneous and the state is shown with an associated uncertainty region depicted by an ellipsoid. The surfaces surrounding the robot have known spring stiffnesses.

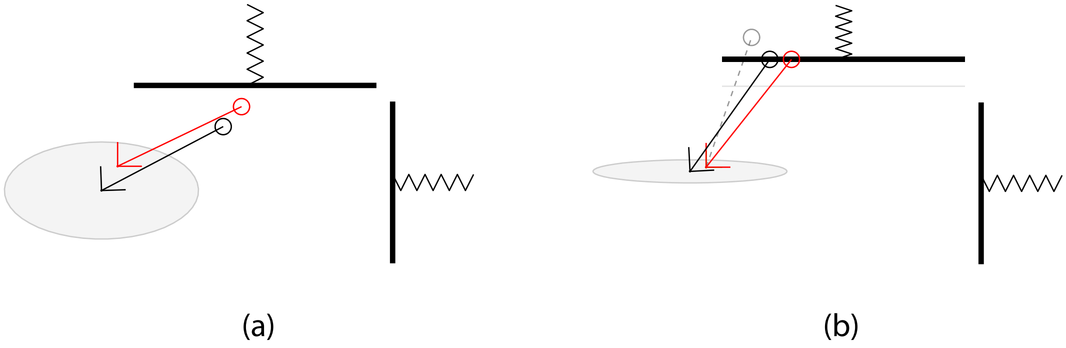


Figure 5.4: The first stage of an example equality constrained problem. The true state of the robot is shown in red and the estimated state is shown in black. In (a) the robot is shown at the start of an experiment. In (b), the arm swings and contacts the flexible environment, thus triggering an equality constraint update that reduces the uncertainty of the state estimate.

In Fig. 5.4-(b), we show the result for our toy example when the robot swings its arm into contact with the environment. The contact is detected because a deflection of the robot is observed (the commanded pose of the robot is shown in gray versus the true pose shown in red). The detection of contact triggers an equality constraint update that is used to move the state estimate to a more likely position. This is a successful constraint update, as the estimated robot is shown to be more closely aligning with the true state in Fig. 5.4-(b). Also the uncertainty has been reduced along the dimension of the constraint. The majority of the uncertainty in the system state is along the x-dimension of the robot position.

In Fig. 5.5-(a), the robot is shown with the arm retracted so that it is no longer in contact with the environment. Also, the robot base has moved to a different position relative to the environment surfaces. When there is no contact with the environment, the conventional filtering process is performed (excluding a constraint update).

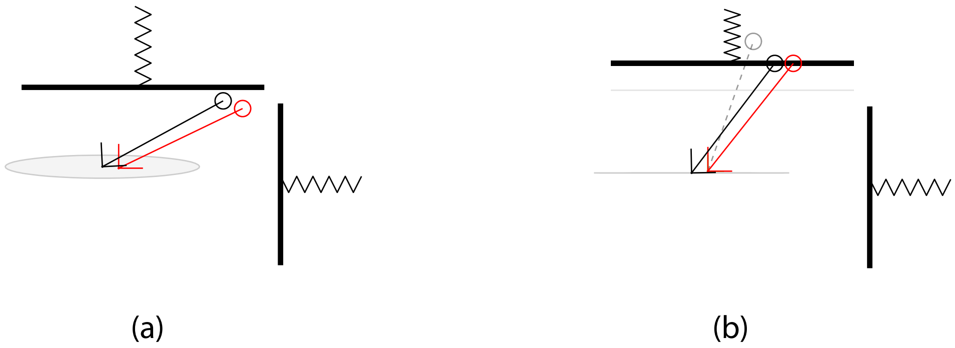


Figure 5.5: This figure depicts the second stage of the example equality constrained problem. After moving out of contact with the environment (shown in (a)), the arm moves into contact a second time (shown in (b)), which shrinks the uncertainty.

In Fig. 5.5-(b), the filtering result is shown when the robot once again swings its arm against a surface in the environment. Contact is detected again and the state estimate of the filter is updated with the equality constraint equation that is derived from the compliance and stiffness models for the environment and the robot. Now that a second point of contact is detected, additional information is obtained that reduces the uncertainty of the state vector. In particular, the orientation of the robot is now known almost exactly due to the fact that two points on a surface define a line, which dictates along which dimension the base of the robot must be located.

In Fig. 5.6-(a), the robot is shown once again with the arm retracted so that it is no longer in contact with the environment. Also, the robot base has moved down to a new position relative to the environment surfaces. Then, in Fig. 5.6-(b), the filtering result is shown when the robot once again swings its arm against a surface in the environment (this time it is the surface which defines a constraint on the horizontal position of the robot). Contact is detected again and the state estimate of the filter is updated with the equality constraint equation that is derived from the compliance and stiffness models for the environment and the robot. Because this constraint defines a hypersurface that

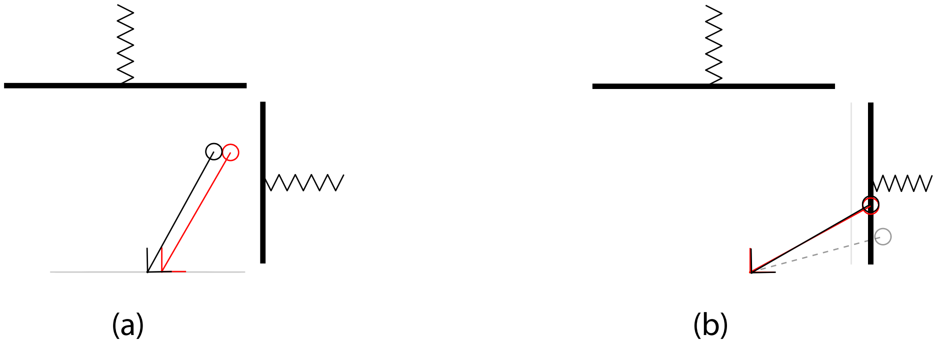


Figure 5.6: This figure depicts the third stage of the example equality constrained problem. After moving out of contact with the environment again (shown in (a)), the arm moves into contact a third time (shown in (b)), which shrinks the uncertainty to a unique solution that overlaps the true state.

intersects with the other constraints at a point, the pose and configuration of the robot is known exactly. In conclusion, by having the robot palpate the environment, the state can be inferred by eliminating regions of the state space that would not be able to explain the sequence of contacts that were observed.

5.3 Experimental Evaluation

Simaan et. al. have developed a multi-backbone continuum robot [4, 5] that is designed for MIS procedures. The robot is a small snake-like robot with three independently actuated segments. Each segment is constructed using three circumferentially located super elastic NiTi secondary backbones and one centrally located super elastic NiTi primary backbone. Actuation, for each segment, is achieved by changing the lengths of the secondary backbones in order to actively bend the segments. We have had the opportunity to test our equality constrained filtering algorithm on this robotic system.

The experimental setup is shown in Fig. 5.1. For this experiment, the end-effector of the robot was driven into forceful contact with a bench-top phantom consisting of a box

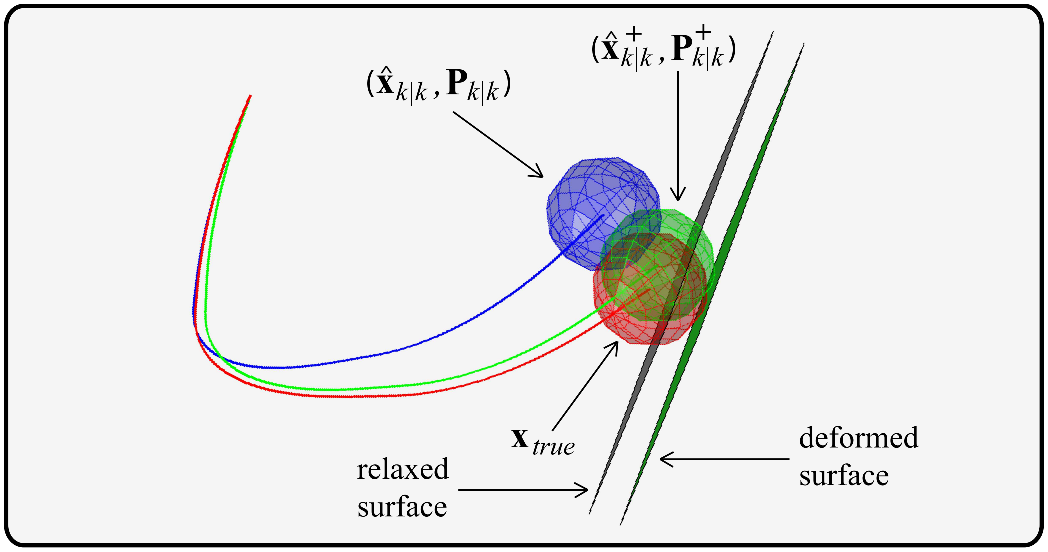

$(\hat{\mathbf{x}}_{k|k}, \mathbf{P}_{k|k})$
$(\hat{\mathbf{x}}^{+}_{k|k}, \mathbf{P}^{+}_{k|k})$
$\mathbf{x}_{true}$
relaxed surface
deformed surface

Figure 5.7: Equality constrained filtering will move the estimate $\hat{\mathbf{x}}_{k|k}$ onto the surface and will properly balance the deformation of the surface with the compliance of the robot. This example shows the new estimate $\hat{\mathbf{x}}^{+}_{k|k}$ near the ground truth configuration after the constraint update.

lined with deformable foam. A laser scanner was used to capture an *a priori* surface model of the phantom. The stiffness of the material was measured by assuming axial stiffness of a uniform material with E young's modulus. To obtain ground truth data with which to compare the performance of our localization algorithm, we used a laser scan that captured the relative positioning of the robot to the phantom and extracted the necessary ground truth registration parameters from the geometry of the object.

We ran the continuum robot to various desired target configurations that were out of reach due to the robot being obstructed by the phantom. This forced the robot to contact the surface at many different points. The data was saved and post-processed using Matlab. During the operation of the algorithm, we chose to only update the filter when the robot was relatively stable (its velocity was below some threshold). This allowed for a more robust estimation of the localization estimates.

In Fig. 5.7, we show an example of an equality constraint shifting the Kalman estimate so that it is both contacting and appropriately deforming the surface. In this figure, we

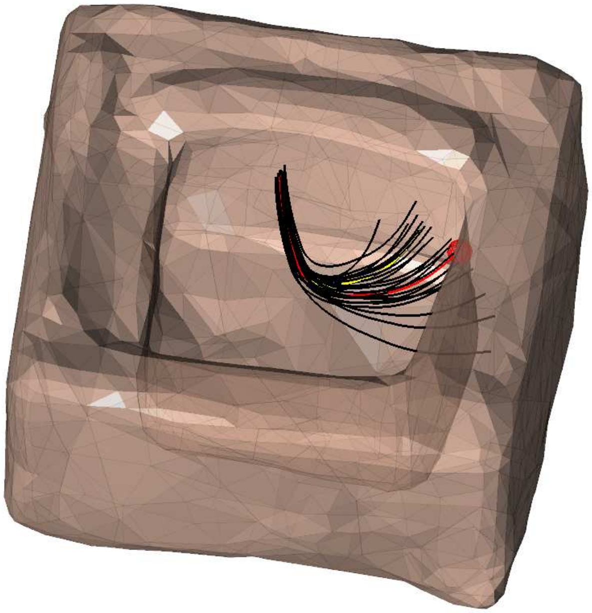


Figure 5.8: Multiple hypotheses are shown, each of which runs the constrained filtering algorithm for registration.

have drawn spheres to model the width and shape of the end effector of the robot. The estimate is shown moving toward the true state because of the added information provided by the constraint. Also shown in this figure is the estimation of surface deformation that is a by-product of our equality constrained filtering approach. Because the state of the robot is estimated and the force on the surface is inferred through the equality constraint equation, the amount of deformation is deduced automatically within the filtering process. This is an advantage of our approach and can help produce more representative models for surgical guidance that display live robot/tissue interaction.

In Fig. 5.8, a set of initialized hypotheses are shown before the data is processed. Each of these hypotheses was evaluated and updated using the constrained filtering algorithms presented in this paper. In Fig. 5.9, it can be seen that many of the hypotheses for the registration and localization have begun to move toward the ground truth configuration. Some hypotheses, though, have diverged due to a poor initial alignment and the local affect

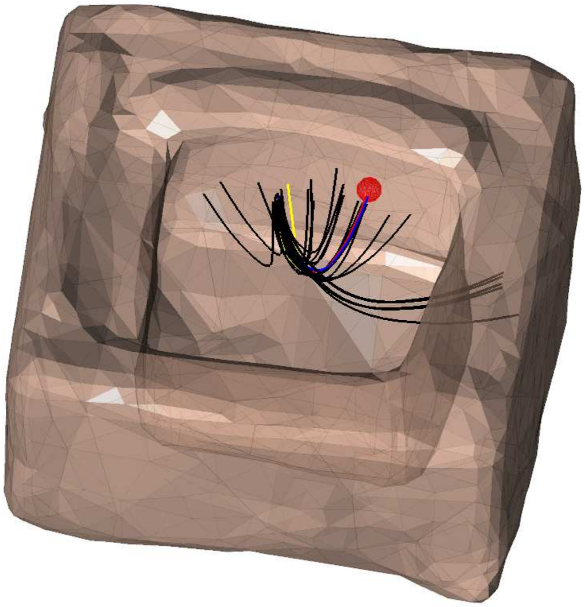


Figure 5.9: During the execution of equality constrained filtering, some hypotheses have diverged.

of the filtering updates. In Fig. 5.10, the resulting configuration of the robot, estimated with our filtering algorithm, is shown overlapping with the ground truth configuration. This estimate emerged with the highest likelihood after imposing all of the constraints.

The initial guess for the registration assumed a 14.6mm error at the position of the robot end effector. This equates to an approximately 10 percent error with respect to the length of the robot. The error was mostly associated with a largely erroneous yaw angle. After processing the contact data and applying the constraints with the Kalman filter, our final registration error was reduced to 1.1mm (with respect to the end effector position), which equates to a less than 0.8 percent positional error with respect to the length of the robot. We believe this result demonstrates the feasibility of our approach for accurate registration in compliant environments.

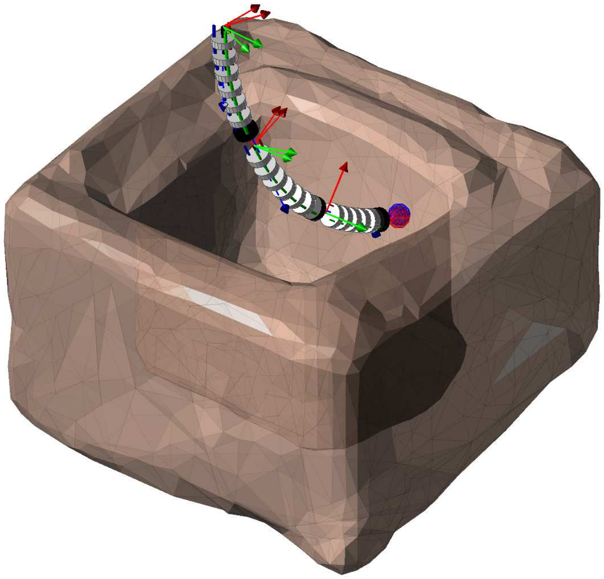


Figure 5.10: The result shows proper alignment between the estimated robot configuration and ground truth.

5.4 Discussion

In this chapter, we present a novel filtering algorithm that leverages geometric constraints within a Kalman filter framework to automatically register and localize a compliant surgical robot with respect to an *a priori* 3D deformable surface model. Our method relies on an equality constrained update step for when the robot is forcefully contacting the environment. With this filtering procedure, we can incrementally eliminate regions of the state space that are decidedly infeasible, eventually resolving the true state that defines the registration of the robot.

The theoretical contributions of this work are its use of equality constrained Kalman filtering to localize a surgical robot, the consideration of elastic force models within a Kalman filter for applying equality constraints, and the introduction of an iterative pseudo-measurement update method for equality constrained filtering. We have shown with a successful experiment that accurate registration can be achieved automatically using our proposed algorithm.

Chapter 6

Periodic Feature Mapping

In the previous sections, we discussed the estimation of shape and the models that we can use to perform image-guided estimation. We also talked about constrained filtering for correcting the estimate of the robot state during a minimally invasive procedure. In general, the algorithms that we have introduced in this thesis, thus far, have essentially solved the localization aspect of our BodySLAM problem.

Now, to round out the idea of BodySLAM, and to close-the-loop for this estimation problem, we will introduce the mapping aspect of our proposed approach. By adding *in vivo* mapping, we can complete the BodySLAM idea. The motivation of the mapping aspect of BodySLAM is to take preoperative surface models and to annotate the models with dynamic moving surface information. We believe that the first step of moving surface estimation within the body is to map the motion of individual surface features.

In [41], the authors introduce an algorithm that performs stereo vision based SLAM within the body. One somewhat unique aspect of this work is that the authors attempt to estimate periodically moving features to account for cardiac and respiratory motion. Unlike the approach that we will present in this chapter, the work in [41] does not use Fourier series parameters to estimate the motion of periodically moving features and only considers motion in terms of depth (in the dimension orthogonal to the image plane of a camera). The authors model motion according to a single sinusoidal parameter that accounts for respiration. This means that their approach does not consider arbitrary and complex motion that can occur when different organs deform and interact in a complex way due to respiratory or cardiac motion.

In [52, 53, 54], a thin-plate spline model is used to to track visually detected periodically moving features in the camera images of a stereo camera. The way that our method extends the work by Richa et. al. in [53] is that we are performing full monocular estimation of moving features. Thus, our approach solves a more challenging nonlinear filtering problem (resolving depth through parallax change with a single camera) than the problem investigated by Richa, which has assumed that the depth of the motion can be estimated via the use of calibrated stereo vision. While Richa is using a stereo camera that is fixed, we are providing a solution to a single moving camera that is tracked by a magnetic tracking sensor, making the SLAM problem more challenging. Lastly, while the solution by Richa can adequately use the EKF to estimate their motion, the filtering task we are investigating has required a filtering technique that involves the use of numerical iteration within the Kalman measurement update step.

Our approach to the mapping of features within the body is to define motion parameters within the state of a Kalman filter for each feature that is detected with a monocular camera system. When the camera is moved, the change in view will provide information concerning the motion of the feature. We have chosen to define the motion of each feature with Fourier series parameters. To eliminate the issue of filter divergence, we have developed an iterative Kalman filter algorithm for performing the measurement update step of this Fourier series filtering scheme.

6.1 Bearing-Only Filtering

The bearing-only SLAM problem, specifically, is when measurements to detected features in the environment are from a bearing sensor (such as a camera). This results in a difficult estimation problem due to the fact that the measurement model is highly nonlinear. Also, the measurements, in the case of a bearing-only sensor, do not provide the information that is necessary to directly observe the position of a feature. This means that a single measurement cannot encode a feature's position in the environment, thus multiple measurements from different vantage points are required to map a feature.

In Fig. 6.1, we show an example of SLAM in which a robot is mapping the locations of two features by acquiring spatial measurements at different vantage points. For conventional monocular SLAM, the problem is to map the location of a feature via the estimation of the following state,

$$\mathbf{x}_k = \left[\begin{array}{ccc} x & y & z \end{array} \right]^T,$$

which encodes the 3D position of the feature. The feature is estimated with a mean and covariance matrix,

$$\mathbf{x}_k \sim \mathcal{N}(\hat{\mathbf{x}}_{k|k}, \mathbf{P}_{k|k}).$$

The bearing-only measurement model that is used within the filtering algorithm is as follows,

$$\mathbf{h}(\mathbf{x}_k) = \left[\begin{array}{c} \text{atan2}(y - y_R, x - x_R) \\ \text{atan2}(z - z_R, \sqrt{(x - x_R)^2 + (y - y_R)^2}), \end{array} \right], \tag{6.1}$$

where (x_R, y_R, z_R) is the position of the robot in the world. This models the mapping of a point feature in the global coordinate frame to a pitch and yaw angle representing the bearing relative to the coordinate frame of the camera image.

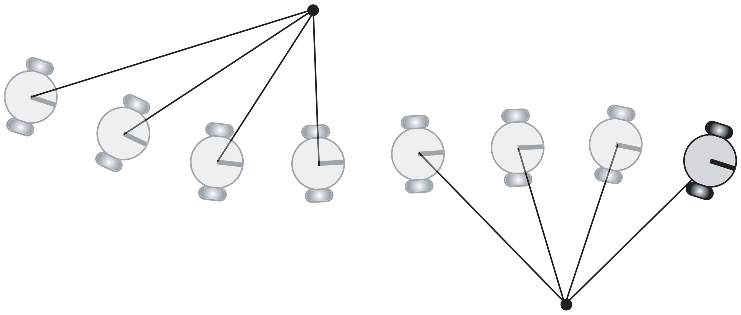


Figure 6.1: This shows a mobile robot mapping features with a bearing-only sensor.

6.2 Iterative Kalman Filtering for Periodic Estimation

For SLAM in the body, the features that are detected with a monocular camera are moving for one or two reasons: respiratory motion and cardiac motion. To capture this motion when mapping visually detected features, our solution is to include dynamic parameters within the Kalman filter state vector and to map the features using an iterative measurement update step.

6.2.1 Filter Formulation

Our approach is to include Fourier series parameters within the formulation of a Kalman filter to encode the motion of each individual feature that is measured by the monocular camera. The following state vector is used to define the motion of a single point feature,

$$\mathbf{x}_k = \begin{bmatrix} x_{DC} & y_{DC} & z_{DC} & {\mathbf{f}_x}^T & {\mathbf{f}_y}^T & {\mathbf{f}_z}^T \end{bmatrix}^T$$
$$\mathbf{x}_k \sim \mathcal{N}(\hat{\mathbf{x}}_{k|k}, \mathbf{P}_{k|k}).$$

This state vector includes the DC terms of the periodic motion (x_{DC}, y_{DC}, z_{DC}) which essentially define the mean of the motion path that the feature takes during a single periodic cycle. Also, the state includes three vectors of additional parameters ($\mathbf{f}_x$, $\mathbf{f}_y$, $\mathbf{f}_z$) that define the dynamic Fourier series components of the motion. More specifically, the vector $\mathbf{f}_x$, includes elements of both the real and imaginary parts of the discrete Fourier transform (DFT) of the periodic signal. We note that only a subset of the elements of the DFT are included in the state vector due to natural redundancies in the DFT of a real signal and due to the lack of high frequency motion for the typical physiological tissue motion that we can expect for our BodySLAM application.

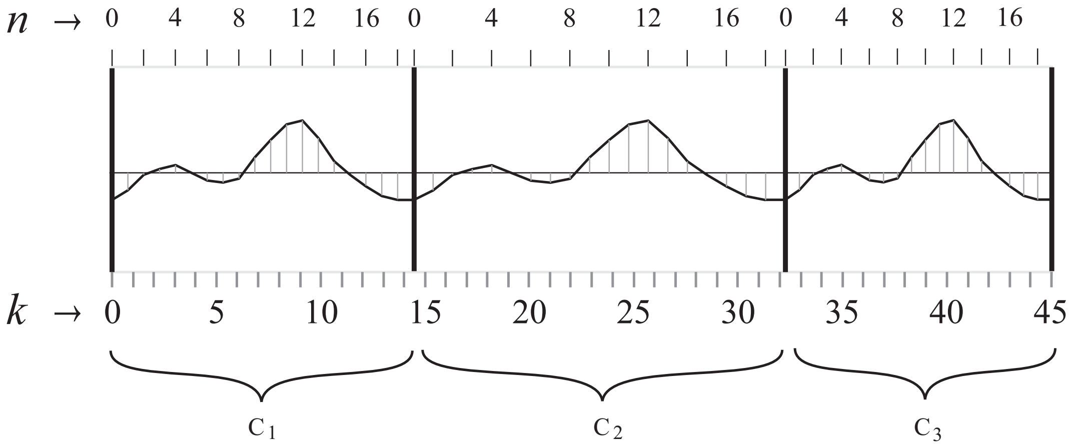


Figure 6.2: This is a depiction of a periodic signal.

In Fig. 6.2, we show a depiction of a periodically moving signal. The length of a cycle in the periodic signal is subject to change. This is an example of a signal that could encode the motion of the x-dimension of a moving feature. A similar signal would be estimated for both the y and z dimensions as well. Measurements are assumed to be acquired asynchronously within a single cycle of the periodic signal but the associated phase of the measurement (when it occurred within a cycle) is assumed to be known. The goal of our filtering approach is to produce an estimate of the parameters $\mathbf{x}_k$ that would best approximate this periodic signal.

In Fig. 6.3, the DFT of a periodic signal is shown. The real part of the DFT is symmetric for real signals, which is an assumption that we can make given that the motion of a periodic feature can be defined with a real signal. The imaginary part of the DFT is odd-symmetric for a real signal and we can also make this assumption due to the fact that the original signal is real. Thus, when defining the necessary parameters in our Kalman filter state vector, we do not need to include the parameters in the DFT that are redundant. In Fig. 6.3, we show with shaded boxes an example of the parameters that would be chosen for inclusion within the Kalman filter state vector. The elements in the middle range of the DFT can be excluded because they do not contribute significantly to the motion.

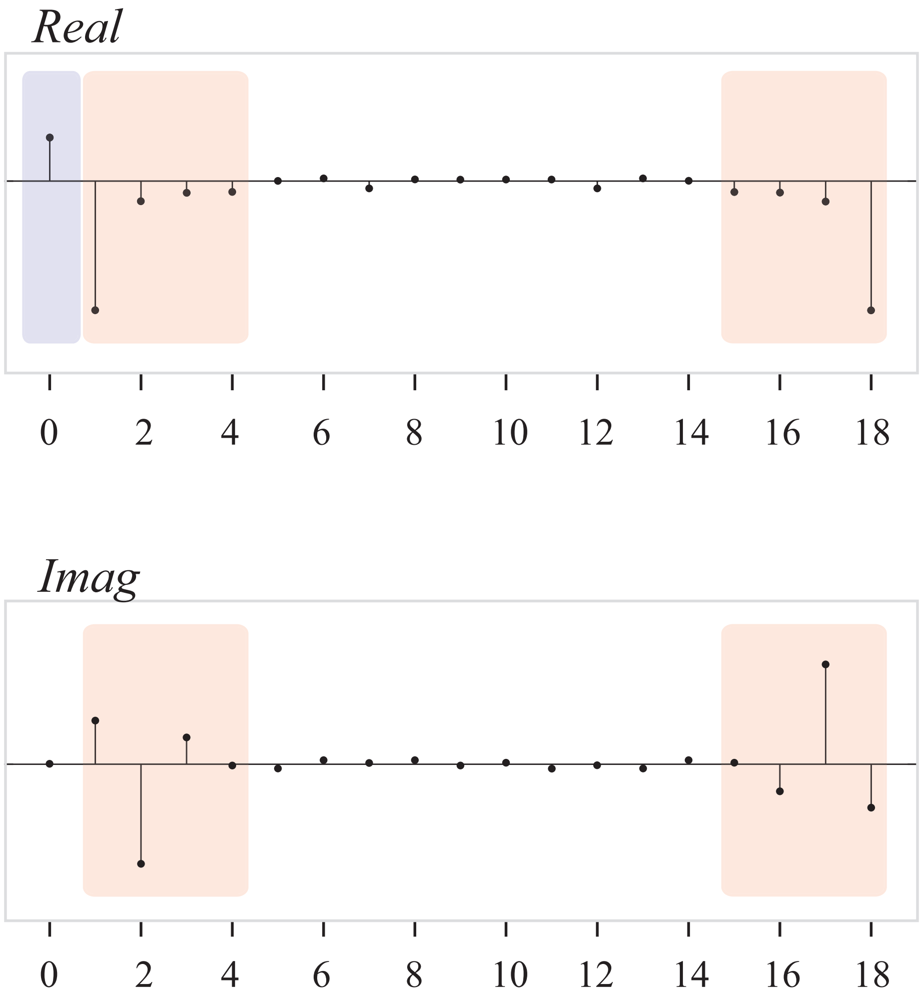


Figure 6.3: This is a depiction of the DFT of a periodic signal.

6.2.2 Filter Prediction Step

The position of a feature, according to our Kalman filter formulation, is completely determined by the state vector $\mathbf{x}_k$ and the phase that denotes the current time within a cycle length. For this reason, despite the fact that there is inherent motion in the position of a feature, there is no motion model for the prediction step of the filter. One way to interpret this concept is to understand that while a feature may be moving from one position to another over the course of one time step, the state vector is static over the course of a periodic cycle. With a static state vector over time, a feature's transition model is the identity matrix, or more simply,

$$\mathbf{x}_k = f(\mathbf{x}_{k+1}) = \mathbf{x}_{k+1}$$

6.2.3 Filter Measurement Correction Step

The measurement model of the Kalman filter maps the state vector $\mathbf{x}_k$ to the measurement space. This means that the measurement model, for a given time within a periodic cycle, will provide the expected bearing angle measurements given a feature state. To implement the measurement function, we first reconstruct the DFT that defines the periodic motion from the Fourier series parameters within the Kalman state vector. This involves filling in zeros for the elements of the DFT that are not represented in the state,

$$\begin{aligned}\mathbf{F}_x &= DFT(\mathbf{f}_x, x_{DC}),\\ \mathbf{F}_y &= DFT(\mathbf{f}_y, y_{DC}),\\ \mathbf{F}_z &= DFT(\mathbf{f}_z, z_{DC}).\end{aligned}$$

Once the DFT is reconstructed, we can obtain the (x,y,z) position of a feature using the inverse Fourier transform (IDFT). We can then figure out the time that a measurement takes place (in the cycle) and extract the corresponding parameters from the IDFT, as follows,

$$\begin{aligned}\mathbf{x}_t &= \mathcal{F}^{-1}\left(\mathbf{F}_x\right),\\ \mathbf{y}_t &= \mathcal{F}^{-1}\left(\mathbf{F}_y\right),\\ \mathbf{z}_t &= \mathcal{F}^{-1}\left(\mathbf{F}_z\right).\end{aligned}$$

The measurement function can then be computed using the bearing-only measurement model that we defined in Eq. 6.1, except now the terms extracted from the IDFT are used in place of the directly observed $(x,\, y,\, z)$ parameters,

$$\mathbf{h}(\mathbf{x}_k) = \left[\begin{array}{c} \text{atan2}(y_t - y_R, x_t - x_R) \\ \text{atan2}(z_t - z_R, \sqrt{(x_t - x_R)^2 + (y_i - y_R)^2}) \end{array}\right].$$

Thus far, we have interchangeably referred to the parameters of the Kalman state vector as Fourier series parameters and as elements of the DFT of a signal. We believe the difference between the two, in this case, is negligible due to the fact that the DFT of a sampled signal is proportional to the Fourier series coefficients of the continuous periodic signal obtained by repeating and interpolating the sampled signal.

Our measurement update method is motivated by the iterated extended Kalman filter (IEKF) [7]. According to [7], the IEKF is equivalent to using Gauss-Newton's method on the cost function associated with the posterior distribution with an initial guess at the previous mean. Stopping after just one iteration of the Gauss-Newton optimization results in the well-known EKF equations. We ask the following question: why would one start an optimization procedure that is meant to optimize the posterior but then stop after one iteration instead of iterating until convergence?

We perform an iterative Kalman update to estimate the motion parameters for observed features on a moving deformable surface. Our Kalman update is formulated as the following cost function minimization,

$$\hat{\mathbf{x}}_k = \arg\min_{\mathbf{x}_k} \left[(\mathbf{x}_k - \hat{\mathbf{x}}_{k-1})^T \mathbf{P}_{k-1}^{-1} (\mathbf{x}_k - \hat{\mathbf{x}}_{k-1}) + (\mathbf{z}_k - \mathbf{h}(\mathbf{x}_k))^T \mathbf{R}^{-1} (\mathbf{z}_k - \mathbf{h}(\mathbf{x}_k)) \right],$$

where $\hat{\mathbf{x}}_k$ is the estimate of the motion parameters for one feature at time step k, $\mathbf{P}_{k-1}$ is the estimated covariance matrix for that feature at time step $k-1$, and $\mathbf{z}_k$ is a bearing measurement to the feature obtained at time step k. Also, in this cost function, $\mathbf{h}(\mathbf{x}_k)$ is the measurement model for the sensor and $\mathbf{R}$ is the covariance matrix for the measurement that defines the expected sensor noise. To optimize this cost function, we use the iterated extended Kalman filter equations, as follows,

$$\begin{aligned}
\mathbf{x}_0 &= \hat{\mathbf{x}}_{k|k-1} \\
\mathbf{K}_i &= \mathbf{P}_{k|k} \mathbf{H}_i^T [\mathbf{H}_i \mathbf{P}_{k|k} \mathbf{H}_i^T + \mathbf{R}]^{-1} \\
\mathbf{x}_{i+1} &= \hat{\mathbf{x}}_{k|k} + \mathbf{K}_i [\mathbf{z}_k - \mathbf{h}(\mathbf{x}_i) - \mathbf{H}_i (\hat{\mathbf{x}}_{k|k-1} - \mathbf{x}_i)].
\end{aligned}$$

The updated estimate $\hat{\mathbf{x}}_k$ is equal to the solution $\mathbf{x}_i$ at convergence.

6.3 Experimental Evaluation

6.3.1 Experiment I

Experiment I involves the use of an open-source available MRI dataset. The dataset was extracted from an anonymous patient and released online for post-processing the data. The motion is recording via the collection of many MRI slices over multiple phases in their cardiac cycle. The resulting post-processed model is a 4D moving surface (representing a human heart) that we can load into our algorithms for testing motion estimation. For this experiment, we used Matlab to create a simulation with a visual camera that observes moving features on the surface of this MRI dataset.

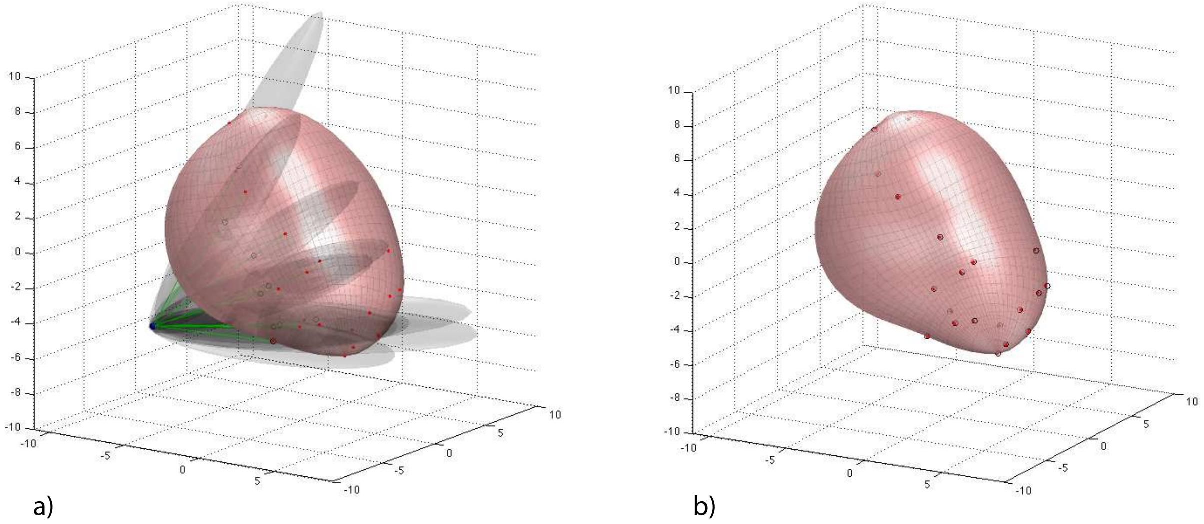


Figure 6.4: This is a photo of the moving heart experiment that we performed that included a simulated robot and motion data from online accessible MRI measurements.

A picture of the simulation is shown in Fig. 6.4. We assumed that the camera traveled a fixed helical path around this structure to observe features on all sides of the beating heart. The camera has a wide view angle and it is assumed that the data association problem is solved. For each bearing measurement that was simulated, artificial noise was added in order to robustly test the performance of our filtering approach. The estimates of the feature locations, upon initialization, are shown in Fig. 6.4-(a). The features are shown with large ellipsoids due to the uncertainty of the initial feature locations.

The result of this experiment is the successful mapping of periodically moving features located on the surface of the heart. When compared to the an EKF implemented with the same Fourier series parameterization, the iterated algorithm that we have developed outperformed the EKF. The EKF was prone to divergence after the simulation had progressed. The iterated approach did not diverge and successfully mapped the motion of the features. The simulation with mapped features is shown in Fig. 6.4-(b). An example of the mapping procedure converging to the true motion is shown in Fig. 6.5.

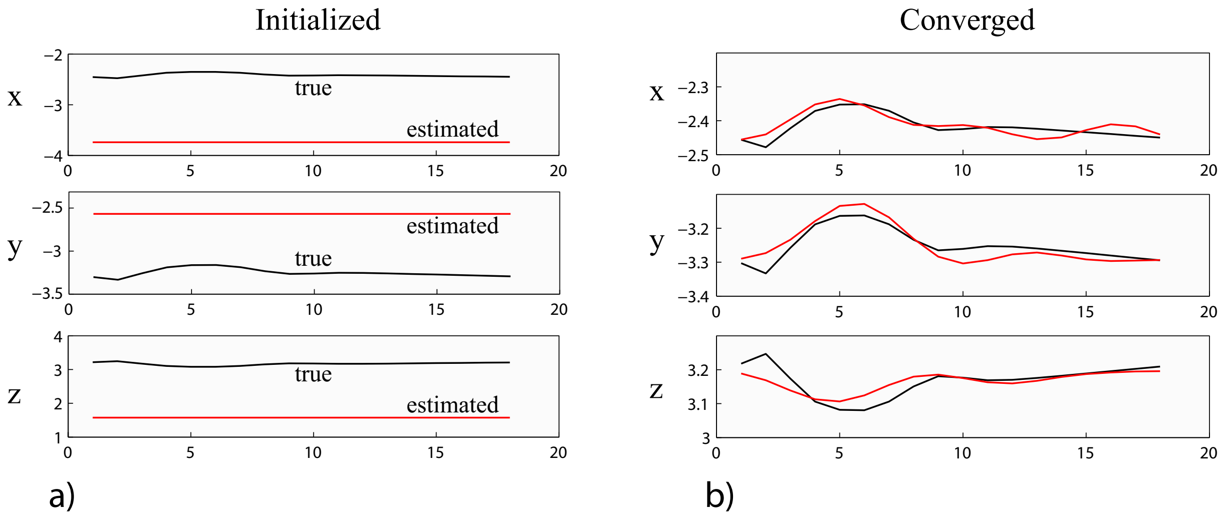


Figure 6.5: This figure shows the convergence of the estimated motion to the true motion after the filter is run to convergence.

6.3.2 Experiment II

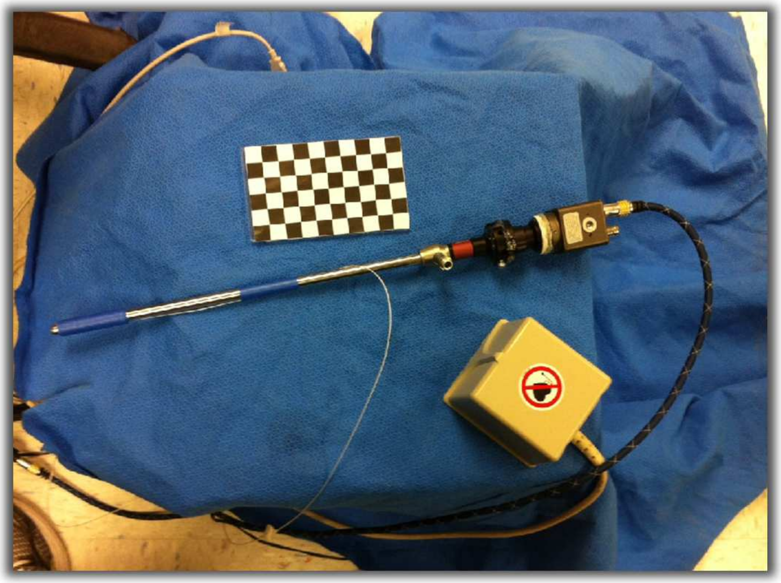


Figure 6.6: This is a photo of the calibration setup, including a laparoscope with a camera and a magnetic tracking system.

Experiment II involves a recorded dataset that we collected by recording the images from a laparoscopic camera (shown in Fig. 6.6). The moving surface that we were experimenting with for this trial was a beating heart phantom, shown in Fig. 6.7. The phantom

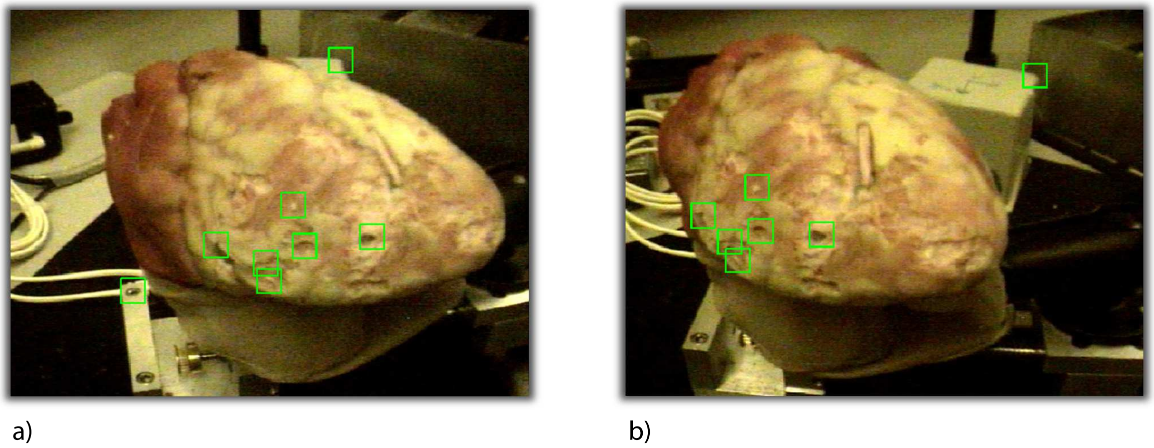


Figure 6.7: This depicts the features that are tracked for the beating heart phantom experiment, which are detected using a template-tracking method.

is a rubber model that beats according to pneumatic oscillations that are introduced by an air pump. For this experiment, we tracked the pose of the laparoscopic camera with the Ascension magnetic tracking system that we also use for snake robot shape estimation. The magnetic field generator is visible in Fig. 6.6.

The checkerboard that we used for the camera calibration process is shown in Fig. 6.6. By recording the Ascension tracker data while we performed the calibration, this provided us with a means to compute the transformation matrix between the coordinate frame of the camera and the coordinate frame of the Ascension tracker.

To track visual features, we relied on template tracking which works by searching in a surrounding region for a template that minimizes a mean square error between itself and an initial template that we created by manually selecting several points of interest on the surface of the heart phantom model. The points were tracked automatically by our feature tracking algorithm until the template tracking failed. An example of the feature extraction is shown in Fig. 6.7.

The result of applying our periodic feature mapping algorithm on this dataset was the accurate mapping of visual features on the surface of a rubber deforming heart phantom.

Although the features in this experiment do not necessarily move in a complex path, we believe that the result demonstrates the efficacy of this method for surgical mapping and the utility that this approach might have for annotating preoperative surface models with live motion data. A quantitative result for this experiment is shown in Fig. 6.9, where the difference in the innovation for an implementation assuming static features and our implementation with dynamic features is plotted. The mostly positive values over time, in this figure, depicts that there is significantly more error in the measurements for a static solution, thus our motion-enabled feature mapping algorithm outperforms the purely static method.

In Fig. 6.8, we show the resulting motion paths that were mapped for the features that were tracked visually for this experiment. The qualitative result is that the motion paths agree with the expected motion of the surface in that they are moving several *mm* in the direction orthogonal to the original camera position. Additionally, all of the feature motion paths agree with each other, which suggests that the motion was accurately mapped.

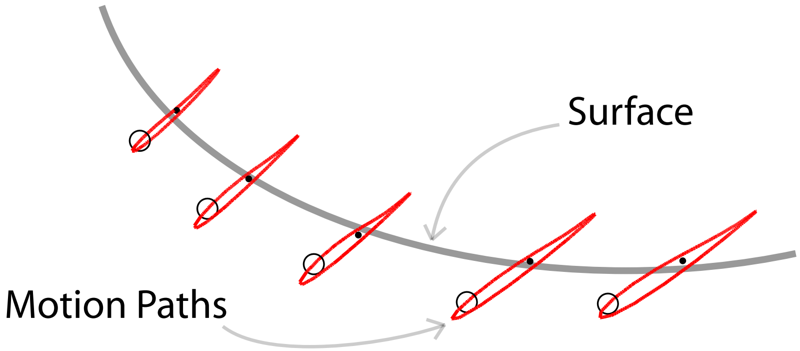


Figure 6.8: This shows a birds-eye view of the mapped features for the heart phantom experiment. The features are shown as points with a surrounding path defining the cyclical motion for each feature. The mapped motion paths agree, suggesting that the surface was mapped accurately.

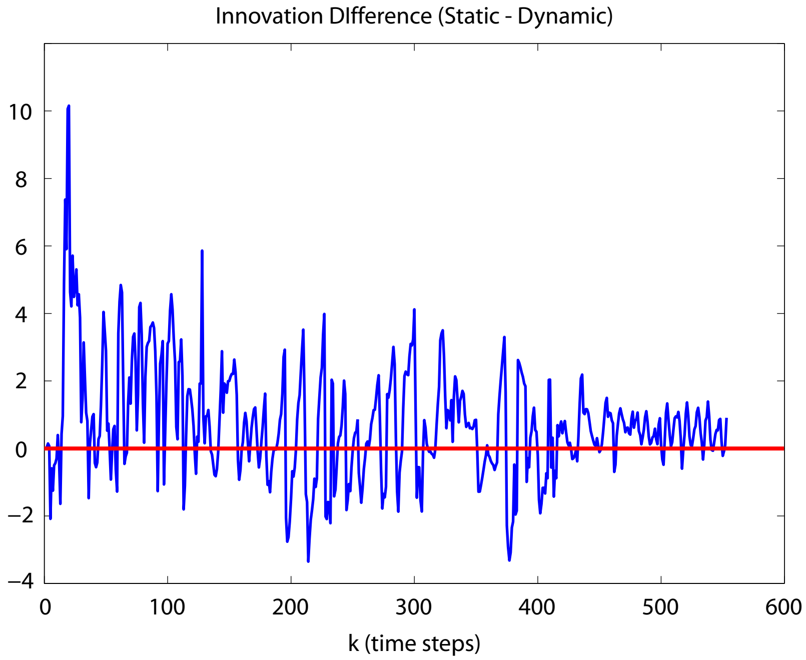


Figure 6.9: This figure depicts the difference in the measurement innovation for static estimation and for dynamic estimation. Larger (positive) values correspond to our dynamic method outperforming a purely static mapping method.

6.3.3 Experiment III

For Experiment III, we tested our periodic motion mapping algorithm on a series of acquired images from one of our NOTES experiments. A figure showing the experiment, the extracted features, and the estimated feature motion (in green) is shown in Fig. 6.10. The dataset is from an *in vivo* animal experiment on a porcine model. The tissue motion that was observed during this experiment was caused by respiratory movement. For these NOTES experiments, it would be valuable to map out the motion of the nearby tissue surfaces in order to gain a 3D understanding of the organ placement in the abdominal cavity. Guiding and delivering the robot to challenging targets like the pancreas can benefit from an increase in the situational awareness via the mapping of surrounding moving surfaces. The result in Fig. 6.10 shows that the successfully mapped features.

Features on the surface of the liver of the porcine model were extracted using a custom

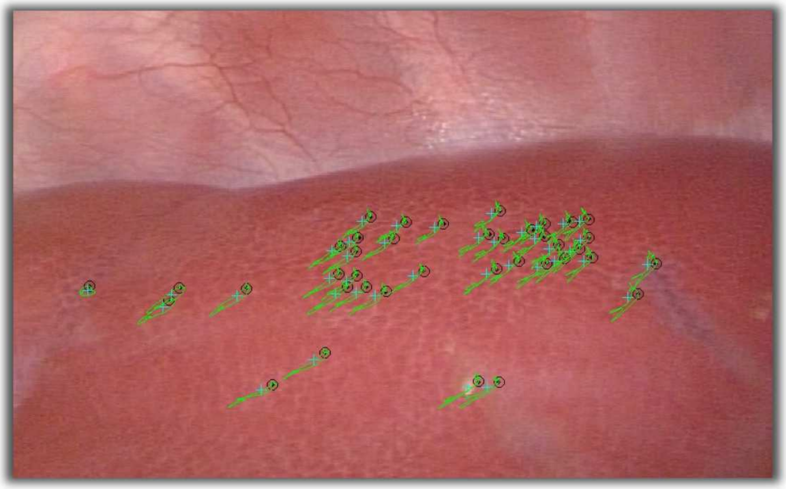


Figure 6.10: The liver of a porcine model with extracted visual features.

algorithm that detects peaks in the difference of Gaussian images that are created by subtracting an image smoothed by one parameters by another image smoothed by another. Unfortunately, for this experiment, the data that we collected did not offer enough parallax to map the full 3D solution of the periodic motion. In the future we will need to record data from the robot itself in such a way that the robot acquires video data of a target from two different vantage points. The result, though, is the accurate mapping of periodically moving features on the liver of a porcine model within the image-plane of the camera. This is still significant because it demonstrates the ability to extend this work to real experimental data.

6.4 Discussion

The periodic feature estimation algorithm that we have introduced is a novel method for inferring the motion of the organ surfaces surrounding the robot. This can be beneficial when attempting to annotate preoperative surface models with live-data that is acquired during an experiment. The advantage of this mapping approach is that it can potentially help our constrained filtering approach. By improving the robot's understanding of the surrounding environment, it will improve the use of geometric constraints that are defined by the positioning of the surrounding surfaces.

The future work of this component of the thesis is to extend the work to real experimental data that includes multiple video sources recording a single target with enough parallax to resolve the motion along the depth dimension of the camera. It is worth noting that the feature mapping aspect of this work has been presented in a way that it is decoupled from the localization of the robot. In the future, it will be advantageous to jointly estimate the two within the same Kalman filter framework. This will translate more appropriately with the theory from SLAM. In this case, we believe that the mapping of periodic features will provide significant environment information that is informative enough to help infer the location of the robot within the Kalman state vector.

Chapter 7

Conclusion

In this thesis, we have introduced a probabilistic approach to the problem of surgical estimation for image-guidance. We call our approach BodySLAM and at a fundamental level, BodySLAM relies on the Kalman filter. The first algorithm that we discussed within our overall vision of BodySLAM was a nonlinear filtering technique that is used to estimate the shape and pose of a highly articulated robotic surgical device. The second algorithm introduced an inequality constrained filtering approach to correct the registration and shape parameters during instances when the robot is discovered to be in an infeasible state given rigid preoperative surface models. Then, we introduced an equality constrained filtering algorithm that constrains the estimation problem when there is forceful contact detected between the robot and the environment. Lastly, the final algorithm of our BodySLAM approach that we introduced in this thesis was a new mapping procedure based on Fourier series parameters to estimate the motion of periodically moving organs.

7.1 Theoretical Contributions

In this thesis, we have introduced several novel algorithms for the application of BodySLAM. The theoretical contribution of our snake shape estimation method is the introduction of a new algorithm to the robotics community that can perform full shape estimation of a surgical tool with only measurements at the distal tip of the robot from a tracking system. The algorithm is fully observable and can accurately track the shape during a live experiment. Likewise, the theoretical contribution of our inequality constrained approach is a new method called uncertainty projection for inequality constrained Kalman filtering. Also, the idea of using constraints to automatically adjust registration parameters during a surgical procedure is a novel approach. The introduction of an equality constrained filtering algorithm that attempts to create a force balance equation is new. We believe that our iterative solution to this problem is a contribution for computing the registration of a robot within a deformable environment. The theoretical contribution of our periodic filtering algorithm is the first extension of bearing-only SLAM to generalized periodic motion estimation.

7.2 Perspectives

7.2.1 Iterative Filtering

Throughout this thesis, we have discussed nonlinear filtering theory and its application to surgical estimation. The Kalman filter is a powerful algorithm for performing state estimation in the presence of observation and process noise. It is efficient for high degree of freedom systems and, as we have shown, can be extended to account for constraints imposed by the geometric relationship between the robot and the surrounding environment.

In our opinion, it has become increasingly apparent that researchers occasionally have the tendency to avoid the Kalman filter as a means for performing nonlinear filtering, and instead have turned to other techniques such as particle filtering or a sample-based unscented version of the Kalman filter. The reason that people are beginning to avoid the Kalman filter for nonlinear problems is that the EKF can often diverge for highly nonlinear problems.

But throughout the development of our BodySLAM approach, we have gained a different perspective on the Kalman filter and its use with nonlinear filtering problems. The EKF definitely has its drawbacks due to nonlinearity and linearization error, but this does not mean that the fundamental Kalman filter framework itself is flawed. The idea of using iteration within the Kalman filter measurement update step has allowed us to realize that the measurement update of the Kalman filter, at a fundamental level, is really an optimization problem over a nonlinear cost function. This optimization problem should be iterated until convergence for proper estimation of the state. The use of iteration prevents divergence of the filter and guarantees at each time step that the updated state estimate has a higher likelihood than the previous time step.

7.2.2 Incorporating All Information

The central idea of image-guidance is to use indirect sources of information that are available in order to create the most realistic visualization possible for surgical feedback. One perspective that we have gained through the development of our BodySLAM approach is that using all sources of available information is important for obtaining a truly representative image-guidance system. This is particularly true when considering that there may be constraints on the system that can provide valuable information to the estimation process.

7.2.3 Sensor-Fusion

Another perspective that we have gained throughout the development of this thesis project is the realization that both SLAM and filtering algorithms for image-guidance are, at a fundamental level, a sensor-fusion problem. The true problem of BodySLAM is how to properly combine all of the information from preoperative medical imaging, live tool tracking, kinematic models, geometric constraints, tissue stiffness models, patient specific hemodynamic readings, robot compliance models, and force measurements. We believe that with all of this information, it will be possible to truly replace the "bird's eye view" that a surgeon is accustomed to with open surgery, but the remaining question is how to combine this information in an intelligent way to produce the a unified visualization for image-guidance. Our approach to BodySLAM is an important step towards this future goal as it includes ways to combine sensor data with geometric constraints and robot kinematic models. For future implementations of BodySLAM to become more advanced, a sensor-fusion algorithm will need to be developed to include even more sources of information.

7.3 Potential Impact and Future Work

The impact of this work is in the future development of image-guidance systems for surgical applications. We believe that with or without the adoption of our BodySLAM approach, image-guidance systems will begin to trend towards the adoption of probabilistic filtering as a means to achieve improved guidance and accuracy due to the use of stochastic models. The idea of BodySLAM, as we presented it, is not complete, but we believe that it can provide a framework that other researchers can build upon to extend these ideas for future image-guidance solutions. Additionally, extending our BodySLAM approach to different robots, different sensing models, and different tracking devices should be a straightforward extension of this work.

Our future work is to combine the filtering algorithms that we have introduced in this thesis into a unified filtering framework. A unified system would benefit the accuracy of the overall approach and would demonstrate the integration of these filtering ideas. The periodic mapping algorithm would benefit from the estimates produced by the localization aspect of BodySLAM as it would continuously improve the geometric constraints that we're using for constrained filtering. On top of that, the localization results can in turn benefit from the periodic mapping algorithm.

Appendix A

Appendix

A.1 Snake Shape Estimation Steering Derivation

For the snake shape estimation filtering algorithm, presented in Chapter 3, we have derived a steering model that computes the offset angles $\phi_k^{N–1}$ and $\theta_k^{N–1}$ for the distal link, indexed N–1, given the differential cable lengths (c_1,c_2,c_3). As discussed in Chapter 3, the parameter θ_k^i defines the angle at which link i is steered away from its nominal orientation and the angle ϕ_k^i defines the magnitude of this orientation change.

The equations that we have derived for the steering model are as follows,

$$\theta_k^{N–1} = \arctan\left(\frac{\sqrt{3}(2c_2 + c_1)}{3c_1}\right) \tag{A.1}$$

$$\phi_k^{N–1} = \left|\ \arcsin\left(\frac{-c_1}{C_R\cos(\theta_k^{N–1})}\right)\ \right|, \tag{A.2}$$

where C_R is a radius term and $(c_1,\, c_2)$ are the measured differential lengths of each of two cables running down the robot.

This model comes from the approximation that the interaction of two links of the HARP robot for steering behaves similarly to a 3-DOF robotic platform (see the depiction in Fig. A.1). For the HARP, as one cable is lengthened and one cable is shortened, the link will be oriented at an angle that follows the geometric model given by the 3-DOF platform.

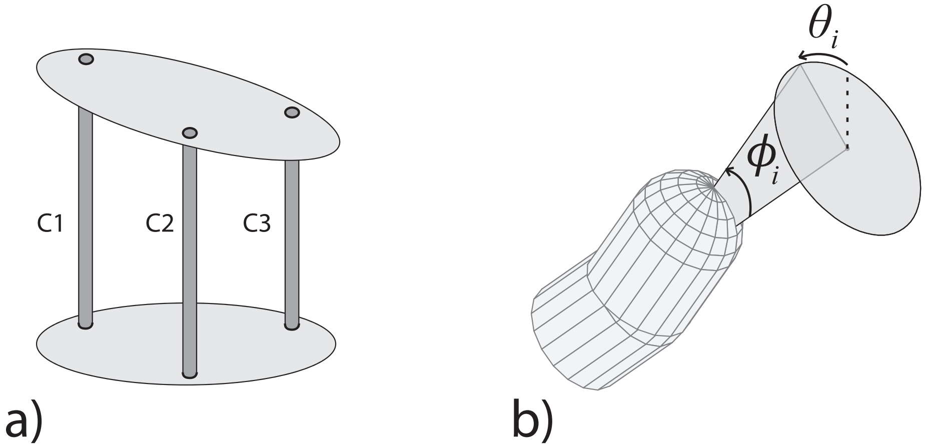


Figure A.1: In (a), a 3-DOF robotic platform is analogous, in terms of steering, with the links of the HARP robot, shown in (b).

A.2 Inequality Constrained Kalman Filtering

A.2.1 Closed form Constrained Filtering Derivation

To perform the uncertainty projection update that we introduced in Chapter 4, as defined in Sec. 4.1, we can write the definition for the mean and variance by evaluating the expectation, as follows,

$$\begin{aligned}\mu_i &= \int_{\beta_i}^{-\infty} f(x)\beta_i dx + \int_{\infty}^{\beta_i} f(x)x dx \\ \sigma_i^2 &= \int_{\beta_i}^{-\infty} f(x)(\beta_i - \mu_i)^2 dx + \int_{\infty}^{\beta_i} f(x)(x-\mu_i)^2 dx,\end{aligned}$$

where $f(x)$, in this case, is the probability density function for a scalar random variable that is zero-mean and has a variance of 1, as follows,

$$f(x) = \frac{1}{\sqrt{2\pi}} e^{-\frac{1}{2}x^2}.$$

This formulation involves two integrals: one that is associated with the infeasible region and one that is associated with the feasible region, with β_i being the point of the constraint. The closed form solution, which can be computed by evaluating the integrals is,

$$\begin{aligned}
\gamma_i &= -2\pi - 2\beta_i e^{\frac{-\beta_i^2}{2}}\sqrt{2\pi} \\
\mu_i &= \frac{1}{\sqrt{2\pi}} e^{\frac{-\beta_i^2}{2}} + \frac{1}{2}\beta_i\left[1 + \mathrm{Erf}\left(\frac{\beta_i}{\sqrt{2}}\right)\right] \\
\sigma_i^2 &= \frac{1}{4\pi}\left[-2e^{-\beta_i^2} + (2+\beta_i^2)\pi + \gamma_i \mathrm{Erf}\left(\frac{\beta_i}{\sqrt{2}}\right) - \beta_i^2\pi\mathrm{Erf}\left(\frac{\beta_i}{\sqrt{2}}\right)^2\right].
\end{aligned} \tag{A.3}$$

A.2.2 Guaranteed Uncertainty Reduction for Linear Constraints

When performing the uncertainty projection update that we introduced in Chapter 4, as defined in Sec. 4.1, it is important that the update procedure reduces the uncertainty of the state. This is a requirement for adopting our method, given that it is beneficial to the estimation process when information is added via the application of geometric constraints within the filtering problem.

The single dimension variance that we defined in Sec. 4.1, can be written in terms of an integral, as follows,

$$\sigma_i^2 \;=\; \int_{\beta_i}^{-\infty} f(x)(\beta_i - \mu_i)^2 dx \;+\; \int_{\infty}^{\beta_i} f(x)(x - \mu_i)^2 dx,$$

This defines the variance of a single random variable when updated for a single inequality constraint at β_i. It is worth noting that the variance σ_i^2 must be less than one. This is due to the fact that a tail of the PDF has been projected onto the point β_i, thus reducing the range of the PDF. It follows that, with σ_i^2 being less than one, the covariance in the transformed state space,

$$\mathbf{\Sigma}_i \;=\; \mathrm{diag}(\sigma_i^2, 1,\; ...\;, 1),$$

must be less than the identity matrix given that all of the diagonal terms are 1 except for the first term that is less than 1. The resulting covariance matrix when transformed back into the original state space is defined as follows,

$$\mathbf{C}_i \;=\; \mathbf{T}\mathbf{W}^{1/2}\mathbf{V}^T\mathbf{\Sigma}_i\mathbf{V}\mathbf{W}^{1/2}\mathbf{T}^T.$$

This equation will equal the previous covariance when the matrix $\mathbf{\Sigma}_i$ is equal to the identity matrix. But we have already shown that $\mathbf{\Sigma}_i$ is less than the identity matrix. Thus, it can be inferred that the uncertainty in the system is guaranteed to be reduced whenever a constraint is applied using our uncertainty projection method.

A.2.3 Uncertainty Projected Mean in the Feasible Region

For the linear uncertainty projection approach, we can show that the mean is guaranteed to move into the feasible region upon updating with the geometric constraint applied to the filtering problem.

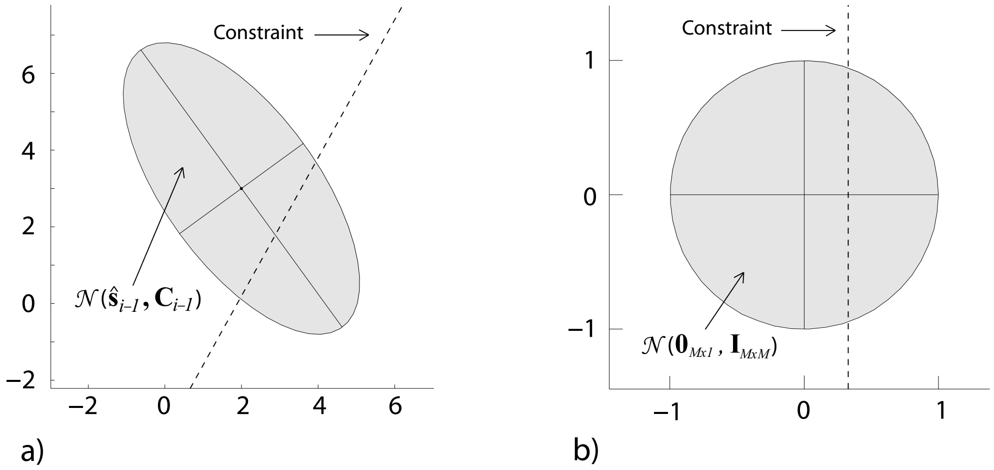

Constraint
6
4
2
0
−2
−2
0
2
4
6
$\mathcal{N}(\hat{\mathbf{s}}_{i-1}, \mathbf{C}_{i-1})$
a)
Constraint
1
0
−1
−1
0
1
$\mathcal{N}(\mathbf{0}_{Mx1}, \mathbf{I}_{MxM})$
b)

Figure A.2: Shown here is an example of the transformation that decouples the constraint so that it only applies to a single dimension of the state space. In (a), the original estimate is shown with its associated covariance matrix and in (b) the new transformed estimate is shown.

Fig. A.2-(a) shows an example of a constrained filtering problem in which a constraint hyper-plane divides the state space into feasible states and infeasible states. In Fig. A.2-

(b), we show an intermediate step of our constrained filtering problem that transforms the problem so that the transformed random variable has white Gaussian noise and only the first dimension of the state is constrained. We note that, no matter what constraint is enforced within the original state space, the problem can be transformed this way.

All points in the transformed state space to the right of the constraint correspond one-to-one to points in the feasible region in the original state space and likewise all points to the left in the transformed space correspond one-to-one to infeasible points in the original state space. Thus, if we can show that the new mean in the transformed state space is guaranteed to be to the right of the constraint, then the new mean that results from the uncertainty projection algorithm is guaranteed to be in the feasible region. To show this, we can look at the equation for the mean of the transformed state space,

$$\mu_i \;=\; \int_{-\infty}^{\beta_i} f(x)\beta_i dx \;+\; \int_{\beta_i}^{\infty} f(x)x dx.$$

This formulation defines μ_i in terms of expectation with integrals over the PDF. This equation can be manipulated as follows,

$$\begin{aligned}
\mu_i &= \int_{-\infty}^{\beta_i} f(x)\beta_i dx \;+\; \int_{\beta_i}^{\infty} f(x)(x-\beta_i)dx \;+\; \int_{\beta_i}^{\infty} f(x)\beta_i dx \\
\mu_i &= \int_{-\infty}^{\infty} f(x)\beta_i dx \;+\; \int_{\beta_i}^{\infty} f(x)(x-\beta_i)dx \\
\mu_i &= \beta_i \;+\; \underbrace{\int_{\beta_i}^{\infty} f(x)(x-\beta_i)dx}_{w(\beta_i)}. && \text{(A.4)} \\
& && \text{(A.5)}
\end{aligned}$$

We note that the term $w(\beta_i)$ in Eq. A.4 must be greater than 0. This is because $f(x)$ is always positive and $(x-\beta_i)$ is always positive for the range of values that the integral is evaluated for. This means that μ_i is greater than β_i and thus the mean in the transformed state space is within the feasible region. Thus, the mean of our uncertainty projection approach is guaranteed to lie in the feasible region.

A.2.4 Guaranteed Uncertainty Reduction for Nonlinear Constraints

When performing the constrained filtering algorithm that we introduced in Chapter 4 for nonlinear constraints, as defined in Sec. 4.2, it is important that the update procedure reduces the uncertainty of the state. This is a requirement for adopting our method, given that it is beneficial to the estimation process when information is added via the application of geometric constraints within the filtering problem.

The final derived covariance update equation that we provided in Sec. 4.2 that defines the solution for updating the covariance $\mathbf{P}_{k|k}$ given a constraint, is as follows,

$$= \mathbf{P}_{k|k} - \mathbf{P}_{k|k}\mathbf{A}_k^T \left(\mathbf{C}_0^{-1} - \mathbf{C}_0^{-1}\mathbf{C}_L\mathbf{C}_0^{-1}\right)\mathbf{A}_k\mathbf{P}_{k|k}. \tag{A.6}$$

It is important to note that Eq. A.6 represents the prior covariance $\mathbf{P}_{k|k}$, which is a positive definite matrix, subtracted by another positive definite matrix. For this reason, the constrained filtering approach that we are introducing here is guaranteed to reduce the uncertainty in the system as long as $\mathbf{C}_L$ is unequal to $\mathbf{C}_0$, and this will always be the case with our uncertainty projection method as we have shown previously that the uncertainty for a linear problem is guaranteed to reduce after the application of a geometric constraint.

Bibliography

[1] A.T. Alouani and W.D. Blair. Use of a kinematic constraint in tracking constant speed, maneuvering targets. *IEEE Transactions on Automatic Control*, 38(7):1107–1111, July 1993. 4, 5, 5.1.1, 5.1.2

[2] T. Bailey. Constrained initialisation for bearing-only SLAM. In *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, volume 2, pages 1966–1971, September 2003. 2.7

[3] T. Bailey and H. Durrant-Whyte. Simultaneous localization and mapping (slam): part II. *IEEE Robotics Automation Magazine*, 13(3):108–117, September 2006. 2.7

[4] A. Bajo, R.E. Goldman, and N. Simaan. Configuration and joint feedback for enhanced performance of multi-segment continuum robots. In *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, pages 2905–2912, May 2011. 2.1, 5, 5.3

[5] A. Bajo and N. Simaan. Finding lost wrenches: Using continuum robots for contact detection and estimation of contact location. In *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, pages 3666–3673, May 2010. 2.1, 5, 5.3

[6] K. R. Beevers and W. H. Huang. Inferring and enforcing relative constraints in SLAM. In *Proceedings of the 7th International Workshop on the Algorithmic Foundations of Robotics (WAFR)*, 2006. 5

[7] B.M. Bell and F.W. Cathey. The iterated Kalman filter update as a Gauss-Newton method. *IEEE Transactions on Automatic Control*, 38(2):294–297, 1993. 2.7, 5.1.2,

6.2.3

[8] J.W. Cannon, J.A. Stoll, I.S. Salgo, H.B. Knowles, R.D. Howe, P.E. Dupont, G.R. Marx, and P.J. del Nido. Real time 3-dimensional ultrasound for guiding surgical tasks. *Computer Aided Surgery*, 8(2):82–90, 2003. 2.3

[9] T. Chia, P. Chow, and H.J. Chizeck. Recursive parameter identification of constrained systems: An application to electrically stimulated muscle. *IEEE Transactions Biomedical Engineering*, 38(5):429–442, May 1991. 5

[10] J. Civera, A.J. Davison, and J.M.M. Montiel. Inverse depth to depth conversion for monocular SLAM. In *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, April 2007. 2.7

[11] K. Cleary, H. Zhang, N. Glossop, E. Levy, B. Wood, and F. Banovac. Electromagnetic tracking for image-guided abdominal procedures: Overall system and technical issues. In *Proceedings of the IEEE International Conference on Engineering in Medicine and Biology Society (EMBS)*, pages 6748–6753, 2005. 2.4, 2.4

[12] A. Davison. Real time simultaneous localisation and mapping with a single camera. In *Proceedings of the International Conference on Computer Vision (ICCV)*, July 2003. 2.7

[13] J. De Geeter, H. Van Brussel, J. De Schutter, and M. Decreton. A smoothly constrained Kalman filter. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 19(10):1171–1177, October 1997. 5

[14] M. Deans and M. Hebert. Experimental comparison of techniques for localization and mapping using a bearings only sensor. In *Proceedings of the 7th International Symposium on Experimental Robotics*, December 2000. 2.7

[15] A. Degani, H. Choset, A. Wolf, T. Ota, and M.A. Zenati. Percutaneous intrapericardial interventions using a highly articulated robotic probe. In *Proceedings of the IEEE/RAS-EMBS International Conference on Biomedical Robotics and Biomechatronics*, pages 7–12, Feb 2006. 1.2.1, 2.2, 3, 4.3

[16] A. Degani, H. Choset, A. Wolf, and M.A. Zenati. Highly articulated robotic probe for

minimally invasive surgery. In *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, pages 4167–4172, May 2006. 1.2.1, 2.2, 3, 4.3

[17] M. Dissanayake, P. Newman, H. Durrant-Whyte, S. Clark, and M. Csorba. A solution to the simultaneous localisation and map building (SLAM) problem. *IEEE Transactions on Robotics and Automation*, 17(3):229–241, June 2001. 2.7, 2.7

[18] J. Dong, H. Calkins, S.B. Solomon, S. Lai, D. Dalal, A. Lardo, E. Brem, A. Preiss, R.D. Berger, H. Halperin, and T. Dickfeld. Integrated electroanatomic mapping with three-dimensional computed tomographic images for real-time guided ablations. *Circulation*, 113:186–194, 2006. 2.4

[19] H.E. Doran. Constraining Kalman filter and smoothing estimates to satisfy time-varying restrictions. *The Review of Economics and Statistics*, 74(3):568–572, 1992. 5

[20] H. Durrant-Whyte and T. Bailey. Simultaneous localization and mapping: part I. *IEEE Robotics and Automation Magazine*, pages 99–108, June 2006. 2.7

[21] M.J. Earley, R. Showkathali, M. Alzetani, P.M. Kistler, D. Gupta, D.J. Abrams, J.A. Horrocks, S.J. Harris, S.C. Sporton, and R.J. Schilling. Radiofrequency ablation of arrhythmias guided by non-fluoroscopic catheter location: a prospective randomized trial. *European Heart Journal*, 27(10):1223–1229, 2006. 2.4

[22] R.D. Ernst, H.S. Kim, A. Kawashima, M.R. Middlebrook, and C.M. Sandler. Near real-time CT fluoroscopy using computer automated scan technology in nonvascular interventional procedures. *American Journal of Roentgenology*, 174(2):319–321, 2000. 1.1, 2.3

[23] K.H. Fuchs. Minimally invasive surgery. *Endoscopy*, 34:154–159, 2002. 1

[24] K. Gary, L. Ibanez, S. Aylward, D. Gobbi, M.B. Blake, and K. Cleary. IGSTK: an open source software toolkit for image-guided surgery. *Computer*, 39(4):46–53, April 2006. 2.4

[25] O. G. Grasa, J. Civera, A. Guemes, V. Munoz, and J.M.M. Montiel. EKF monocular SLAM 3D modeling, measuring and augmented reality from endoscope image

sequence. In *Medical Image Computing and Computer-Assisted Intervention (MICCAI)*, 2009. 2.5

[26] N. Gupta. Kalman filtering in the presence of state space equality constraints. In *Chinese Control Conference*, pages 107–113, July 2007. 4, 5.1.1

[27] K.A. Horvath, M. Li, D. Mazilu, M.A. Guttman, and E.R. McVeigh. Real-time magnetic resonance imaging guidance for cardiovascular procedures. *Seminars in Thoracic and Cardiovascular Surgery*, 19(4):330–335, 2007. 2.3

[28] L. Joskowicz, M. Sati, and L.P. Nolte. Fluoroscopy-based navigation in computer-aided orthopaedic surgery. In *Proceedings of the IFAC Conference on Mechatronic Systems*, 2000. 1.1, 2.3

[29] R.E. Kalman. A new approach to linear filtering and prediction problems. *Journal of Basic Engineering*, 82(1):35–45, 1960. 2.6

[30] R.E. Kalman. On the general theory of control systems. In *Proceedings of the 1st Int. Cong. of IFAC*, volume 1, pages 481–492, 1961. 2.6, 3.2

[31] A. B. Koolwal, F. Barbagli, C. Carlson, and D. Liang. An ultrasound-based localization algorithm for catheter ablation guidance in the left atrium. *The International Journal of Robotics Research*, 29(6):643–665, May 2010. 1.1, 2.3, 2.5

[32] N.M. Kwok, G. Dissanayake, and Q.P. Ha. Bearing-only SLAM using a SPRT based Gaussian sum filter. In *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, April 2005. 2.7

[33] T. Levison, J. Moody, B. Jaramaz, C. Nikou, and A. DiGioia. Surgical navigation for THR: A report on clinical trial utilizing HipNav. In *Medical Image Computing and Computer-Assisted Intervention (MICCAI)*, volume 1935, pages 207–224. 2000. 2.1

[34] M.G. Linguraru, N.V. Vasilyev, P.J. Del Nido, and R.D. Howe. Statistical segmentation of surgical instruments in 3d ultrasound images. *Ultrasound in Medicine and Biology*, 33(9):1428–1437, 2007. 2.3

[35] M.J. Mack. Minimally Invasive and Robotic Surgery. *JAMA: The Journal of the*

American Medical Association, 285(5):568–572, 2001. 1.1

[36] R. Manzke, V. Reddy, S. Dalal, A. Hanekamp, V. Rasche, and R. Chan. Intra-operative volume imaging of the left atrium and pulmonary veins with rotational x-ray angiography. In *Medical Image Computing and Computer-Assisted Intervention (MICCAI)*, pages 604–611, 2006. 2.3

[37] R. Manzke, L. Zagorchev, A. dAvila, A. Thiagalingam, V.Y. Reddy, and R. Chan. Rotational x-ray angiography: a method for intra-operative volume imaging of the left-atrium and pulmonary veins for atrial fibrillation ablation guidance. In *Proceedings SPIE*, pages 1–9, 2007. 2.3

[38] M. Montemerlo, S. Thrun, D. Koller, and B. Wegbreit. FastSLAM: A factored solution to the simultaneous localization and mapping problem. In *Proceedings of the AAAI National Conference on Artificial Intelligence*, 2002. 2.7

[39] T. Moon and W. Stirling. *Mathematical Methods and Algorithms for Signal Processing*. New Jersey: Prentice Hall, Upper Saddle River, 2000. 4.1.2

[40] P. Mountney, D. Stoyanov, A. Davison, and G. Z. Yang. Simultaneous stereoscope localization and soft-tissue mapping for minimal invasive surgery. In *Medical Image Computing and Computer-Assisted Intervention (MICCAI)*, 2006. 2.5

[41] P. Mountney and G. Z. Yang. Motion compensated SLAM for image guided surgery. In *Medical Image Computing and Computer-Assisted Intervention (MICCAI)*, 2010. 6

[42] P. Moutarlier and R. Chatila. An experimental system for incremental environment modeling by an autonomous mobile robot. In *1st International Symposium on Experimental Robotics*, June 1989. 2.7

[43] H. Nakagawa and W.M. Jackman. Use of a three-dimensional, nonfluoroscopic mapping system for catheter ablation of typical atrial flutter. *Pacing and Clinical Electrophysiology*, 21(6):1279–1286, 1998. 2.4

[44] S. Nazarian, A. Kolandaivelu, M.M. Zviman, G.R. Meininger, R. Kato, R.C. Susil, A. Roguin, T.L. Dickfeld, H. Ashikaga, H. Calkins, R.D. Berger, D.A. Bluemke, A.C.

Lardo, and H.R. Halperin. Feasibility of real-time magnetic resonance imaging for catheter guidance in electrophysiology studies. *Circulation*, 118(3):223–229, 2008. 2.3

[45] D. Noonan, P. Mountney, D. Elson, A. Darzi, and G. Z. Yang. A stereoscopic fibroscope for camera motion and 3D depth recovery during minimally invasive surgery. In *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, 2009. 2.5

[46] P.M. Novotny, J.A. Stoll, P.E. Dupont, and R.D. Howe. Real-time visual servoing of a robot using three-dimensional ultrasound. In *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, pages 2655–2660, 2007. 2.3

[47] R.A. Omary, J.D. Green, B.E. Schirf, Y. Li, J.P. Finn, and D. Li. Real-time magnetic resonance imaging-guided coronary catheterization in swine. *Circulation*, 107(21): 2656–2659, 2003. 1.1, 2.3

[48] T. Ota, A. Degani, D. Schwartzman, B. Zubiate, J. McGarvey, H. Choset, and M.A. Zenati. A novel highly articulated robotic surgical system for epicardial ablation. In *Proceedings of the IEEE International Conference on Engineering in Medicine and Biology Society (EMBS)*, pages 250–253, Aug 2008. 2.2, 4.3

[49] Y-L. Park, S. Elayaperumal, B.L. Daniel, E. Kaye, K.B. Pauly, R.J. Black, and M.R. Cutkosky. MRI-compatible haptics: Feasibility of using optical fiber bragg grating strain-sensors to detect deflection of needles in an MRI environment. In *Proceedings of the International Society for Magnetic Resonance in Medicine (ISMRM)*, 2008. 3

[50] T. Peters, B. Davey, P. Munger, R. Comeau, A. Evans, and A. Olivier. Three-dimensional multimodal image-guidance for neurosurgery. *IEEE Transactions on Medical Imaging*, 15(2):121–128, April 1996. 2.4

[51] B.D. Powell and D.L. Packer. Does image integration improve atrial fibrillation ablation outcomes, or are other aspects of the ablation the key to success? *Europace*, 11 (8):973–974, 2009. 1, 1.1.2

[52] R. Richa, A.P.L. Bo, and P. Poignet. Beating heart motion prediction for robust visual tracking. In *Proceedings of the IEEE International Conference on Robotics and*

Automation (ICRA), pages 4579–4584, May 2010. 6

[53] R. Richa, P. Poignet, and C. Liu. Efficient 3D tracking for motion compensation in beating heart surgery. In *Medical Image Computing and Computer-Assisted Intervention (MICCAI)*, volume 5242, pages 684–691. 2008. 6

[54] R. Richa, P. Poignet, and C. Liu. Three-dimensional motion tracking for beating heart surgery using a thin-plate spline deformable model. *The International Journal of Robotics Research*, 29(2-3):218–230, 2010. 6

[55] N. Shimada, Y. Shirai, Y. Kuno, and J. Miura. Hand gesture estimation and model refinement using monocular camera - ambiguity limitation by inequality constraints. In *Proceedings of the International Conference on Automatic Face and Gesture Recognition*, pages 268–273, 1998. 4, 4.1.2

[56] T.B. Shope. Radiation-induced skin injuries from fluoroscopy. *Radiographics*, 16(5): 1195–1199, 1996. 1.1, 2.3

[57] D. Simon. Kalman filtering with state constraints: a survey of linear and nonlinear algorithms. *Control Theory Applications, IET*, 4(8):1303–1318, August 2010. 4, 5.1

[58] D. Simon and T. Chia. Kalman filtering with state equality constraints. *IEEE Transactions on Aerospace and Electronic Systems*, 39(1):128–136, January 2002. 4, 5, 5

[59] D. Simon and D.L. Simon. Constrained Kalman filtering via density function truncation for turbofan engine health estimation. *International Journal of Systems Science*, 41(2):159–171, February 2010. 4, 4.1.1, 4.1.2, 4.1.2, 4.1.3, 4.1.3

[60] R. Smith, M. Self, and P. Cheeseman. Estimating uncertain spatial relationships in robotics. *Autonomous Robot Vehicles, Springer*, 1990. 2.7, 2.7

[61] J. Solà, A. Monin, M. Devy, and T. Lemaire. Undelayed initialization in bearing only SLAM. In *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, April 2005. 2.7

[62] W.G. Stevenson, E. Delacretaz, P.L. Friedman, and K.E. Ellison. Identification and ablation of macroreentrant ventricular tachycardia with the CARTO electroanatomical

mapping system. *Pacing and Clinical Electrophysiology*, 21(7):1448–1456, 1998. 2.4

[63] M. Tahk and J.L. Speyer. Target tracking problems subject to kinematic constraints. *IEEE Transactions on Automatic Control*, 35(3):324–326, March 1990. 4, 5, 5.1.1, 5.1.2

[64] B.O.S. Teixeira, J. Chandrasekar, L.A.B. Torres, L.A. Aguirre, and D.S. Bernstein. State estimation for equality-constrained linear systems. In *Proceedings of the IEEE Conference on Decision and Control*, pages 6220–6225, December 2007. 4, 5.1.1

[65] B.O.S. Teixeira, L.A.B. Torres, L.A. Aguirre, and D.S. Bernstein. On unscented Kalman filtering with state interval constraints. *Journal of Process Control*, 20(1): 45–57, 2010. 4

[66] A. Thiagalingam, R. Manzke, A. dAvila, I. Ho, A.H. Locke, J.N. Ruskin, R.C. Chan, and V.Y. Reddy. Intraprocedural volume imaging of the left atrium and pulmonary veins with rotational x-ray angiography: implications for catheter ablation of atrial fibrillation. *Journal of Cardiovascular Electrophysiology*, 19(3):293–300, 2008. 2.3

[67] S. Tully, A. Bajo, G. Kantor, H. Choset, and N. Simaan. Constrained filtering with contact detection data for the localization and registration of continuum robots in flexible environments. In *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, May 2012. 5

[68] S. Tully, G. Kantor, and H. Choset. A single-step maximum a posteriori update for bearing-only SLAM. In *Proceedings of the 2010 AAAI Conference on Artificial Intelligence*, July 2010. 2.7

[69] S. Tully, G. Kantor, and H. Choset. Inequality constrained Kalman filtering for the localization and registration of a surgical robot. In *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, September 2011. 4

[70] S. Tully, G. Kantor, M.A. Zenati, and H. Choset. Shape estimation for image-guided surgery with a highly articulated snake robot. In *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 1353–1358,

Sept. 2011. 3

[71] S. Tully, H. Moon, G. Kantor, and H. Choset. Iterated filters for bearing-only SLAM. In *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, May 2008. 2.7, 2.7

[72] P. Vachhani, S. Narasimhan, and R. Rengaswamy. Robust and reliable estimation via unscented recursive nonlinear dynamic data reconciliation. *Journal of Process Control*, 16(10):1075–1086, 2006. 4

[73] M.-A. Vitrani, G. Morel, and T. Ortmaier. Automatic guidance of a surgical instrument with ultrasound based visual servoing. In *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, pages 508–513, April 2005. 1.1, 2.3

[74] L.S. Wang, Y.T. Chiang, and F.R. Chang. Filtering method for nonlinear systems with constraints. *Control Theory and Applications, IEEE Proceedings*, 149(6):525–531, November 2002. 4, 5.1.1, 5.1.2, 5.1.2

[75] W. Wein, A. Khamene, D.A. Clevert, O. Kutter, and N. Navab. Simulation and fully automatic multimodal registration of medical ultrasound. In *Medical Image Computing and Computer-Assisted Intervention (MICCAI)*, pages 136–143, 2007. 2.4, 2.4

[76] W. Wen and H.F. Durrant-Whyte. Model-based multi-sensor data fusion. In *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, May 1992. 5

[77] C. Yang and E. Blasch. Kalman filtering with nonlinear state constraints. In *Proceedings of ISIF, Fusion 2007*, 2007. 5

[78] X. Yi, J. Qian, L. Shen, Y. Zhang, and Z. Zhang. An innovative 3D colonoscope shape sensing sensor based on FBG sensor array. In *Proceedings of the International Conference on Information Acquisition (ICIA)*, pages 227–232, July 2007. 3

[79] L. Zhang, J. Qian, Y. Zhang, and L. Shen. On SDM/WDM FBG sensor net for shape detection of endoscope. In *Proceedings of the IEEE International Conference on Mechatronics and Automation*, volume 4, pages 1986–1991, 2005. 3

[80] W.Y. Zhang, R.N. Rohling, and D.K. Pai. Surface extraction with a three-dimensional freehand ultrasound system. *Ultrasound in Medicine and Biology*, 30(11):1461–1473, 2004. 2.3

[81] H. Zhong, T. Kanade, and D. Schwartzman. Sensor guided ablation procedure of left atrial endocardium. In *Medical Image Computing and Computer-Assisted Intervention (MICCAI)*, pages 1–8, 2005. 2.4