A MINIMALLY INVASIVE SURGICAL SYSTEM FOR 3D ULTRASOUND GUIDED ROBOTIC RETRIEVAL OF FOREIGN BODIES FROM A BEATING HEART

by

Paul Thienphrapa

A dissertation submitted to The Johns Hopkins University in conformity with the requirements for the degree of Doctor of Philosophy.

Baltimore, Maryland

March, 2014

Abstract

The result of various medical conditions and trauma, foreign bodies in the heart pose a serious health risk as they may interfere with cardiovascular function. Particles such as thrombi, bullet fragments, and shrapnel can become trapped in a person's heart after migrating through the venous system, or by direct penetration. The severity of disruption can range from benign to fatal, with associated symptoms including anxiety, fever, cardiac tamponade, hemorrhage, infection, embolism, arrhythmia, and valve dysfunction. Injuries of this nature are common in both civilian and military populations. For symptomatic cases, conventional treatment is removal of the foreign body through open surgery via a median sternotomy, the use of cardiopulmonary bypass, and a wide incision in the heart muscle; these methods incur pronounced perioperative risks and long recovery periods.

In order to improve upon the standard of care, we propose an image guided robotic system and a corresponding minimally invasive surgical approach. The system employs a dexterous robotic capture device that can maneuver inside the heart through a small incision. Visualization and guidance within the otherwise occluded internal regions are provided by 3D transesophageal echocardiography (TEE), an emerging form of intraoperative medical imaging used in interventions such as mitral valve repair and device implantation.

A robotic approach, as opposed to a manual procedure using rigid instruments, is motivated by the various challenges inherent in minimally invasive surgery, which arise from attempts to perform skilled surgical tasks through small incisions without direct vision. Challenges include reduced dexterity, constrained workspace, limited visualization, and difficult hand-eye coordination, which ultimately lead to poor manipulability. A dexterous robotic end effector with real-time image guidance can help overcome these challenges and potentially improve surgical performance.

However promising, such a system and approach require that several technical hurdles be resolved. The foreign body must be automatically tracked as it travels about the dynamic environment of the heart. The erratically moving particle must then be captured using a dexterous robot that moves much more slowly in comparison. Furthermore, retrieval must be performed under 3D ultrasound guidance, amidst the uncertainties presented by both the turbulent flow and by the imaging modality itself. In addressing such barriers, this thesis explores the development of a prototype system capable of retrieving a foreign body from a beating heart, culminating in a set of demonstrative *in vitro* experiments.

ABSTRACT

Advisor

Russell H. Taylor, Ph.D.

John C. Malone Professor of Computer Science, Johns Hopkins University

Co-Advisors

Peter Kazanzides, Ph.D.

Associate Research Professor of Computer Science, Johns Hopkins University

Aleksandra Popovic, Ph.D.

Senior Member Research Staff, Philips Research North America

Acknowledgments

It is my great fortune and honor to have this opportunity to recognize those individuals that have guided my pursuits, be it directly, indirectly, or in ways yet to be understood. First I thank my Ph.D. advisors Russell Taylor, Peter Kazanzides, and Aleksandra Popovic for their tireless investments in me and my work. Their insights and wisdom have helped me grow as a member of the community, and their passion for advancing surgical technology has inspired me to expand my knowledge and abilities. I appreciate them for providing me the opportunity to work with them in the first place, and for procuring the resources to fuel it. This work was supported in large part by Philips Research North America, with the balance supported by NSF EEC 9731748, NSF MRI 0722943, and Johns Hopkins University internal funds.

Next I thank the Johns Hopkins community at large for its environment encouraging intellectual curiosity and innovation without barriers, whether logistical, cognitive, or otherwise. I thank Cathy Thornton, Debbie DeFord, and Emily Evans of the CS Department; Jamie Meehan, Alison Morrow, Elisa Ahmanson, and Anita Sampath of the LCSR; Felicia Roane of the ECE Department; as well as Walt Krug and Mike

Franckowiak of the Maryland Hall machine shop for their unwavering and patient support of my activities. For devoting clinical expertise from their respective domains I thank School of Medicine scholars Paul Flint, Jeremy Richmon, and Kaushik Mandal, and for graciously volunteering their technical expertise as part of my Graduate Board exam committee, Greg Hager and Louis Whitcomb.

Despite the risk of inevitably committing errors of omission, I must specifically acknowledge the following collaborators, colleagues, and friends for the profound impression each has made on me during my time at Hopkins. In approximately the order in which I discovered them: Dr. Sotiriadis, Hyun, Yi, Ankur, Nd, Fran, Hamid, Kai, Henry, Tomo, Noel, Dr. Okamura, Sue Sue, Ioana, Tian, Marcin, Omar, Babak, Kurt, Chin-Hsing, Brian, Iulian, Anton, Nathan, Min Yang, Tamas, Seth, Wen, Yixin, Xingchi, Eric, Tze-Yuan, Yan Yan, Amy, Shahin, Hongliang, Carol, Kevin, Keith, Kel, Manu, Ryan, Hari, Bharat, Doug, Chris, Emil, Jay, Vijay, Ameet, Andrew, Francois, Haytham, Antonio, Dr. Simaan, Wei Wei, Nishikant, Zihan, Anand, Tutkun, Ehsan, Jade, Simon, Purnima, Lyelin, and Hui. Each of these individuals, and others I fail to recall, have granted me the gift of their time, and in doing so have supported my pursuits, enriched my knowledge, and shaped my thinking. Some interactions have been as direct as entrusting me with their valuable personal property, others have transpired behind the scenes, and everything in between. I would be remiss not to acknowledge the Free Agents, Team Sick Nasty, All But Defense, Baltimore United, and Empty-Netter Headers as well. I thank the city of Baltimore for its shelter and for never letting me feel unwelcome. I was delighted for the city when the Orioles finally clinched a playoff spot, and celebrated in the streets when the Ravens won the Super Bowl. I am proud to call Charm City home.

Thank you Drs. Liu, Dong, Boussalis, Rad, and Warter; and Guy, Craig, Tom, Quinn, and Mike for catapulting me into graduate school via the opportunities preceding it. I would like to credit the open source community for the free software that I use extensively. Thank you Prof. Jenkins, Khan Academy, and Dr. Hoff for the (respectively) FPGA, math, and computer vision lessons that arguably changed my life, and at the very least helped me through degree requirements.

Special thanks go to Leng for her unconditional belief in me as a person, and to John V, Mike, Suk, Tuan, Kendall, Zack, Sal, Binh, Henry, Malisa, Tida, and Sheng for keeping me in their lives despite the perpetual progression of the years.

My greatest gratitudes go to my mom and my sister. Our struggles define my existence, and our accomplishments are what I live for.

Dedication

To my family.

Contents

List of Tables

List of Figures

List of Algorithms

Chapter 1

Introduction

Penetration of a foreign body into the heart is an injury that commonly arises in both civilian and military populations [2, 3]. There are multiple avenues for the development of the condition. Thrombi can appear in the heart following myocardial infarction. Debris can enter the heart due to an iatrogenic cause or by embolization from the venous system after an injury in the torso or extremities. Additionally, external foreign bodies can emerge by direct penetration through the chest. Such a perforating heart injury may cause irreparable damage to the muscle, which is often fatal, while a non-perforating direct injury usually results in a foreign body being lodged in the pericardial wall; however, small caliber bullets and shell fragments with low velocity tend to circulate freely in the chambers. Embolized objects typically circulate in the left ventricle or right atrium, and can subsequently become entrapped in the pericardial trabeculations and fatty tissue [4, 5].

1.1 Clinical Problems and Significance

1.1.1 Background

A direct injury is usually accompanied by severe life-threatening symptoms such as hemorrhage and cardiac tamponade, requiring a fast response to stabilize the patient [2]. Once the patient is stabilized, cardiac surgery is performed to remove foreign objects and, if needed, repair the pericardium. Embolized foreign bodies can be symptomatic or asymptomatic. Asymptomatic foreign bodies are entrapped in the pericardium and are treated conservatively with frequent follow-ups to evaluate if the foreign body has been released [6]. Free-moving foreign bodies have the potential to cause arrhythmia, occlusion, neurotic manifestations, and death, necessitating surgical intervention [4, 5].

The most common surgical approach for removing lodged or freely circulating foreign bodies from the heart is a highly invasive one. First, localization of foreign objects is performed using chest X-ray or ultrasound imaging [3]. A median sternotomy, followed by an incision in the pericardium, is then performed to expose the heart chamber containing the object [2, 6–8]. A left or right thoracotomy is an alternative though less commonly used incision. The difficulty of manipulating cardiac catheters to retrieve moving targets precludes such percutaneous treatment for the vast majority of patients.

Median sternotomy is a risky procedure requiring a long recovery period. The potential risks include bacterial mediastinitis, inflammation of the tissues in the mid-chest, and bone fracture. Moreover, patients with a history of sternotomy are prone to serious injury in the future due to the sternotomy wires and staples [9]. In a standard surgical setting, cardiopulmonary bypass may be used to stop the heart during surgery. The use of cardiopulmonary bypass is associated with extended perioperative time along with risks of hemolysis, clotting, and air embolism.

1.1.2 Summary of Approach

The objective of this work is to develop a minimally invasive surgical system for retrieval of foreign bodies from a beating heart. From a clinical perspective, a minimally invasive approach can help improve the management of cardiac foreign bodies by mitigating the risk of perioperative and postoperative complications associated with sternotomy and cardiopulmonary bypass, and potentially by reducing operating room times as well.

There are, however, a number of technical challenges associated with constructing such a system. The approach suffers from limited surgical access and lack of direct visualization. To combat the issue of limited surgical access, we propose a procedure using a dexterous robotic device that is inserted transapically into the heart, after identification of the foreign body under preoperative imaging. Visualization of the heart is provided by 3D transesophageal echocardiography (TEE). Under intraoperative 3D ultrasound guidance, the robot moves to secure the target. The scenario is illustrated in Figure 1.1. Though not used in the current study, integration with cone beam CT and intraoperative fluoroscopy imaging into the system represents a future direction of interest.

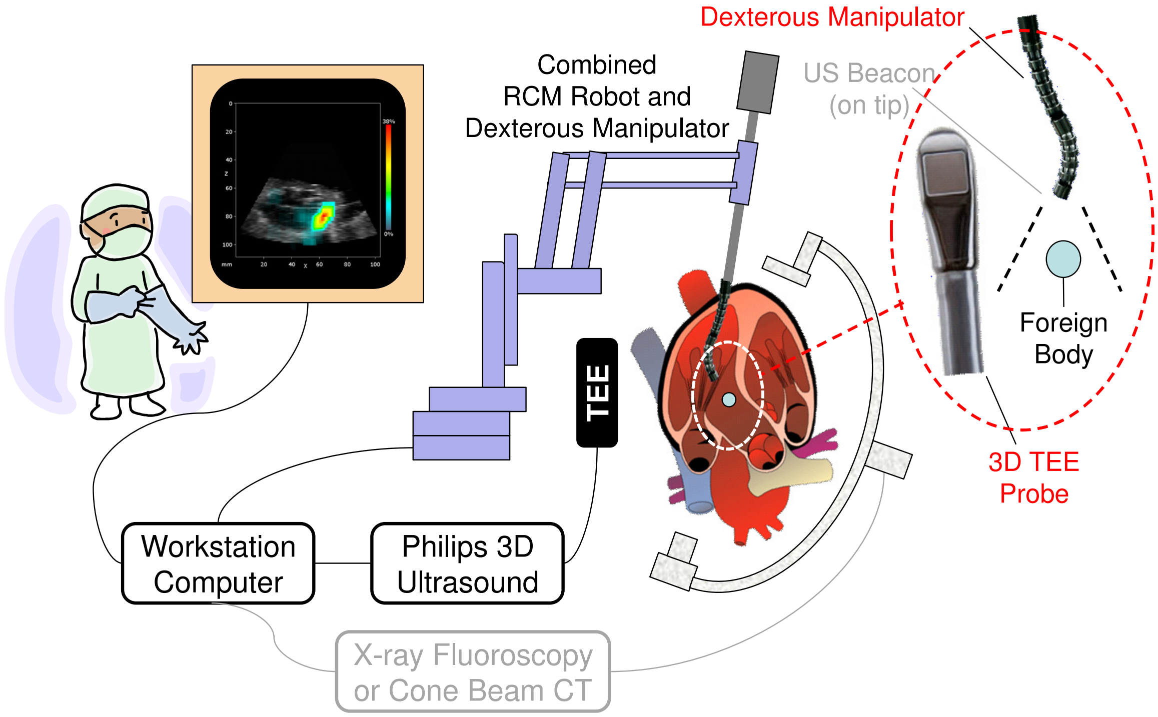


Figure 1.1: Envisioned scenario of minimally invasive robotic retrieval of foreign bodies from the heart under 3D transesophageal echocardiography (TEE) guidance.

Commonly used for diagnosing cardiovascular diseases, 3D TEE is emerging as interventional guidance in cardiac procedures such as mitral valve and atrial septal defect repair, as well as device implantation. It offers the benefits of soft tissue contrast as well as volumetric imaging, thus enhancing the visualization of the cardiac anatomy over B-mode ultrasound. The use of 3D TEE for real-time tracking has the potential to improve the intelligence and accuracy of interventions.

1.1.3 Motivations

Minimally invasive surgery, when available, is an attractive option for patients due to its markedly reduced traumatic effects, both physical and psychological. However, it requires surgeons to perform skilled surgical manipulations through small access ports, using rigid instruments, and without direct vision. Such an arrangement limits the dexterity, workspace, and field of view available for the task, and imposes unnatural hand-eye coordination schemes. These well-documented shortcomings of minimally invasive surgery make it a suitable candidate for robotic assistance.

As an image-guided robotic surgical system, the proposed solution represents a significant technological leap from existing practices, but advances in other areas of computer integrated surgery can be leveraged to make the gap in sophistication more tenable. There are several motivating considerations for pursuing the proposed solution; among them:

- To improve the standard of care for the management of cardiac foreign bodies
- To improve outcomes for both patients and providers
- To highlight new capabilities for potential innovation in various domains

1.1.4 Technical Barriers

From a technological standpoint, this work seeks to extend the existing capabilities of, and introduce new capabilities using, intraoperative imaging and minimally invasive robotic manipulation. The performance requirements of the application highlight the need for real-time acquisition, processing, and analysis of 3D ultrasound images, which are known to suffer from noise, artifacts, distortion, and in general poor image quality relative to other medical imaging modalities such as CT and endoscopy. Furthermore, the computational demand in 3D processing and tracking is significantly greater than in the conventional 2D case. In this application, vision oriented retrieval must be performed amidst the uncertainties presented by the modality.

After considering the difficulties of using ultrasound for image guidance, a number of technical hurdles remain. First, a technique must be devised for tracking the foreign body as it travels about the heart, against a dynamic background. Next, the behavior of the foreign body must be quantified in order to inform the design of a sufficiently functional robotic system. Retrieval strategies based on intraoperative information must then be formulated; the irregular motion of the foreign body in turbulent flow, combined with the slow speed of the robot, magnifies the difficulty of the task. Outlined heretofore are the nonexistent capabilities to be created; meanwhile a versatile dexterous robot and an associated closed-loop control system are the key technological resources that must be engineered.

In addressing these barriers, this thesis explores the development of a prototype system for retrieving a foreign body from a beating heart, culminating in a set of demonstrative *in vitro* experiments. The list below summarizes the intermediate stages of the effort, encapsulating the technical barriers involved.

- Establish an experimental platform for investigation of the problem
- Track a foreign body in a dynamic environment using 3D ultrasound
- Characterize the motion and behavior of cardiac foreign bodies
- Devise strategies for capturing a foreign body in turbulent flow
- Perform autonomous guidance of a dexterous robot using 3D ultrasound
- Ensure real-time performance of the tracking and guidance computations

1.2 Contributions

1.2.1 List of Contributions

The emergence of a variety of ultrasound imaging devices with native 3D capability, such as laparoscopic ultrasound, transthoracic echocardiography (TTE), transesophageal echocardiography (TEE), and intracardiac echocardiography (ICE) probes, encourage the exploration of 3D ultrasound for use in computer integrated interventions. In this effort, the underlying theme is the development of vision, control, and computational strategies that establish 3D ultrasound as an effective real-time tracking and guidance tool in the dynamic environment of a beating heart.

In contrast to related works which simplify the instrument guidance scenario to manual guidance (e.g., [10]), teleoperated robot guidance (e.g., [11]), or motion in one dimension (e.g., [12]), this work details the development of a dexterous robot capable of autonomous guidance inside the heart. This capability can potentially help reduce the surgeon's cognitive load with regards to relating the robot frame of reference to the patient anatomy. In contrast to related works, this requires tracking of surgical targets with a control bandwidth at or near frame rate; in broad terms, all avenues for achieving the requisite performance should be considered, including efficient algorithms, high performance computing hardware, and novel approaches to the problems.

Though beyond the scope of the current effort, an interesting example of the latter category is visual enhancement of targets using an ultrasound transceiver, or via induced vibrations. This can help relax assumptions on the properties of said targets from a vision perspective, paving the way for improved robustness and safer operation. Still, detectable properties unique to particular targets should be exploited. The following list outlines the research contributions embodied in this work.

System and application. A new approach to removing cardiac foreign bodies is proposed, namely a minimally invasive intervention on a beating heart using a high dexterity manipulator guided by 3D ultrasound. This is in contrast to existing practices, which include a highly invasive median sternotomy and cardiopulmonary bypass, or no removal of the foreign body at all due to the significant trauma and risks involved. Non-robotic, minimally invasive treatments (Section 2.1) have been investigated in the past, but these options are limited to foreign bodies that are long and stationary, thus making the targets easy to grab with minimal dexterity. As cardiac foreign body retrieval is a technically demanding procedure, the development of a capable ultrasound guided robotic surgical system would help improve other procedures on the heart, such as mitral valve and atrial septal defect repair, and enable new procedures as well. The outcomes of this specific effort are thus applicable in a more generalized context.

Real-time tracking in 3D ultrasound. The targeted application imposes a variety of new challenges to the development of real-time tracking, guidance, and control systems for foreign body capture. The erratic movement of the foreign body requires that tracking be updated at or near the approximately 20 Hz frame rate of the 3D ultrasound stream; this fast algorithm must perform amidst the dynamic environment that is the beating heart. This is in contrast to the lower tracking rates (e.g., 2–8 Hz) previously reported in 3D ultrasound-based tracking studies (Section 2.3.2.1). Image-based tracking in 3D is a computationally demanding process. Additionally, ultrasound images contain large quantities of noise, have low resolution, and suffer from artifacts. In response to these challenges, we suspend the assumption that each frame of the ultrasound stream is a conventional image with exact types and amounts of information. Instead, we use a time-integrated, statistical view to combat imperfections in the imaging and tracking.

Characterization of cardiac foreign body motion. Tied to the non-existence of an image-guided system for removal of cardiac foreign bodies is the absence of information concerning the motion of such bodies in a live heart. This is to be expected, since knowledge of foreign body motion has not been needed heretofore. The conventional surgical approach provides such invasive access and vision that the surgeon can overwhelm the object with brute force. Furthermore, the use of cardiopulmonary bypass would remove motion altogether, but these are exactly the risk-laden procedures that this work proposes to circumvent. We conduct an analysis of foreign body motion for the purpose of appropriately designing a surgical retrieval system, but the analysis itself can be of general interest.

Vision oriented, minimally invasive capture strategies. The application calls for visual servoing in a turbulent flow that causes the motion of the foreign body to be complex and difficult to predict. Compounding the difficulty of the task, the sensitivity of the surrounding heart tissue prescribes the use of a dexterous robot that moves slowly and deliberately in comparison. Capture of an erratically moving target using a slow device is a new problem in computer integrated surgery research, and solutions proposed to date have circumvented this technical challenge by including a human in the loop, as opposed to using computer vision oriented task execution (Section 2.3.2.2). In devising a set of minimally invasive capture strategies, we define new measurements that can meaningfully assist in the quantification of expected outcomes and the evaluation of the proposed methods. We use a time-integrated, statistical view of the foreign body motion to give the system the ability to learn the behavior of motion that is otherwise arbitrary in appearance, thereby resolving a seemingly contradictory situation.

Autonomous 3D ultrasound guidance of a high dexterity robot. In previous work in robotic cardiac surgery, the dexterity of the robot is limited either by the degrees of freedom (as in the one-DOF heartbeat motion compensation devices) or by nonholonomic constraints (as in the concentric tube robots) that lead to reduced versatility in the paths of the robots (Section 2.2). Other dexterous robots have been investigated that use intraoperative 3D ultrasound, endoscopy, or fluoroscopy for guidance; these have relied on a human in the loop for teleoperation (Section 2.3.2.2). The proposed system uses a high dexterity robot, guided by 3D ultrasound, to attain an enhanced dexterous workspace in the heart from a fixed entry port. The proposed system has sufficient awareness of the environment that the robot can navigate without an intermediary agent.

1.2.2 Broader Impacts

With ongoing advances in robotics, imaging, sensing, and other technologies, there is significant room for improvement in the delivery of healthcare. This work identifies a clinical problem—foreign bodies in the heart—that has an associated treatment approach with pronounced risk—a median sternotomy. In such cases, both patients and providers can benefit from efforts to improve the state of care.

The design of a novel surgical system incorporating emerging technologies demands that technical barriers, such as those outlined in Section 1.1.4, be surmounted. In the course of solving this specific clinical problem, new technologies, methods, and measurements are created. The lessons gleaned may potentially apply to a wider scope of problems, including those in related domains as well as to similar challenges in other domains. Some of the broader impacts of this effort are presented below.

Use of 3D ultrasound to guide interventions. Today, 3D ultrasound is predominantly used as an intermittent visualization tool to verify correct placement of tools and catheters with respect to anatomical areas of interest. In this work, we demonstrate the capabilities of emerging 3D ultrasound in tracking live targets and guiding an intracardiac robot. This forms the basis for further expansion of the role of 3D ultrasound in both intracardiac (e.g., mitral valve repair, atrial septal defect repair) and other (e.g. throat, prostate, abdominal) interventions.

Dexterous guidance of a minimally invasive robot. Straightforward guidance of a dexterous robot can have a transferable impact on other cardiac procedures due to the ability to traverse complex paths, and thus perform tasks with greater dexterity, in a minimally invasive manner.

Characterization of a fast target for capture with a slow robot. Exploration of this problem led to the determination of meaningful parameters and metrics for statistical characterization of erratic targets, for capture using a slow robot, thereby introducing new perspectives on tracking in chaotic environments. This may be useful in tracking heart structures, such as the valves and muscles, for establishing guidance boundaries and for direct contact. More generally, this type of strategy may be useful in tracking animate targets (e.g., insects, birds) and other projectiles. By sacrificing the time needed to build a real-time statistical understanding of a target, it may be possible to relax the speed and acceleration requirements of a robot that would be needed for direct pursuit.

In beating heart mitral valve repair, the state of the art features manual navigation of the straight and rigid NeoChord device [10]. There is a marked improvement in outcomes under augmented 3D ultrasound guidance, but even so the results show that the paths taken by the device are irregular, and furthermore some uncertainty remains in the endgame of clamping down on the valve. By using a dexterous robot guided by 3D ultrasound, a more direct path can be taken because the device can move internally rather than having to pivot about the surgical entry port, and because the registration between the device and the image can be handled automatically by the system rather than cognitively by the interventionalist. Uncertainty in the endgame, i.e., positioning of the device relative to the mitral valve, is also removed by the same mechanism.

A systems-based approach, thematically analogous to a top-down design methodology wherein all components needed for a functional system are developed up front, is helpful in appropriately delineating the subsystems and highlighting those in need of further development. This method may combat the tendency and temptation to over-analyze subproblems and over-engineer solutions, phenomena which can arise when a high-level understanding has yet to be reached.

Chapter 2

Related Work

2.1 Non-Robotic, Minimally Invasive Treatment of Cardiac Foreign Bodies

The conventional approach to removing foreign bodies from the heart involves a highly invasive sternotomy and cardiopulmonary bypass, as these facilities provide surgeons with the maximum of access for a skilled task. Considering the heightened risk and complications of open surgery, it is often elected to leave the condition untreated. Minimally invasive intervention has thus been a subject of interest, even absent such advanced resources as robots and other specialized devices.

The use of catheters to retrieve cardiac foreign bodies has been a popular strategy (e.g., [13–25]) due in part to the availability of and familiarity with cardiac catheters in the clinical space. Catheter-based approaches are attractive due to the minimal surgical footprint, but as revealed through the example efforts below, they require that the foreign body be relatively long and stationary, as catheters are difficult to control precisely whether by hand or by robot.

Iatrogenic venous foreign body emboli, as [18] explains, usually lodge in the great systemic veins, the right heart chambers, or the major pulmonary arteries. These foreign objects typically originate from broken distal portions of catheters introduced into the venous system to monitor pressures or administer fluids. Venous emboli can remain with prolonged patient survival, but most authors recommend removal. During an attempt to place a central venous catheter from the right subclavian vein, 15 cm broke off, with the sharp end in the superior vena cava and the distal end in the apex of the right ventricle. To effect retrieval, one catheter was inserted from the antecubital vein, and another from the femoral vein. Several initial attempts to secure the object went unsuccessful under 20 minutes of fluoroscopy. Afterwards, the proximal end of the foreign body was dislodged from the superior vena cava, such that it could be snared by the other catheter. This two-catheter approach can be used when both ends of a broken catheter are lodged in tissue.

Intracardiac catheters can break due to unexpected movements of patient limbs, or physician error. In the study of [19], a catheter broke off in a neurosurgery patient, and the fragment could not be found in the arm where it was initially believed to be. Radiological investigation localized it in the trunk of the pulmonary artery and right atrium. An endoscopic 7-fr polypectomy snare (thin, flexible, and atraumatic to tissues) was used. The catheter was introduced into the left atrium under X-ray guidance, and the foreign body was extracted after 30 minutes.

Other alternative retrieval approaches may use flexible endoscopic forceps with transjugular entry [26,27], Dormia ureteral stone dislodgers [28], or forceps for vascular, urinary, and biliary foreign body removal [29]. In the vast majority of procedures, whether minimally invasive or not, authors have found the use of preoperative and intraoperative echocardiography to be beneficial in localizing foreign bodies [30–37].

2.2 Robots in Cardiac Surgery

Early efforts in robotic cardiac surgery employed the AESOP and ZEUS systems (Computer Motion Inc., Santa Barbara, CA). AESOP, or Automated Endoscopic System for Optimal Positioning, was a robotic endoscope manipulator that accepted foot pedal or voice commands, thus freeing a surgeon's hands from having to hold and adjust the operative field of view during a minimally invasive surgery. AESOP received FDA approval in 1994 and assisted in a surgery the same year. ZEUS was a bimanual telemanipulator that included AESOP as its third arm. In 1999 it was used in the first robotic minimally invasive surgery on a beating heart, and in 2001 it received FDA approval.

The da Vinci surgical system superseded ZEUS in 2003, and has been used extensively in a variety of cardiac procedures, including coronary artery bypass grafting (CABG), mitral valve repair, aortic valve repair, arrhythmia treatment, and atrial septal defect repair. Mohr *et al.* [38] reported on 148 da Vinci-assisted cardiac surgery cases, spanning totally endoscopic coronary artery bypass (TECAB) and mitral valve repair procedures. This early study found that robotic surgery could be performed safely provided conditions amenable to use of the robot are met. Advantages of a robotic system such as reduction in fatigue were tempered by disadvantages such as lack of tactile feedback.

Meanwhile, research on beating heart motion synchronization was already underway. A very high speed camera (1,000 frames per second) was used for tracking heart motion in [39] for the purposes of improved robotic control and visualization. The tracking information, refined using an autoregressive filter for error prediction, was fused with video from an NTSC camera for rendering. Experiments in tracking a laser dot at 1.5 Hz showed perfect synchronization.

The Sensei X robotic catheter system (Hansen Medical Inc., Mountain View, CA), FDA approved in 2007, allows for precise teleoperated positioning and force sensing (via the IntelliSense feature) of a catheter for procedures such as ablation and 3D cardiac mapping. The physician's workstation contains views of electrophysiology data, mapping systems, and the fluoroscopic scene. Similar in concept are the CorPath vascular robotic system (Corindus Vascular Robotics Inc., Waltham, MA), which focuses on manipulation of guidewires and stents in percutaneous coronary interventions, and the Stereotaxis magnetic navigation system, in which remote steering of catheters is accomplished using a magnetic field.

HeartLander [40] is a miniature mobile robot designed to enter the body through a subxiphoid entry and crawl along the surface of the heart to deliver therapy. The robot consists of two suction legs that adhere to the heart to remain stationary and alternate between adhesion and release to effect worm-like travel, as well as a tether to furnish power and communication. In addition to being a minimally invasive device, HeartLander naturally attains heartbeat motion compensation for both manipulation and visualization since it is fixed to the moving reference frame of the heart.

CardioARM [41–46] is a 12-mm diameter, 300-mm long snake-like robot that can assume complex curvatures, allowing it to be maneuvered through confined spaces around the heart. The mechanism consists of two concentric snakes, each composed of several links attached by universal joints, which are in turn driven by actuating wires. Tensioning the wires stiffens the manipulator, while relaxing the wires has the opposite effect. The inner snake is controlled by only one wire, and the outer snake is controlled by three, making it steerable.

Advancement of the end effector proceeds as follows. With the inner snake stiff to maintain shape, the limp outer snake is advanced by one link increment and simultaneously steered. Once the outer snake reaches the desired incremental displacement, it is stiffened to freeze the overall shape. The inner snake is then made limp, advanced one link increment such that its tip is flush with the outer snake's tip, then stiffened again, and so forth. Porcine experiments demonstrated that the robot was able to traverse clinically relevant paths around the heart without any damage to the surrounding tissue.

Several robots have been devised for mitral valve procedures, including a 1-DOF heartbeat compensation device [12, 47, 48], a heartbeat compensation catheter with force sensing capability [49–53], and the NeoChord DS1000 [10]; while the latter is not a robot, its use has been demonstrated as part of a computer integrated surgical suite. Concentric tube robots for guidance of MEMS patent foramen ovale closure devices [11, 54] have been investigated as well. These devices are further discussed below in the context of 3D ultrasound guidance.

2.3 Ultrasound in Surgical Navigation

Ultrasound has found great success in medical applications such as diagnostic imaging and interventional guidance, both as a standalone and as a complementary modality. Its ability to image soft tissues inside the body without emitting harmful ionizing radiation makes ultrasound a valuable resource in biomedicine. However, the technology presents challenges that have inhibited its use in many scenarios. The noise, low resolution, and artifacts inherent in ultrasound images make visualization of fine anatomical details difficult, and relatively low acquisition rates limit its ability to track moving targets and perform robot guidance. As a result, research in the use of ultrasound for surgical tracking and guidance has been, and continues to be, an ongoing topic of interest. The present work focuses on expanding the roles and capabilities of 3D ultrasound in clinical interventions and robotic assistance.

2.3.1 Related Work using 2D Ultrasound

Research in ultrasound integrated systems has focused predominantly on 2D (B-mode) images, so robust registration with patient anatomy, preoperative models, or other types of images has been a challenge, as has out-of-plane guidance of surgical instruments. These issues have been addressed in part by the reconstruction of 3D volumes from mechanical or manual sweeps of 2D images, but the acquisition rates under these approaches have remained prohibitively low for real-time navigation. Nevertheless, the body of work that has been done using 2D ultrasound probes serves as inspiration for further work. Despite the difference in dimensionality, 2D and 3D ultrasound share several similarities in the context of surgical tracking and robot guidance, such as the ability to image soft tissue cross sections (albeit with relatively low resolution), thus there is the potential for sharing lessons learned between them.

A developed 2D ultrasound-guided robotic system was reported in [55–57]; a Star algorithm and a temporal Kalman filter were used to track the center of a carotid artery in B-mode ultrasound image sequences. The Star-Kalman algorithm appeared to work best out of the algorithms tested, which also included normalized cross correlation, sum of squared differences, Star, and snake algorithms. First, a seed point interior to the artery was chosen based on the previous center point, from which radially emanating rays were formed. Candidate artery boundaries were then selected along each ray, and a uniform probabilistic estimate was used to determine the best estimate of each boundary point. The centroid of these boundary points was then chosen as the artery center, and a spatial Kalman filter was used to estimate the carotid artery contours. For both phantom and *in vivo* cases, errors were less than 0.7 mm.

Several control methods based on force, images, and teleoperation were implemented; for the experiments, image guidance was used exclusively. Ultrasound guidance experiments were designed to keep the artery cross section centered in the image in one dimension, while the probe was manually guided in another. The desired 3D motion was reconstructed from the 2D image slices based on the kinematics of the robot and probe. Effective performance was shown in this collaborative human-robot guidance scenario based on 2D ultrasound.

An early example of autonomous robot guidance under 2D ultrasound involved an electromagnetically tracked 2D ultrasound probe on a Phantom Omni robot [58]. Touching of targets defined in ultrasound images allowed for determination of the frame transformation. The targets were the centers of spherical cross sections of adequate intensity. Calibration was obtained by alignment of the probe to known image patterns. Guidance of the robot to touch targets was 88% successful, and errors measured by sweeping the probe over the target and measuring the corresponding robot positions yielded figures of 2–3 mm for slow scans.

A treatment more focused on a robotic platform for 2D ultrasound was presented by Masuda *et al.* [59]. A patient-mounted robotic system was developed to hold an ultrasound probe against a patient's abdomen with a controlled level of force. The robot was teleoperated over a network, motivated by scenarios in which an expert is not physically present with the patient. Control commands from a joystick flowed in one direction while ultrasound images traveled in the other. Position and orientation commands were decoupled via separate joystick handles. Different network speeds were used, with resulting teleoperation control and video feedback latencies of 2 and 5 seconds for 10 Mbps LAN and 128 Kbps ISDN respectively. Such delays, predominantly attributed to the encoding/decoding of motion-JPEG images, were found to be cognitively stressful for operators and a learning period was required for operators to adapt.

Vilchis *et al.* [60] proposed a similar architecture in which an ultrasound probe-holding robot was designed for control by teleoperation. Some absolute accuracy of the robot could be sacrificed because the application called for scanning of pregnant women. Specifically, translations were actuated by a parallel strap mechanism that allowed for flexibility in positioning. The straps held a wrist that provided rotation and insertion axes of motion. Demonstrations of the system used a Phantom Omni robot teleoperated over different types of communication links and video codings.

In a more clinically-oriented effort, the work of [61] explored the use of B-mode ultrasound to manually guide a needle towards a gall bladder. Image processing was performed to identify and track the contour of the target, with the consideration that the target, and hence the contour, moves. The computation had a 51% duty cycle of the 130-ms control period in phantom experiments. Primarily due to this latency, errors of 7.49 mm were found, and the needle path was updated at an approximately 3 Hz rate. The ensuing animal experiment thus did not meet expectations, and furthermore the needle was difficult to localize in the image. It was found that such a system worked best as an effective planning tool to aid the interventionalist.

The state of the art was further advanced in [62], in which a computer assisted biopsy application used a LARS robot to guide a 2D ultrasound probe, and used another robot to control a needle guide. The end effectors in both cases, namely the ultrasound probe in the former case and the needle guide in the latter, were tracked using electromagnetic sensors. Control and data processing for the probe, tracker, and both robots were performed on a surgical workstation computer. 3D Slicer software was used for visualization, planning, and reconstruction of 3D volumes from adjacent 2D slices. To simulate cancer lesions, a pit in a plum and olives in calf liver were used, submerged in a water tank. A mechanical phantom with plastic pins was used to facilitate measurements, and tracking was performed entirely using the tracker as opposed to the robot joint encoders.

First, the LARS-held probe was manually guided to both explore the target volume and to train the robot scanning protocol for the region of interest. The LARS then autonomously repeated the process of moving in 1D to acquire an ultrasound image. A 3D volume representation was reconstructed from the collected slices to enable surgical planning. The planned depth of the cancer lesion was manually marked on a needle, which the needle guide robot automatically placed at the entry point.

The needle was inserted into the phantom manually under ultrasound monitoring.

Measurements were performed using a separate ultrasound probe and console, and errors between the needle tip and targets of about 1.2 mm were found. Ratio-wise, the 3D reconstruction of 2D ultrasound images was about twice as good and much more reliable using the robot than with freehand probe guidance. Total system error was 3 mm, determined by touching the needle to the mechanical phantom and measuring the distance from the images. With this allowed margin of error, robotic assistance led to 7/7 successful hits, while the success rate for freehand guidance was 3/4 due to gaps in volume data and synchronization inaccuracies.

A number of other studies have performed 3D volume reconstruction from 2D slices acquired through manual or robotic sweeping of the probe. Real-time 3D reconstruction has also been accomplished using two probes fixed at right angles [63] to facilitate robot guidance. These concepts may be applied, for example, to virtually enlarge the field of view of a 3D probe.

It was found in [64] that camera projection methods from classical visual servoing were not well suited for ultrasound-based servoing. As part of the proposed method, 2D ultrasound image slices were first acquired that contained a target egg shape, representing a cancer lesion. Using 20 equispaced radial lines interior to the target in each slice, the best edge points were selected where the lines intersected the edge using a Gaussian. The edge points were fit to a third-order polynomial to create a contour representing the original egg shape. A least squares fit to an egg model was used to determine the 6-degree of freedom (DOF) transformation and associated velocity screw. Simulation experiments showed convergence to a final configuration in about 100 iterations; robustness to noise and model errors were demonstrated as well.

Later advances saw the use of 2D ultrasound to deduce the configurations of instruments or other surgical targets for the purpose of robot guidance. In [65–68], B-mode ultrasound views were acquired containing the image of a tool jaw (two blobs) intersecting with the image plane. The tool was required to be kept perpendicular to the image plane such that both sides of the jaw were always visible in the image. Control laws were developed based on a geometric model of the tool and real-time appearance of the tool in the image to estimate the tool configuration. In the envisioned operative scenario, the user would select a target in an interactive image view, and the robot would proceed to guide the tool to meet the target. Each servo task was completed in approximately one minute; preliminary *in vivo* testing on a beating heart was reported as well.

A jawed instrument in a fixed B-mode ultrasound plane was guided to a specified configuration in [69] using nonlinear model predictive control (NMPC). The system was solved as an optimization problem over a finite prediction horizon, with a quadratic cost function incorporating system states and inputs, subject to nonlinear system dynamics and constraints; simulations showed robustness to noise and model errors. The tool was assumed to be roughly perpendicular to the image plane, and its configuration was determined by segmenting the blobs where the jaws intersected the plane.

The work of [70–73] presented model-free ultrasound guidance based on the moments of an egg-shaped target in B-mode images to servo an ultrasound probe to a desired cross section of the target. The computation of the image Jacobian (or interaction matrix) included parameters such as area, surface gradient, moments, and changes in these parameters between consecutive frames. Efforts were taken to combat noise in taking derivatives during online estimation of some out of plane vectors. In robot experiments, guidance to desired targets were completed in roughly half a minute for an ultrasound phantom and in one minute for a lamb kidney.

2.3.2 3D Ultrasound-Based Tracking and Guidance

The first commercial 2D (B-mode) ultrasound scanner was released in 1963 (Physionic Engineering Inc., Longmont, CO). Reconstructed 3D ultrasound was first demonstrated by Sonicaid Ltd. (Glasgow, Scotland) in 1973, then by the University of Tokyo in 1984, and made commercially available by Kretztechnik AG (Zipf, Austria) in 1989. However, these systems reconstructed 3D volumes from multiple 2D images, leading to mechanical and computational delays of several seconds to several minutes between scans. Such limitations meant that these systems served best as visualization and diagnostic tools. Real-time 3D ultrasound (also known as 4D ultrasound) based on matrix array transducers did not emerge in research until

1991 (Duke University) and as a product until 1998 (Volumetrics Medical Imaging Inc., Durham, NC). The introduction of streaming made interventional use of 3D ultrasound viable, and early studies (e.g., [74] in 2003) reported on benefits over the 2D counterpart. Now commonplace for both cardiac diagnosis and intervention, 3D ultrasound technology continues to improve alongside advances in research exploring its capabilities.

2.3.2.1 Surgical Tracking using 3D Ultrasound

Tool tracking and target tracking are fundamental components of the system proposed in this work, and the enhancement of tracked objects for visibility in ultrasound can help improve its efficacy. The appearance of an instrument can be enhanced in 3D ultrasound via induced vibrations. A piezoelectric buzzer was attached to an instrument at a carefully determined position in [75]. Optimizations of the acoustic frequency and display filters were also made, thereby improving the detection capability. The pilot nature of the study resulted in relatively low acquisition rates of about 4 Hz, but the results were promising for transesophageal, transthoracic, and intracardiac echocardiography probes (TEE, TTE, and ICE respectively).

In [76], a variable magnetic field was used to vibrate ferrous shrapnel, and color flow ultrasound Doppler was used to locate the target. A vibration frequency in the range 90–120 cm/s was invisible to the naked eye. The field strength was sufficient to reach a distance of a few inches, and was collinear with the transducer axis. In a water bath, shrapnel was placed near a hyperechoic rubber surface to demonstrate the advantages of the technique; in conventional ultrasound, visibility would have been poor. An average RMS localization error of 1.06 mm was achieved.

Substantial improvements in tracking and visualization can be achieved by incorporating sensors onto instruments. Doing so, however, is often disruptive to the workflow; for example, electromagnetic tracking places a field generator in the workspace, while optical tracking imposes line-of-sight restrictions. A non-disruptive sensing technique was proposed in [77], wherein a miniature acoustic sensor was installed on the tip of a catheter to receive signals from a 3D ultrasound probe already part of the procedure. Since the timing and sequence of the beams were known, the catheter tip could be localized with an accuracy of 0.36 mm.

While these methods enhanced the appearances of objects in 3D ultrasound in order to improve tracking performance, tracking of surgical instruments was performed in earnest in [78]. Straight shaft tools were tracked by first examining connected neighbor voxels to find candidate tool volumes. After a principle component analysis on the volumes, the candidate with the highest ratio between the principle axes was considered to contain the tool, with position and orientation computed as part of the analysis. The location of the tip was then found as the endpoint of the voxel region along the primary axis. Evaluation was performed by overlaying an ideal tool shape onto the image; over several samples, the error was 0.7 mm. These efforts evolved into instrument segmentation techniques that considered the composition of the environment from statistical and spatial standpoints [79–81].

Ref. [82] performed needle detection by fitting thresholded 3D ultrasound voxels to a polynomial curve. Metrics for tool axis localization included distance of the detected tool from the curve or the probability of a voxel being a tool given its intensity and distance from the curve, then using RANSAC to determine the best polynomial fit. Finally, the tip was localized by searching for a drop in intensity along the localized tool axis. Experiments were performed on a simulation phantom, PVA phantom, and turkey breast. Errors in position and orientation of about 1 mm or less was achieved with a run time of about one second. The results were more robust than those found in the literature based on projection- and Hough transform-based methods.

Noting that a 6D parameter space is generally computationally expensive, the work of [54] demonstrated real-time, automatic detection of curved instruments and robots in 3D ultrasound. A standard preprocessing pipeline was used to skeletonize the tool, detect the plane of tool residence using RANSAC, and approximate the mean square distance of the circle traced by the arc-shaped tool. Simulation results showed a 45-degree arc and less than 3.3% of radius noise would be necessary for effective detection. Water tank experiments with a 3D printed arc and a linearly translating probe showed errors of 2.3 mm in the radius, 2.5 degrees in the plane angle, and 2.1 mm in the circle center. Qualitative *ex vivo* results with a porcine heart were encouraging.

Another related capability is the tracking of anatomical features in 3D ultrasound.

Segmentation of the mitral valve was investigated in [83–86] and showed good spatial performance at 0.77 mm of error, with a temporal performance suited for offline diagnosis and modeling applications. Atrial septal defects were tracked in 3D ultrasound at 25 Hz in [87]. A number of methods were evaluated including block matching, optical flow, and block flow schemes using sum of squared differences, inverse Pearson correlation coefficient, and maximum likelihood estimation as similarity measures. The performance achieved was an error of 2 voxels at a rate of 2 Hz, and it was determined that a block flow technique, combining velocity computation from optical flow with template matching, was the superior approach. Subsequent developments [88–90] examined improvements in tracking robustness and computational efficiency.

In [91], a real-time algorithm was proposed for tracking fast-moving anatomical targets in 3D ultrasound. The method demonstrated a 3D extension of a common 2D method [92] used in computer vision for searching temporal correspondences. A sum of squared differences formulation was set up to solve a least squares optimization problem to find the best match between a live image and an affine-transformed volume of interest. The processing time averaged 35 ms per frame on a conventional workstation computer, and the algorithm was able to track a region of the heart wall that moved as much as 3.98 mm between frames.

A Philips Sonos 7500 with an X4 3D echocardiography probe was used to quantify the shape of the left ventricle in real time in [93]. The approach did not rely on a model of the anatomy, while improving upon methods that produce estimates based on 2D cross sections. After initialization, the endocardial surface of the heart was tracked automatically, and the results were used to calculate a shape index. The shape was represented as a 1D signal by helical sampling of the surface (a morphed binary volume) with cylindrical coordinates, with moments of inertia corresponding to the principal axes. The function used was the distance from the surface to the long axis over angle, while reference signals were derived from a sphere, an ellipsoid, and a cone with identical moments of inertia. These quantified shapes were tested for correlations with various phases of ventricular function throughout the cardiac cycle, marking the first time such correlations had been demonstrated in real time. Ejection fraction, a measurement readily available in many clinical echocardiography systems, was also used to evaluate shape correlation with ventricle size and function. The shape index correlated well with ejection fraction, though ejection fraction may be better correlated to cardiac function. Since the shape index described the similarity between the shape of the ventricle at a particular phase of the cardiac cycle to a reference shape, it could be used to detect pathological conditions. Spheres were associated to overloading and cones to good health, while ellipsoids were independent because both spheres and cones could be described as ellipsoid variants. It was found that a diseased heart attempts to restore normal function by changing shape.

2.3.2.2 Robot Guidance using 3D Ultrasound

Leading up to robotic systems with 3D ultrasound-based guidance, imaging devices of various form factors were developed and tested. In [94], laparoscopic 3D TEE was used to visually locate targets and manually guide a robot to those positions. While the guidance aspect was rudimentary in the context of contemporary work, experiments were built atop the Smith *et al.* [95, 96] efforts of the early 1990s in matrix array transducers. Subsequent work [97, 98] used two variants of 3D ultrasound catheter transducers to locate needle tips in a water tank and command a robot to move one of these tips towards the other, stationary one. The trials succeeded with approximately 3 mm of error (it was noted that catheter transducers are of very low resolution). Several ring array transducers have also been developed for catheters [99]. Stated applications include stent grafts for coronary artery disease and deep vein thrombus.

Robot guidance using several variants of 3D ultrasound devices were demonstrated in [100]. An experimental transducer array was used to guide a 3-DOF Cartesian robot, with errors of 1.58 mm. Also shown was use of catheter transducers, which have relatively low spatial resolution due to their compact size. Guidance of the same robot with a forward viewing catheter was achieved with 3.41 mm of error, while an improved error figure of 2.36 mm was found when using a side scanning version. In all cases, transfer and processing of volume data were noted as difficult issues for real-time ultrasound-based servoing.

In [101], a 6+1-DOF manipulator held a 3D ultrasound probe-needle parallel assembly, with the needle pointing downward and registered manually. Boneless turkey breast on a rubber acoustically-absorbing pad in a water tank was used to simulate a prostate for biopsy procedures. The first voxel in each image line of the pyramidal scans was used to find the prostate surface. The surface center was approximated from the resulting voxels. Using this point, the phantom was divided into eight sectors. Experiments involved touching each sector with the needle and measuring the success rate of needle touches. Real-time 3D ultrasound was later used to guide the robot to touch a needle to a 10×3-mm metal rod and a simulated cyst, both in turkey breast [102]. The experiments allowed the use of a simple robotic setup with prefixed registration and automated segmentation via thresholding. Neighborless voxels were eliminated to reduce speckle. RMS errors of 1.15 mm were achieved over an overall procedure time of 1–3 minutes.

Robotic guidance of an ultrasound probe was performed in [103–105] to image a crosswire phantom until convergence of the lines in the image. The interaction matrix was adaptively updated to estimate the image scaling factor and probe/robot frame transformations over several image acquisitions, using least squares optimization. Subsequently in [106], speckle decorrelation was used to separately compute in-plane and out-of-plane motion using B-mode ultrasound images. Images were acquired at 12 Hz and good static error was found in planar guidance experiments. In 6-DOF experiments, dynamic tracking errors were 1.4 cm and 3 degrees. Using a robotic probe, one test used a robotically positioned phantom, and in another test the phantom was manually displaced and tracked with vision.

An example of 3D transthoracic ultrasound for visual guidance involved an instrument equipped with markers visible in ultrasound images, as described in [107]. The efforts led to the creation of an instrument marker whose orientation could be computed, with the assumption that the instrument always pointed away from the imager. Experiments performed in a water tank with a Phantom Omni robot driving the instrument to a series of image-based landmarks were used to measure the tracking error. Orientation measurements demonstrated the invariance of the method to probe orientation. Errors were under 1 mm with a robot speed of 3 mm/s and a 2 Hz update rate.

Three-dimensional ultrasound at a 25 Hz frame rate was used to guide a robot with a tracked instrument to touch a tracked target in a water tank in [108–110]. First, the projection through the ultrasound volume with the maximum intensity was found to localize the instrument shaft. A modified Radon transform was used for instrument shaft detection, with a graphics processing unit (GPU) implementation of the transform enabling real-time tracking. Markers were installed onto the instrument tip to allow its orientation to be determined. The fiducials consisted of an echogenic nylon cross further covered in heat shrink wrap to improve its appearance in ultrasound; template matching was used to detect the raised features. The remaining two degrees of freedom, namely the tip position and roll orientation, were found via image processing on the passive image markers.

The robot could be guided at a maximum speed of 2 mm/s, governed in part by the small search space (5 mm^3) of the tracking algorithm. In the experiments, the robot was homed to a distance of 1 cm above the targets, then commanded to detect and move to four such targets placed on the floor of a water tank. Initial target and instrument detection completed in two seconds. Total robot travel was approximately 10 seconds in duration for 1.2 cm of distance, with an average error of 1.2 mm.

In a water tank experiment, a five-axis stage served as a ground truth. At each yaw orientation in five-degree increments over 60 degrees, five positions were measured within a 2-cm square. The measurement error was 1.07 degrees; the tip position was found to within 1.8 mm, excluding outliers of substantial error. In a beating porcine heart experiment, electromagnetic tracking was used as a ground truth. Each trial was run for five seconds, and the per-trial measurement errors were 1.4, 0.5, and 0.9 mm.

In [12, 47, 48], which addresses mitral valve repair procedures, 3D ultrasound was recognized as an effective modality for intraoperative cardiac imaging because it is real time and does not emit harmful ionizing radiation. Acknowledged disadvantages of 3D ultrasound included the size of the volume data and the resulting processing delays, which could be potentially prohibitive for robotic guidance at heartbeat rates.

The work proposed a 1-DOF device that follows heartbeats in the principle axis of motion, relying on the surgeon to compensate for off-axis motions. Motions pertinent to the mitral valve repair task were characterized from manually- segmented 3D ultrasound streams, finding the requisite speed, acceleration, range, and frequency specifications to fabricate a handheld interventional device.

To combat 3D ultrasound acquisition delays that limited the update rate to 8 Hz, a predictive model of the mitral valve motion was used. Methods studied included an autoregressive filter, an autoregressive filter with fading memory, and an extended Kalman filter, with the latter selected as the operative method. The prediction algorithm accounted for variabilities in the motion such as noise.

In user trials, subjects were asked to draw a circle on a piece of paper programmed to pulsate with simulated noise and delay, and the accuracy and axial force were measured. Separate water tank experiments using ultrasound with a pulsing marker were performed as well. As expected, precise timing was found to be essential; an extended Kalman filter worked well in this regard, with errors on the order of one millimeter.

As an extension of this work, a catheter was servoed in one degree of freedom in [49–53] to track the mitral valve in a beating heart. The work focused on combating kinematic issues such as friction and backlash in order to achieve the performance of the earlier, more specialized device [12], then developed force- and position-based control of cardiac catheters.

Recent developments feature applications of 3D ultrasound in cardiac surgery. A concentric tube robot was designed for minimally invasive beating heart surgery to guide a microelectromechanical (MEMS) device for patent foramen ovale closure [11]. An advantage of this type of robot is that it can apply greater forces at the tip than catheters do while having a similar form factor. The MEMS device was specially designed to be removable in case of failure during surgery, and stiff enough to be driven into tissue without breaking. Teleoperated delivery is performed under hybrid imaging: 3D ultrasound for coarse positioning and fluoroscopy for deployment of the device at the end effector. In pull-out force testing, the device was found to be safe to 10 N, a wide margin over the requirement of 1 N, in both *ex vivo* and *in vivo* experiments. Three *in vivo* trials were successful.

Addressed in [10] was degenerative mitral valve disease (DMVD), in which incomplete valve closure occurs as a result of diseased leaflets. The disease affects 2% of the general population, and conventional treatment includes sternotomy and cardiopulmonary bypass. A minimally invasive surgical solution was proposed based on the recently announced NeoChord DS1000 and 3D transesophageal echocardiography for manual guidance of the surgical instrument. In the stated procedure, the instrument is guided to the vicinity of the mitral valve, where it secures a diseased valve leaflet and attaches a suture to force the valve to close properly. Proper positioning of the device is examined using the ultrasound console with image overlays, while grasp of the leaflet is verified using an optical sensor. With the TEE probe tracked by an electromagnetic tracker, the surgeon identified the mitral valve in the image, allowing the system to create an image overlay of the structure. The instrument position was also tracked and incorporated into an augmented reality view, assisting the surgeon in guiding the instrument. Two porcine studies were conducted involving five cardiac surgeons each (six in total), categorized as two novices, three experts without NeoChord experience, and one expert with NeoChord experience. The subjects were asked to guide the tool from the apex of the heart to the target in the left atrium, with and without augmented reality assistance. While it was found that augmented reality improves navigation in general, 10/10 trials succeeded without it while only 9/12 did so with it. This was explained by the reduced risk of injury when augmented reality is available due to less deviations in the tool trajectory, as well as the increased repeatability of the trajectories.

2.3.2.3 Other Related Work in 3D Ultrasound

The works described above are those most pertinent to the current topic, but several other efforts are relevant for 3D ultrasound-based robot guidance as well. For example, 3D stereo vision for teleoperation of the da Vinci surgical robot was evaluated in [111, 112]. In comparing user performance under conventional endoscopic stereo, 3D ultrasound stereo, and 2D ultrasound stereo, the expected results of decreasing efficacy in the stated order were found.

Electromagnetic sensors have been used to track intraoperative fluoroscopy and a TEE probe to register 3D ultrasound volumes to 2D X-ray images [113]. Errors were on the order of 2 mm; the total error was found to be less than the sum of per-subsystem errors, possibly due to cancellation effects.

Two methods of generating strain image volumes were evaluated in [114]: (1) manual collection of adjacent B-mode slices, and (2) mechanical collection of adjacent 3D volumes. Simulation settings were fixed such that the main influence on spatial resolution was the post-volume filtering, and the resolution was fixed between datasets to compare the signal-to-noise ratio (SNR), which was used as a basis for comparison. Results indicated that the SNR for the 3D probe was superior due to more reliable image acquisition, and therefore provided better reconstruction. On the other hand, it was acknowledged that smaller 2D probes may be more suited in certain clinical scenarios.

Decisions behind the design of a robot used to hold an ultrasound probe in a teleoperation scheme were detailed in [115]. A serial spherical joint was selected based on such parameters as workspace, accuracy, dexterity, manipulability, structural and computational complexity, mass, size, singularities, and geometric constraints. Decisions were optimized over these parameters and the resulting kinematic model was thoroughly developed.

2.3.3 Summary

The work discussed herein bears some key distinctions from the aforementioned studies. The first is the 3D ultrasound-based guidance of a robot with distal dexterity, leading to enhanced workspace coverage inside the heart. This is in contrast to rigid instruments that rely predominantly on pivots about the surgical entry point (remote center of motion, or RCM); tools whose distal degree of freedom travels along its axis of insertion [12]; dexterous devices with motion constraints such as catheters [50] and concentric tubes [11]; and devices that are guided manually [10,11]. Internal dexterity further implies that manipulation respecting the RCM point requires appropriate redundancy resolution, as opposed to the technique of mirroring motions about said point as was sufficient in [110].

Another distinguishing factor relates to the proposed task, i.e., demonstration of 3D ultrasound in guiding a robot to capture a fast, erratically moving target in a dynamic environment, as opposed to a static or slowly moving target (e.g., a tumor) or one whose motion is highly patterned (e.g., the mitral valve). Finally, this project requires that robot guidance be performed with real-time feedback, at the 20 Hz frame rate of the ultrasound stream, rather than at the 2–8 Hz control rates that was adequate for predictive mitral valve tracking [12].

Chapter 3

Robotic Surgical System with 3D Ultrasound Guidance

3.1 System Requirements

3.1.1 Functional Requirements

To improve the standard of treatment, we propose a minimally invasive robotic system that captures a moving cardiac foreign body by maneuvering a dexterous robot under 3D TEE guidance. A robotic approach, as opposed to a manual procedure using rigid instruments, is motivated by the various challenges inherent in minimally invasive surgery, which arise from attempts to perform skilled surgical tasks through small incisions without direct vision. Challenges include reduced dexterity, constrained workspace, limited visualization, and difficult hand-eye coordination, which ultimately lead to poor manipulability. A dexterous robotic end effector with real-time image guidance represents a unification of otherwise disparate technologies that can help overcome these challenges and potentially improve surgical performance.

To achieve these goals, the following components and capabilities are required:

- Robot—An end effector that can be introduced transapically into the heart, with sufficient dexterity to reach arbitrary positions within the chamber. Dexterity is particularly important in this situation because the access port and surrounding tissue do not offer the pivot range often used in laparoscopy.

- Imaging—Intraoperative 3D ultrasound for tracking the foreign body and guiding the robot to capture it. Ultrasound is preferred over other imaging modalities because it does not present the ionizing radiation of fluoroscopy or the restrictive environment of MRI. It features superior acquisition rate and soft tissue contrast, and is typically cost effective to use.

- Tracking—Ability to track an erratically moving object in the dynamic environment of a beating heart, in real-time, using an imaging modality with known shortcomings.

- Guidance—Ability to determine a capture strategy for a slow robot and translate capture decisions into robot guidance.

An experimental system for minimally invasive evacuation of foreign bodies from a beating heart is depicted in Figure 3.1. The key components include a beating heart phantom to recreate the clinical scenario *in vitro*, an ultrasound system to image the scene, a high dexterity robot to capture the foreign body, and a PC to perform tracking and guidance based on the ultrasound volume stream. Though the system builds upon a number pre-existing resources, this chapter provides a detailed description for informational purposes and identifies the novel contributions explicitly. The main research contribution found here is in the targeting of a new clinical application. Meanwhile, the technical contributions reside in the extension of existing resources for said application as well as the products engineered to unify the components into a functional entity; some of these products have since been incorporated into open source projects. A photograph of the setup is shown in Figure 3.2.

3.1.2 Performance Requirements

3.1.2.1 Dexterity

In minimally invasive surgery, the remote center-of-motion (RCM) is a common theme in which an instrument is maneuvered inside the body by pivoting about a small incision, the RCM, in the exterior tissue. Often modeled as a point, the RCM limits the workspace of a straight tool because motions are confined to pivots about the point. Additionally, the anatomy of the entry port, or a trocar used to guide instruments through the port, may further reduce the workspace by restricting the available pivot range.

Cardiac surgery using a transapical approach differs from traditional minimally invasive procedures such as laparoscopy in that the use of the surgical entry point

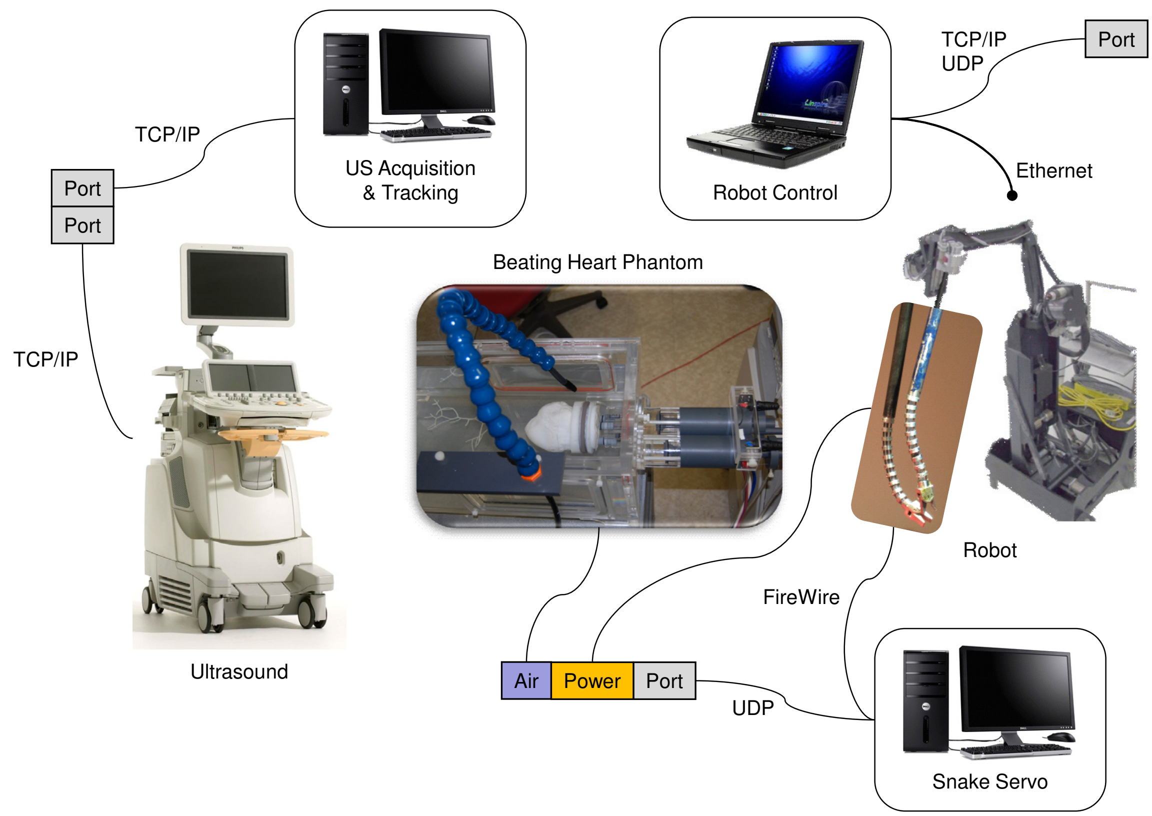


Figure 3.1: Conceptual diagram of a 3D ultrasound guided robotic system for capturing foreign bodies from a beating heart phantom. The high dexterity robot uses two controllers as it is a hybrid of two separate robots. Robot Control communicates with Snake Servo over UDP.

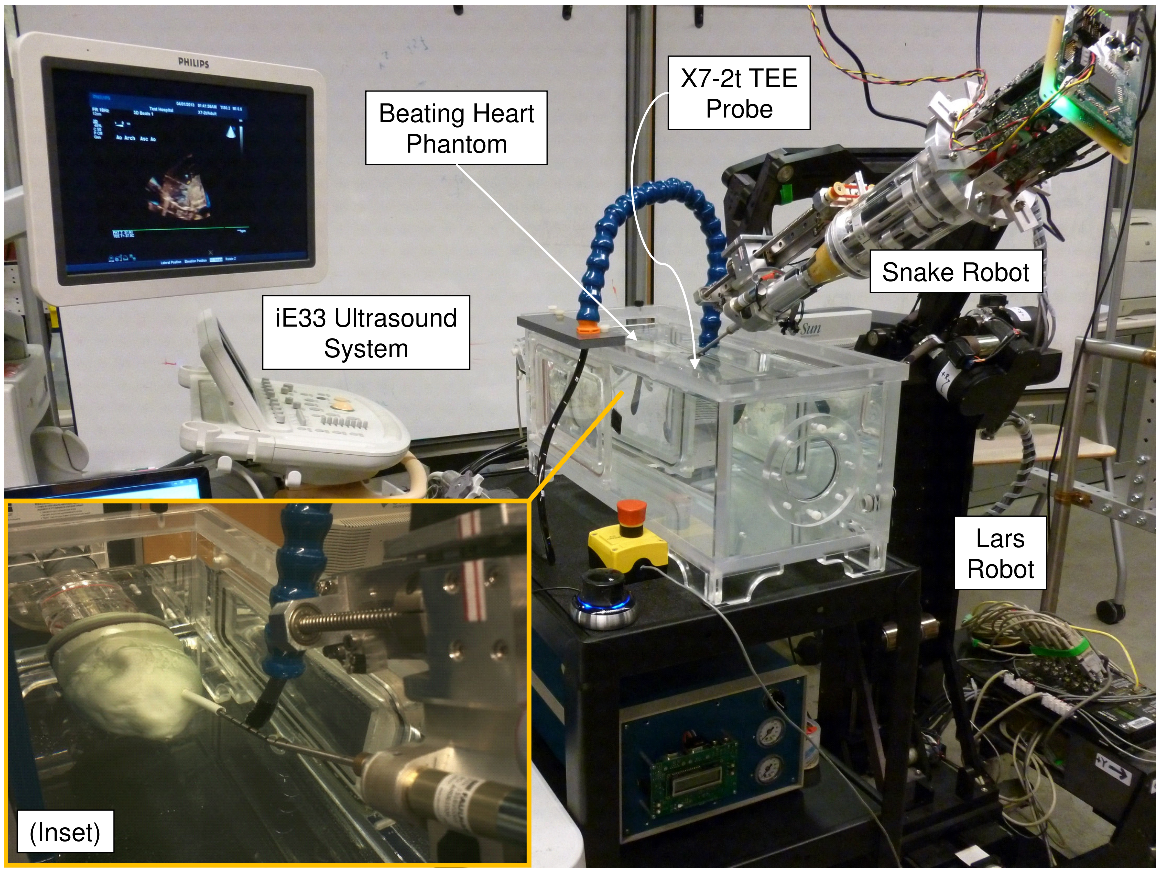


Figure 3.2: Photograph of the setup diagrammed in Figure 3.1. The TEE probe head is held underwater through the arching blue tube (not labeled); the snake robot, with actuation unit and controller seen suspended above the water tank, is attached to the LARS robot. (*Inset*) Snake robot stem inserted into heart phantom (white blob).

as a fulcrum cannot be taken for granted. The apex of the heart in particular has a thickness of 12–15 mm that denies free pivot, with only minimal allowances given by the flexibility of the tissue. Furthermore, it is desirable to minimize pivots about the apex in order to reduce trauma to the tissue and blood loss through the incision, so such motions cannot be relied upon to provide substantial reach or manipulation.

Dexterity is thus needed so that a device can move within the heart without the entirety of the tool having to move. As depicted in Figure 3.3, a dexterous tool can reach various places in the heart with minimal pivot, while a straight tool would potentially require a substantial amount of pivot in order to reach the same places, displacing the tissue around the apex in the process.

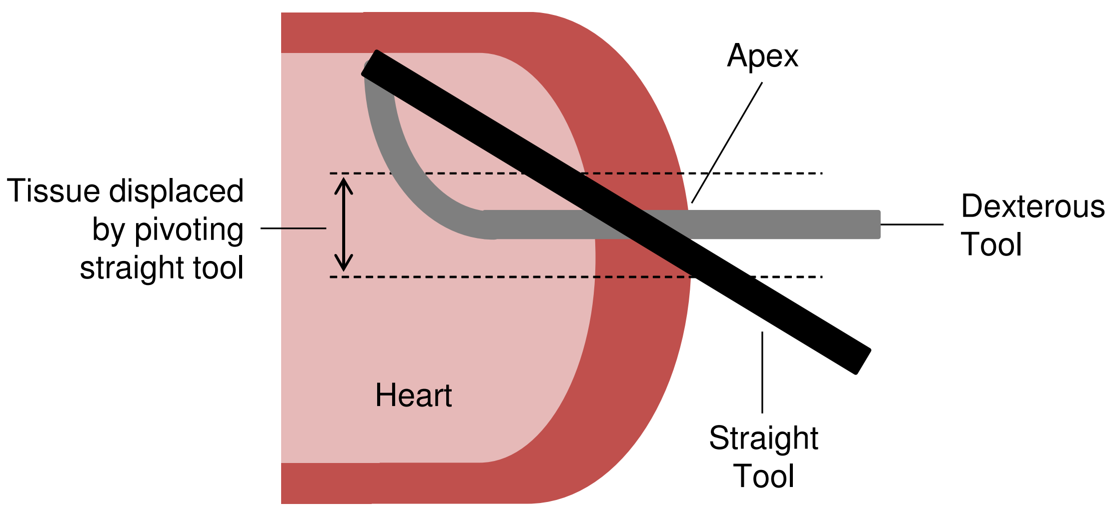


Figure 3.3: Conceptual diagram of a minimally invasive surgical scenario, showing both a straight tool and a dexterous tool entering the heart from the same incision and reaching the same interior point. The straight tool must pivot about the insertion point in order the achieve the same positioning as the dexterous tool.

Because it travels erratically, a cardiac foreign body may be found at arbitrary locations inside the heart, so a minimally invasive device designed to capture the particle must have adequate workspace coverage without the benefit of free pivot. Dexterity thus becomes paramount under these circumstances because it would allow tools to explore regions that are unreachable by straight tools. The added advantage of dexterity is the reduction in the motion envelope of the mechanism outside the body, due to the transfer of mobility to the inside—this can be a favorable characteristic in the operating room.

3.1.2.2 Speed

In order to chase a foreign body in a beating heart, the robot would have to be capable of moving at very high speeds. However, this immediately raises safety concerns because any inaccuracies in the control could result in the robot striking the heart, the heart contracting into the robot, or both. Given the need for dexterity as mentioned above, the risk of collision is amplified because high speed end effectors are difficult to accurately control, particularly with high dexterity.

Injury to the heart tissue can occur if it impacts a device that is stationary but ill-controlled (precluding an evasive maneuver) at high speed (300 mm/s [116]). With an approximate mass of 50 g (the mass of the left ventricle is around 121–453 g [117]) and an acceleration of 3.8 m/s^2 [12], the heart tissue would impact the robot with roughly 0.19 N of force. While this amount of force may not cause rupture directly, tears are still likely to occur if there are sharp edges on the device, and the blunt force may damage blood vessels. In order to ensure that a dexterous robot does not hit the inside of the heart, the robot speed must be maintained at a conservative level, necessitating a foreign body capture strategy other than direct pursuit (Chapter 5).

3.2 Beating Heart Phantom

The Philips beating heart phantom (Figure 3.4) is a custom-made multi-modality phantom compatible with X-ray, ultrasound, and MR imaging. It is a replica of a human heart made of polyvinyl alcohol (PVA) cryogel, and resides submerged in an acrylic glass water tank. Two servo-actuated pneumatic pistons pump water into and out of the heart chambers, and the deformability of PVA leads to the effect of a heartbeat and blood flow. Each piston, 31.75 mm in diameter, is programmed to pump about 18 ml of water per heartbeat following the motion plotted in Figure 3.5. The chosen stroke volume is lower than in healthy humans, reflecting surgical [118] and post-injury cardiovascular conditions. Due to these attributes, the beating heart phantom is considered to provide an adequate representation of an actual heart for the purposes of the current study. Additional details on the phantom can be found in [119].

3.3 Ultrasound System

The ultrasound system consists of a Philips iE33 xMATRIX Echocardiography System and an X7-2t 3D transesophageal echocardiography (TEE) probe, the latter

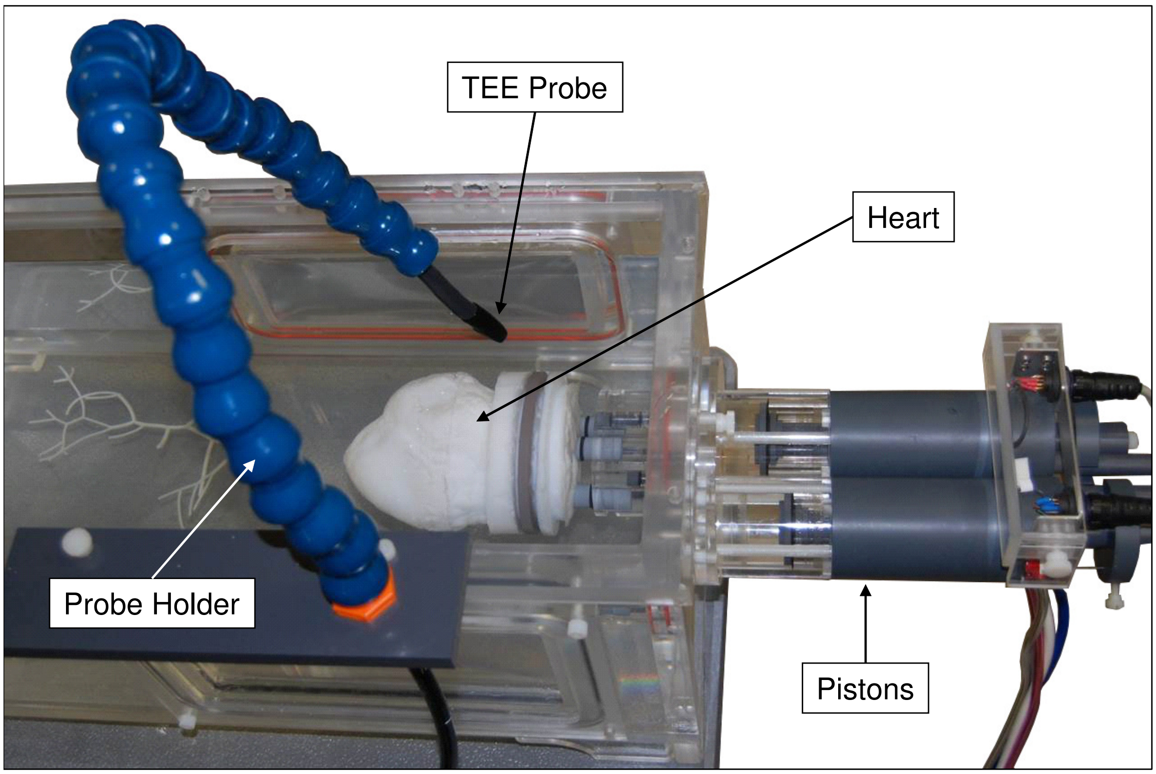


Figure 3.4: Enlarged view of the beating heart phantom and TEE probe inside the water tank. Also shown are the pneumatic servo-driven pistons that pump water to simulate heartbeats.

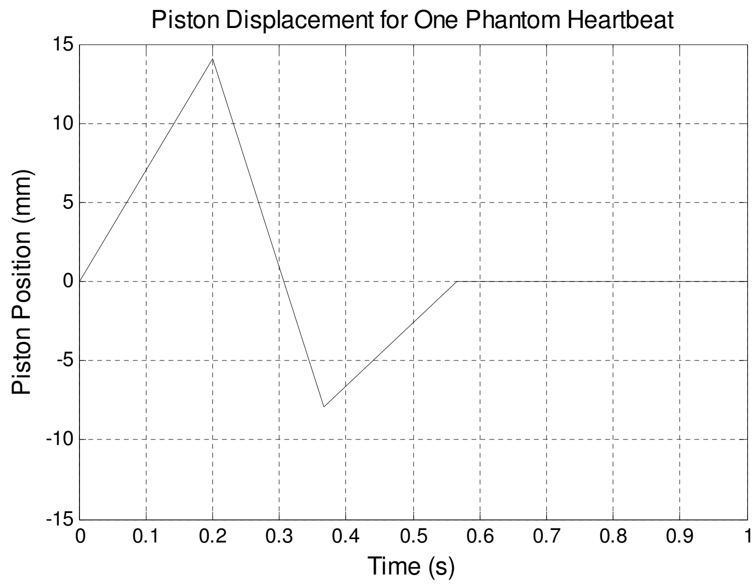


Figure 3.5: Piston motion for each heartbeat of the phantom.

a diagnostic and interventional imaging device designed for visualization of the heart. Image volumes are streamed at approximately 20 frames per second over a TCP/IP network connection to a PC (2 GHz Xeon dual-core CPU, 4 GB of RAM) for image processing, tracking, and guidance computations. The volumes are of resolution $176 \times 96 \times 176$ voxels, spanning a field of view of 60° azimuth, 30° elevation, and 12 cm depth respectively. Gain is set at 47%, compression at 40 dB.

3.4 High Dexterity Robot

The high dexterity robot (Figure 3.6) is an 11-DOF serial combination of the IBM/JHU LARS robot and the JHU snake robot. The LARS is a seven-DOF remote center-of-motion (RCM) robot that is well suited for coarse positioning of the end effector, while the snake is a four-DOF miniature (length 39.5 mm, diameter

4.2 mm) manipulator that enables enhanced workspace coverage inside of the heart. The kinematic redundancy and internal dexterity offered by this union provides the flexibility to implement different modes of operation tailored to specific parts of the procedure. Control requirements, such as task objectives, redundancy resolution, and motion constraints, are formulated as a constrained least squares optimization problem [120] that solves for a set of incremental joint displacements at each control step. This framework provides a convenient interface for implementing different modes of operation by adjusting the weights of the optimization problem. As a result, different applications can be supported; while the combined robot was originally developed for the present work, it was first introduced by Sen *et al.* for robotic palpation with an ultrasound transducer [121].

The two robots are run logically and physically as a single robot through a hybrid motion controller. Control software for the unified robot is run on one PC. A Galil DMC-4080 motion controller serves as an Ethernet-based interface to the LARS joints, while the snake controller [122–126] (Appendix A) is a custom-developed device connected via IEEE 1394 (FireWire) to a separate Xenomai-based Linux PC (2.8 GHz Pentium 4 CPU, 1 GB of RAM). The Linux PC in turn presents a TCP/IP-based joint-level interface between the snake robot and the main PC. The *cisst* libraries [127], a collection of software libraries for computer assisted intervention systems, are used extensively in the robot control programs.

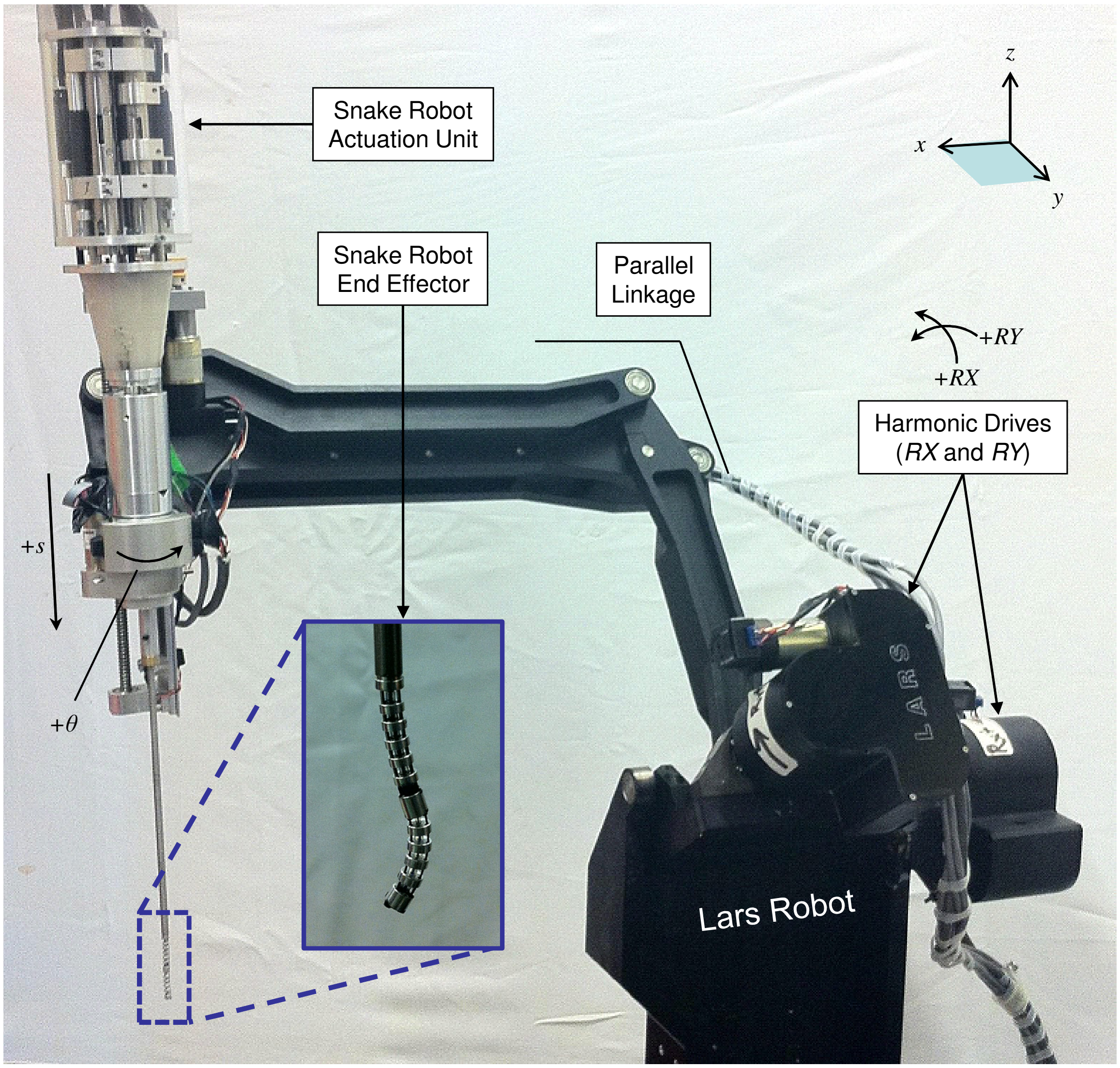


Figure 3.6: The 11-DOF high dexterity robot, formed by a seven-DOF LARS holding a four-DOF snake. LARS DOFs RX, RY, s, and θ are indicated near their corresponding joints, while X, Y, and Z follow the coordinate axes shown. The snake robot consists of two two-DOF snake-like segments, giving rise to an S-bend capability (*inset*).

3.4.1 LARS Robot

The IBM/JHU LARS robot [128] is a seven-DOF tool holder with three translational DOFs (X, Y, and Z), three rotational DOFs (RX, RY, and a rotation θ about the tool axis), and an insertion DOF s along the tool axis. Rotational DOFs RX and RY are constructed in a parallelogram-like structure that imposes physical restrictions on the motion of the mounted end effector. This structure is referred to as the parallel linkage RCM [120]. In minimally invasive surgery, the robot can be positioned such that the RCM coincides with the surgical entry point, avoiding potentially traumatic injuries to the surrounding tissue due to motions lateral to this point. Alternatively, a virtual RCM can be incorporated into the kinematic model of the robot if the mechanical RCM is not coincident with the surgical entry point.

3.4.2 Snake Robot

The JHU snake robot [1] features a miniature, dexterous four-DOF manipulator via a series connection of two (2) two-DOF snake-like segments. Totaling 39.5 mm in length with a 4.2-mm diameter, the manipulator appears at the distal end of a 284.0-mm long stem enclosing the actuation wires. The actuation unit resides at the proximal end of the stem. Figure 3.7 shows a photo highlighting this arrangement.

As shown in Figure 3.8, each snake-like segment is constructed from four superelastic nitinol wire backbones running through a series of discs. The central *primary*

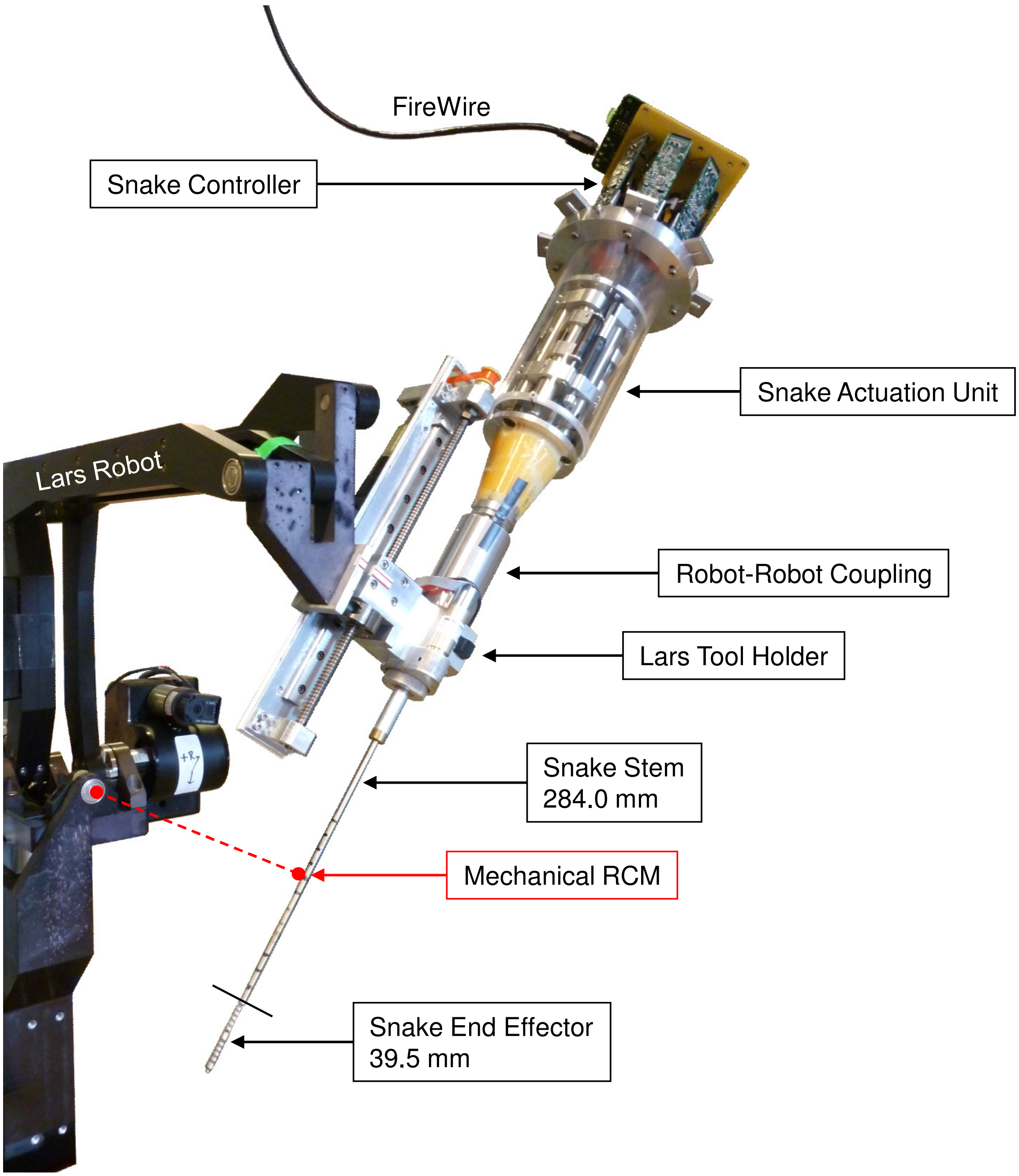


Figure 3.7: The snake robot pictured here has been adapted for the present study, allowing it to be maneuvered by another robot. Previous renditions, targeting throat procedures, were less mobile and thus did not require a compact controller. At home, mechanical RCM is 206 mm from tool holder, and stem is $-31°$ in RX.

backbone runs parallel to three surrounding *secondary* backbones at equally-spaced radial distances and angles. All four backbones are fixed to the distal end disc, while only the primary backbone is fixed to the proximal base disc. The secondary backbones are free to glide through holes in the base disc and intermediate spacer discs. Manipulation is realized by the push-pull motions of these secondary backbones.

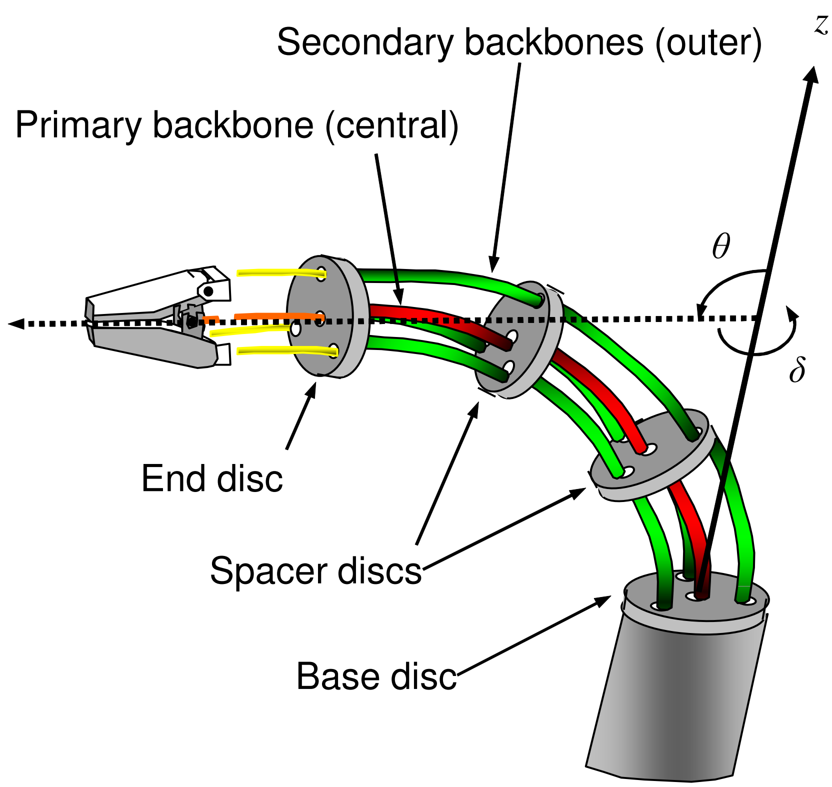


Figure 3.8: Anatomy of a two-DOF snake-like segment [1]. When bending plane $\delta = 0°$, a bending angle $| \pm \theta| > 0°$ bends the snake along the $\pm x$-axis ($y = 0$), defining the local coordinate system. Two such segments in series form the four-DOF device. A gripper is shown in reference to a parent design and possible future work.

In the home configuration, the manipulator is straight and collinear with the stem, parallel to the z-axis of the snake robot coordinate system which is rooted at the base disc. The two degrees of freedom of each snake-like segment i include the angle of bend *from* the z-axis (θ_i), and the plane of bend, i.e., the angle *about* the z-axis (δ_i). The robot and its kinematics are analyzed more thoroughly in [1, 129–132].

3.4.3 Robot Control

Control of the robot follows an existing approach and implementation wherein the motion of the end effector is considered to be roughly linear for small incremental displacements, over small time steps. It is also assumed that the motion is sufficiently slow such that dynamic effects can be excluded from consideration. Accordingly, the Cartesian end effector velocity $\dot{x}$ can be approximated from joint velocities $\dot{q}$:

$$\dot{x} = J(q) \cdot \dot{q}\,, \tag{3.1}$$

where $J(q)$ is the Jacobian matrix relating the workspace and joint space velocities evaluated at the instantaneous robot configuration q. Jacobian matrices were previously established separately for the LARS and snake robots, based on the partial derivatives of their respective forward kinematics equations. Given that the LARS-snake attachment at the distal LARS joint results in a series connection of the two robots, the joint motions of one robot remain decoupled from those of the other, and the final end effector pose can be found by a rigid transformation of the snake distal frame into the LARS distal frame. This independence means that the Jacobian of the combined robot is a concatenation of the individual robot Jacobians:

$$J(q) = [J_{LARS}(q_{LARS}) \mid J_{snake}(q_{snake})] \tag{3.2}$$

with

$$q = \begin{bmatrix} q_{LARS} \\ q_{snake} \end{bmatrix} . \tag{3.3}$$

Control of the combination robot using this method by Sen, Thienphrapa, and Taylor (first reported in [121]) represents a technical contribution as it is simpler than the approach taken by Kapoor *et al.* [133] using complete kinematic equations, yet usable.

Equation (3.1) is computed frequently (at 100 Hz) in discrete time. The fixed time interval between computations leads to

$$\Delta x = J(q) \cdot \Delta q \,, \tag{3.4}$$

that is, an incremental end effector displacement can be found given a set of incremental joint displacements. Of interest is the inverse operation: finding a set of joint motions that yield a desired motion at the end effector. A direct way to solve this is via the generalized inverse of the Jacobian.

$$\Delta q = J^{+}(q) \cdot \Delta x \tag{3.5}$$

However, a unique solution exists (the Jacobian is invertible) if and only if the Jacobian is square and full rank. This requires that the number of robot degrees of freedom be equal to the number of Cartesian (workspace) degrees of freedom, and that no joint can become aligned or redundant with another in configuration space. A simple example of a mechanism that fulfills these requirements is an XYZ Cartesian stage.

Indeed, the 11-DOF LARS-snake robot configuration space has a higher dimensionality than its six-DOF workspace, so a unique solution to Equation (3.4) does not exist in general. We are interested in a framework for resolving the spatial redundancy while approximating task goals as faithfully as possible. We would also like the ability to place restrictions on the motion of the robot, whether in configuration space (e.g., joint limits) or the workspace (e.g., forbidden regions). These requirements can be formulated as a constrained optimization problem [120], which can resolve multiple-solution situations as well as no-solution cases (e.g., when a command puts the robot outside of its reachable or dexterous workspace).

3.4.3.1 Objective Functions

This subsection on objective functions, and the next subsection (3.4.3.2) on constraints, provide background on the constrained optimization approach to robot control by Funda, Taylor, *et al.* [120] as it applies to our system. Several groups have used this method to control surgical robots. Li *et al.* used it in conjunction with boundary and guidance virtual fixtures to enable safe guidance of the Steady Hand robot in ear, nose, and throat (ENT) surgery [134–138]. Kapoor *et al.* developed a library of virtual fixture primitives within the constrained optimization framework to assist in suturing and other complex surgical tasks [139–141]. Kapoor *et al.* and Xia *et al.* applied the framework to the da Vinci robot for bimanual teleoperation [131, 142]. The same algorithm was used by Simaan *et al.* to control the snake robot as well [1, 129, 132, 133]. The purpose of this review is to facilitate discussion on the specific implementation for our system [121], starting on Section 3.4.3.3.

The primary objective of the robot controller is to move the end effector in response to an input motion command τ, or $\Delta\tau$ for discrete time intervals. (For our purposes, $\Delta\tau$ can be treated as synonymous with Δx of Equation (3.4), with τ signifying commands of external origin, such as from a manual input device in a teleoperation setup.) Following Equation (3.4), the corresponding objective function then is to find a set of joint displacements Δq that minimizes the 2-norm between the commanded and actual motions:

$$\underset{\Delta q}{\text{minimize}} \quad \left\| W_h \cdot (J(q) \cdot \Delta q - \Delta\tau) \right\| , \tag{3.6}$$

with

$$W_h = diag(w_{h,1}, \ldots, w_{h,6}) \, , \tag{3.7}$$

where $w_{h,i}$ is a scalar weight that adjusts the importance of each (three-dimensional)

Cartesian workspace degree of freedom in determining the optimal solution. We generally set $W_h = I$ to maintain a neutral workspace. However, it may be important for the end effector to reach the foreign body position, for example, but less important for it to do so through a specific orientation. In this case we can use $W_h = diag(1, 1, 1, w_{h,or}, w_{h,or}, w_{h,or})$ with $w_{h,or} < 1$. Different sets of weights are associated with different objective functions; the selection of weights is further discussed below.

Due to the kinematic redundancy of the robot, there is in general a continuum of possible joint displacements for a given desired end effector displacement. A solution that results in the least amount of overall joint motion can be obtained using an objective function that minimizes the 2-norm of joint displacements, as shown in Equation (3.8). W_j here serves a similar purpose to W_h above—it is a diagonal matrix where the (11) nonzero elements represent the weight of each joint.

$$\underset{\Delta q}{\text{minimize}} \quad \|W_j \cdot (\Delta q - 0)\| \tag{3.8}$$

Often times the home configuration of a robot is one in which the reachable or dexterous workspace is greatest. When a given robot motion is achievable via multiple solution sets of joint motions, it may be favorable to select the one that keeps the robot closest to the home or to some other preferred joint configuration. Preference for a certain configuration may also prevent the robot from drifting into joint limits, which can help avoid possible misbehavior as a result of inadequate solutions, or more practically, avoid damage to the mechanisms or cabling. This can be implemented as an objective function to minimize the 2-norm between the preferred and actual configurations (q_p and $q + \Delta q$ respectively):

$$\underset{\Delta q}{\text{minimize}} \quad \left\| W_p \cdot (\Delta q - (q_p - q)) \right\| . \tag{3.9}$$

3.4.3.2 Constraints

The low-level controllers, namely the Galil for the LARS and the FireWire-based device for the snake, are capable of preventing the breaching of joint position and velocity limits. It would be preferable, however, for such prevention to be built into the control calculations to facilitate smooth motion and reliable performance. As an example, an unconstrained controller may compute a set of joint motions of which a subset would exceed joint limits. The low-level controllers would disallow these displacements, thereby avoiding potential damage to the robot, but the end effector position would be erroneous as a result. There is no guarantee that the error would eventually be corrected because commands could continue to issue to the effectively immobile joints, and there is no mechanism for the remaining joints to compensate.

To enforce solutions to respect position and velocity limits at a higher level, we impose constraints on the solution Δq in the form shown in Equation (3.10).

$$A \cdot \Delta q \geq b \tag{3.10}$$

As mentioned, the basic set of constraints includes joint position and velocity limits,

$$\begin{aligned} q_{lower} \leq q + \Delta q \leq q_{upper} \\ -\Delta q_{max} \leq \Delta q \leq \Delta q_{max} \,, \end{aligned} \tag{3.11}$$

with q_{lower} and q_{upper} referring to the lower and upper joint limits respectively, and Δq_{max} to the maximum allowed joint displacements within one control step. These constraints can be rephrased to the format of Equation (3.10) as follows:

$$\begin{aligned} I \cdot \Delta q &\geq q_{lower} - q \\ -I \cdot \Delta q &\geq q - q_{upper} \\ I \cdot \Delta q &\geq -q_{max} \\ -I \cdot \Delta q &\geq -q_{max} \,. \end{aligned} \tag{3.12}$$

Workspace constraints can also be constructed in a similar fashion to the creation of the workspace objectives, but these constraints have not yet appeared to be necessary and thus have not been pursued.

A key virtual fixture in minimally invasive surgery is the virtual RCM, implemented in the LARS-snake robot following [140]. It is realized as a set of constraints, each of which limits the lateral movement of the robot in a particular direction from the RCM point. First, the point on the robot closest to the RCM is found; this point is assumed to reside along the snake stem. The Jacobian $J_r(q)$ of this point is computed, as are several perpendicularly radiating vectors. At least three vectors are needed in order to laterally constrain the stem to the inside of a polygon, and improved accuracy can be obtained by increasing the number of vectors (the present robot uses eight). Each radial vector $H_i \in \mathbb{R}^6$ is a unit normal with the last three elements set to zero. Then

$$H_i^T \cdot J_r(q) = A_{r,i} \in \mathbb{R}^{11}\,, \tag{3.13}$$

and the dot product $A_{r,i} \cdot \Delta q$ is the distance from the robot in the associated direction, so in constructing the constraint, $A_{r,i}$ is paired with the distance h_i of the robot from the RCM. If the snake stem passes through the RCM, $h_i = 0$. The constraint can be relaxed using a tolerance (e.g., 1 mm in any direction) if the structure imposing the RCM is compliant. With N radial vectors, the RCM constraint is

$$\begin{bmatrix} -H_1^T \\ \vdots \\ -H_N^T \end{bmatrix} \cdot J_r(q) \cdot \Delta q \geq \begin{bmatrix} h_1 \\ \vdots \\ h_N \end{bmatrix}. \tag{3.14}$$

3.4.3.3 Use of the Optimization Framework

The optimization framework enables the creation of virtual fixtures as described above, as well as various modes of operation prescribed by application requirements, as described below. In the standard form, n objectives and m constraints are composed as follows:

$$\begin{aligned} \underset{\Delta q}{\text{minimize}} \quad & \left\| \begin{bmatrix} W_1 & & \\ & \ddots & \\ & & W_n \end{bmatrix} \begin{bmatrix} C_1 \\ \vdots \\ C_n \end{bmatrix} \Delta q - \begin{bmatrix} d_1 \\ \vdots \\ d_n \end{bmatrix} \right\| \\ \text{subject to} \quad & \begin{bmatrix} A_1 \\ \vdots \\ A_m \end{bmatrix} \Delta q \geq \begin{bmatrix} b_1 \\ \vdots \\ b_m \end{bmatrix}. \end{aligned} \tag{3.15}$$

To illustrate, Equation (3.16) is a near-complete formulation for the LARS-snake.

$$
\begin{aligned}
\underset{\Delta q}{\text{minimize}} \quad & \left\| \begin{bmatrix} W_h & 0 & 0 \\ 0 & W_j & 0 \\ 0 & 0 & W_p \end{bmatrix} \begin{bmatrix} J(q) \\ I \\ I \end{bmatrix} \Delta q - \begin{bmatrix} W_h \cdot \Delta\tau \\ 0 \\ q_p - q \end{bmatrix} \right\| \\
\text{subject to} \quad & \begin{bmatrix} I \\ -I \\ I \\ -I \\ -H_1^T \cdot J_r(q) \\ \vdots \\ -H_N^T \cdot J_r(q) \end{bmatrix} \Delta q \geq \begin{bmatrix} q_{lower} - q \\ q - q_{upper} \\ -q_{max} \\ -q_{max} \\ h_1 \\ \vdots \\ h_N \end{bmatrix}
\end{aligned}
\tag{3.16}
$$

3.4.3.3.1 Selection of Objective Weights

The optimization framework provides a convenient interface for implementing various modes of operation by adjusting the weights of the optimization problem. As described in [120], higher weight values can be applied to certain degrees of freedom to penalize them during the optimization process, while lower weight values can be used to allow more freedom in other motions. However, the tuning of these weights remains an ill-defined manual process—an art—particularly as the number of joints increases. It follows that in practice, creating a new mode of operation for a complex or highly redundant robot can be an arduous undertaking.

We do not investigate the matter rigorously, but nevertheless an attempt to discuss the considerations is included here. Perhaps a large contributing factor to the difficulty in selecting an appropriate set of weights relates to the size of the parameter space. Empirical evidence of this explanation is found in the fact that many robots using this control approach have weights set incorrectly. In some cases this manifests as a mysterious but subtle malfunction that does not immediately alert the user of the condition, in contrast to more easily noticed abnormalities arising from software bugs, hardware problems, etc. More commonly, incorrect weights simply nullify the associated objective functions. If the purpose of the objective function is to condition the optimizer near a singularity, for example, the error may never be caught if the robot does not reach the singularity, or if the errors resulting from the singularity are misclassified and attributed to other, more common problems.

The size of the parameter space is due in part to the coupling of seemingly disparate objective functions. Minimization of total joint motion may appear to be a goal unrelated to following handle motion commands, but because the optimizer collapses all objectives into a single value, adjusting the weight of a single joint in one objective function necessarily affects the behavior of the entire controller. Additionally, the weights must be balanced among objective functions, lest one objective function be inadvertently given preference over others, and adjustment of one weight, as previously alluded, requires re-balancing of all weights over all objectives. This care is likewise required when new modes of operation are introduced, and is easily overlooked in the absence of tools or a systematic approach. Finally, if weights are to be configured dynamically, re-balancing must be performed on-the-fly as well.

Another aspect of the parameter space challenge is that weights are highly specific to a particular robot, making it difficult to reuse values between different mechanisms. Ref. [120] considers the equivalence of a unit translation of a prismatic joint with a unit rotation of a revolute joint—a good starting point. In general, the importance of a joint depends on parameters such as link lengths and units of measure, as well as on the behavior desired, which is tightly driven by the application and environment and can be to a large extent a subjective matter.

An additional challenge of tuning optimization weights for a specific robot and desired mode of operation is the potentially nonlinear and complex relationship between the weights and the resultant behavior. Objective functions may be in conflict with constraints and with other objectives as well; for example, configuration space constraints may make it difficult to judge the correctness of workspace objectives during development. Despite these difficulties, we have established a set of control modes described in the subsequent section. Though the values chosen may not be entirely correct given the current discussion, these modes appear to perform with sufficient adequacy for the intended application.

3.4.3.3.2 Modes of Operation

Coarse positioning of the robot can be achieved by assigning higher priorities to the LARS joints; this may be useful, for example, in the initial placement of the end effector at the surgical entry port, or for pulmonary motion compensation. Coarse positioning is obtained via the following weight assignments:

$$W_{j,coarse} = diag(L, L, L, L, L, H, L, H, H, H, H)\,. \tag{3.17}$$

Each diagonal element of W_j corresponds to a joint, $L = 0.0001$ is a value of very low penalty, and $H = 1000$ is a very high one. Thus $W_{j,coarse}$ allows all LARS joints (joints 1–7) except for joint 6 to move freely, while almost entirely disallowing movement of the snake joints (joints 8–11). Under teleoperation, this simulates the sensation of a one-to-one mapping between the six-DOF input device and the individual LARS joints. Joint 6, the LARS insertion stage s, is effectively disabled to prevent motion coupling with other joints due to the kinematic redundancy.

A dexterous-only mode is implemented using the weights shown in Equation (3.19). As the functional complement of coarse positioning, only the snake is allowed to move while the LARS joints are effectively disabled. Different values besides L are used to achieve fine-grain behavior. A joint weight $L < w_{j,i} \ll H$ has the effect of allowing subdued motion in joint i, a method that is redundant with setting a velocity constraint. We preferred this method in this case to encapsulate the subdued motion to this mode, but this redundancy further highlights the difficulty of setting weights. Besides its utility during the development and debugging phase, this mode is often used for demonstrations and lab tours.

$$W_{j,dex} = diag(H, H, H, H, H, H, H, 50, 25, 25, 25) \tag{3.18}$$

The main mode of operation used in the dexterous, minimally invasive surgical task is a fine positioning scheme that relies primarily on the snake joints for workspace coverage, but incorporates a small amount of RCM-respecting LARS motion to enhance reachability and dexterity. The joint weights that enable this behavior are:

$$W_{j,rcm} = diag(1, 1, 1, 1, 1, 0.2, H, 50, 25, 25, 25)\,. \tag{3.19}$$

Of note is that joint 6, the LARS insertion stage s, is heavily favored to allow motion along the tool axis to dominate over other coordinated joint motions that would otherwise achieve the same outcome. Also, joint 7, the LARS tool-roll stage θ, is heavily penalized to prevent tangling of the FireWire and power cables leading to the snake controller. The motion otherwise afforded by this joint is redundant with the capabilities of the snake, so the reduction in dexterity is not noticeable. Joints 8–11 corresponding to the snake appear to carry a heavier penalty than do LARS joints 1–5, which would suggest the preference of those LARS joints over the snake joints. In practice, however, the snake moves much more prominently as desired in this mode than the LARS does, again reflecting the non-obvious nature of weight selection.

Upon a user-initiated transition to this mode, a fixed offset along the snake stem in the present robot configuration is established as the RCM point, which can be disengaged and reset as desired. An RCM constraint is then constructed as described previously in subsequent iterations of the control loop. In this application, the snake is bent by image-guided control and must then be retracted via teleoperation, so the system must maintain awareness of the RCM point in order to automatically unbend the snake while the teleoperated retraction takes place. This scenario is not commonly encountered in minimally invasive surgery because robotic tools are typically rigid or completely teleoperated. To enable teleoperated insertion and retraction within the optimization framework, (1) a command $\Delta\tau_{insert}$ is formed such that the motion is along the snake stem (the z-axis in the tip frame):

$$\Delta\tau_{insert} = v_{insert} \cdot [F_{tip}(1,3), F_{tip}(2,3), F_{tip}(3,3), 0, 0, 0]^T \ , \tag{3.20}$$

where v_{insert} indicates both the magnitude and direction of the insertion speed (negative for retraction) and $F_{tip}(i,j)$ is the ij^{th} element of the tip frame; (2) the joint weights are set as follows to compute objective motion in only the LARS insertion stage s:

$$W_{j,insert} = diag(H, H, H, H, H, 1, H, H, H, H, H) \, ; \tag{3.21}$$

and (3) the constraints on snake joints θ_{L1} and θ_{L2} are modified to force the snake to gradually straighten as it is retracted through the RCM. It is not a strict point so the snake need not straighten immediately upon reaching it. Rather, it can gradually do so (a square function of the insertion depth was used) until it is fully straight when fully retracted through the RCM. This prevents the snake from snapping to a new position, possibly avoiding damage due to the sudden motion or due to collisions with the environment. There are two snake segments, so modification of the bend constraint proceeds in two phases. In short, the distal segment may remain fully bent if retraction has progressed only to the proximal segment.

Lastly, a safety mode in the event of communication loss with the snake sets all snake joint weights to H. This allows for continued functioning of the robot after the snake robot has been disconnected, whether by intention or otherwise. For all detected error conditions, an artificial command of $\Delta\tau = 0$ is used; after the debugging stage, most errors encountered were transient and recoverable in nature, so this strategy proved to be more convenient than program abort.

3.4.4 Dexterity Tests

When the end effector is in the heart it is preferred that motion be accomplished primarily by the smaller distal joints, with the larger proximal joints serving only to supplement workspace coverage. This section demonstrates the advantage of distal dexterity in terms of pivots about the apex (through which the robot enters), as well as reachability inside the heart phantom.

3.4.4.1 Reduction of Pivots about the Apex

Five arbitrary targets interior to the still heart phantom were selected, and the robot, initially positioned at the surgical entry point, was triggered to hit the targets under 3D ultrasound guidance in both dexterous and straight modes. Table 3.1 lists the resulting robot joint angles. Joint values for Target #2 are not shown because the straight tool failed to reach the target altogether. Target #3 was in a location unreachable by both tools (too shallow and lateral relative to the insertion point), so its joint angles are also not included.

Table 3.1: Robot joint displacements after reaching arbitrarily selected targets in a heart phantom. The bottom row computes the resulting amount of pivot about the apex. The straight tool was unable to reach Target #2, while neither tool was able to reach Target #3, so the corresponding joint angles are not shown.

	Target #1		Target #4		Target #5	
Joint	Dex	Straight	Dex	Straight	Dex	Straight
LARS X (mm)	1.67	1.69	1.69	3.41	1.69	3.41
LARS Y (mm)	42.13	39.55	35.71	41.66	35.71	51.23
LARS Z (mm)	5.50	10.93	6.01	28.54	7.10	13.69
LARS RX (deg)	21.04	16.20	24.50	37.05	25.33	23.77
LARS RY (deg)	5.61	3.83	4.80	2.44	4.35	2.69
LARS s (mm)	14.80	27.33	24.90	9.29	35.32	16.68
LARS θ (deg)	0	0	0	0	0	0
Snake θ_1 (deg)	3.85	0	13.13	0	2.43	0
Snake δ_1 (deg)	8.03	0	38.04	0	45.12	0
Snake θ_2 (deg)	40.00	0	39.73	0	33.48	0
Snake δ_2 (deg)	67.39	0	86.28	0	104.13	0
Apex Pivot (deg)	0.26	12.43	0.99	0.78	2.93	8.06

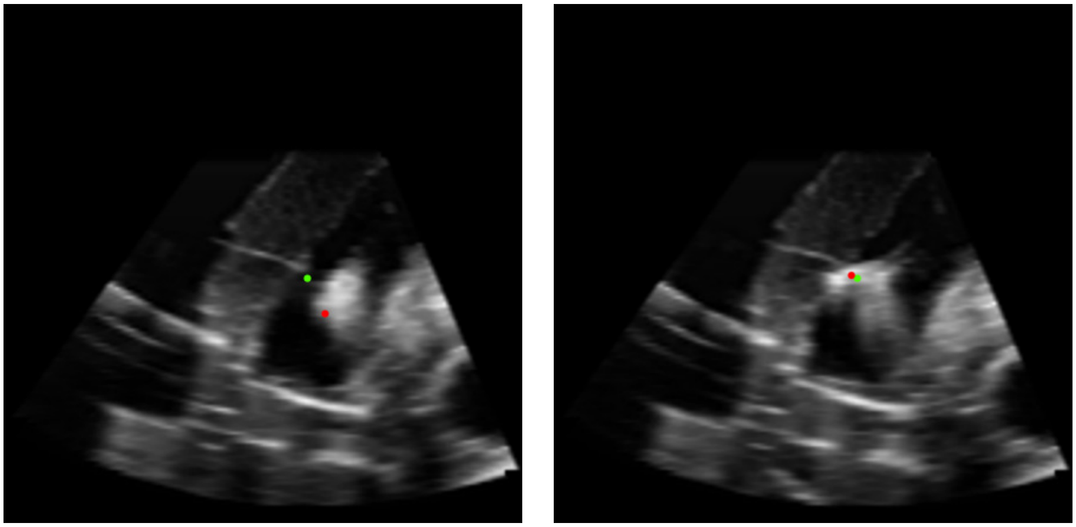
Table 3.1 illustrates the advantage that dexterous tools have over straight tools in terms of reduction in the amount of pivot about the apex, which can help decrease trauma to the tissue and blood loss through the incision. Correspondingly, the oper-

ational space of the robot outside the body is also reduced. Target #5 was reachable by both tools, but the straight tool needed to pivot slightly more (8.06°, vs. 2.93° for the dexterous tool) in order to orient itself to the same target. The straight tool for Target #1, however, pivoted much more. Following the conceptual diagram of Figure 3.3 and a wall thickness of 12–15 mm, 12.43° of pivot would displace the surrounding tissue by about 1.65 mm in both directions, or 3.30 mm in total. This quantity is significant relative to the tool diameter of 4.2 mm and the fluid pressures present in the heart. Target #4 was easy to reach for both dexterous and straight tools, so the respective pivot angles were less than one degree.

3.4.4.2 Reachability inside a Heart Phantom

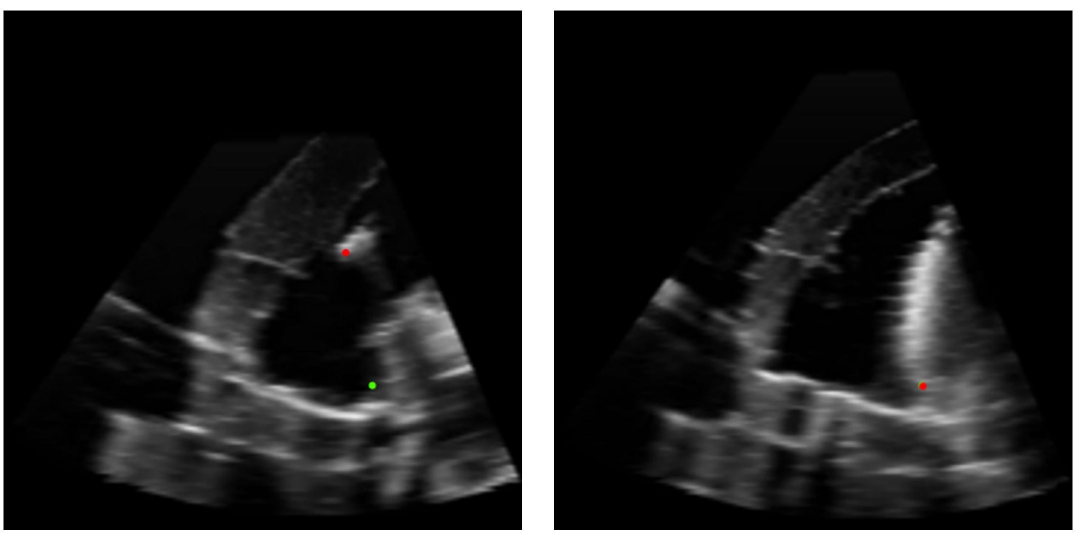
Using the same five targets, Figures 3.9–3.12 illustrate the advantage that a robotic tool with distal dexterity has over conventional rigid minimally invasive tools in terms of reachability. In two cases (Targets #1 and #5) the dexterous tool was able reach the target with less error. In one case (Target #2) the dexterous tool reached the target while the straight tool failed entirely. Target #3 is not shown because, as noted above, it was in a location unreachable by both tools (too shallow and lateral relative to the insertion point). Both tools were able to reach Target #4 due to the direct path from the insertion point to the target.

For qualitative verification, a 2.4-mm steel ball (smaller than the foreign body surrogate used in Chapter 4) was placed at a target location, and the magnetic robot



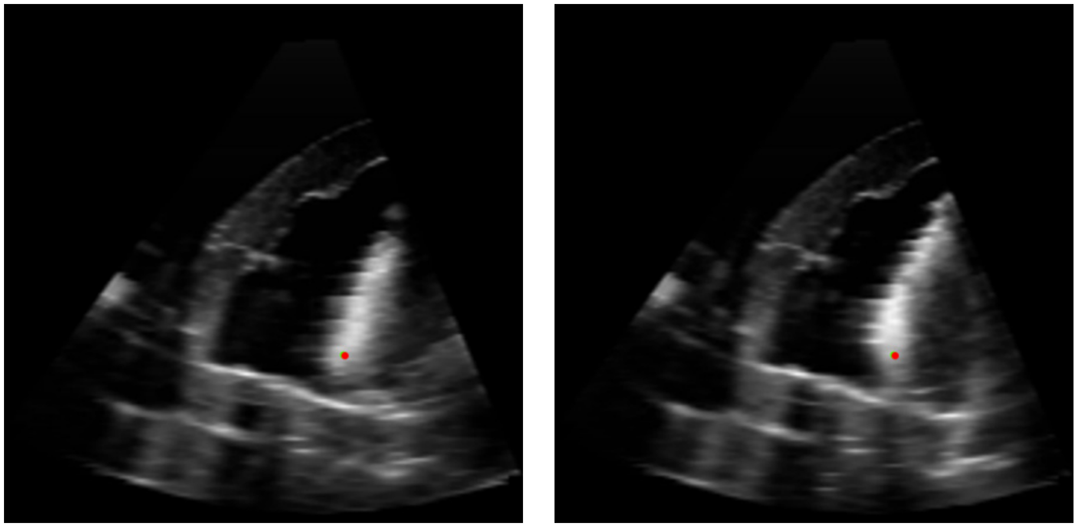
Straight Tool Dexterous Tool

Figure 3.9: Dexterity test in heart phantom, Target #1: Dexterous tool better able to reach target. The green dot represents the desired target, while the red dot indicates the actual position the robot was able to reach.



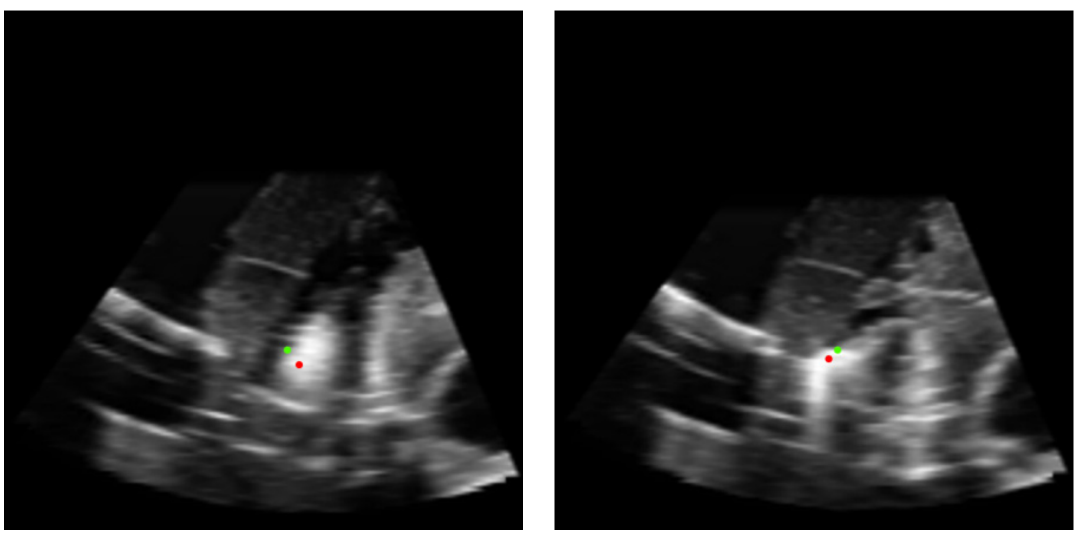
Straight Tool Dexterous Tool

Figure 3.10: Dexterity test in heart phantom, Target #2: Straight tool completely unable to reach target.



Straight Tool Dexterous Tool

Figure 3.11: Dexterity test in heart phantom, Target #4: Direct path, both tools able to reach target.



Straight Tool Dexterous Tool

Figure 3.12: Dexterity test in heart phantom, Target #5: Dexterous tool better able to reach target.

tip was guided to retrieve it. This is shown as the bright spot in the dexterous tool image for Target #5. The ball was collected successfully.

3.5 3D Ultrasound-Robot Registration

3.5.1 Preoperative Registration

Registration between the 3D TEE probe and the robot is performed with the snake tip acting as a fiducial. The TEE probe is set up to view the phantom in the water tank, and the robot is then moved to five non-coplanar positions within the imaging field of view. The image coordinates of the robot are manually segmented and paired with the corresponding robot coordinates. Procrustes analysis is performed on these point correspondences to determine the scaled but otherwise rigid transformation between the two coordinate systems.

Due to the possibility of minor inadvertent probe movements between experiment sessions, the registration process was performed at the beginning of each run. The fiducial registration error (FRE) over a total of four sessions was approximately 1.0 mm, a reasonable figure given the image resolution of about 0.8 mm/pixel. Sample registration points for each session are plotted in Figure 3.13, along with the corresponding FREs. We note the importance of streamlining the registration process for future work and will address the issue in its own right. The ensuing section describes our efforts in automatic online registration for the purpose of easing the workflow complications of image-guided robotic surgery.

The feed-forward accuracy of a robot with compliant segments such as the snake robot is in general a challenging problem and an ongoing topic of research. Additionally, errors in end effector position can arise due to imperfections in the intermediate link connections, and these errors can accumulate over link lengths. We refer to the respective literature mentioned previously for details regarding the open-loop accuracy of the LARS and snake robots. From the imaging perspective, errors in tracking and segmentation can arise, particularly with relatively low-resolution ultrasound images. In light of these potential sources of error, our system controls the robot position incrementally based on image feedback, using the difference between actual and desired positions as a control input. Thus the system is less sensitive to positioning errors, and the need for absolute accuracy is relaxed.

3.5.2 Online Registration

Registration is a standard task in surgical robotics and in robotics generally. Preoperative registration, as described in the previous section, is commonly used when coordinate systems remain fixed relative to one another, but such registration becomes invalid once any of the coordinate systems moves. In ultrasound-based robot guidance, it is possible for the devices to move with respect to each other during a surgical procedure. A TEE probe in particular is mounted on a flexible cable, so maintaining it in a fixed position can be challenging. Breathing will cause periodic

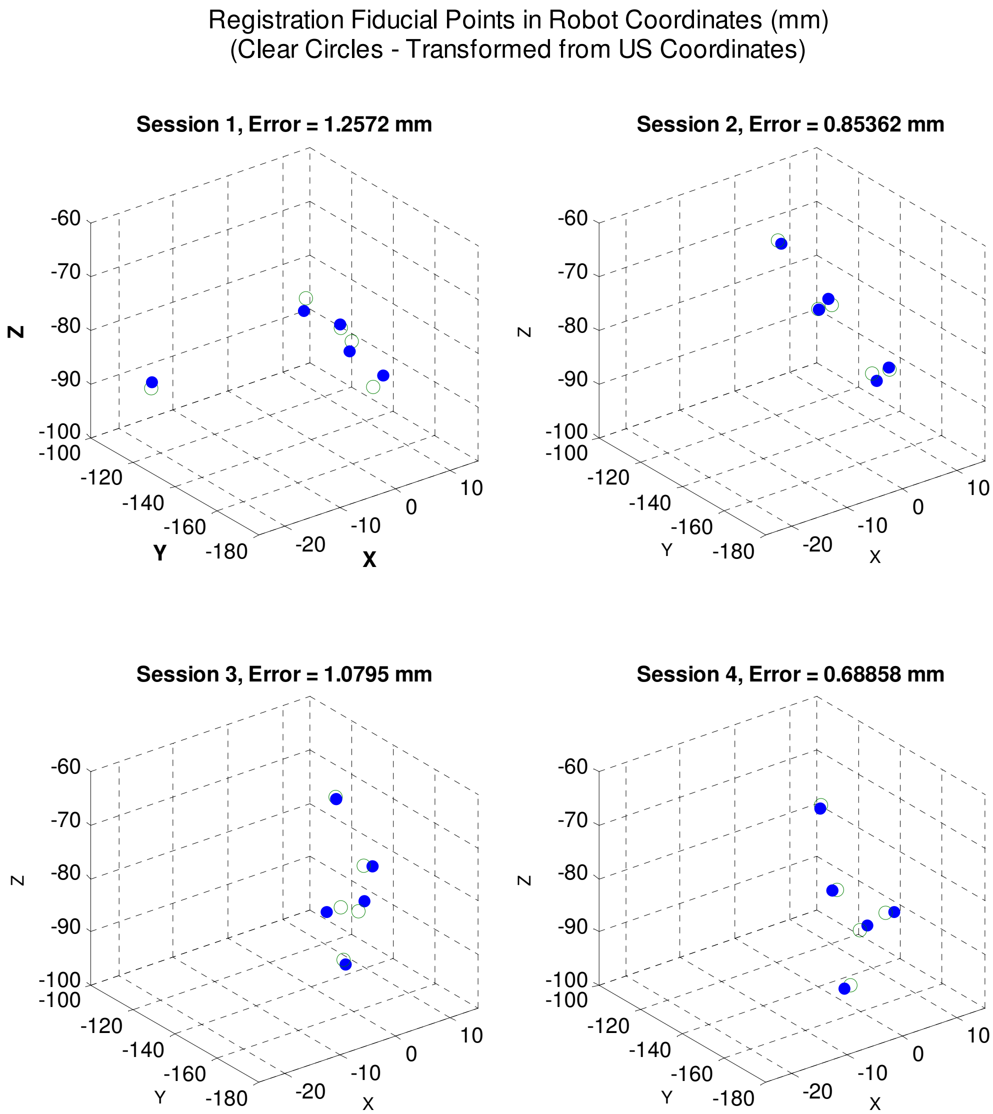


Figure 3.13: Sample points used in TEE probe-to-robot registration. Filled circles represent robot positions as measured in the robot coordinate system, while clear circles represent the same positions extracted from the ultrasound images and transformed into robot space. As a reference, the ventricle is about 45–55 mm in length, while the ultrasound volume is roughly a 120-mm-high pyramid. Because the pyramid is only about 75 mm wide at the base, fiducial points are not as spread out as normally preferred. The FRE is 1.0 mm on average.

movement of the probe, and once the probe is in motion, aperiodic shifts can occur as well. Even if a fixed position for the ultrasound probe can be enforced, it is beneficial to have flexibility in positioning, given that ultrasound often fulfills the multiple roles of visualization and guidance. Furthermore, preoperative registration adds complexity to an already complex surgical workflow. If the installation of fiducial markers in the workspace is required as part of the registration process, the surgical duration and complexity are correspondingly increased.

One solution to maintaining continuous registration of a moving probe is through the use of an electromagnetic (EM) tracker, as in [62], provided metallic components are kept out of the EM field. If the probe is tracked using a stationary EM tracker, the probe becomes free to move because the EM and robot coordinate systems remain fixed relative to each other. However, tracking coordinate systems is cumbersome due to the additional hardware requirements. Thus an elegant solution for automatic online registration that takes advantage of existing components is desirable.

In an image-guided surgical robot system, the robot itself, with a known tool shape, can be used as a mechanically tracked fiducial. The known robot orientation based on its kinematics, combined with its orientation computed from live 3D ultrasound images, can be used to compute and update the registration between the two coordinate systems in real time. We propose an automatic online registration method [143] that proceeds in three major steps: (1) detection of the instrument in the 3D ultrasound image stream, (2) computation of the instrument configuration in the ultrasound coordinate system, and (3) computation of the frame transformation.

3.5.2.1 Instrument Detection

Surgical instruments are commonly metallic, making them visible under ultrasound imaging. Furthermore, they typically feature a long shaft to allow a surgeon or robot to operate in confined spaces. Many algorithms exist that can take advantage of these properties to detect an instrument in an ultrasound image in an automated fashion, so we leverage the abundant body of work in this space. One often used technique is the Hough Transform, under which a line or rod shape can serve as an input parameter; more complex shapes can be specified for advanced detection. For non-rigid tools such as the snake robot, optical shape sensing fibers can be used to determine the orientation of the robot.

It is often the case that instruments are visible in ultrasound images, but their boundaries are difficult to discern, leading to errors in automated instrument detection. To combat such issues, tools can be augmented with markers that are easy to distinguish by automated algorithms, but that do not interfere with the task at hand. Placing rings around the tool shaft in the manner shown in [107] results in an ultrasound-visible pattern of bumps that is distinct for each orientation of the tool. However, this approach requires specialized machining of surgical tools. Vibration of the instrument could also aid in its visibility in ultrasound, but the approach would require additional resources to effect the vibration.

We propose the practical workaround of retrofitting of existing tools with a small detachable marker that has a distinct echogenic pattern. Two points along the length of a shaft are needed to determine its 5D orientation (the sixth being the roll degree of freedom). A very simple style of marker, such as two spherical shapes (Figure 3.14), can be used in this case. Normalized cross-correlation (NCC) performs sufficiently well in tracking such features, as we demonstrate in detail in Chapter 4. NCC with a sphere template was used to automatically track the two markers in ultrasound images. Although these markers are not visually clear in Figure 3.14 (*right*), the algorithm was able to detect them during our tests (Section 3.5.2.4) nonetheless. The detected marker coordinates are used to compute the orientation of the instrument in the manner described next (Section 3.5.2.2).

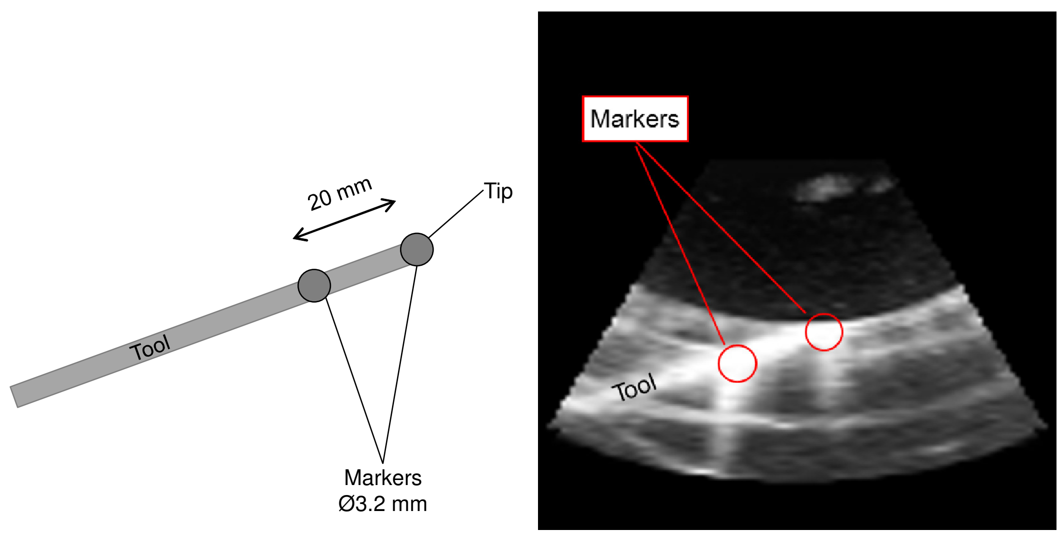


Figure 3.14: Surgical tool retrofitted with two spherical markers for online registration testing. (*Left*) Diagram of a tool with markers attached. (*Right*) View of the prototype under ultrasound imaging.

3.5.2.2 Computation of Instrument Configuration

After segmenting the instrument from the image, the next step is to use this knowledge to compute its configuration. We exploit the fact that surgical tools commonly have long shafts between the handle and end effector in determining a vector v_{tip} describing the instrument's 5D orientation in ultrasound coordinates. Equation (3.24) ultimately leads to the instrument pose ${}^{u}T$ in ultrasound coordinates (subscripts and superscripts u and r are used to denote the ultrasound and robot coordinate systems respectively; Figure 3.15 illustrates some of the pertinent coordinate systems). Beforehand, v_{tip} is used to determine the orientation relative to the negative z-axis in axis-angle form,

$$\begin{aligned} \bar{\omega} &= |-\hat{z} \times v_{tip}| \\ \theta &= \cos^{-1}\left(\frac{-\hat{z} \cdot v_{tip}}{|v_{tip}|}\right), \end{aligned} \tag{3.22}$$

from which a rotation matrix can be obtained using Rodrigues' rotation formula:

$$\begin{aligned} {}^{u}R &= P + \cos\theta(I - P) + \sin\theta[\bar{\omega}]_{\times} \\ P &= \bar{\omega} \otimes \bar{\omega}, \end{aligned} \tag{3.23}$$

where $[\cdot]_{\times}$ is the skew operator and $\otimes$ the outer product operator. Since the tool tip ${}^{u}p_{tip}$ was found as part of the instrument detection,

$${}^{u}T = \begin{bmatrix} {}^{u}R & {}^{u}p_{tip} \\ 0 & 1 \end{bmatrix}. \tag{3.24}$$

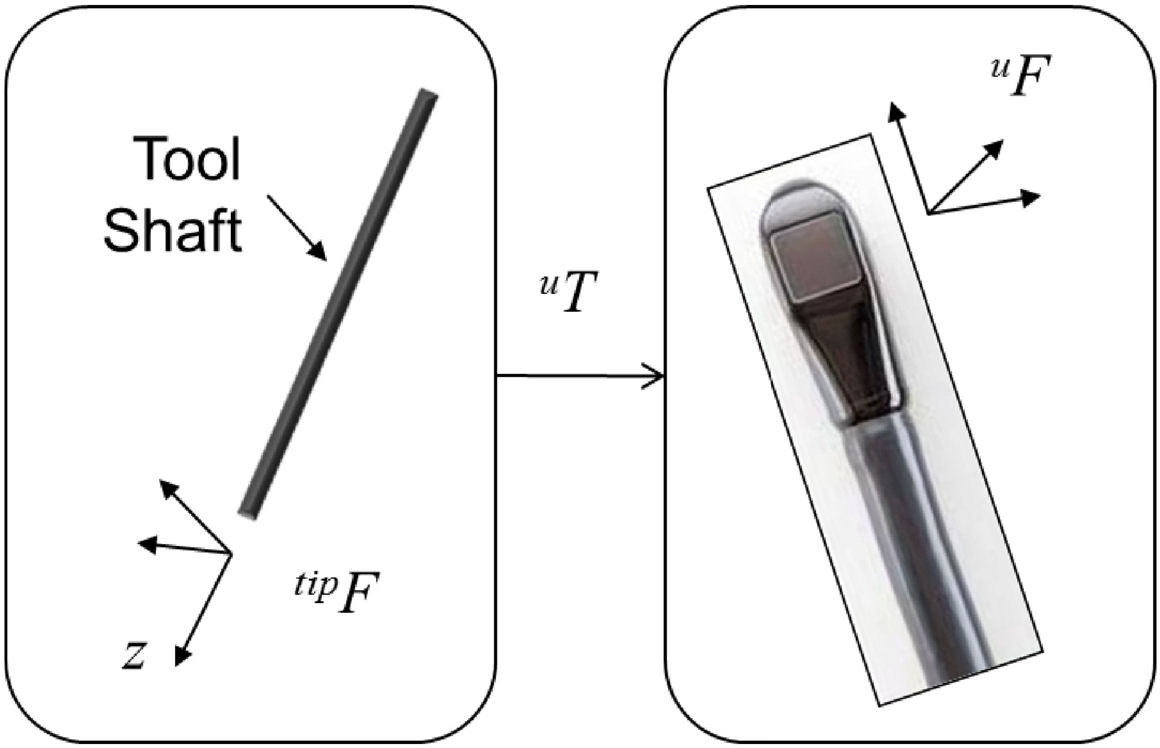


Figure 3.15: Illustration of the tip frame measurement in ultrasound coordinates.

3.5.2.3 Computation of the Frame Transformation

As mentioned, the end effector orientation in terms of the robot coordinate system, ^{r}T, can be readily queried, so the desired coordinate transform $^{r}T_u$ is obtained automatically and online:

$$^{r}T_u = {}^{r}T \cdot {}^{u}T^{-1} \, . \tag{3.25}$$

3.5.2.4 Basic Tests and Discussion

A basic test was performed to examine the feasibility of the proposed approach. First, a preoperative registration was performed to obtain a transformation for comparison; the FRE was 1.0 mm in this case. A central workstation computer was used to control the robot and acquire ultrasound images. Acquisition was carried out in a separate thread of control, with new images buffered in a continuous loop. Simultaneous access to the buffer by the both the acquisition and main threads was prevented using semaphores provided by the Windows operating system. Since the main thread was able to query the most recent image from the acquisition thread as well as the instantaneous robot state from the controller, synchronization between ultrasound and robot measurements could be maintained to within one frame of the 20 Hz image stream.

With the tool maintaining a fixed orientation ($R_x = 34.4°$), the robot was guided to nine different arbitrary positions within the imaging field of view, upon which the above online registration method was applied. In computing the orientation of the tool, use of the online method resulted in an average difference of 3.1° in the rotation axis and 2.4° in the rotation angle versus use of the constant transformation. Comparison of position was not performed because both registration methods used the same tracked tip feature.

Next, the tool was guided to a known location in the robot coordinate system, namely the origin (in the home configuration of the robot, the end effector is not at the origin by design due to an angle offset in the mechanism, so the robot had to travel about 80 mm). The target registration error (TRE) was 0.93 mm using the preoperative registration and 1.17 mm due to the online registration. The ability to produce reasonable estimates of position and orientation using the proposed online registration method suggests further work can be done in this direction.

Simplifying assumptions are made that the surgical instrument has a distinct long axis, and that the instrument is symmetric about this axis. It is also assumed that the tool enters the field of view from the "top" of the image. In [144], similar assumptions are made with regards to the nature of laparoscopic instruments and their entry into the field of view of an endoscope, so the assumptions made here are not unreasonable. Nevertheless, the tests described above are admittedly simple, so further development and rigorous analysis is necessary. To start, an extension to 6D, the extra dimension being a rotation of the instrument about its shaft, can be achieved by either (1) attaching a geometrically asymmetric marker, or (2) making inferences based on the specific circumstances of the application. The assumed entry direction of the tool into the image is one example.

Other challenges that online registration faces in practical use include noisy data and errant segmentation of instruments, events which can be expected due to the motion of the tools and changing appearance of the environment. An estimation approach, such as with an (extended) Kalman filter, may be appropriate to address these conditions. Thus, issues that must be resolved for the proposed online registration approach to be useful include robust segmentation of the robot in the images, detection of robot departure (partial or full) from the field of view and correct segmentation in such situations, and determination of when assumptions, such as the direction of robot entry into the image, are breached. It is also important that the computations be performed in real time in order to faithfully reflect live conditions.

With automatic online registration, the need for preoperative registration, fiducials, manual intervention, or a fixed imaging probe would be eliminated, thereby simplifying the workflow of robotic surgery significantly. An additional potential benefit is the real-time monitoring of registration accuracy. Assuming the scale between the imaging and robot coordinate systems is known, changes in the tool configuration between multiple image frames, as measured in the individual coordinate systems, can be compared with each other, as shown in Equation (3.26). A disparity between the measured displacements seen in each coordinate system can be used, for example, to trigger a dead man switch.

$$e = \sqrt{\| {}^{r}p_{tip}(t) - {}^{r}p_{tip}(t-1)\|^2 - \| {}^{u}p_{tip}(t) - {}^{u}p_{tip}(t-1)\|^2} \tag{3.26}$$

3.6 Image Guidance

3.6.1 Proposed Workflow

A brief summary of the system operation is outlined below. Greater details are provided later in the text as part of the description on real-time experiments.

- Preoperatively, the foreign body is coarsely localized, and the TEE probe is positioned such that the moving foreign body is visible in the field of view.

- Registration between the probe and robot coordinate systems is performed, following the procedure described above.
- The interventionalist identifies the foreign body in the 3D ultrasound stream, providing the system with the target to track.
- The robot is introduced through a transapical access port; upon completion, the surgical entry region, or RCM, is established.
- 3D ultrasound-based tracking of the foreign body begins.
- Once the system decides on where to capture the foreign body, it uses the transformation obtained from preoperative registration to translate tracked image coordinates into robot coordinates.
- The tracking subsystem uses these coordinates to guide the robot to capture the foreign body.
- After the foreign body is captured, the robot is withdrawn from the heart along with the foreign body. This step can be executed autonomously or by teleoperation.

3.6.2 Visual Servoing

A major advantage of an image-guided robotic system is the ability of the robot to sense its position with respect to the environment in real time, and adjust its trajectory accordingly. The advantage applies to manual guidance of instruments as well. Such a capability is particularly useful for robots with kinematic models or calibrations that are imperfect, such that the actual position of the end effector tends to deviate from what is computed from kinematics alone. In the snake robot, friction in the tubes and wires, along with minor mechanical tolerances, can affect the joint bend angles in difficult-to-model ways. Additionally, visual servoing can help overcome imperfect registrations between coordinate systems.

In light of these considerations, our system is provisioned to correct the path of the robot based on comparison of the visually observed robot tip position with the expected position. However, while snake robot open loop positioning errors are expected, thus motivating the use of vision feedback, information from ultrasound is also subject to uncertainty. We begin task execution by first sending to the robot controller a desired motion vector directed from the initial robot position to the goal position. We then use vision to update the motion vector, provided:

1. The robot position is tracked properly in the ultrasound image: normalized cross-correlation is used to track the robot tip, and proper tracking in a given ultrasound frame is assumed when the correlation value is at least 0.9—the method is fully explained in Chapter 4 for tracking a foreign body;

2. The error between the tracked and expected positions is greater than 3.0 mm, a threshold reflecting the snake robot diameter (4.2 mm, Section 3.4.2) and the tracking error (2.3 mm, Section 4.3); and

3. At least 0.5 seconds has elapsed since the previous update, though tracking is performed in every frame. This interval was chosen to allow for a few infrequent corrections (if necessary) during the 3–4 seconds of robot travel time (Sections 5.3.4.2 and 5.4.3.2).

This intermittent style of visual servoing is used because the robot kinematic model is trusted as good enough compared to ultrasound tracking; the latter is used for the initial trajectory and to repair significant trajectory deviations along the intermediate path. The robot performed well throughout the experiments of Sections 5.3 and 5.4. However, a more elegant fusion of multiple unreliable information sources for robot control, e.g., open loop robot positioning and ultrasound sensing, is an interesting improvement to pursue in the future.

3.6.3 The Role of Autonomy

Image guidance and visual servoing are technologies that lead naturally into the theme of autonomous control. The use of autonomy must be considered carefully in critical applications such as surgery. A premature system that encounters an unmodeled circumstance may create harmful situations, but risks can be mitigated via partial autonomy. For example, autonomous control can be limited to the most complex or dangerous part of a task, or perhaps the most repetitive and mundane part, while the remainder of the procedure can be performed in a conventional manner. Other possibilities include having an otherwise autonomous task progress only when explicitly commanded by human input, or partitioning between autonomous and manual responsibilities. In any case, a human expert should have full control over surgical procedures, be it through supervision of robot actions or a fall back to manual mode.

Autonomous capabilities are of interest because they offer several benefits. Robots can enable tasks to be performed in extreme scenarios, including hazardous environments, and those that exceed human physical capabilities. Partial or full autonomy can also assist procedures that are cognitively demanding. Minimally invasive retrieval of a cardiac foreign body under ultrasound guidance, for example, requires (1) a mapping between the ultrasound and device coordinate systems, (2) a mapping between the redundant degrees of freedom of the device and its overall configuration, and (3) guidance of the device with respect to the anatomy. Autonomous control would be advantageous in this situation because image registration, dexterity, and virtual fixtures can be challenging to manage intuitively while performing a complex task. Shared autonomy and cooperative control are possible solutions, and depending on the situation, preferable. Nevertheless, increased autonomy presents tangible benefits. A surgeon may be able to focus on task performance as opposed to device manipulation, and the accuracy of procedures can be improved, which together can help decrease staff time and reduce costs. Finally, the repeatability of procedures can be enhanced by means of smoother device trajectories—Li *et al.* [136] found that due to stiffness in the handle, cooperatively controlled motions may be less smooth than those produced under teleoperation. By extension, autonomous control may be most favorable in this regard.

In this work, a system and strategy for a new surgical application are proposed, and the efficacy of the solution is demonstrated by autonomous guidance under 3D ultrasound. One rationale for this choice is that the task is a candidate for an autonomous approach due to its complexity, as discussed above. Also, because the application is new, we can better focus on the system itself and assess its baseline performance by decoupling the possible human factors. On the other hand, the system is agnostic to the origin of commands, be they from automatic tracking, from a manual input device, or some intermediate hybrid. This perspective echoes that of Li *et al.* [134], wherein the algorithms were tested using autonomous control while a human-machine collaborative system was the ultimate goal.

3.7 Chapter Summary

This chapter describes the components and interconnections of an image-guided surgical system for minimally invasive retrieval of foreign bodies from a beating heart. The system consists of a robot with a highly dexterous end effector, a 3D ultrasound (TEE) system, and an intermediary PC to perform tracking and guidance computations. We explain that dexterity is needed in order to reach arbitrary locations in the heart without pivoting the instrument about the apex. The robot is then required to be slow in order to ensure safe operation. Methods for preoperative and online registration between the ultrasound images and the robot are described, followed by the envisioned information flow and system operation. Also introduced is a beating heart phantom for experiments.

The research contributions embodied in the system development leans predominantly towards the engineering effort. We were able to leverage the accomplishments of subcomponents developed previously, so we describe them briefly in the chapter and defer to external literature for further details. Greater emphasis is placed on those innovations specific to the current effort, including the physical attachment of the hybrid LARS-snake robot and its kinematic model. Also discussed is the extension of an existing control framework to support a virtual RCM as well as different modes of operation. With a system in place, the next step is to develop the ability to track a foreign body in a beating heart using 3D ultrasound, in order to understand the behavior of the motion and subsequently devise a strategy for securing the target.

Despite the prior availability of core subcomponents, the system described in this chapter represents the first known instance of robot steering using 3D TEE, guidance of a highly dexterous end effector using 3D ultrasound, and 3D ultrasound-based robot guidance with a large control bandwidth (e.g., 20 frames per second). Thus the development of the system alone represents a meaningful contribution to the research community.

Table 3.2: Technical barriers overcome (Section 1.1.4)

• Establishment of a platform for investigation of the problem, with real-time interplay between 3D ultrasound, tracking, guidance, and the robot
• Demonstration of a combined robot with only the individual robot Jacobians, as opposed to a full kinematic model

Table 3.3: Research contributions (Section 1.2)

• First known system for dexterous robot steering using 3D TEE
• Creation of robot operation modes not previously found in the literature
• Development of a FireWire-based motion controller with minimal cabling and open source components

Chapter 4

Foreign Body Tracking and Motion Characterization

In much of the previous work using 3D ultrasound for robot guidance, attempts have been made to maintain the tracking problem at a manageable level of complexity. For example, in [98, 102, 107, 109], the targets are static in an empty water tank. Tracking of moving cardiac structures is a considerably more complex problem; in [12], initial motion measurements on the mitral valve were extracted manually from 3D ultrasound sequences to inform the design of a heartbeat compensation device. As part of the operation of the device, intraoperative 3D ultrasound and an extended Kalman filter-based predictive model were used to track the motion of the mitral valve in one degree of freedom. Still, cardiac structures, such as the valves and chamber walls, have motions that are amenable to predictive modeling because their behavior is generally periodic and well understood. While positive results were attained by employing predictive control to track heart structures in one dimension in [12, 50], the requirements for retrieving a foreign body from a beating heart differ from these examples in multiple ways.

A foreign object within the heart exhibits a less constrained behavior, being subject to collisions with deformable and irregularly-shaped surfaces, amidst heartbeats and turbulent blood flow. Thus, a foreign body may at times exhibit periodic-like motion in some or all spatial dimensions, and at other times the motion can appear arbitrary [145]. A reliable intraoperative model for such a complex scenario would be difficult to compute in real time because it would need to incorporate a geometrical model of the heart, computational fluid dynamics, and interruption of blood flow around the foreign body, among other influences. Indeed, modeling the clinical problem would be an involved subject of investigation in its own right, beyond the scope of the present work.

Thus, in developing a minimally invasive surgical system for capturing a foreign body from a beating heart, an experimental study of motion characteristics would be informative. Pulmonary motion is presently decoupled from the analysis as it is much slower and more predictable in relative terms. Such motion does not add significant complexity to the tracking problem, and in a surgical setting a robot can compensate to keep the end effector steady with respect to the surgical insertion point. This section reviews the findings of our efforts in understanding the behavior of foreign bodies in a beating heart [145].

4.1 Real-Time Foreign Body Tracking

4.1.1 Selection of Normalized Cross-Correlation

A fundamental system requirement is the ability to track, using 3D ultrasound, the location of a foreign body in real time as it travels about a beating heart. For motion characterization alone, this capability is not absolutely necessary because slower methods would suffice (indeed, as explained below, a ground truth for assessing the tracking method is provided by manual segmentation results). However, a real-time tracking method for characterization would be beneficial for a number of reasons: (1) it would enable greater quantities of data to be analyzed, (2) it would transition more readily to robotic retrieval, and (3) it would facilitate comparisons of tracking performance between the characterization and retrieval implementations.

Prior to experimentation, we supposed that a free moving foreign body in the heart could move at speeds comparable to that of the heart wall—on the order of 300 mm/s [116]. We require a tracking method that can update at a rate that is fast enough to monitor this activity. In the present case, the limiting factor is the frame rate of the 3D ultrasound image stream (approximately 20 frames per second), so the algorithm execution time should be less than 50 ms for each image volume, i.e., as fast as possible.

During our initial evaluation of tracking methods, we focused on those that could be implemented to run in a computationally efficient manner. One candidate approach used sum-of-squared differences (SSD), a standard technique that measures the similarity between a template and a candidate signal based on the distance between the two. Its effectiveness has been demonstrated in tracking atrial septal defects [87] and other internal organs [91] in 3D ultrasound. However, while we found SSD to be capable of tracking a point on the heart wall [146], its ability to track a foreign body was greatly diminished as it was unable to follow an object against a changing background without drifting. Next we considered normalized cross-correlation (NCC), a well-known template matching algorithm that can avoid the drift problem by exploiting the foreign body shape. Thus, as part of the work in [145] we devised a tracking technique based on NCC.

In NCC, the basic form of the correlation coefficient at each image voxel, $R(u, v, w)$, matching a template $t(x, y, z)$ against a candidate sub-image $f(x, y, z)$, is given in Equation (4.1). Normalization, accomplished by subtracting the mean from both the template and the sub-image under it ($\bar{t}$ and $\overline{f}$ respectively) and dividing by their standard deviations, provides for robustness against changes in intensity, which can occur as both the foreign body and background move. A thorough description of the technique can be found in standard image processing texts (for example, [147]).

$$R(u, v, w) = \frac{\sum_{x,y,z} \left[f(x, y, z) - \overline{f}\right] \left[t(x - u, y - v, z - w) - \bar{t}\right]}{\sqrt{\sum_{x,y,z} \left[f(x, y, z) - \overline{f}\right]^2 \sum_{x,y,z} \left[t(x - u, y - v, z - w) - \bar{t}\right]^2}} \tag{4.1}$$

NCC in this conventional form incurs considerable computational overhead. It is at its core a convolution operation in 3D, so with template of size n^3 and a search space of size m^3, the computational complexity is on the order of $O(n^3m^3)$, or roughly $O(n^6)$ for similarly-sized templates and search spaces. Each step in turn involves several additions and multiplications.

Because NCC is highly parallelizable, graphics processing unit (GPU) and parallel computation can be used to reduce the computational latency. We chose an alternative (though not mutually exclusive) route of using the formulation presented in [148], with a Matlab implementation provided by [149]. A brief outline describing the mechanics behind the computational efficiency of this algorithm follows.

Referring to Equation (4.1), let $f'(x) = f(x) - \bar{f}$ and $t'(x) = t(x) - \bar{t}$ (abbreviated to 1D)—each signal with its mean removed. The numerator then becomes

$$\sum_x f'(x)t'(x-u)\,, \tag{4.2}$$

a standard convolution that can be computed equivalently and generally more quickly in the frequency domain via

$$\mathcal{F}^{-1}\left\{\mathcal{F}(f')\mathcal{F}^*(t')\right\}\,, \tag{4.3}$$

where $\mathcal{F}$ denotes the Fourier transform.

In the denominator, the template factor $\sum_x \left[t(x-u) - \bar{t}\right]^2$ can in fact be precomputed, leaving only $\sum_x \left[f(x) - \bar{f}\right]^2$ for consideration. Computation of this factor can be optimized by noting that many of the operations for each voxel overlap with those of adjacent voxels, so on the first encounter with a voxel the result can be tabulated for subsequent operations. Construction of this table requires $O(n^3)$ operations, a vast improvement over the $O(n^6)$ runtime of the straightforward algorithm.

NCC can be considered a fundamental similarity measure as it is widely applicable and requires minimal if any parameter tuning. It is easy to understand, implement, and test, and its shortcomings, such as inability to accommodate rotation and scale, are well understood. As a result, it is often the initial method of choice for applications that do not focus on matching or tracking specifically. Our initial consideration of NCC was guided by the aforementioned influences, as well as by the accessibility of resources and support for it. During evaluation it performed adequately for our purposes, and furthermore we were able to devise workarounds for its weaknesses. It was therefore selected as the operative method.

4.1.2 Tracking Approach

We initialize the tracking system by interactively delineating the foreign body in a still image frame, thus defining the template for automatic correspondence searches in all subsequent frames. A template size of $10 \times 10 \times 10$ voxels was chosen as it is suitable for enclosing the target to be tracked, and a search space of $40 \times 30 \times 30$ voxels was determined to be large enough to find the target as it moves between frames. In a continuous loop, the computer obtains the most recent ultrasound image from the stream as described in Section 3.5.2.4, analyzes the image to determine the instantaneous foreign body position as explained next, and repeats the process.

The similarity measure $R(u, v, w)$ is a real number in $[-1, 1]$, with higher values indicating a positive correlation and values tending toward -1 signifying a match of inverted intensity. The position of the foreign body in each image frame i is taken to be that with the highest correlation ($R_{i,max}$) with the template in that frame. A search for the foreign body in each frame is performed within a cube-shaped sub-image, referred to above as the search space. The position of the search space in each frame is centered around the tracked foreign body position from the previous frame. Computational efficiency can be further improved by placing a smaller search space intelligently based on an estimated direction of travel; this represents an avenue for further improvement.

Particularly with ultrasound data, reliance on a predefined template can make NCC vulnerable to identification errors as noise and artifacts may cause the appearance of the target to vary. Indeed, our treatment of inherently acoustic information as optical images for the purpose of using vision-based tracking methods leaves room for future improvement. Nevertheless, the effects of errant tracking can be mitigated by discarding image frames associated with sufficiently low values of $R_{i,max}$ implying misidentification of the target. A frame i is discarded if $R_{i,max}$ is less than c standard deviations below the mean correlation value of the current trial. The condition for discarding the tracking result of a frame is stated in Equation (4.4) and discussed further in Section 4.3.

$$R_{i,max} < R_{mean} - c \cdot R_{SD} \tag{4.4}$$

Recovery from lost tracking is performed by expanding the search space to a box of size $60 \times 45 \times 45$ voxels and recentering it to the centroid of past foreign body positions. The functionality of this recovery method was found to be satisfactory in practice as it allows tracking to resume in a relatively short amount of time. Sporadic discontinuities due to imperfect tracking can be tolerated because the lack of instantaneous certainty can be overcome in the application.

Given the challenges that ultrasound presents as an imaging modality, tracking results may be intermittently erroneous even when using more sophisticated techniques. For frames in which the reported foreign body location is unavailable or doubtful, it is preferable to have the system wait until tracking is recovered, allowing the robotic retrieval instrument to maintain its current position, or to move into a known safe configuration if the uncertainty is prolonged over several frames.

4.2 Experimental Setup

An experimental setup for studying the motion of foreign bodies in a beating heart is illustrated in Figure 4.1. It includes an ultrasound system, a beating heart phantom, and a PC (2.3 GHz Xeon CPU, 4 GB of RAM) that acquires live streaming ultrasound volumes over a wired TCP/IP network connection. This system does not include a robot, but is otherwise identical to the one presented in Section 3.

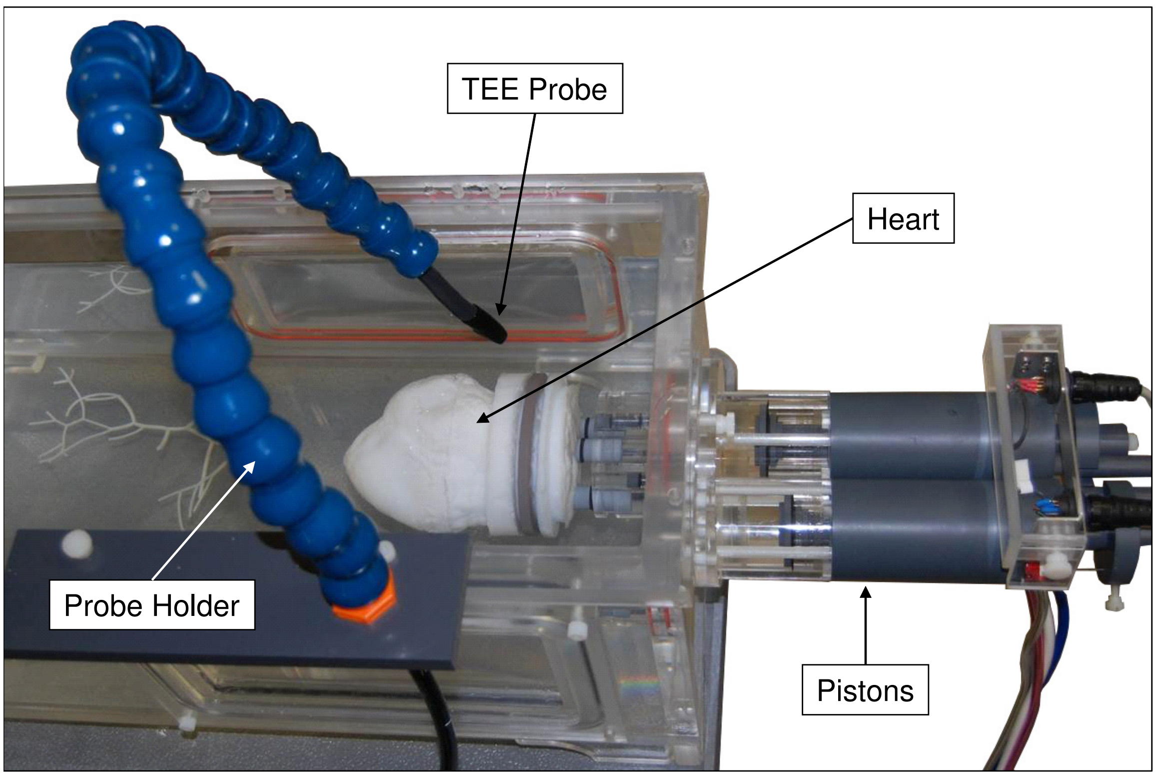


Figure 4.1: Beating heart phantom and TEE probe inside the water tank (Figure 3.4, reproduced here for clarity).

In foreign body characterization experiments [145], a 3.2-mm steel ball was inserted into the right ventricle of the beating heart phantom (detailed in Section 3.2) to mimic a clinical trauma case due to similarities in size (typically 2–5 mm) and material of the foreign body. Figure 4.2 shows ultrasound image slices of the phantom with the foreign body outlined, while Figure 4.3 provides a 3D visualization. The foreign body can be difficult to discern from the heart wall in a single image. If available, image artifacts unique to the foreign body can be used by our NCC-based method to aid in tracking. Motion of the foreign body over consecutive frames also reveals information about its position that further allows for improved tracking performance.

Submerged in a water tank, the heart was activated with a motion programmed as shown in Figure 3.5. Five 20-second sets of ultrasound data were acquired at a rate of 20 frames per second. The duration was empirically chosen based on the amount activity that would occur during a given trial. In addition, 20 heart cycles appear to provide adequate data for robust testing of online tracking algorithms, based on the literature. For example, [110] evaluated instrument tracking in a beating heart using a five-second window.

4.3 Tracking Accuracy

The value of c in Equation (4.4) can be tuned to the level of confidence desired; we found that $c = 0.9$ exhibited a reasonable compromise between spatial and temporal accuracy. As shown in Figure 4.4 (*top*), choosing a smaller c would lead to greater

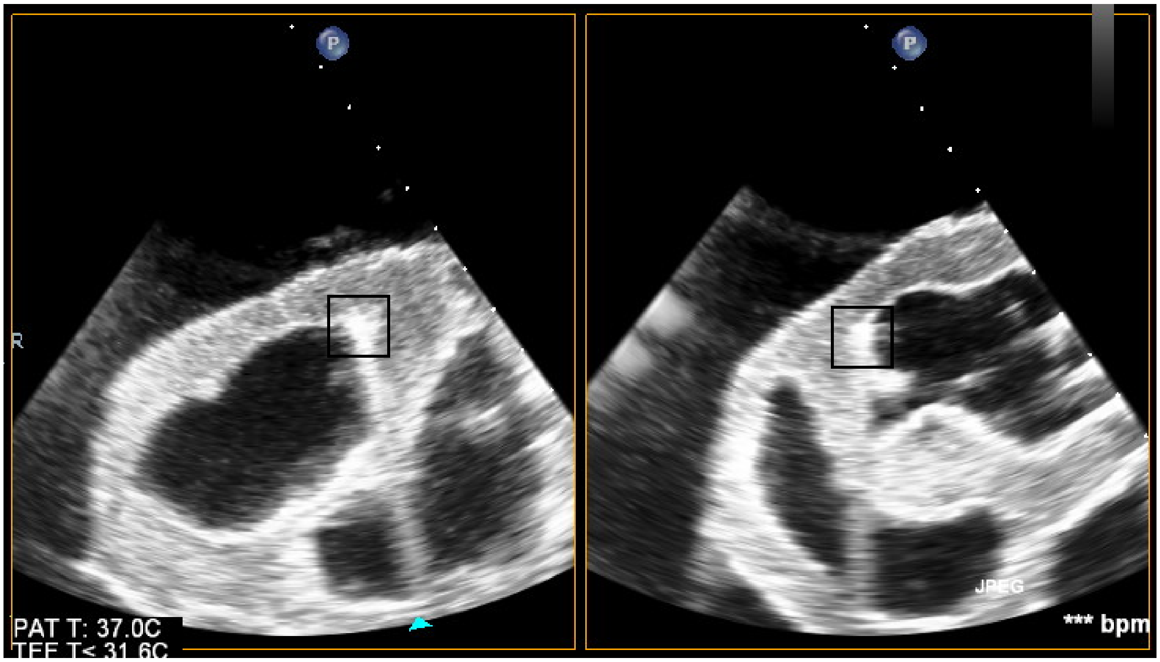


Figure 4.2: Orthogonal ultrasound image slices showing the foreign body (outlined) inside the heart phantom.

spatial accuracy at the expense of dropping more frames (*bottom*), and vice versa. A value of 0.9 was selected because it was the greatest value of c before a sharp rise in error. Table 4.1 lists the number of discarded ("dropped") frames and accuracy for all datasets. NCC-based foreign body tracking with detection of low confidence results (post-drop error) was accurate to within 2.3 mm RMS after dropping 18.5% of frames. As expected, in the case of NCC without frame dropping (pre-drop error), tracking was lost in some frames resulting in mean errors that were almost twice as high (4.3 mm).

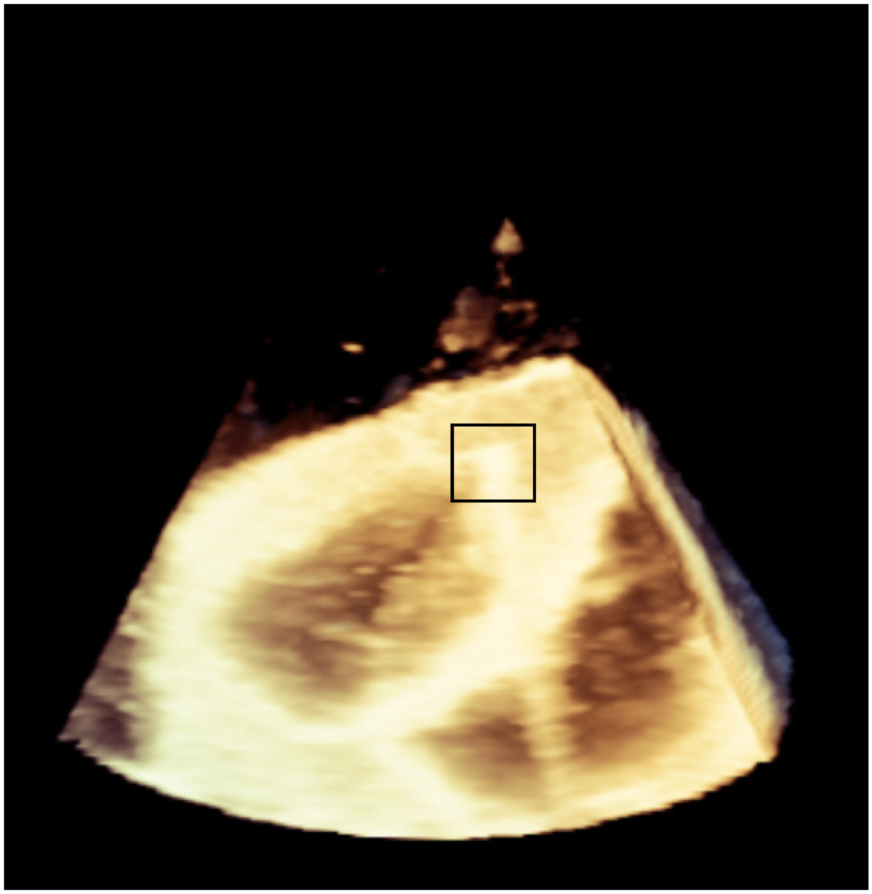


Figure 4.3: 3D rendering, generated by the ultrasound system, of the ultrasound volume with foreign body outlined.

Table 4.1: Foreign body tracking accuracy

	Datasets					Aggregate			
Measurement	A	B	C	D	E	Min	Mean	Max	SD
Pre-Drop Error	6.2	4.8	5.3	2.5	2.6	2.5	4.3	6.2	1.7
Post-Drop Error	2.0	3.1	2.6	1.8	2.1	1.8	2.3	3.1	0.5
% Frame Drop	20.9	15.5	20.1	22.9	13.3	13.3	18.5	22.9	4.0
$100 \times R_{mean}$	80.4	71.0	83.6	85.1	84.0	71.0	80.8	85.1	5.8
$100 \times R_{SD}$	9.2	8.5	10.3	4.8	3.7	3.7	7.3	10.3	2.9

Note: Errors in mm RMS

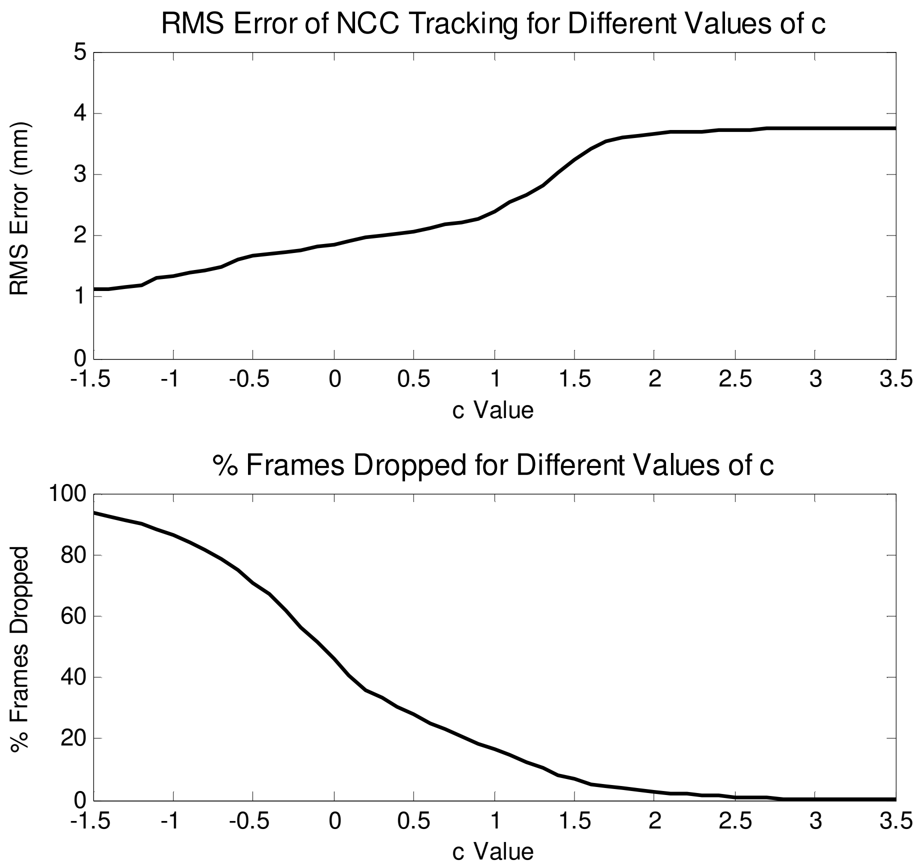


Figure 4.4: NCC tracking performance for different values of c of Equation (4.4). Increasing c effectively reduces the requirement for valid tracking, resulting in greater errors (vs. manual segmentation; *top*), but fewer frames dropped (*bottom*).

4.4 Motion Characteristics

Figure 4.5 shows a trace of the foreign body positions as determined by NCC, with manually-segmented data serving as a basis for evaluation. The corresponding errors between the traces are shown in Figure 4.6. A correspondence between the foreign body motion and the 1-Hz heartbeats is roughly discernible, but there is also a considerable of amount of motion that is seemingly arbitrary. In a spectral view of the motion, shown in Figure 4.7, spectral power at lower and higher frequencies (corresponding to drift and heartbeat harmonics respectively) suggest a complex behavior that supports this observation. Studies on the mitral valve have also reported on these higher frequency components [12].

The foreign body travels a distance of 40.2 mm in its principle axis of motion. Further inspection of the motion indicates speeds of 343.5 mm/s and accelerations of about 7.8 m/s^2. These figures are consistent with those reported for the heart wall (300 mm/s [116]) and mitral valve (200 mm/s speed and 3.8 m/s^2 acceleration [12]). Table 4.2 summarizes the extracted motion parameters. The fast and seemingly unpredictable nature of the motion highlights the need for carefully planned foreign body retrieval strategies, as these are fairly demanding requirements for existing robots to achieve under surgical conditions.

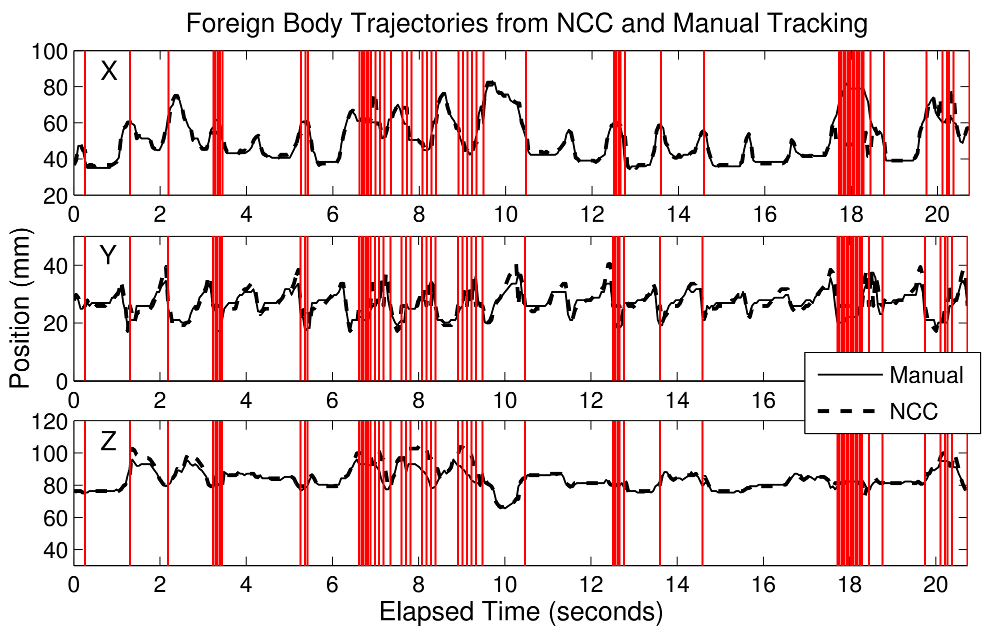


Figure 4.5: Motion traces of a foreign body in the beating heart phantom, tracked using 3D ultrasound. Red vertical bars indicate frames dropped due to low correlation values.

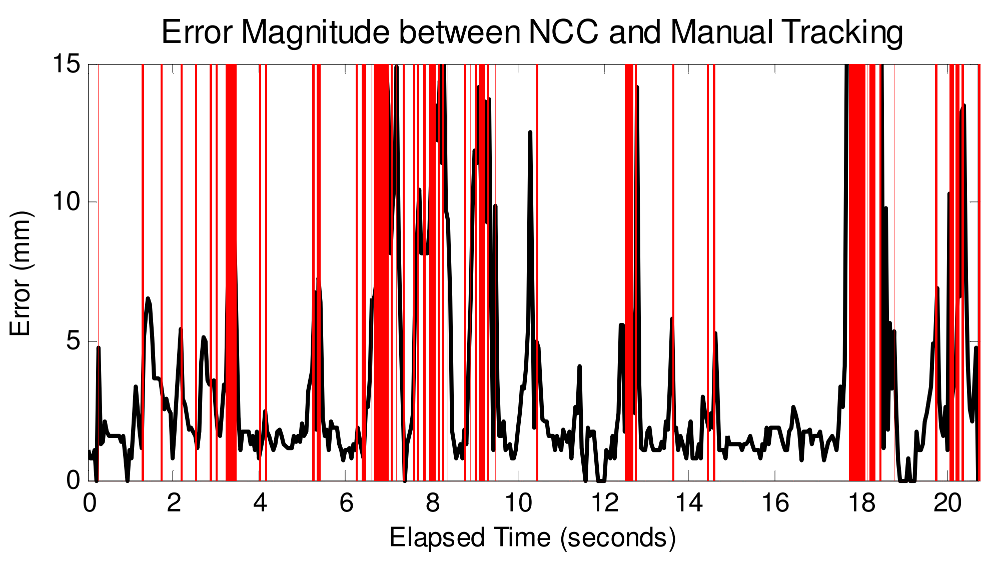


Figure 4.6: Error magnitude between NCC and manual tracking for the foreign body positions shown in Figure 4.5. Red vertical bars indicate frames dropped due to low correlation values, coinciding with large errors.

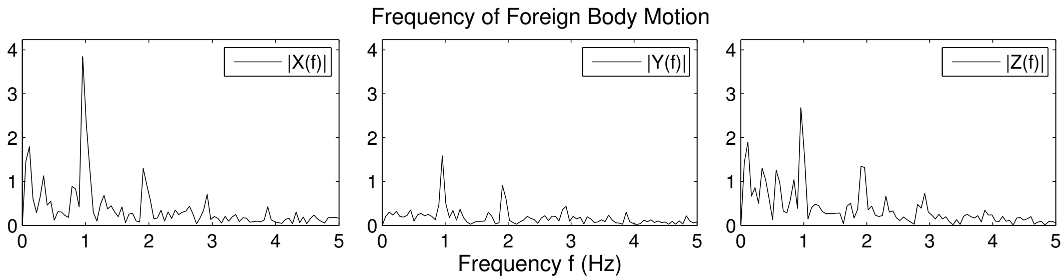


Figure 4.7: Frequency components of foreign body motion.

Table 4.2: Foreign body motion measurements obtained from 3D TEE stream

	Datasets					Aggregate			
Measurement	A	B	C	D	E	Min	Mean	Max	SD
Range	51.4	49.0	47.5	23.5	29.8	23.5	40.2	51.4	12.7
Speed (max)	336.2	334.2	530.0	208.7	308.3	208.7	343.5	530.0	116.5
Acceleration	5.3	9.5	9.1	6.2	8.7	5.3	7.8	9.5	1.9

Notes: Range in mm, speed in mm/s, acceleration in m/s^2

4.5 Chapter Summary

In designing a minimally invasive surgical assistant for removal of free moving foreign bodies from a beating heart, it is necessary to understand the motion and behavior of the target to be captured. It was surmised that the motion is fast with speeds comparable to those of cardiac structures, but no known studies of this phenomenon had been conducted.

This chapter presents a real-time algorithm for tracking a foreign body in a beating heart based on an efficient implementation of normalized cross-correlation. The accuracy of the algorithm was assessed against a ground truth, and a phantom experiment was performed to determine motion parameters; the key results are summarized in Table 4.3.

Table 4.3: Summary of foreign body motion measurements

Tracking Accuracy		Motion Characteristics		
RMS Error	2.3 mm	Range	40.2	mm
		Speed	343.5	mm/s
		Acceleration	7800.0	mm/s^2

The results show that the foreign body motion is fast and abrupt, and its instantaneous position is generally difficult to predict. This provides valuable information regarding the requirements of an interventional system that is capable of catching the particle. Indeed, simpler conditions exist from a technical standpoint: the foreign body may embed in stationary or predictably-moving tissue, or it may leave the heart altogether (though this may lead to complications such as embolization). We explore the challenging cases in the interest of improving interventions and transferring the findings to similar applications, such as mitral valve repair.

The next step involves the design of a robot control scheme for retrieving the foreign body, with the possibility that the speed of the foreign body lies beyond the capabilities of contemporary surgical robots to directly pursue it.

Table 4.4: Technical barriers overcome (Section 1.1.4)

- Establishment of an experimental platform for investigation of the problem
- Tracking of a foreign body in a dynamic environment using 3D ultrasound
- Characterization of motion and behavior of cardiac foreign bodies in a phantom
- Ensuring real-time performance of the tracking computations

Table 4.5: Research contributions (Section 1.2)

• First use of 3D TEE for tracking a moving target
• Real-time tracking in 3D ultrasound
• Characterization of cardiac foreign body motion in a phantom

Chapter 5

Foreign Body Capture Strategies

5.1 Direct Pursuit

To explore the feasibility of 3D ultrasound guided foreign body retrieval, a preliminary set of experiments was devised using the seven-DOF LARS robot (Section 3.4.1) holding a rigid instrument [150]. The robot was commanded to chase a foreign body moving inside a beating heart phantom, based on positions of the particle tracked using streaming ultrasound. The purpose of these experiments includes the following: (1) development of a framework of resources and capabilities that enable careful investigation of the problem; (2) examination of the ability of a robotic system to pursue the moving target, considering the behavior of a foreign body in a beating heart (Chapter 4); and (3) demonstration of 3D transesophageal echocardiography (TEE) in tracking a target for the purpose of autonomous robot guidance.

5.1.1 Experimental Setup

A prototype system for minimally invasive evacuation of foreign bodies from a beating heart using direct pursuit is illustrated in Figure 5.1. Its primary components include an ultrasound system, a LARS robot, a beating heart phantom, and a workstation computer. This is a nearly identical setup to the one described in Section 3. Instead of a snake robot, the LARS robot here holds a more conventional straight tool.

The foreign body motion characteristics, detailed in [145] and reviewed in Chapter 4, indicate that catching such an object is a demanding task that appears to warrant novel strategies and robot designs. Nevertheless, as an early approximation we use an existing surgical robot, the LARS Robot [128] (Section 3.4.1). This 7-DOF robot (three translations, three rotations, and one insertion) was designed for laparoscopic surgery. It can reach speeds of 50 mm/s and accelerate at a rate of 375 mm/s^2 in the x- and y-directions. The z-axis is the limiting direction, with a maximum speed of 20 mm/s and acceleration of 90 mm/s^2. Implementations of the robot forward/inverse kinematics and operation modes are based on the *cisst* libraries [127].

A 3.0-mm diameter rod held by the robot serves as a surrogate for a surgical tool, while a 3.2-mm diameter steel ball is attached to the rod tip to enhance tracking, thus decoupling the tracking problem from the visual servoing task; in future work we will address increasingly realistic scenarios, where the robot is equipped with a functional end effector.

Figure 5.1: Experimental setup for 3D TEE-guided robotic foreign body retrieval by direct pursuit. System is similar to that of Section 3, except the LARS robot holds a straight rigid tool rather than a snake robot.

5.1.2 Experimental Procedure

The tracked foreign body positions ${}^{u}p_{fb}$, obtained from characterization experiments, are used to guide the robot, effectively in pursuit of a virtual foreign body moving in the beating heart phantom. First, with the robot (tool tip) in the field of view of the ultrasound probe, the transformation ${}^{r}T_{u}$ between the robot and ultrasound coordinate systems (designated r and u respectively) is determined by moving the robot to nine known positions and recording an image for each. The tip positions ${}^{u}p_{reg}$ in ultrasound image coordinates are extracted manually, and a transformation matrix is computed using standard approaches. The average fiducial registration error (FRE) over the five datasets is 1.0 mm.

The transformation ${}^{r}T_{u}$ is then applied to the tracked foreign body positions ${}^{u}p_{fb}$ to place them into the robot frame. The robot is commanded via position control per ${}^{r}p_{fb}$, the foreign body positions in terms of robot coordinates. As the robot moves, ultrasound images of the tip are acquired at a rate of 20 frames per second for retrospective analysis. The steps described heretofore are summarized as follows:

$$\begin{aligned} &Reg({}^{u}p, {}^{r}p) \\ &{}^{r}p_{fb} = {}^{r}T_{u} \cdot {}^{u}p_{fb} \end{aligned} \tag{5.1}$$

where

$$Reg \Leftrightarrow {}^{r}T_{u} : {}^{u}p \rightarrow {}^{r}p \tag{5.2}$$

is a paired-point registration.

Tracking of the robot tip in the ultrasound images is performed in an identical fashion to the tracking of the foreign body. The target (the tip in this case) is selected interactively in the first frame and tracked automatically in subsequent frames using normalized cross-correlation (Section 4.1). The tracked tip positions in ultrasound space are transformed into robot space:

$$^{r}p_{tip} = {}^{r}T_{u} \cdot {}^{u}p_{tip} \tag{5.3}$$

We compare the foreign body positions $^{u}p_{fb}$ with those of the tracked tip positions of the pursuing robot ($^{u}p_{tip}$) in order to understand the feasibility of the proposed system—a close match would suggest that the robot can follow and thus capture the foreign body successfully.

5.1.3 Results

While the foreign body can attain speeds of 343.5 mm/s, the robot prototype is only capable of 20 mm/s along its limiting axis. There is a mismatch in acceleration as well: 7800 mm/s^2 for the foreign body versus 90 mm/s^2 for the robot. The robot was thus effectively sped up by reducing the speed of the virtual target to account for the speed mismatch. This simulated experiment has the effect of decoupling the tracking and guidance tasks, thereby exposing the robotic performance in the context of the application. Alternatively, a simulated robot at full speed could be used to gauge performance requirements from a dynamics viewpoint, but the former approach was chosen for this study because it enables us to (1) perform 3D ultrasound-based guidance on a real robot, which can potentially reveal practical insights; (2) test the hypothesis that chasing the foreign body exceeds the ability of surgical robots; and (3) track the robot under 3D ultrasound. Two reduction factors are applied to highlight the influence of velocity.

A sample foreign body trajectory and its robot counterpart are shown together below. The motion shown in Figure 5.2 results from reducing the foreign body speed by a factor of 9.0, roughly related to the maximum speed of the robot. A reduction in speed results in a reduction in acceleration by the same factor. The RMS error over all datasets is 2.1 mm. Reduction of the speed by a factor of 20.0 (Figure 5.3) yields a reduced error of 1.6 mm over all datasets. The foreign body in this case is virtually motionless, analogous to a near-continuum of static targets along a 3D path. Excerpts of these plots are shown together for comparison in Figure 5.4. Table 5.1 lists results from all five datasets.

5.1.4 Discussion

The 1.6 mm error in 3D ultrasound-based dynamic target tracking suggests the viability of the proposed system for static or slow targets, in light of previous work reporting accuracies of 1–1.2 mm [12, 102, 107, 109]. Direct pursuit of faster targets

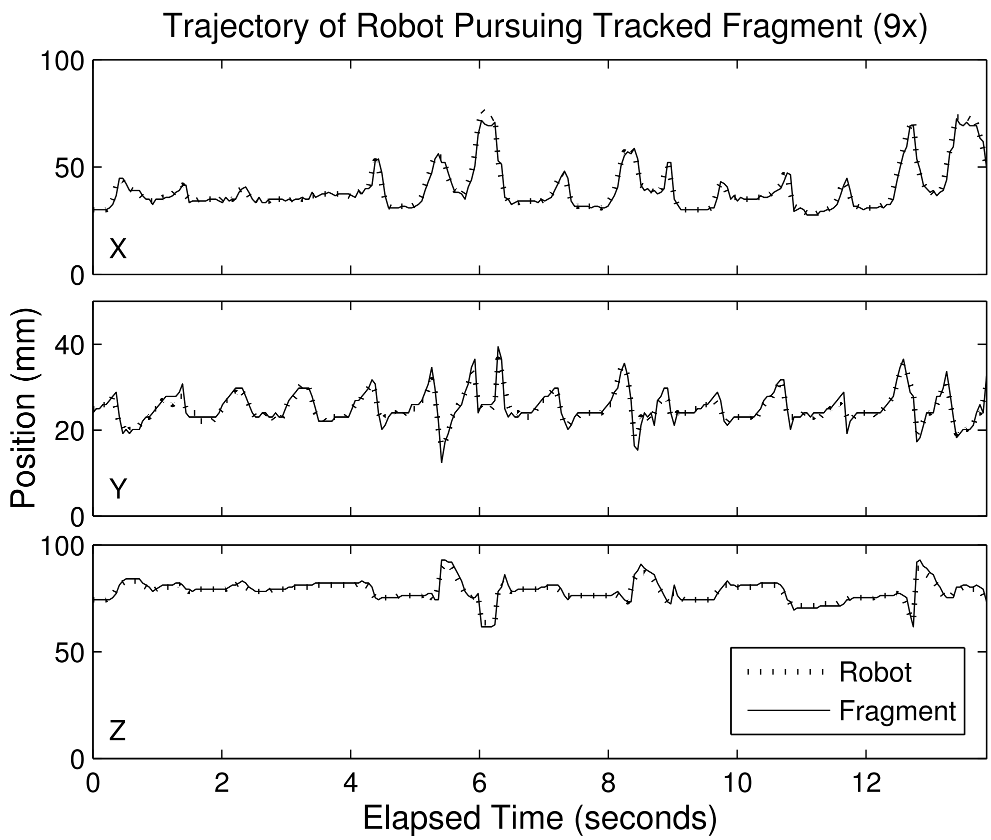


Figure 5.2: Motion traces of a foreign body, and that of a robot in pursuit, with virtual target speed reduced by a factor of 9.0.

Table 5.1: Robot RMS Position Error from Foreign Body Path

		Speed Reduction	
Units in mm		9x	20x
Datasets	A	2.2	1.6
	B	2.5	1.9
	C	2.2	1.9
	D	1.8	1.2
	E	1.8	1.6
Aggregate	Min	1.8	1.2
	Mean	2.1	1.6
	Max	2.5	1.9
	SD	0.3	0.3

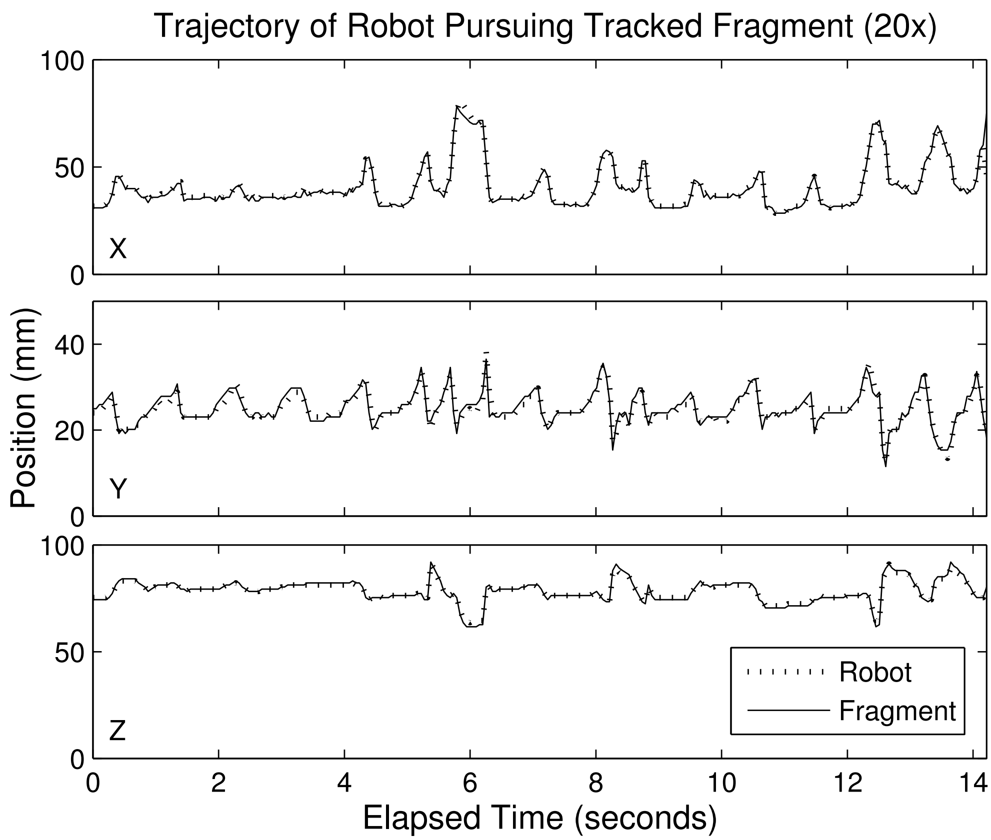


Figure 5.3: Scenario of Figure 5.2, but the virtual target speed is reduced by a factor of 20.0.

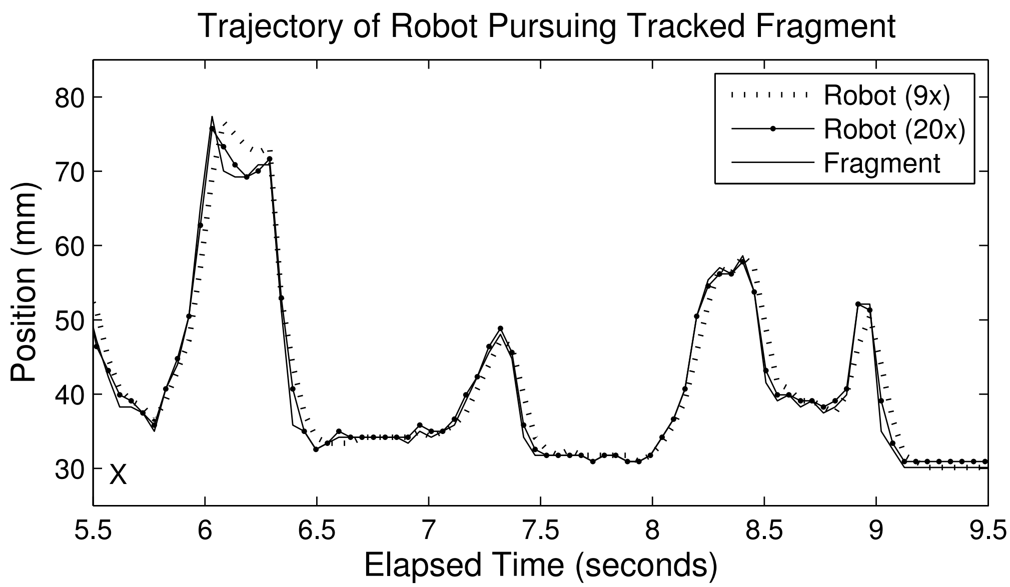


Figure 5.4: Superimposed view of the foreign body motion with those of the pursuing robot, when the speed of the target is reduced by factors of 9 and 20.

may be feasible with a faster robot, but as explained in Section 3.1.2, a fast robot can be unsafe inside the heart. We further decided to use a slow physical robot rather than a fast simulated one in order to incorporate the image feedback and processing delays of the closed loop system and their effects on tracking performance.

The error increase from 1.6 to 2.1 mm in the fastest case sheds additional light on the inherent accuracy of the system. Comparison of these two accuracy figures suggests that the main contributor of errors in the dynamic task is the inability of the robot to change directions as abruptly as the foreign body does. Robot speeds of 17 mm/s (20x less than 343.5 mm/s) have been observed in the dynamic test cases, while [107] and [109] use robot speeds of 3 and 2 mm/s respectively, and the robot of [102] is jogged manually. The high-speed 1-DOF device (290 mm/s) of [12] achieves a 1.0 mm accuracy aided by a predictive model, which is appropriate for heart valve motion but less suitable for a foreign body, as shown in [145].

In contrast to the previous work in 3D ultrasound-based robot guidance reporting relatively low acquisition and processing frequency (e.g., 2 Hz [107] or offline servoing [102]), the 20 Hz frame rate achieved with our setup was sufficient to track a heart wall [146] and a foreign body, and to steer a robot. The latter represents the first known use of 3D TEE for robot steering. Despite the limited field of view, particularly in comparison to a transthoracic probe, it appears feasible to use the 3D TEE probe for this application, as motion of the foreign body is typically confined to a single ventricle.

Insights gained from this study motivate the need for novel robot designs, in particular one with a flexible end effector with distal dexterity to navigate the heart chamber. The experiments show that millimeter-scale accuracy may be difficult to achieve at the speeds required to follow a foreign body, so careful retrieval planning will be preferred over direct pursuit.

5.1.5 Further Lessons in Direct Pursuit

We conducted a detailed study of image-guided direct pursuit using a robotic endoscope to chase targets with various motion profiles [151]. The motivating application for the effort was robotic steering of an endoscope to conveniently maintain visualization of surgical targets and instruments in minimally invasive coronary artery bypass graft (CABG) surgery. One of the challenges of CABG surgery is the unintuitive mapping between the view, endoscope, surgeon, and endoscopy assistant; an image-guided robot has the potential to enable solo surgery.

In the work, also detailed in Appendix B, a set of virtual phantom experiments was performed to assess the performance of the robot motion. A stationary red laser was projected onto a blue surface such that when the robotic endoscope was moved, it would appear in the video as though the laser had moved with respect to a stationary camera. The robot could thus "guide" the laser in pursuit of virtually defined targets; the simple imaging setup helped direct the focus to the robot performance. Different motion profiles were tested, including straight and curved paths, constant and varying speeds, as well as pulsed speeds with varying pulse widths. Also tested was the initial distance between the robot and the target.

The experiments indicated that a change in target speed was more difficult for the robot to chase than a change in target direction. This follows from the main observation that the robot tends to lag behind the target. Based on this lesson, the robot control algorithm was improved in the endgame segment to enable catching of the target. The revised method was then tested in a more realistic experiment on an endoscopic beating heart phantom. The experiments showed the ability of the robot to reach all clinically relevant targets within 1.2 s and 7 pixels of accuracy at a safe and conservative robot speed.

This work highlighted some of the difficulties of direct pursuit of a moving target from the perspective of minimally invasive CABG surgery. In particular, prior to the study we expected that the difficulties of chasing irregularly moving targets would relate to determining the instantaneous position and velocity of the target. It was through the experiments, on the other hand, that we realized the robot always lags behind the target due to the image acquisition and processing latencies. Additionally, a conventional PD control scheme would generally not allow the gap to be closed because as the robot converges on the target, its velocity is correspondingly reduced, resulting in a steady-state error. To overcome this phenomenon, either the robot must be capable of exceedingly fast speeds, or an alternate control scheme must be devised. The endoscopy robot work followed the latter approach via online optimization of gain parameters. Subsequent sections detail the approach used to capture an erratically moving cardiac foreign body using a dexterous end effector under 3D ultrasound guidance.

5.2 Ambushing Capture Locations

5.2.1 Criteria

A retrieval strategy based on chasing the foreign body using a robotic end effector would be extremely challenging given the high speeds and accelerations. Furthermore, a dexterous end effector is necessary for adequate reachability inside the heart, as explained in Section 3.1.2, and it is difficult to control a small dexterous robot at high speed. As well, the foreign body tends to exhibit a preference towards certain subsections of, and paths within, the overall ventricle volume. These factors motivate a more relaxed retrieval approach wherein the end effector is aimed at a location where likelihood of successful intercept is greatest. The criteria for computing such *capture locations* [145, 152] are outlined below.

Spatial probability. Which locations are expected to contain the foreign body the most? This metric may be applicable to a slower end effector with a capture mechanism that activates relatively quickly, such as a gripper.

Dwell time. How long does the foreign body dwell in a certain location before it leaves (t_{dwell})? When it leaves, how long does the system have to wait before the foreign body reenters said location (t_{wait} or t_{empty})? These metrics may be useful for capture mechanisms that are slower to activate, such as suction.

Visit frequency. How often does the foreign body visit, or traverse, a certain location? This metric may be suitable for a mechanism such as a net that best captures a foreign body in transit.

The ensuing sections describe the algorithms used to compute capture locations along with the results of applying these algorithms to the foreign body tracking data of Chapter 4, which also includes details on the tracking method and experimental setup for studies on a beating heart phantom. Performance limitations due to the speed of the robot and the sensitive environment that it operates in gave rise to an indirect capture strategy, in contrast to a more natural strategy in chasing. From a design perspective, it is dutiful to use the findings of capture location experiments to iterate the design of the robot. Indeed, it was in part through this route that the robot operation modes (Section 3.4.3.3.2) were refined. The beating heart phantom setup has proven to be a valuable resource in developing various capture location computation techniques.

5.2.1.1 Spatial Probability

The probability of containing the foreign body at each voxel in the ventricle volume is computed as the histogram of foreign body positions accumulated over the tracking duration. The pseudocode in Algorithm 5.1 demonstrates how this computation is performed. In essence, the spatial probability measures the fraction of time that the foreign body is present at a particular voxel, in relation to the total time elapsed.

Algorithm 5.1 Spatial probability computation.

$P \leftarrow \mathbf{0}$ ▷ same size as US volume
$k \leftarrow 0$
while tracking **do**
  $x_{fb} \leftarrow$ current foreign body voxel position ▷ queried from tracker
  for all $p_x \in P$ **do**
    $d \leftarrow |x - x_{fb}|$
    if $d \leq R$ **then** ▷ foreign body radius
      $p_x \leftarrow p_x + 1$
    end if
  end for
  $k \leftarrow k + 1$
  $P_k \leftarrow P / \sum |p_x|$ ▷ available for query
end while

The foreign body is approximated as a sphere with radius $R > 1$ voxel, so multiple histogram bins (corresponding to voxels) are incremented at each time step k. Dividing the histogram by the elapsed time could then yield probabilities greater than one, so the sum of the counts, $\sum |p_x|$, is used as the divisor instead. The capture location is the voxel of maximum spatial probability.

Algorithm 5.1 treats the foreign body as a uniform object. If weighing is desired to emphasize, for example, the center of the object over its edges, the increment amount on Line 8 can be modified to be some function $0 \leq f(d) \leq 1$ that decreases with distance d from the centroid of the foreign body. Depending on the form of $f(d)$, the condition on Line 7 that the voxel be within range of the foreign body may be superfluous. Given the noise and uncertainties in ultrasound imaging, however, this approach was forwent in favor of simplicity of the implementation.

Clearly, this manner of computing spatial probability is inefficient because not all voxels need to be examined at each time step. Only those voxels within the range of the foreign body have their respective counts updated, so a substantial increase in performance can be achieved by limiting the computation to this subset of the volume. The requirement for real-time performance leads to the following reformulation.

While the speedup attained by the latter version over the former is significant, the number of voxels processed at each time step may remain substantial due to the $O(n^3)$ complexity in the size of the foreign body. Porting of software to a higher performance programming language such as C++ represents one possibility for alleviating this

Algorithm 5.2 Spatial probability computation, second version.

1: $P \leftarrow \mathbf{0}$
2: $k \leftarrow 0$
3: **while** tracking **do**
4: $\quad x_{fb} \leftarrow$ current foreign body voxel position
5: $\quad S \leftarrow \{x : |x - x_{fb}| \leq R\}$
6: $\quad$ **for all** $x \in S$ **do**
7: $\quad\quad p_x \leftarrow p_x + 1$
8: $\quad$ **end for**
9: $\quad k \leftarrow k + 1$
10: $\quad P_k \leftarrow P / \sum |p_x|$
11: **end while**

concern. Noting that Matlab is optimized for matrix operations, however, we were able to obtain a real-time response with a Matlab implementation by recasting the problem accordingly. As a result, Line 7 of Algorithm 5.2 can be performed in a single step without the expensive `for` loop overhead of Line 6, as illustrated below, with variable names following those of Algorithm 5.2. Note that in Matlab, matrix elements can be accessed using a scalar index, and multiple elements can be accessed in one statement using a vector of indexes.

```
% Matlab code excerpts for accumulating spatial probability

% first version: with loop construct
for i=1:numel(S)
    P(S(i)) = P(S(i)) + 1;
end

% second version: without loop construct
P(S) = P(S) + 1;
```

A qualitative view of the foreign body motion is shown in Figure 5.5. Red regions indicate where the foreign body is more likely to be located based on the spatial probability metric. The second row of Table 5.2 shows that on average, the foreign body spends up to 50.5% of the time in the most spatially probable voxel. Although it is shown in the frequency domain (Figure 4.7) that the motion of the foreign body is more irregular than the motion of the heart wall, predictive models using the spatial position of the foreign body can be used to move the robot to a position of high spatial probability.

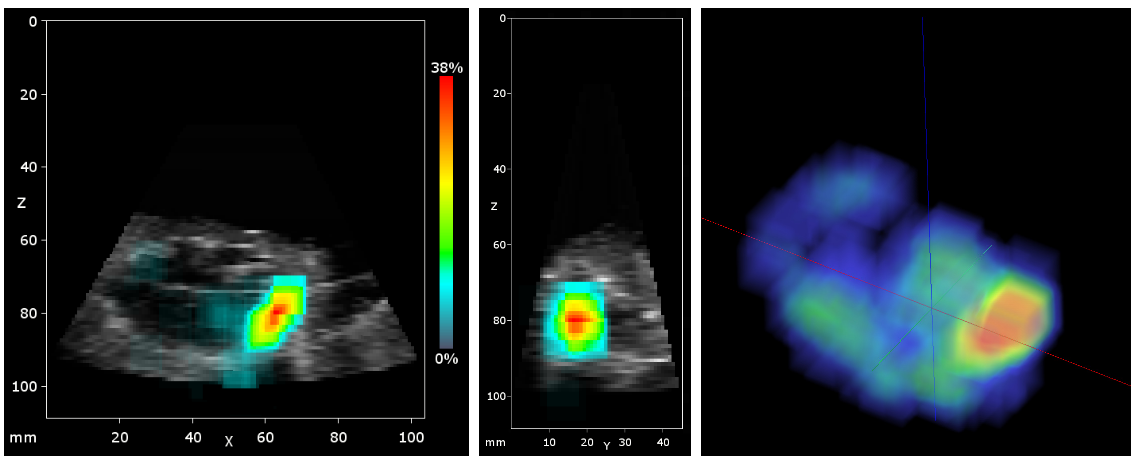


Figure 5.5: Spatial probability map of the foreign body position. (*Left and center*) Coronal and sagittal slices of the map. (*Right*) A 3D rendering obtained from the ultrasound system.

These measurements support the notion that it is feasible to capture the foreign body by a calculated ambush, even though the motion can at times be quick, abrupt, and irregular. Figure 5.5 illustrates a distinct locus of the foreign body position that appears to be exploitable. The first row of Table 5.2 lists the percentage of the image volume in which the foreign body is present at least 25% of the time. As the data show, this is a minute proportion—only 0.14% of the entire volume is expected to host the particle to this extent, implying that there exists a salient region where capture by ambush has a meaningful chance of success. Were the foreign body to transit the heart uniformly, the cumulative histogram curves of Figure 5.6 would be linear; that is, as the spatial probability of the capture location increases, the size of the subvolume(s) with at most that spatial probability would grow linearly. Instead, the curves show a sharp increase in the size of the subvolume(s) with spatial probability. For example, voxels that never see the foreign body (x-axis at 0%) applies to 94% (y-axis) of the volume, suggesting that foreign body distribution is not uniform, which in turn implies that capture need not be attempted in this region. As indicated on the plot, a mere 5% spatial probability level applies to almost the entire volume—99% of it, so a capture strategy should focus on the remaining 1% of the volume. Finally, the plots plateau beyond 30% spatial probability; as mentioned, if the distribution were uniform, such a phenomenon would not be observed.

Table 5.2: Foreign body spatial tendencies. (*Top row*) Percentage of volume with $\geq$ 25% spatial probability. (*Second row*) Maximum spatial probability found.

	Datasets					Aggregate			
Measurement	A	B	C	D	E	Min	Mean	Max	SD
25% Probability	0.07	0.03	0.17	0.19	0.24	0.03	0.14	0.24	0.09
Max. Probability	39.8	38.0	57.5	54.0	63.0	38.0	50.5	63.0	11.1

5.2.1.2 Dwell Time

During the development of capture strategies, it was surmised that the efficacy of a capture location is dependent on the device used to execute the endgame, a topic

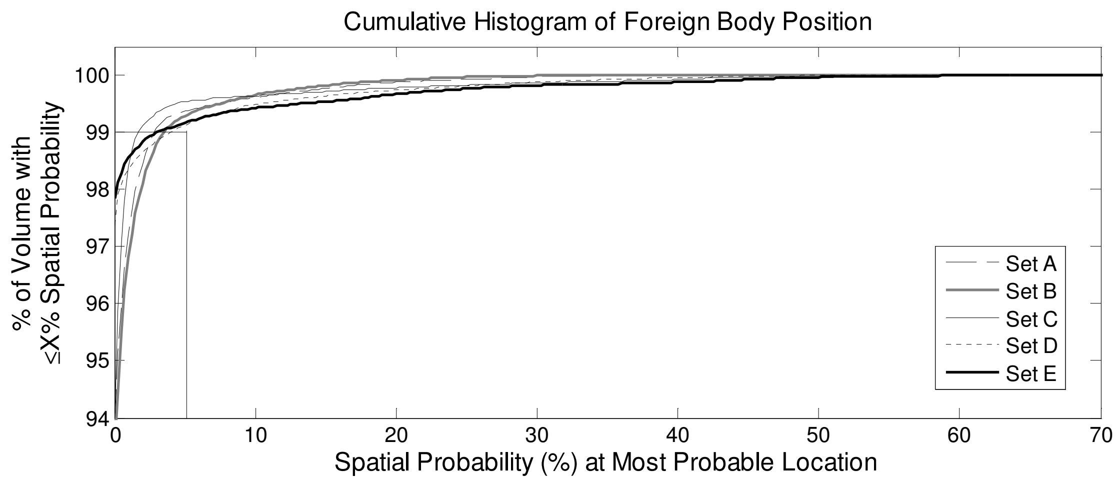


Figure 5.6: Cumulative histograms of foreign body spatial probability. Around Y=94% of the volume is never touched (X=0%), and roughly Y=99% of the volume is occupied at most X=5% of the time.

for future investigation in greater depth. Nevertheless, to accommodate a variety of potential capture mechanisms, we proposed additional criteria for computing capture locations. This section describes a method based on *dwell time*, measuring the duration that the foreign body resides in a particular location before leaving it. A complementary figure is the *empty time*, or the duration that a location remains absent of the foreign body between appearances; we refer to this duration synonymously as the *wait time* because it is equivalent to the amount of time that the system would have to wait for the particle to return to a chosen location.

A scenario in which dwell time may produce a more meaningful capture location than does spatial probability is one in which there is a slight latency for the mechanism to take effect, such as suction. In this case, dexterous targeting of the end effector remains important as it can help reduce perioperative time, improve capture certainty, and minimize blood loss over undirected suction (lost blood can be restored through intraoperative autotransfusion or perioperative blood salvage, but minimizing the dependence on such facilities is preferred due to contraindications such as sensitivity to the anticoagulants used, or risks such as infusion of external contaminants [153]).

The steps for computing dwell and empty (wait) times are listed in Algorithm 5.3. At each time step k, the voxel positions of the ultrasound image are checked against that of the foreign body. An *enter* event occurs in a voxel when the particle appears following an absence. The entry time is recorded along with the duration of the preceding absence (i.e., the empty or wait time). Similarly, a *leave* event occurs when the particle leaves (vacates) the voxel, triggering the recording of the leave time and the dwell time of the preceding presence. No records are kept for the two possible non-events, namely the continuances of the absent and present states.

Similar to the situation with spatial probability, there is the potential to speed up the computation of dwell time by examining only the eventful subset of voxels. As suggested above, this subset includes the set of voxels entered or vacated by the foreign body since the previous time step. The computational complexity remains $O(2n^3)$ in the size of the foreign body, but a speedup is realized because the size of the image is much greater than the size of the foreign body—on the order of 100^3 voxels to 10^3 (this discrepancy applies to spatial probability as well).

To capitalize on the above observation, we define the quantities shown in Table

Algorithm 5.3 Dwell and empty (wait) time computation.

```
D, E ← 0                                          ▷ same sizes as US volume
t_{enter}, t_{leave} ← 0
t_{dwell}, t_{empty} ← 0
n_{dwell}, n_{empty} ← 0
state ← absent
k ← 0
while tracking do
    x_{fb} ← current foreign body voxel position
    for all state_x ∈ state do
        d ← |x − x_{fb}|
        if d ≤ R then                             ▷ foreign body present in voxel x
            if state_x = absent then              ▷ absent → present ⇒ enter event
                state_x ← present
                t_{enter,x} ← k
                t_{empty,x} ← t_{empty,x} + k − t_{leave,x}
                n_{empty,x} ← n_{empty,x} + 1
            end if
        else                                      ▷ foreign body absent from voxel x
            if state_x = present then             ▷ present → absent ⇒ leave event
                state_x ← absent
                t_{leave,x} ← k
                t_{dwell,x} ← t_{dwell,x} + k − t_{enter,x}
                n_{dwell,x} ← n_{dwell,x} + 1
            end if
        end if
    end for
    k ← k + 1
    D_k ← t_{dwell}/n_{dwell}                     ▷ dwell time for query
    E_k ← t_{empty}/n_{empty}                     ▷ empty time for query
end while
```

5.3 and illustrated in Figure 5.7.

Table 5.3: Definitions for second version of dwell time computation.

Symbol	Definition
k	Discrete time variable associated with ultrasound image frames
B	Set of voxels occupied by FB (at time k unless otherwise noted)
V	Set of voxels vacated by FB between time $k-1$ and k $V = B_{k-1} \setminus B_k$
E	Set of voxels entered by FB between time $k-1$ and k $E = B_k \setminus B_{k-1}$
N	Set of voxels neither vacated nor entered between time $k-1$ and k, i.e., voxels that remain vacant or occupied in both frames $N = \overline{(V \cup E)} = \overline{(B_{k-1} \cup B_k)} \cup (B_{k-1} \cap B_k)$

Only voxels in the set $V \cup E$ need to be updated at each time step. Specifically,

$$\begin{aligned} t_{enter,e} &= k \, \forall e \in E \\ t_{dwell,v} &= k - t_{enter,v} \, \forall v \in V \end{aligned} \tag{5.4}$$

is computed to obtain dwell times, and

$$\begin{aligned} t_{leave,v} &= k \, \forall v \in V \\ t_{empty,e} &= k - t_{leave,e} \, \forall e \in E \end{aligned} \tag{5.5}$$

for empty times. As before, these times are divided by the number of like events to produce an average figure. The real-time performance obtained under this approach enabled the use of a prototyping programming environment; pseudocode for a Matlab implementation is listed in Algorithm 5.4. Variables are set up identically to Algorithm 5.3. In this case we found that the use of a loop construct to iterate through

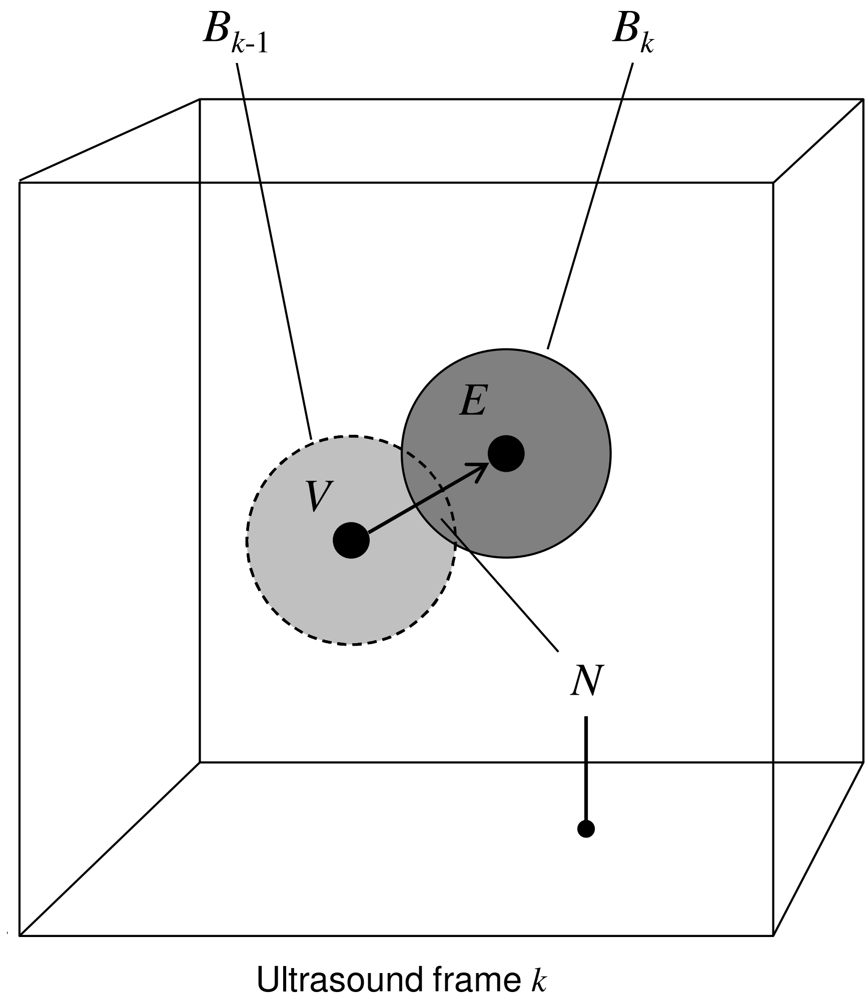


Figure 5.7: Depiction of the pertinent quantities for computing dwell time (second version), as defined in Table 5.3. Only those voxels that have been vacated (V) or entered (E) between frames need to be updated; all others (N) remain unchanged.

the eventful voxels yielded faster performance than vectorizing the indexes. Further speedup is possible with GPU and parallel programming because the record keeping process is highly parallelizable by voxel. The foreign body tracking subsystem (Section 4.1) stands to benefit the most from advanced computational resources because, for example, multiple (transformed) templates could be tracked simultaneously over a larger search space, thereby improving tracking robustness.

Results of dwell time measurements on the foreign body data of Chapter 4 are displayed in Table 5.4. In the longest dwelled location, the expected dwell time is about 0.84 seconds. One scenario for this value is that after the foreign body comes to rest during the diastolic phase, it remains so for less than one second before being propelled during the systolic phase. If the longest dwelled location is selected as the capture location, the robot may have to wait about 2.28 seconds for the foreign body to emerge. An interesting future pursuit is the use of minimum empty time in the decision making process.

Table 5.4: Dwell and empty times in the longest dwelled location.

	Datasets					Aggregate			
Measurement	A	B	C	D	E	Min	Mean	Max	SD
t_{dwell} (s)	0.33	0.54	2.00	0.49	0.85	0.33	0.84	2.00	0.83
t_{empty} (s)	3.58	1.75	1.59	3.40	1.07	1.07	2.28	3.58	3.55

Algorithm 5.4 Dwell and empty (wait) time computation, second version.

$D, E \leftarrow \mathbf{0}$
$t_{enter}, t_{leave} \leftarrow \mathbf{0}$
$t_{dwell}, t_{empty} \leftarrow \mathbf{0}$
$n_{dwell}, n_{empty} \leftarrow \mathbf{0}$
$state \leftarrow$ **absent**
$k \leftarrow 0$
while tracking **do**
    $x_k \leftarrow$ current foreign body voxel position
    $B_{k-1} = sphere(R, x_{k-1})$ ▷ sphere of radius R, centered at x_{k-1}
    $B_k = sphere(R, x_k)$
    $E = B_k \setminus B_{k-1}$
    $V = B_{k-1} \setminus B_k$
    for all $e \in E$ **do** ▷ entered voxels
        if $state_e$ = absent **then**
            $state_e \leftarrow$ present
            $t_{enter,e} \leftarrow k$
            $t_{empty,e} \leftarrow t_{empty,e} + k - t_{leave,e}$
            $n_{empty,e} \leftarrow n_{empty,e} + 1$
        end if
    end for
    for all $v \in V$ **do** ▷ vacated voxels
        if $state_v$ = present **then**
            $state_v \leftarrow$ absent
            $t_{leave,v} \leftarrow k$
            $t_{dwell,v} \leftarrow t_{dwell,v} + k - t_{enter,v}$
            $n_{dwell,v} \leftarrow n_{dwell,v} + 1$
        end if
    end for
    $k \leftarrow k + 1$
    $D_k \leftarrow t_{dwell}/n_{dwell}$
    $E_k \leftarrow t_{empty}/n_{empty}$
end while

5.2.1.3 Visit Frequency

Some devices such as nets or other passive mechanisms may require that the foreign body be in motion in order to effect a capture. A viable strategy in this case would be to place the end effector at the location that the particle transits most often. Computation of this *visit frequency* metric can be likened to a simplified dwell time procedure that considers only the number of enter events without regard to duration. This is shown in Algorithm 5.5, with some variables renamed to align with the context. Computational considerations are similar to those described previously.

In order to properly place the capture device (e.g., a net), the direction of travel must be taken into consideration. One possible approach is to record the previous and subsequent positions of the foreign body through the most visited location in order to deduce the direction of travel. If multiple directions are found, a workaround is to deploy the device when the foreign body is in the most dwelled location, with the net facing said location, and allow the foreign body to arrive.

Table 5.5 reports the results of visit frequency measurements on foreign body data. The foreign body traverses the most frequently visited location about 1.54 times per second. This figure is greater than the nominal 1-Hz heart rate; one possible sequence for this phenomenon is that the particle travels through a point, hits the heart wall soon thereafter, and bounces back through the same point, suggesting possibly that such a location emerges due to systole, and is near a surface that is normal to the direction of travel of the particle. The tradeoff in using a simpler passive capture

Algorithm 5.5 Visit frequency computation.

$V, n_{visit} \leftarrow \mathbf{0}$ ▷ same sizes as US volume
$state \leftarrow$ **absent**
$k \leftarrow 0$
while tracking **do**
    $x_{fb} \leftarrow$ current foreign body voxel position
    for all $state_x \in state$ **do**
        $d \leftarrow |x - x_{fb}|$
        if $d \leq R$ **then** ▷ foreign body *present* in voxel x
            if $state_x =$ absent **then** ▷ absent → present ⇒ *visit* event
                $state_x \leftarrow$ present
                $n_{visit,x} \leftarrow n_{visit,x} + 1$
            end if
        end if
    end for
    $k \leftarrow k + 1$
    $V_k \leftarrow n_{visit}/(\text{elapsed time})$ ▷ available for query
end while

device is that the end effector must be positioned and oriented very accurately, in a place where the motion may be most erratic.

Table 5.5: Visit frequency in the most frequently visited location.

	Datasets					Aggregate			
Measurement	A	B	C	D	E	Min	Mean	Max	SD
Visits/s	1.25	0.95	1.70	2.00	1.80	0.95	1.54	2.00	0.43

5.2.2 Distances between Capture Locations

Having devised multiple ways of computing a capture location, a question that arises is whether these locations are in reality distinct from one another. That is, although the computations are guided by different criteria, they may ultimately lead to the same result. An example in which this is true is when the foreign body is stuck in a stationary position. In this case, capture locations based on spatial probability, dwell time, and visit frequency should yield the same result, as only one part of the heart volume would contain the foreign body, while the rest would remain uneventful. Furthermore, there should not be a more suitable place for a capture attempt. The existence of multiple capture locations can provide alternative avenues for capture based on situations that arise preoperatively and intraoperatively.

From a conceptual standpoint, the capture locations are generally expected to be different from one another. Whereas spatial probability measures the total fraction of the time that the particle is present at a particular location, the dwell time measures the duration that the particle is expected to remain there. A high spatial probability is possible with a low dwell time if the object moves quickly but returns to the same place often. The most frequently visited location is also at odds with dwell time, because by our definition, the former measure is more pronounced when the foreign body moves at a high rate of speed and returns often. Extending this assessment, capture locations based on spatial probability and visit frequency may be more similar to each other than to those based on dwell time.

The three types of capture locations were determined for each of the foreign body datasets, and the distances between the locations were found. These distances, detailed in Table 5.6 and summarized in Figure 5.8, demonstrate that the capture locations are indeed distinct, though such distinctions are variable. The data appear to confirm our projection that spatial probability and visit frequency are more similar to each other than to dwell time. In considering the similarities and differences between the primary capture locations, i.e., those examined in this section, the possibility is evoked of making capture decisions based on a combination of these criteria. For example, rather than acting on the greatest dwell time, it may be preferable to use a location with both a long enough dwell time and a high enough spatial probability to improve the chance of success. This is an interesting topic of future work.

Table 5.6: Distances between capture locations.

	Datasets					Aggregate			
Measurement	A	B	C	D	E	Min	Mean	Max	SD
$\lvert CL_A - CL_B \rvert$ (mm)	10.77	5.96	5.47	6.74	6.55	5.47	7.10	10.77	2.11
$\lvert CL_B - CL_C \rvert$ (mm)	12.87	4.35	4.38	7.43	4.65	4.35	6.74	12.87	3.66
$\lvert CL_C - CL_A \rvert$ (mm)	4.07	2.70	2.12	1.26	2.13	1.26	2.46	4.07	1.04

$CL_{A,B,C}$ - Capture location based on spatial probability, dwell time, or visit frequency.

5.2.3 Capture Range

In addition to influencing the choice of capture location, the type of device ultimately deployed will determine the degree of targeting accuracy needed to capture a moving body. Some mechanisms may have larger functional ranges than others, as illustrated in Figure 5.9. Equivalently, some mechanisms may tolerate larger positioning errors than others. In this section, we examine the effect that the inherent *capture range* of a device may have on the measurement of a capture location. As an estimate, we model the operative volume of the end effector as a binary sphere centered at the tip, wherein the foreign body is successfully captured if it falls within it.

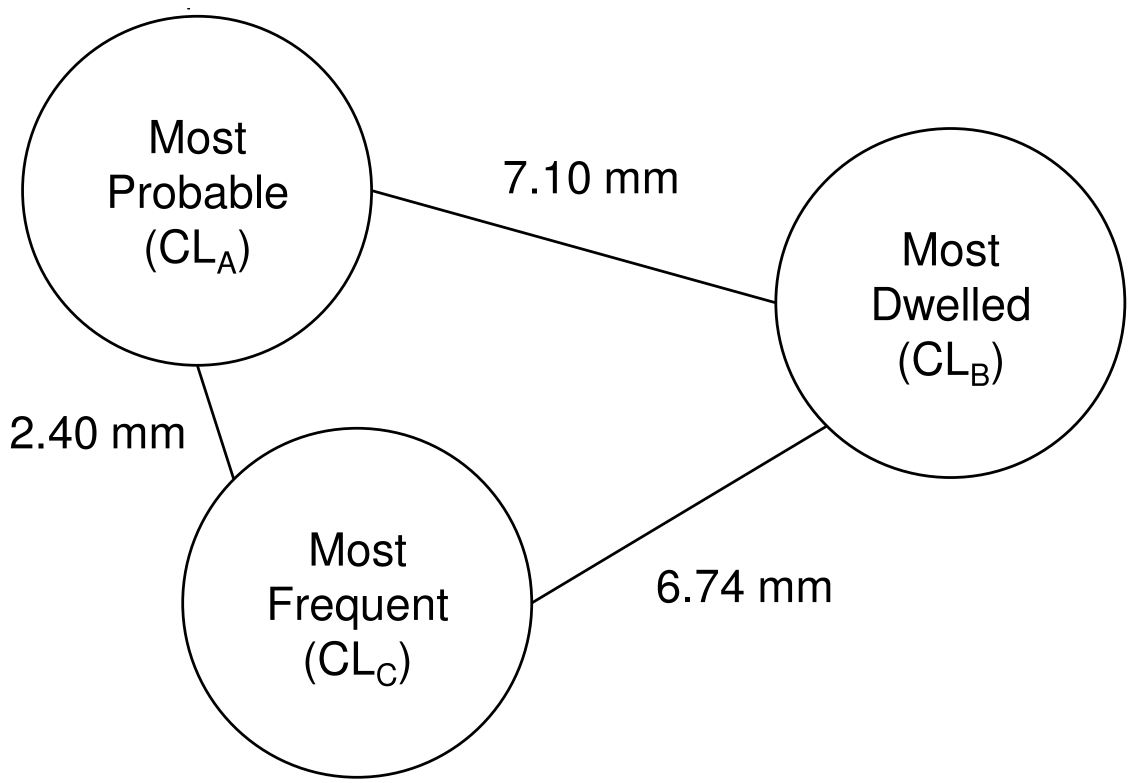


Figure 5.8: Diagram of Table 5.6, illustrating that capture locations are distinct. Multiple capture locations can be computed simultaneously so that an interventional system can make selections based on preoperative and intraoperative criteria.

We are interested in being able to predict the performance of our methods for devices with different capture ranges (5 mm is an initial estimate of a nominal range).

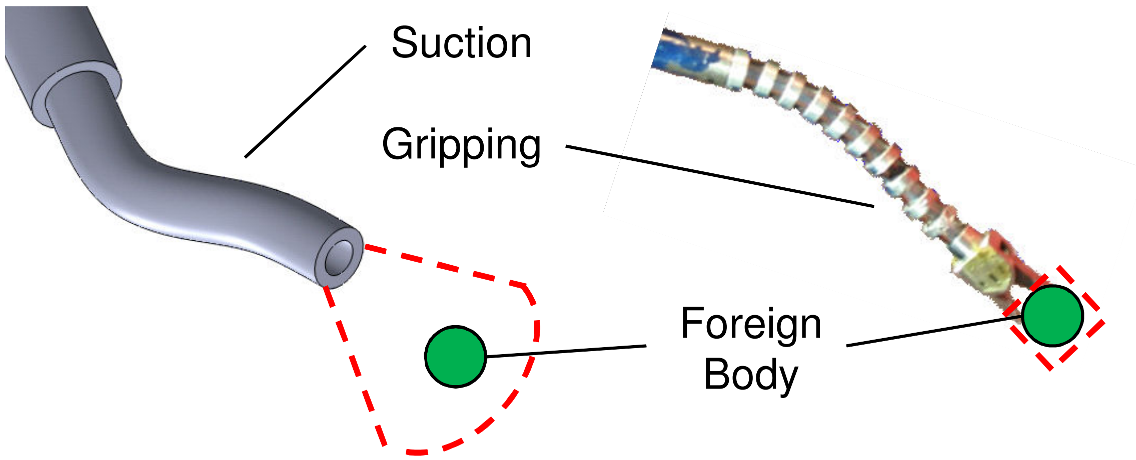


Figure 5.9: Two potential capture scenarios: suction (*left*) and gripping (*right*), with associated capture ranges indicated by dashed lines. Suction may tolerate positioning error, but a gripper may activate quickly.

Figure 5.10 (*left*) shows how the spatial probability at the most probable location increases as a function of capture range (R). In our simulation, Algorithm 5.2 is run with different values of R on prerecorded foreign body tracking data, and the maximum spatial probability found for each R is plotted. Results for multiple datasets (A–E) are shown simultaneously to qualitatively convey the variations between them. In a similar fashion, Algorithm 5.4 is run with different values of R to determine the dwell time at the most dwelled location for each R (Figure 5.10, *center*), and likewise for visit frequency (Figure 5.10, *right* and Algorithm 5.5).

From a design perspective, one observation is the saturation of performance above a 7 mm range in many cases. Dwell time (Figure 5.10, *center*) is computed as an average over the capture range and thus appears to decrease above 7 mm. When instead computed as a maximum, the expected monotonic increase is restored, as shown in Figure 5.11. The average view is presented first, as maximum values are more sensitive to per-trial variabilities.

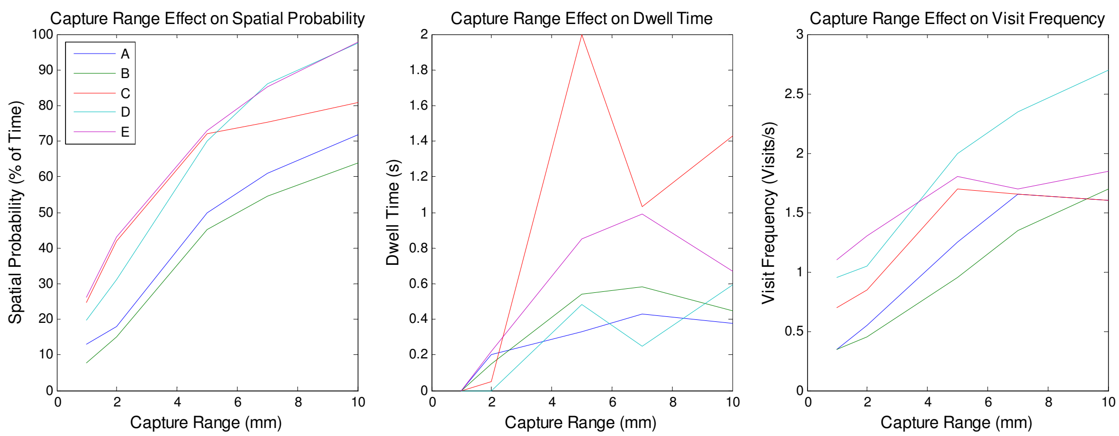


Figure 5.10: Effect of capture range on capture locations, determined using Algorithms 5.2 (spatial probability, *left*), 5.4 (dwell time, *center*), and 5.5 (visit frequency, *right*) with different capture ranges R to measure the respective capture locations. Series A–E correspond to individual datasets. Dwell time result C (*center*) is possible when the foreign body becomes trapped in a stationary part of the heart. A larger range leads to increased capture opportunity (e.g., greater perceived probability); of note is saturation at a range of roughly 7 mm.

Noticeable deviations from the average behavior suggest that the motion can fall into different, possibly aperiodic, modes. It is thus advisable to augment the aforementioned calculations by recording secondary locations (e.g., the second-highest probability) to increase the number of candidates, and by retaining sliding-window histories to adapt to slow, aperiodic changes in foreign body behavior.

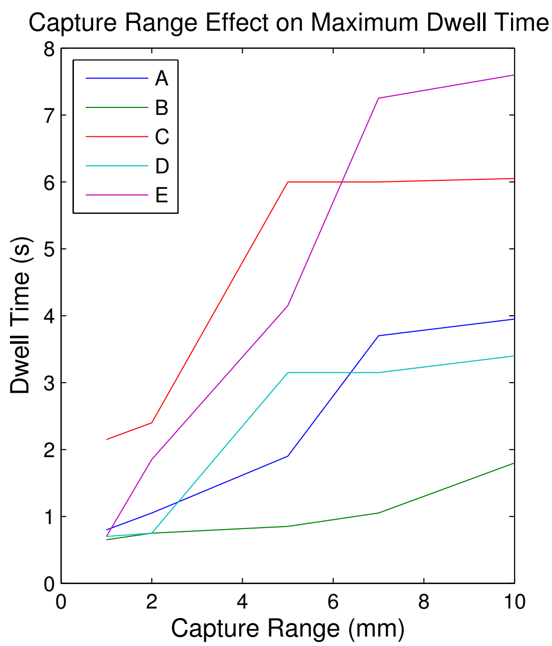


Figure 5.11: Effect of capture range on *maximum* dwell time (Figure 5.10, *center* applies to *average* dwell time). The figures increase monotonically with capture range as expected, with saturation above 7 mm.

5.2.4 Pre-Experiment Simulations

Capture location measurements reveal that the foreign body exhibits preferential regions of presence. We next examined the real-time evaluation of capture locations and their evolution over time [154]. Each row in Figures 5.12 and 5.13 shows the evolution of each type of capture location. The top rows show how the most probable location develops; on average, 18.5 seconds of tracking is needed for the measurement to reach 50%. The most dwelled location (*center rows*), meanwhile, can be determined after 6–8 seconds of tracking. Finally, as the bottom rows indicate, the visit frequency measurement stabilizes after 14–16 seconds of tracking.

Table 5.7 lists estimates of the minimum time intervals for which capture location measurements repeat. This is determined by dividing the data into equal time slots to find the duration with the maximum correlation of measurement profiles. On the whole, the results suggest a motion behavior periodicity of 3–4 seconds. Spatial probability prediction requires a longer observation period than do the others (4.5 seconds compared to 3.5), implying that although the behavior may be fairly consistent, the specific trajectories may vary. This is a preliminary finding to be more rigorously studied in the future.

The results lend insight regarding the tracking duration required to discover capture locations, thus providing some preparation for robot experiments. The predictability of figures based on prior measurements is revealed as well, and is a topic of interest for further investigation.

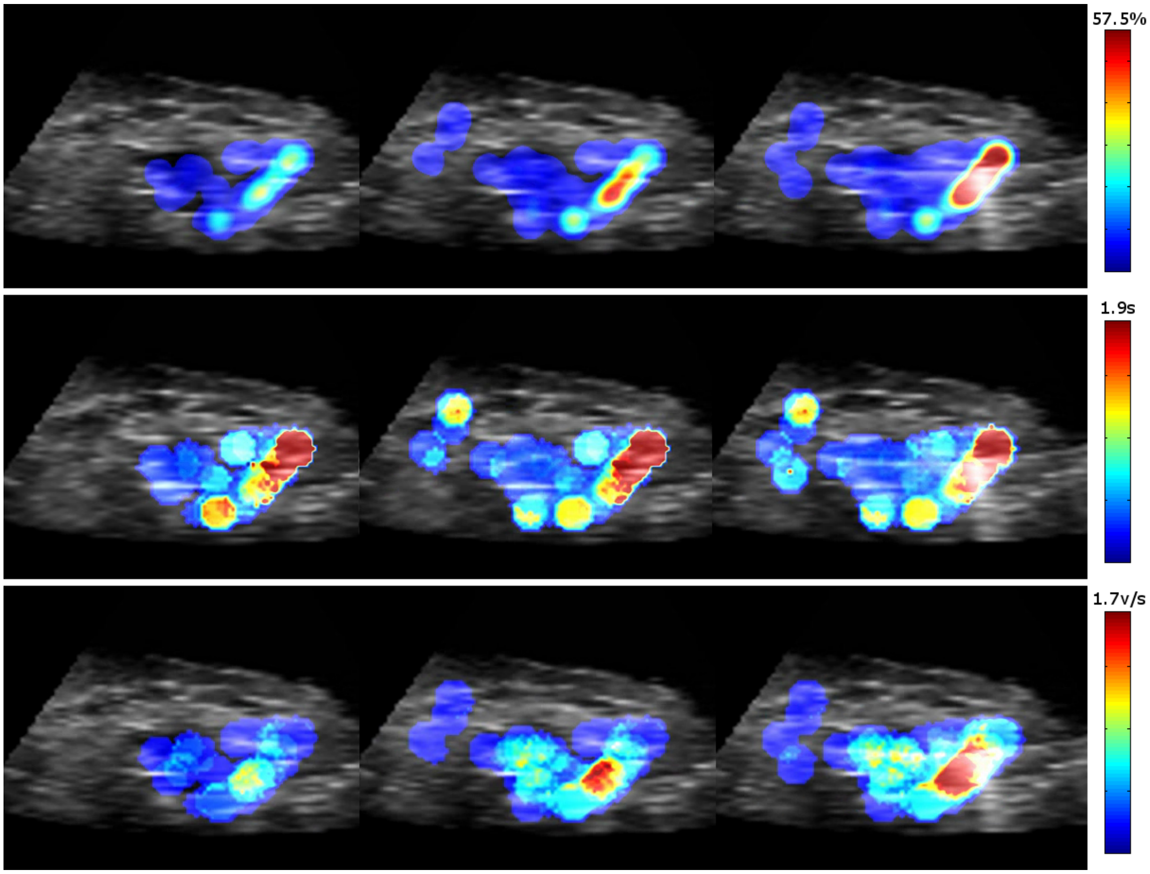


Figure 5.12: Capture location values from left to right at t = 6, 12, and 18 s. (*Top row*) Map of spatial probabilities, showing distinct regional preferences of the foreign body. (*Center row*) Map of dwell times, showing faster evolution than spatial probability. (*Bottom row*) Map of visit frequencies; highly-traveled sections are not necessarily the most probable or most dwelled.

Table 5.7: Minimum independent time intervals with repeating capture location estimates.

Measurement	Interval (seconds)
Spatial Probability	4.5
Dwell Time	3.5
Visit Frequency	3.5

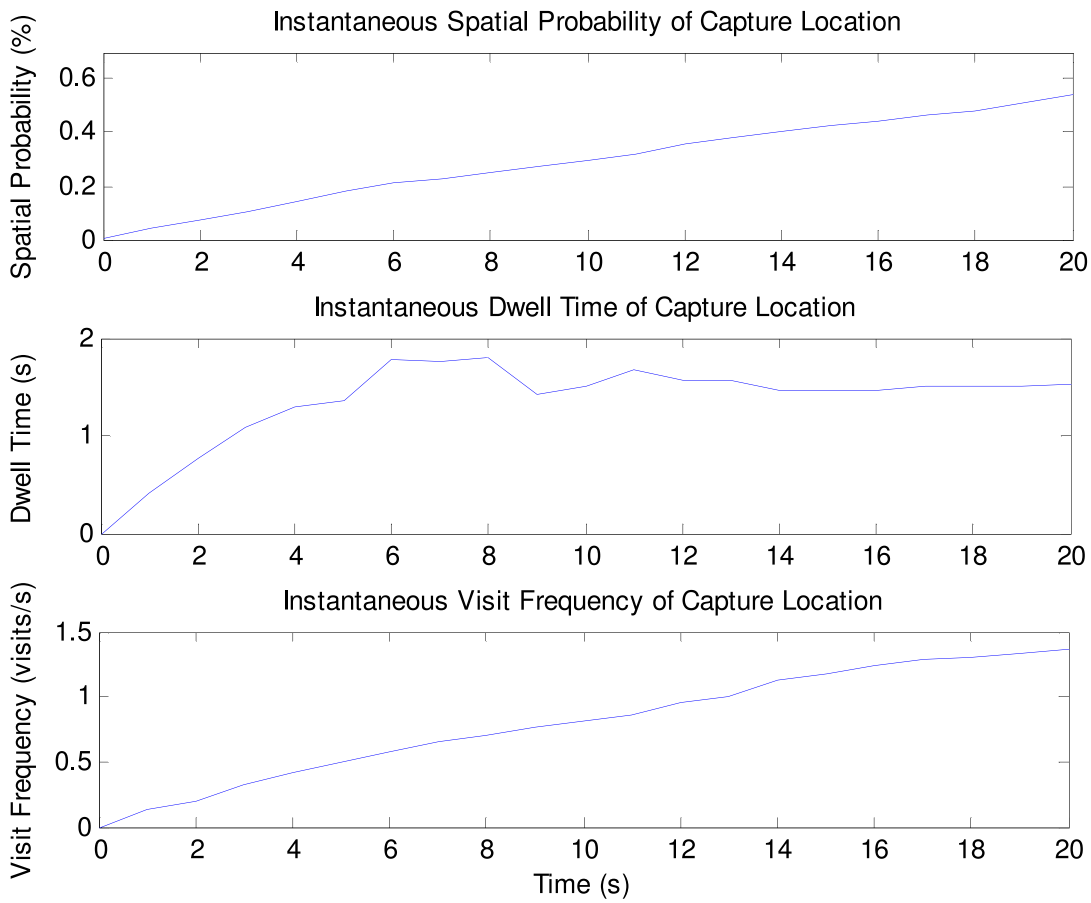


Figure 5.13: Development of capture locations over time. (*Top*) Spatial probability. (*Center*) Dwell time. (*Bottom*) Visit frequency; a slight plateau at t = 14–16 s appears more clearly in plots of maxima, not shown here.

5.2.5 Summary of Capture Location Criteria

A robot designed for minimally invasive foreign body retrieval must have adequate distal dexterity to reach arbitrary points in the heart. If the foreign body is to be chased down for capture, such a robot must be capable of high speeds and accelerations as well. Analyses of dexterity and speed requirements (Section 3.1.2) suggest that direct pursuit would be difficult to achieve. Given the tendencies that emerge after a period of observing the particle, we hypothesized that a viable strategy would be to aim a slow, dexterous end effector towards a carefully planned location in the heart and wait for the foreign body to travel into the device.

Multiple methods for determining these capture locations were devised, based on spatial probability, dwell time, and visit frequency criteria. Application of these measurements on foreign body tracking data helped improve our understanding of the motion, which can in turn be used to iterate the system design.

The particle is found in the most spatially probable location 50.5% of the time; dwells in the most dwelled location for 0.84 s at a time (with 2.28-s gaps in between), and transits the most frequently visited location 1.54 times per second. These values are informative in the design of a robotic retrieval system. For example, inter- and intra-dataset variances suggest that estimates be computed online so as to adapt to aperiodic changes in the motion. The ensuing steps will involve the design of a control scheme that computes capture locations in real time and guides a relatively slow robot to retrieve a fast, erratic target.

5.3 Initial Capture Experiments

5.3.1 Experimental Setup

The setup for capture experiments is shown in Figure 5.14. It consists of a beating heart phantom (Section 3.2) to recreate the clinical scenario, an ultrasound system (Section 3.3) to image the scene, a high dexterity robot (Section 3.4) to capture the foreign body, and a PC (2 GHz Xeon dual-core CPU, 4 GB of RAM) for tracking and guidance. Three-dimensional ($176 \times 96 \times 176$) ultrasound images are streamed at about 20 frames per second over TCP/IP.

A small 1.6×3.2 mm magnet is affixed to the snake robot tip to act as an abstract capture device with high attractive force within the capture range and negligible effect beyond it. The magnet was tested by advancing it vertically downwards towards a foreign body resting in water. The attraction distance was found to be 4.2 mm over 20 repetitions. While the use of a magnet is a simplification of the clinical scenario, the functionality and performance required of the robot guidance system is expected to be preserved.

5.3.2 Experimental Procedure

1. *Preoperative registration*: Registration is performed by moving the robot to various points within the field of view of the 3D TEE probe. Corresponding

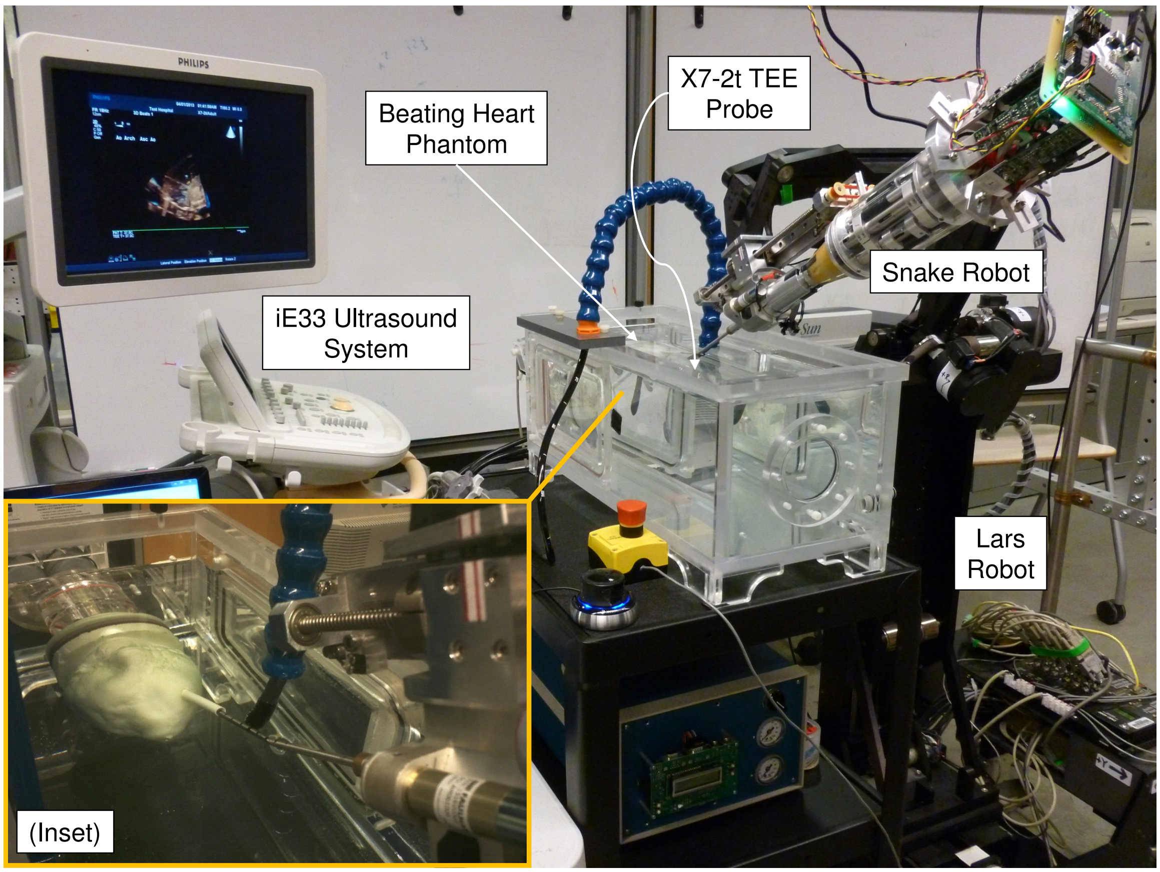


Figure 5.14: Setup for 3D ultrasound (TEE) guidance experiments using a high dexterity robot to capture a foreign body in a beating heart phantom (Figure 3.2, reproduced here for clarity). The inset shows a closeup of the snake robot inserted into the heart through the apex, with the TEE probe in place.

points in ultrasound image and robot coordinates are paired to determine the transformation between the two coordinate systems. The fiducial registration error (FRE) was found to be 1.0 mm, a fair result given the image resolution of about 0.8 mm/pixel. Further details on the registration process are included in Section 3.5.1. Though somewhat crude, this initial step suffices for the present study. We note the importance of streamlining the registration method for clinical use and address the issue in its own right in Section 3.5.2.

2. *Remote center of motion (RCM) placement*: In coarse positioning mode, the robot is teleoperated to the surgical entry point at the apex of the heart phantom. The RCM is defined upon user-initiated switch to dexterous/RCM mode (Section 3.4.3.3.2).

3. *Tracking initialization*: The foreign body (a 3.2-mm steel ball) is interactively selected from live ultrasound images of the heart phantom, to be used as a template for automatic tracking henceforth. This step is identical to its counterpart in the foreign body motion tracking and characterization experiments of Chapter 4.

4. *Capture location computation*: The foreign body tracking loop is activated, and the tracked positions are used to compute a capture location in real time.

5. *Robot guidance*: Once a capture location is found, the robot is guided to capture the foreign body.

5.3.3 Capture Success Criteria

Under the spatial probability method, a capture location emerges once the probability reaches 50%; when using the dwell time approach, a peak measurement (minimum 0.5 seconds) that is twice that of the runner up is used as the decision threshold. These values are based on measurements from [145, 152] and expected computation times from [154]. The visit frequency method is not included in this set of experiments as it is more suited for a net-like mechanism that captures a target in transit. We record the elapsed time in determining a capture location as well as the elapsed time to perform a successful capture. A capture is successful if the foreign body attaches to the magnetic tip without disengaging for at least five seconds, and unsuccessful if this condition is not met within 15 seconds of finding a capture location.

5.3.4 Initial Capture Results

The goal of this experiment is to demonstrate, via a closed-loop system, real-time analysis of foreign body motion in a beating heart phantom using 3D ultrasound, and to correspondingly guide a dexterous robotic end effector to capture it. To help evaluate the proposed methods, we record the success rate of capture attempts as well as the times required to execute the different phases of successful captures. Additionally, ultrasound image streams are recorded for further analysis and for offline verification of times and events.

5.3.4.1 Success Rate

An example capture sequence is shown in Figure 5.15. The system was able to retrieve the foreign body in 86.4% of attempts, including those based on spatial probability (14/17) and dwell time (5/5). Failed captures (13.6%) were due to prolonged displacement of the foreign body to a different location after a capture location had been identified, but before the robot was able to reach it; the possibility of such a phenomenon has been noted in our prior studies. We elected a single-shot style of approach in this experiment to avoid conflating issues of system design and implementation, with the intention of using the lessons learned to establish the necessary improvements. The outcomes suggest that even a rudimentary retrial strategy, such as operator-initiated reset, can improve the success rate to 100% after a modest number of reattempts. The 86.4% success rate for single-shot trials was thus encouraging, and the lessons learned were used to devise an automatic retrial procedure (Section 5.4).

The same 15-second threshold is used to decide failure during automatic retrial. In the future, more advanced metrics such as expected wait times may be used to make decisions. For example, for each position in the volume, we can compute the amount of time that the robot can expect to wait before the foreign body reemerges within capture range. This can be used to identify capture locations based on best wait times, or to report expected wait times for capture locations computed by other means. Furthermore, the expected wait time, or some multiple thereof, can be used to recognize a failed attempt and trigger a new computation.

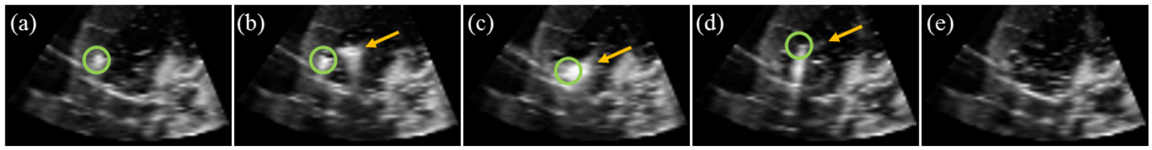


Figure 5.15: Image sequence showing different phases of foreign body capture. (a) Foreign body (*circle*) in heart phantom before computation. (b) Dexterous robot (*arrow*) approaches capture location. (c) Robot captures foreign body. (d) Robot leaves heart with foreign body attached. (e) Heart empty after foreign body is extracted.

5.3.4.2 Execution Times

Table 5.8 lists retrieval times broken down by phase. During the observation phase, the system passively tracks the foreign body to compute a capture location. This is perhaps the most critical part of the robotic procedure, and the measured time is particularly meaningful when retrial becomes necessary. Because heart surgery can last several hours, an average observation time of less than one minute suggests that the system can afford more time on observation and analysis for improved outcomes.

Table 5.8: Foreign body retrieval execution times (in seconds).

	Type of Capture Location Used	
Phase	Spatial Probability	Dwell Time
Observation	29.6 ± 6.9	54.3 ± 33.1
Insertion	2.8 ± 0.3	3.1 ± 0.5
Waiting	3.7 ± 2.0	2.2 ± 1.5
Total[1]	97.7 ± 21.6	124.5 ± 68.4

[1] Includes retraction time

The insertion phase encapsulates the travel time of the robot, starting with the moment a capture location is discovered and ending when the robot reaches its destination. Insertion times were fairly consistent at around 3 seconds throughout the experiments, as the robot was automatically servoed at a preset speed. The motion was slow enough to be monitored in the ultrasound stream, but not so much as to profoundly affect procedure times.

Upon reaching a capture location, the robot may need to wait for the foreign body to return to its capture range; this waiting phase was found to last 2.2–3.7 seconds on average. This measurement is important because it provides a bound on the least certain aspect of the procedure. As presented in the next section (Section 5.4), excessive waiting periods can be combated by continuing to form capture locations even after advancement to capture, then repositioning the robot upon expiry of a wait limit.

The bottom row of Table 5.8 states the total execution times, which include the per-phase times as well as the time to retract the robot from the heart. These final times vary widely because the retraction motion was triggered and sustained with user input. Nevertheless, total retrieval times of 2–3 minutes provide a rough estimate of the duration of the robotic part of the procedure.

Observation and waiting times exhibit large variances as well, in agreement with expectations noted in previous studies. The variabilities can be attributed to the irregular paths traced by the foreign body, as well as its tendency to get stuck on occasion, either in a transient pattern of motion or in some area of the heart chamber.

Figure 5.16 highlights the irregular motion of the foreign body during an example attempt; retrieval phases are indicated for reference.

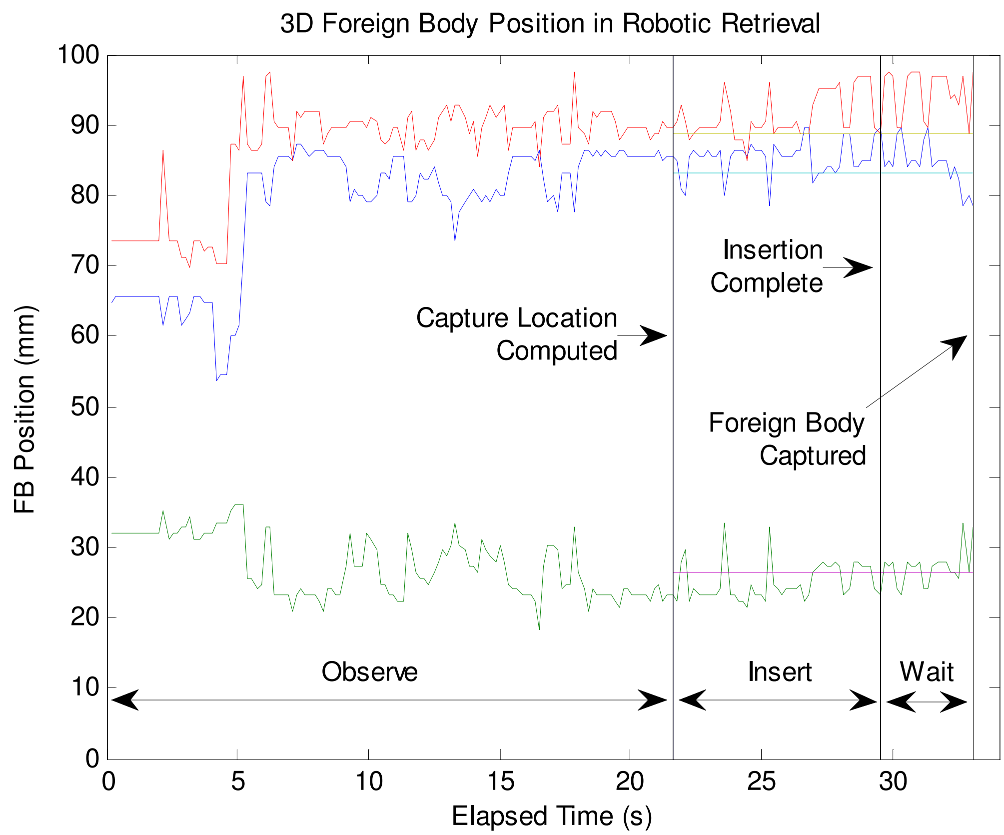


Figure 5.16: Example three-axis foreign body motion trace during retrieval, with observation, insertion, and waiting phases annotated. The blue, green, and red series correspond respectively to the X-, Y-, and Z-axis positions in ultrasound image coordinates.

5.3.4.3 Comparison of Capture Locations

Both spatial probability and dwell time methods were computed for each retrieval attempt, even though only one was used. As expected based on previous experiments, capture locations differed depending on the method used to compute them [152]. On average there was a 9.1 mm distance between the two types of capture locations.

Different methods of computing capture locations are, in general, expected to yield different results, as they approach the problem of intercepting an erratically moving target from different perspectives. But the degree to which they differ may be tied to each individual set of circumstances. For example, in the trivial case in which the foreign body becomes trapped and immobilized, all methods of computing a capture location should be in agreement.

A literal interpretation of success rates might suggest that use of the dwell time measure leads to better outcomes compared to the spatial probability criteria. However, the cause of failures in the latter case, i.e., transient foreign body behavior, can conceivably afflict dwell time estimates as well. At this stage of development, the proposed approaches are retained for continued assessment, and new ones may be introduced in the future.

Capture locations based on probability manifested over a range of 8.2 mm, while for dwell time locations this range was 11.7 mm—respectively 22.1% and 31.5% of the average extent of travel (37.1 mm) along the dominant axis. Observation times for probability were notably lower than for dwell time, and with less deviation (Table 5.8, *first row*). Taken together, these observations suggest a greater element of variability in the dwell time metric.

Such results can be expected as probability is effectively a figure integrated over time based on a histogram of foreign body positions, while dwell time is more sensitive to particular trajectories. Of note, waiting times were lower and less varying when dwell time was used (Table 5.8, *third row*), hinting at the existence of a tradeoff between the stability of determining a capture location and that of capturing the foreign body once a decision has been made.

Thus, any particular method may be preferred over the others depending on several factors, such as desired system behavior, the capture mechanism used, and environmental variables. It is also possible that all methods are adequate to the task, especially when retrial strategies are included. Expected wait times can be incorporated into capture location algorithms if greater certainty is desired.

5.3.5 Lessons Learned

In these initial capture experiments, only one attempt to secure the foreign body was allowed in each trial. This was done to emphasize the raw effectiveness of the proposed capture strategies. Under these conditions, the system achieved successful capture 86.4% of the time. Without further improvements, however, surgeons would have to withdraw the robot and repeat the process in 13.6% of cases. Procedures would become cumbersome, time consuming, and potentially more risky, introducing disadvantages that could outweigh the benefits of an image-guided robotic system. Resolving the failure scenarios is thus of the utmost importance.

The experiments revealed that captures failed due to shifting of the foreign body to a different place after the computation of a capture location, but prior to the arrival of the robot to that location. Such transient shifts in position away from seemingly regular motion patterns were consistent with our prior experiments [145, 152] without the use of a robot. Given the relatively high chance of success in a single attempt, we used this lesson to conclude that a success rate of 100% could be achieved by reattempting capture on the fly, without necessitating a restart of the procedure. An automatic retrial strategy is the topic of the next section (Section 5.4).

5.4 Capture with Automatic Retrial

5.4.1 Failure Analysis

Prior to the live foreign body capture experiments of the preceding section, studies on computing and assessing various forms of capture locations were performed without the presence of a robot [145, 152]. In these studies we noted that the foreign body, while appearing to move in a periodic fashion for the most part, was occasionally prone to large aperiodic shifts, either by being stuck in some region of the heart, or by assuming a different (typically periodic) trajectory. These types of events were cited as the major causes of measurement variabilities between experiment sessions. While potential problems for robotic retrieval were acknowledged, the matter was not resolved at the time because its ultimate effect was not clearly known. It was through experimentation that the failure mechanisms, and thus potential solutions, became identifiable. Via annotated motion traces, Figure 5.17 shows the sequence of events leading up to a failed capture.

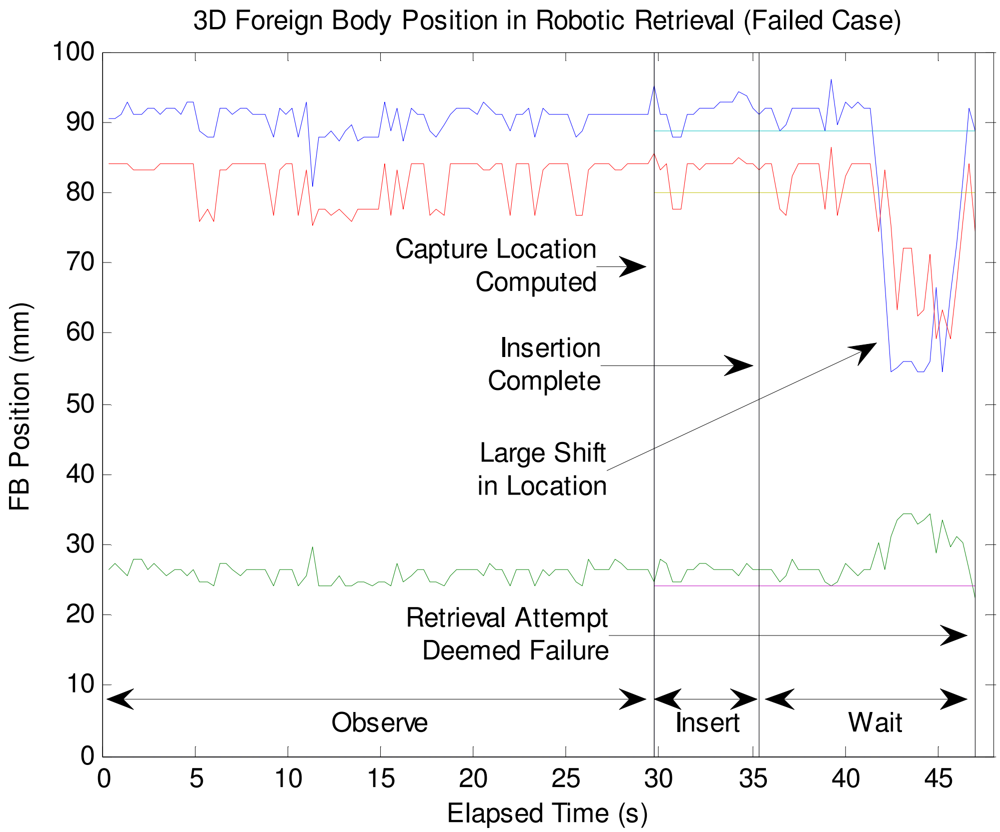


Figure 5.17: Three-axis foreign body motion trace for a failed retrieval attempt. The blue, green, and red series correspond respectively to the X-, Y-, and Z-axis positions in ultrasound image coordinates.

A direct pursuit scheme would be advantageous in this regard because it is relatively agnostic to particular positions or directions. However, the option had to be abandoned in part due to speed requirements, as explained in Section 3.1.2. Another way to resolve a missed capture is to withdraw the robot to the entry port and repeat the process. However, this approach can be cumbersome and time consuming in an operating room setting. It is thus preferable to utilize the capabilities and resources already available in the system to include an automatic retry feature that can resolve a shifted foreign body on the fly.

5.4.2 Automatic Retrial Procedure

The procedure for robotic capture of a cardiac foreign body, including automatic retrial after a failed attempt, is depicted in the flowchart of Figure 5.18. The colored boxes and regions denote functions in common with the initial (non-retrial) experiments from the previous section, while plain boxes contain auxiliary actions that enable the retrial capability. The green start box incorporates the major prerequisites of the procedure, such as placement of the robot at the RCM point and selection of the foreign body from ultrasound images to act as a tracking template. The yellow and blue regions signify the tracking and guidance processes respectively.

After a capture location is found and the robot is ready to be servoed, control branches into two concurrent processes, as indicated by the upper junction. A new acquire/track/compute process is executed concurrently with any robot guidance so that a new capture location is available if a retrial is needed. The algorithm seeks a new capture location to target if capture does not succeed within 15 seconds of the start of the attempt; in the initial experiments, this deadline was used to mark the trial as failed.

The workflow as shown can technically enter a non-terminating loop if every at-

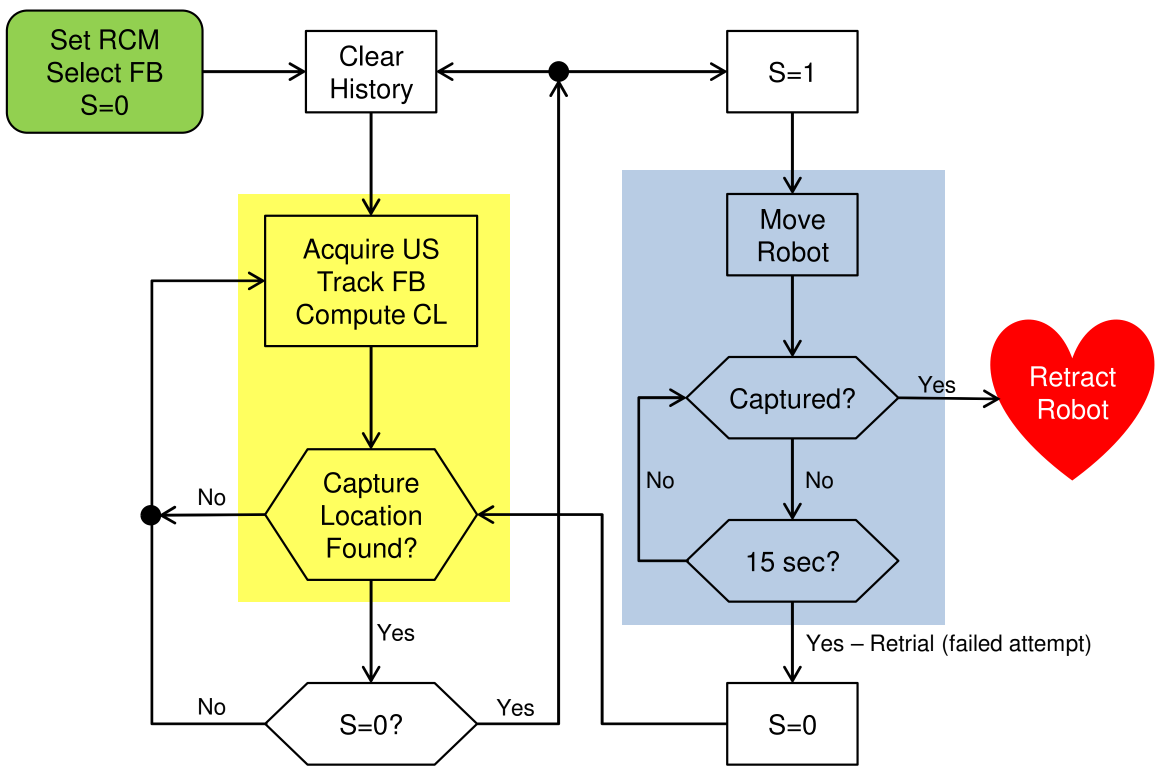


Figure 5.18: Workflow for robotic foreign body capture with automatic retrial.

tempt fails to capture the foreign body. This ultimately did not evolve into a practical concern because, as the forthcoming section details, at most one retrial was needed throughout the experiments. Nevertheless, the ability to abort on demand was available should the trial remain unsuccessful for effectively infinite time, i.e., the earlier of 10 retrials or 10 minutes.

5.4.3 Automatic Retrial Results

The goal of this experiment is to test the earlier hypothesis that the success rate of capture can be improved dramatically, from 86.4% to perhaps 100% with just a modest number of retrials. As mentioned, the robot is automatically guided to the next capture location, which had begun accumulating during the current attempt, if the current attempt fails. As in the non-retrial experiments, the condition for failure is no capture within 15 seconds of the beginning of the attempt, marked by both the completion of capture location computation and guidance of the robot. Aside from parameters related to retrial, all experimental conditions were maintained from the prior experiment, as were the types of data recorded.

5.4.3.1 Retrial Analysis

Per design, the system was able to retrieve the foreign body in 100% of attempts, including those based on spatial probability (14/14) and dwell time (22/22). An errant but otherwise arbitrary initial capture location was assigned in the first two spatial probability trials and first three dwell time trials in order to artificially trigger the retrial mechanism. Successful capture was observed in all five cases and in a lone natural retrial instance as well, each within a single retrial. The results are summarized in Table 5.9.

Table 5.9: Outcome of foreign body capture experiments with automatic retrial.

Capture Type	Trials	Retrials[1]	Success
Spatial Probability	14	3[2]	14
Dwell Time	22	3	22
Total	36	6	36

[1] Implies failure on initial attempt
[2] One case with natural initial failure, i.e., automatic retrial not artificially triggered

In one of the spatial probability trials, one natural miss occurred due to the foreign body shifting locations as the capture decision was being made. Snapshots of the critical moments of the ensuing retrial are shown in Figure 5.19. When the robot reached its initial destination, the target was elsewhere in the volume and came within 3.4 mm of the end effector without being caught. Although the capture range is quoted as 4.2 mm, the effective range is less when the foreign body is in motion. After a 15-second waiting period, a retrial decision was made, and a new capture location became available after 7.1 additional seconds. The robot repositioned itself and came to rest 1.8 s and 22.6 mm later, where it met the foreign body without any wait. The system was thus able to recover from an initial miss and capture the foreign body within one retrial. Figure 5.20 shows a plot of the motion traces corresponding to the preceding description, while Figure 5.21 summarizes the timeline of events.

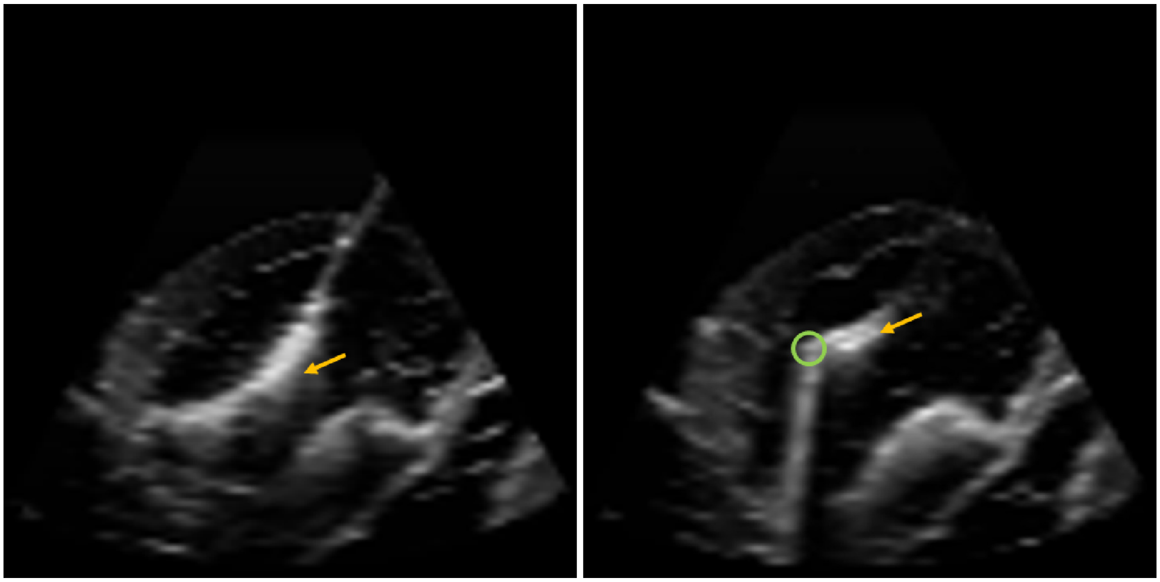


Figure 5.19: Ultrasound images showing two fundamental phases of automatic retrial. (*Left*) Dexterous robot (*arrow*) waiting at initial capture location while foreign body travels about heart, out of view. (*Right*) Robot captures foreign body (*circle*, recognizable by its long acoustic shadow) at a new capture location 22.6 mm away.

5.4.3.2 Non-Retrial Execution Times

We repeat the temporal breakdown of the retrieval process in order to examine the similarities and differences, if any, between sessions of experiments. The per-phase average execution times are given in Table 5.10. Since the preceding section has addressed cases involving retrial, the table below represents only trials that succeeded on the first attempt. Note that cases requiring retrial (e.g., Figure 5.21) are likely to take longer due to the extra time needed to wait and readjust.

Many results from the current set of experiments are within a reasonable distance of those from the prior run. Although the spatial probability observation time (24.4 s on average here vs. the 29.6 s seen previously), overall insertion time (3.6 vs. 2.9

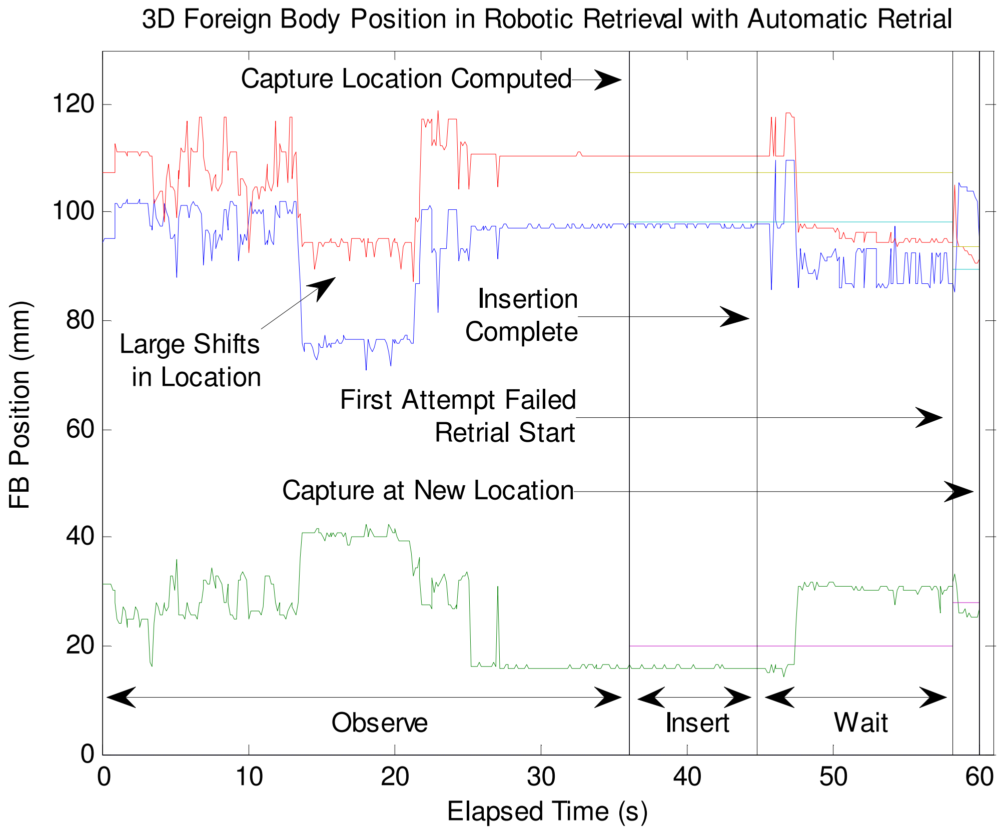


Figure 5.20: Three-axis foreign body motion trace during a retrieval attempt in which automatic retrial was engaged naturally. Large shifts in position caused the capture location estimate to become outdated. A new location was found and the robot repositioned itself to capture the target. The blue, green, and red series correspond respectively to the X-, Y-, and Z-axis positions in ultrasound image coordinates.

Table 5.10: Foreign body retrieval execution times (in seconds) for cases not requiring automatic retrial.

	Type of Capture Location Used	
Phase	Spatial Probability	Dwell Time
Observation	24.4 ± 8.2	18.6 ± 12.8
Insertion	4.1 ± 0.5	3.3 ± 0.4
Waiting	4.5 ± 1.7	3.9 ± 2.8
Total[1]	100.5 ± 46.5	65.3 ± 17.3

[1] Includes retraction time

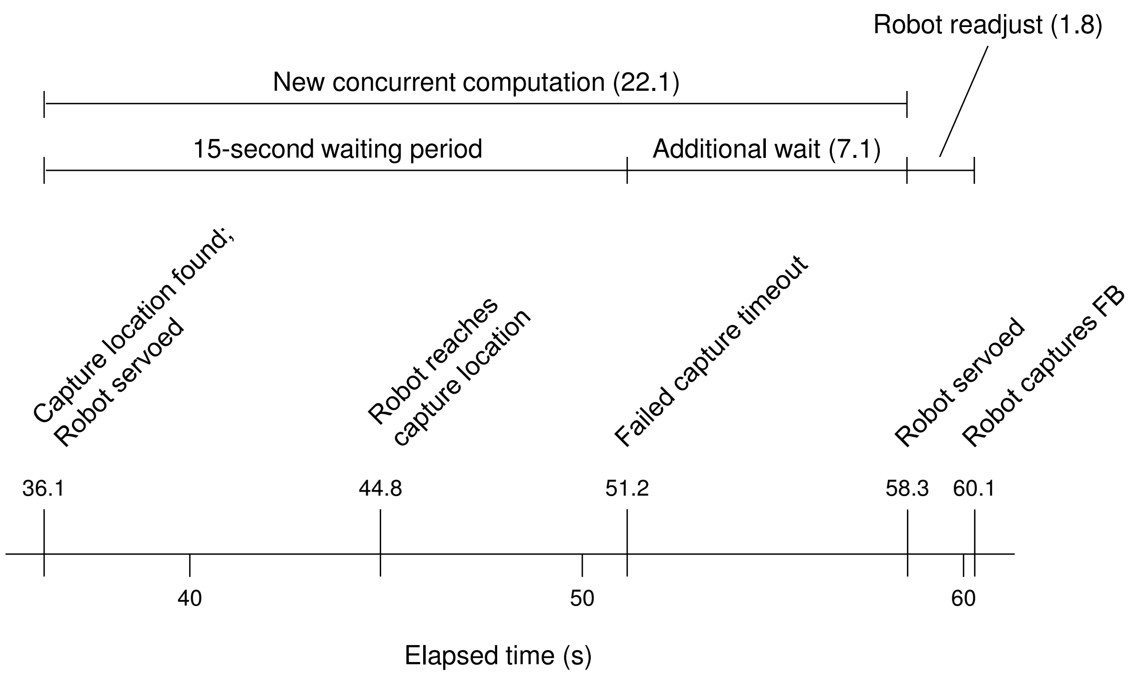


Figure 5.21: Timeline of events in robotic foreign body capture with automatic retrial, summarizing the trial shown in Figure 5.20. Execution times for retrial cases are likely to be longer than times listed in Table 5.10 (non-retrial execution times) due to the extra time in waiting and readjusting.

s), and overall waiting time (4.1 vs. 3.3 s) appear dissimilar in a relative sense, minor absolute differences of about five seconds during observation and ≤ 1 second apiece for insertion and waiting are to be expected given the reported deviations and the variability in the activity in general.

The most notable difference between Table 5.8 from the previous section and Table 5.10 above is the observation period for computing dwell time, which reduced from 54.3 to 18.6 seconds without any changes to the algorithm, parameters, or conditions. Dwell time is the more variable of the two tested characteristics by far, with high standard deviations owing to a sensitivity towards particular positions and paths, as noted before.

Because one of the dwell time criteria includes dominance (specifically, twice the value) of the primary candidate over runners up, lack of a clear leader can lead to extended observation time. The present session saw a greater distribution of dwell time capture locations throughout the volume—21.7 mm in the major axis as opposed to the previously found 11.7 mm range (respectively, 58.5% and 31.5% of the 31.7-mm extent of foreign body travel). A denser gathering of capture locations hints that the foreign body may be confined to a smaller region, which in turn may lead to the presence of several viable candidates. Conversely, a wider scattering of capture locations suggests greater foreign body travel and thus fewer positions that can serve well as dwell time capture locations. Under these conditions, capture locations can more quickly emerge, resulting in the reduced observation periods seen here.

While variabilities make inter-session evaluations somewhat challenging, spatial probability and dwell time results within the same trial are more readily compared because they are measured simultaneously on identical data. Accordingly, we found that dwell time capture locations consistently yield shorter waiting periods than do spatial probability ones: 3.9 seconds of wait to 4.5 in the current set, and 2.2 against 3.7 seconds in the previous. Again, there may be a tradeoff between the temporal certainty of producing a capture location and the temporal certainty of capture once the system has committed to an action.

5.4.4 Section Summary

A failed capture is overall an atypical event, but it is likely enough to occur that a contingency plan must be in place. The experimental results indicate that the success rate of capturing a foreign body from a beating heart can be driven to 100% with a minimal number of retrials. This section describes a mechanism for automatic retrial and demonstrates its feasibility, but other implementations may be suitable as well. One possibility is to allow the operator to override a computed capture location with a manually selected one. The operator can also be allowed to manually trigger a retrial or abort any action in progress.

5.5 Discussion

Improvement in the temporal determinism of the system is a topic of interest, though the degree of determinism, i.e., at a more precise level than an upper bound, may be a subjective matter. One possibility is to eliminate the "double dominance" dwell time criterion, which was instituted as a countermeasure against premature decisions and the lack of determinism in space, as several places in the heart can share a common dwell time in the first few seconds of tracking. Nevertheless, the primary goal of this study was to successfully capture a foreign body in the heart as it moves erratically due to heartbeats and blood flow. While the system should perform the task in a reasonable amount of time, an explicit time constraint was not defined. In the context of cardiac surgery lasting several hours, a reasonable duration for the robotic part of the procedure is at least a few minutes, which was achieved in the experiments presented.

Over multiple sessions of experiments, dwell time continues to exhibit greater variability in computation time as compared to spatial probability, yet in the live capture experiments provides better performance in terms of success rate and waiting time for the foreign body to meet the end effector at the capture location. While natural misses have taken place 4/29 times (13.8%) under spatial probability, none have been witnessed in 24 dwell time attempts (artificial misses are excluded because it is not known whether they would have failed naturally). An advantage of applying spatial probability to cardiac foreign body tracking is the relative stability in producing an answer; that is, because it is simply an accumulation of a histogram, the computation time is less varied. The associated disadvantage then is that older data plays an equal role in the decision making process as current and likely more relevant data. This is a common issue in probability-based prediction, and one way to counter this effect is to deemphasize data points over time. Nevertheless, the presence of the phenomenon in foreign body tracking speaks to the often predictable, sometimes transient nature of cardiac foreign bodies, as we have assessed previously in qualitative terms. Spatial probability, as implemented in this work, appears to handle 86.2% of cases natively.

In contrast, dwell time appears to be a more suitable metric in capturing a moving foreign body, given its higher success rate and lower wait times. This, however, comes at the expense of having less certainty in knowing when a capture location will emerge. Under certain environmental conditions, such as multiple dwell loci or constant movement, it is possible for the observation aspect of the procedure to continue for a very long period unless interrupted by some other condition. The distinction between these two types of capture locations may be less important when retrial capabilities are available. Furthermore, multiple types of calculations can be performed simultaneously, and the chosen result can be a function of time constraints, reachability, etc. As our experiments demonstrate, the success rate of robotically capturing a foreign body from a beating heart can be driven to 100% with a minimal number of retrials, regardless of the base algorithm. From a broader perspective, it may be more effective to develop the capabilities needed for a variety of circumstances, in addition to optimizing on certain subtleties.

The proposed approach calls for tracking of the foreign body to accumulate statistics on its motion, then deploying a robot to capture it accordingly. One question that arises is how introduction of the robot into the heart affects the flow of blood and/or motion of the object; this question is rooted in the concern that such alteration of the environment has the potential to invalidate any computed capture location.

In our experiments, we believe that the robot causes negligible disruption to the natural activity within the heart. The snake robot is narrow (4.2 mm in diameter) and has a skeleton profile (see Figure 3.6, *inset*) that offers minimal impedance to the circulation. In tandem with the very high surrounding flow rate (on the order of 300 mm/s), these factors make diversion of the fluid difficult; thus the flow pattern is expected to remain largely unchanged. Any effect that the robot has on its surroundings is further mitigated because in the first 1–2 seconds of insertion, only a fraction of the robot is in the heart. The robot speed of approximately 17 mm/s is sufficiently conservative to avoid inducing new currents as well.

Some influence of the robot on the blood flow and foreign body motion is ultimately expected, but the effect appears to be below detection for the present set of experiments. Empirically, the 86.4% success rate indicates that computed capture locations remain relevant after insertion of the robot, and it is not suspected that this success rate can be achieved with undirected guidance. In guiding the robot to arbitrary non-capture locations, we did not observe a capture by chance over trials of

4:44, 4:30, and 5:05 minutes—14:19 minutes in total. Also, the source of the 13.6% failure rate, i.e., sudden displacement of the foreign body away from the computed capture location, was noted in our previous experiments that did not involve a robot. Thus, the robot's connection to the failed attempts is at least inconclusive if not unlikely.

Though not observed, collision of the foreign body with the robot is a realistic event; if the above rationale is true, however, the resulting change in path would be transient and the prior behavior would resume. Nevertheless, the effect of the robot on the foreign body should continue to be noted. In fact, minding clinically acceptable limits, it may be possible to devise a system that alters the motion of the foreign body by design, in a controllable manner that facilitates capture.

Another question that arises is whether a smart tracking and guidance system is worthwhile at all—that is, whether the end effector can simply be placed arbitrarily for capture by coincidence. The possibility exists that the foreign body will eventually chance into capture after a sufficiently long wait. However, as mentioned above, such an incident was not encountered in 14:19 minutes over three sessions. Wait times of this magnitude may be difficult to tolerate during cardiac procedures, and may be unacceptable altogether with a time-sensitive device such as suction. Overall, the results demonstrate the viability of using a slow, dexterous robot to capture an erratic cardiac target under 3D ultrasound guidance.

5.6 Chapter Summary

This chapter combined a 3D ultrasound-guided, high dexterity robot (Chapter 3) with real-time 3D ultrasound-based tracking (Chapter 4) to realize the goal of retrieving a foreign body from a beating heart in a minimally invasive manner. First, direct pursuit of the foreign body by the robot was attempted, even though a significant mismatch between the foreign body speed (343 mm/s) and robot speed (17 mm/s) was evident. (The robot speed was limited by its physical capabilities and by safety concerns.) We learned that for direct pursuit to be feasible, either the robot must be remarkably fast, or the target must be substantially slower. A related study on an endoscope robot provided the insight that not only must the robot be fast, it must be faster than the target in order to overcome latencies in the control pipeline. These factors effectively ruled out the viability of a direct pursuit approach.

Thus originated the strategy of indirect pursuit, or ambushing, whereby the foreign body is passively tracked in the ultrasound stream in order to accumulate a statistical measure, which in turn is used to guide the robot to a *capture location*, where the particle can be expected to appear and be caught by the waiting robot. Criteria outlined for the computation of capture locations included spatial probability, dwell time, and visit frequency. Pre-robotic phantom evaluation of capture locations provided estimates that (1) helped establish expectations for performance, and (2) inform the creation of the protocol for the robotic experiments.

In the initial set of robotic capture experiments, each trial was limited to a single attempt in order to assess the raw performance of the system and strategies. That is, if a capture could not complete within 15 seconds after a capture location was found, the trial was marked as failed. Under this protocol, successful capture was achieved in 86.4% of the trials. We observed that the 13.6% failure rate was caused by sudden but infrequent shifts of the foreign body after a capture location had been determined. The implication was that a foreign body behaving in such a manner could eventually be captured if the system was able to compute new capture locations on the fly. An automatic retrial mechanism was devised and tested via a fresh set of experiments, the result of which was a successful capture rate of 100%. This outcome demonstrated the feasibility of real-time tracking and dexterous guidance under 3D ultrasound, as well as the ability to capture a fast, erratic foreign body from a beating heart (phantom) using a slow, dexterous robot.

Table 5.11: Technical barriers overcome (Section 1.1.4)

• Devising strategies for capturing a foreign body in turbulent flow
• Performing autonomous guidance of a dexterous robot using 3D ultrasound
• Ensuring real-time performance of the tracking and guidance computations

Table 5.12: Research contributions (Section 1.2)

• First feasible MIS solution for capturing cardiac foreign bodies
• First system for autonomous, dexterous guidance in a beating heart
• Capture of a fast target with a slow robot under 3D ultrasound guidance

Chapter 6

Conclusions

Foreign bodies in the heart, such as thrombi and shrapnel that migrate through the venous system, can interfere with cardiovascular function. The resulting adverse effects must be weighed against the conventional treatment, often a highly invasive surgery incurring pronounced perioperative and postoperative health risks. In an effort to improve the state of care, we proposed a 3D ultrasound-guided, high dexterity robotic system to help surgeons detect and remove foreign bodies from a beating heart in a minimally invasive manner.

The ability to track a foreign body in a beating heart phantom, in real time, using live 3D TEE images was developed, enabling characterization of the particle motion. Straightforward pursuit using a non-dexterous surgical robot proved difficult, as the foreign body motion was found to be fast and erratic. As well, we reasoned that dexterity was a requirement in this application because the apex of the heart, the entry point of the device, does not allow free pivot. Because highly dexterous robots are difficult to control at high speed, it would be potentially unsafe to operate a robot fast enough to chase a foreign body inside the heart.

When observed over time, distinct tendencies in the foreign body motion emerged, offering the possibility of using a slow, dexterous robot to ambush the target at carefully planned locations. Alternative, real-time (at the 20 Hz ultrasound frame rate) methods of computing these so-called capture locations were then formulated. Results of robotic retrieval experiments demonstrated the viability of capturing an erratic, fast-moving foreign body, in a beating heart, using a slow, dexterous robot guided by 3D ultrasound. The 100% success rate achieved with an automatic retrial mechanism suggested that continued refinement can lead to an effective solution for the treatment of cardiac foreign bodies. With robotic aspects incurring perioperative times on the order of a few minutes, additional observation/computation time can be used to potentially increase robustness and reliability.

6.1 Next Steps

An important step in the development of this work is to design an end effector to secure the foreign body once the former coincides with the latter. Section 5.2, and Figure 5.9 in particular, suggest a few reasonable options: suction, gripping, and netting. Parameters that may influence the selection process include capture range, required accuracy, and speed of deployment. It may be necessary to evaluate multiple alternatives in order to determine the relative importance of these factors and their respective contributions to overall effectiveness.

The ability to automatically verify capture would be beneficial to a robotic workflow because it would remove a degree of uncertainty in a sensitive surgical procedure. One possible verification method would be to continue tracking the foreign body even after the robot reaches the capture location. If the foreign body is last seen entering the robot's capture range and no longer seen thereafter, it can be inferred that capture has occurred. A sufficiently transparent capture device would allow the foreign body to be tracked even post-capture, making this inference unnecessary. Different techniques may be appropriate depending on the nature of the mechanism. For example, in a gripping or clamping device, an optical or contact sensor between the jaws can detect the presence of a foreign body. For suction, verification may be accomplished by direct inspection at the proximal end of the tube. These per-device methods can also be combined with image-based detection to improve the verification certainty.

The next major milestone is to test the system *in vivo*. As part of the preparations for an animal trial, questions on the setup and workflow can be addressed and optimized, and the sterilization process for different components can be refined. Such a test would help highlight areas in need of improvement, be they logistical, mechanical, or algorithmic. X-ray integration may help with the initial localization of foreign bodies.

To facilitate a feasible workflow, the preoperative registration process must be improved, possibly in the form of improved online registration. One approach consists of automatic segmentation of the dexterous robot in images, and comparing the extracted configuration against robot kinematics to deduce the image-to-robot registration. Some challenges that online registration faces in practical use include noisy data and errant segmentation of instruments, events which can be expected due to the motion of the tools and changing appearance of the environment. An estimation approach, such as with an (extended) Kalman filter, may be appropriate to address these conditions. Remaining issues include robust segmentation of the robot in the images, detection of robot departure (partial or full) from the field of view and correct segmentation in such situations, and determination of when assumptions, such as the direction of robot entry into the image, are breached. It is also important that the computations be performed in real time in order to faithfully reflect live conditions.

Provisions for improved visualization should be considered. Though clinicians are typically comfortable with 3D imaging, 3D information remains more difficult to interpret than 2D images due in part to information overload. With the addition of complex information such as computed capture locations and probabilities, intuitive presentation of the data is important to allow clinicians to respond appropriately. For more general use cases, the tracking component should be expanded to accommodate multiple simultaneous foreign bodies, as well as foreign bodies of different shapes, sizes, and compositions.

6.2 Future Directions

6.2.1 Online Image-Based Registration and Virtual Fixtures

The goal of this study is to introduce computationally efficient techniques for tracking motion in a beating heart in real-time. In contrast to related work focusing on delineation of specific anatomical features, the intent of this study is to augment the robot guidance system with safety constraints, so that the end effector can execute the foreign body retrieval task without impacting the surrounding tissue. Additionally, 3D TEE probes are long and flexible, making them susceptible to inadvertent displacements; as found in previous studies (e.g., [10]), unintended probe motion may disrupt the surgical workflow by requiring re-registration, or complicate the surgical setup by requiring extra sensing equipment.

An image-based approach for tracking both heart and probe motions would be advantageous as it would not require extra hardware or steps in the surgical procedure; natural fiducials exist in the unique structure of the heart. However, existing image-based registration techniques (e.g., mutual information maximization) targeted predominantly at preoperative scenarios are prohibitively slow for intraoperative computation. Real-time tracking of heart surfaces would allow for (1) online registration between the robot and ultrasound probe in the face of minor probe motions, and (2) intraoperative adjustment of virtual fixtures to reflect the live patient anatomy. We propose an image-based method for achieving these goals without further complicating the surgical setup and workflow.

In the context of real-time control of a surgical robot, we expect to make compromises between the speed and the accuracy of registration and tracking. We hypothesize that such tradeoffs can be accommodated while maintaining functional and safe robotic guidance. A conservative safety barrier can be constructed by observing regions traversed by the heart wall at any point in time, and avoiding any such regions at all times. While computationally simple, this approach has the potential drawback of inhibiting the robot from capturing the foreign body. A preferred, computationally efficient method for determining a larger operational workspace is sketched below.

1. Filter speckle and other forms of noise.

2. Perform morphological operations to enhance edges and surfaces.

3. Identify and reject imaging artifacts such as strong reflections and shadows.

4. Detect surfaces and classify objects, e.g., tissue, robot, and foreign body.

5. Extract sample points along the internal surface of the heart, enabling the system to construct a linearized surface approximation for use as a barrier virtual fixture.

Assuming known, fixed transformations from the patient, to the bed, to the world, to the robot, the relatively sparse surface approximation can be used to detect, in real time, any probe movement from its initial (known) configuration. This measurement can then be used to update the registration between the probe and the robot, allowing the surface to be transformed from image to robot coordinates. Once the surface is expressed in robot coordinates, it can be used within the robot controller to establish a safety barrier by solving a virtual fixture problem. At each control step, points on the surface near the robot would be interpolated and cast as geometric constraints to disallow the robot from breaching the surface.

In performing this tracking it is assumed that changes between consecutive image frames are small, so the results of one frame can be used to constrain the estimation in the next frame. This assumption also allows for reduced regions of interest if a spatial search is involved. In a similar vein, variations in the appearance of the tissues and tool are expected to be bounded, so comparison of the current image frame against an image averaged over multiple heart cycles can be used to infer objects and their locations. Such variations are also expected to be periodic; the efficacy of synchronizing the image stream to a small database of images populated and updated intraoperatively can be explored.

6.2.2 Advanced Modeling and Guidance

Section 5.2 outlined a few key methods for computing capture locations, but additional criteria of interest were unexplored in this work, such as (weighted) combinations of capture locations. For example, rather than acting on the greatest dwell time, it may be preferable to use a location with both a long enough dwell time and a high enough spatial probability to improve the chance of success. More advanced metrics such as expected wait times may also be used to make decisions. For example, for each position in the volume, we can compute the amount of time that the robot can expect to wait before the foreign body reemerges within capture range. This can be used to identify capture locations based on best wait times, or to report expected wait times for capture locations computed by other means. Furthermore, the expected wait time, or some multiple thereof, can be used to recognize a failed attempt and trigger a new computation. These criteria attempt to make coarse statistical predictions, but predicting the trajectories of a foreign body using conditional probabilities, Markov, and Bayesian models represents a different direction of interest. It would be interesting to find associations between foreign body motion and existing models, and to test their effect on capture outcomes.

With regards to guidance, we demonstrated near-autonomous capability, but support for a spectrum of cooperative control is a possible future direction. For example, capture locations can be defined manually through the operator console, or worst- and best-case expectations for dwell/wait times can be used for automatic operation; in between these extremes, the system can provide a set of intelligent recommendations that the operator can select from. Similarly, in situations requiring retrial, the adjustment can be performed manually, autonomously after an expected wait time has been exceeded, or manually with system suggestion.

As a final note, a system may compute several different capture locations simultaneously, and plan a capture path based on factors such as the reachability of the location, the expected time to reach and wait, and obstacles. On the other hand, minding clinically acceptable limits, it may be possible to devise an *active* system that, rather than waiting for the foreign body, alters the motion of the foreign body by design, in a controllable manner that facilitates capture.

Appendix A

An IEEE 1394-Based Motion Controller for the Snake Robot using a Centralized Processing, Distributed I/O Architecture

Preamble

This chapter originated as a project report [125] submitted to the Department of Computer Science, in fulfillment of a Ph.D. degree requirement. This writing includes minor updates and corrections, as well as a section (A.9) devoted to subsequent developments [126]. Nevertheless, details may remain or subsequently become outdated.

Besides enabling use of the Snake Robot (Section 3.4.2), the outcomes of this effort formed the basis for *JHU Open Source Mechatronics* [155, 156], which publicly hosts a set of electronics design files, FPGA code, and basic software for a FireWire-based motion controller. This in turn is a component of the Intuitive Research Kit [157,158]. These projects are advised or co-advised by P. Kazanzides.

Abstract

Research in surgical robots often calls for multi-axis motion controllers and other I/O hardware for interfacing various devices with computers. To facilitate convenient prototyping of robots with large numbers of axes and I/O lines, it would be beneficial to have controllers that scale well in this regard. We would like to incorporate additional components into a system without necessitating an unwieldy increase in I/O devices, nor introduce an excess of disjoint software interfaces and environments.

Field-programmable gate arrays (FPGAs) enable low-latency mechatronic interfaces. High speed serial buses such as IEEE 1394 (FireWire) make it possible to consolidate multiple data streams into a single cable, and contemporary computers with real-time operating systems have the computational resources to process such dense data streams. The accessibility of these technologies motivate a centralized processing, distributed I/O control architecture, which is particularly advantageous for education and research.

This paper documents the design, implementation, testing, and deployment of a motion controller and its associated API using this design approach. A real-time controller for the Johns Hopkins University Snake Robot, a novel, miniature, and dexterous surgical manipulator with many degrees of freedom, is developed that combines previous experience and the aforementioned motivations with readily available but powerful new technologies.

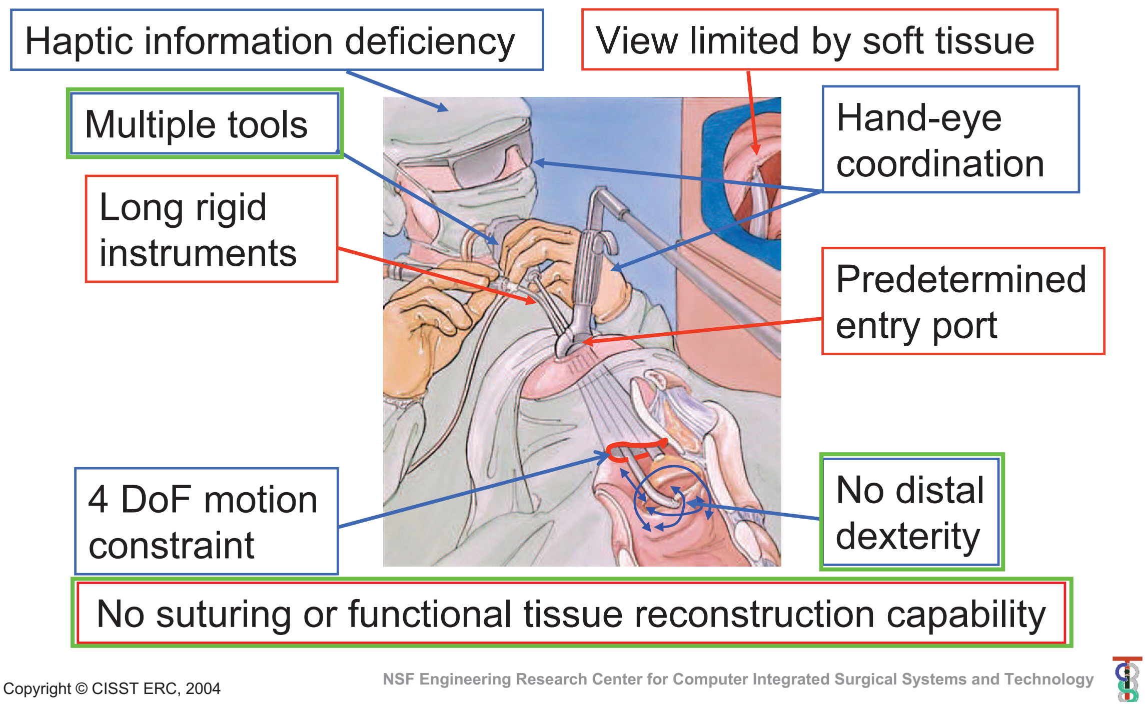


Figure A.1: A minimally invasive surgery setup (*Credit: A. Kapoor*).

A.1 Introduction

A.1.1 Background

Minimally invasive surgery (MIS) is often beneficial for patients due to reduction of trauma, leading to fewer complications and shorter hospital stays. However, MIS poses a number of challenges for surgeons, including constrained workspaces, limited field of view, and lack of dexterity at the distal end. These challenges remain despite the availability of manual MIS-specific instruments, which are rigid, difficult to manipulate through narrow insertion tubes, and they lack adequate suturing and tissue reconstruction capability. The situation is illustrated in Figure A.1. In such situations, the efficacy of a surgical robot is strongly tied to its dexterity.

Research on the Snake Robot [1] seeks to improve MIS of the throat and upper airways by providing surgeons with highly dexterous robotically-controlled tools. This dexterity is achieved by incorporating more degrees of freedom (dof). More sophisticated surgical tasks can be accomplished by increasing dof, but the corresponding hardware increase imposes a practical limit on the exploration of these ideas. Similarly, research on different types of multi-axis surgical robots is often mired in the hardware construction effort. In response to these difficulties, this paper presents the development of a system that is well suited for real-time control of robots with many axes of control.

A.1.2 Snake Robot

A unique design targeted for MIS of the upper airways, the Snake Robot addresses these issues by introducing a small, dexterous end effector that can be teleoperated. To avoid obscuring the work area, the end effector is attached to its actuators via a hollow, narrow, meter-long shaft containing its wires, and appears at the distal end of a laryngoscope; this shaft is also teleoperable, as described below. Figure A.2 depicts the configuration.

The Snake Robot is an eight-dof (11-actuator) manipulator. Multiple Snake Robots may be used for surgical tasks including bimanual suturing, suction, and

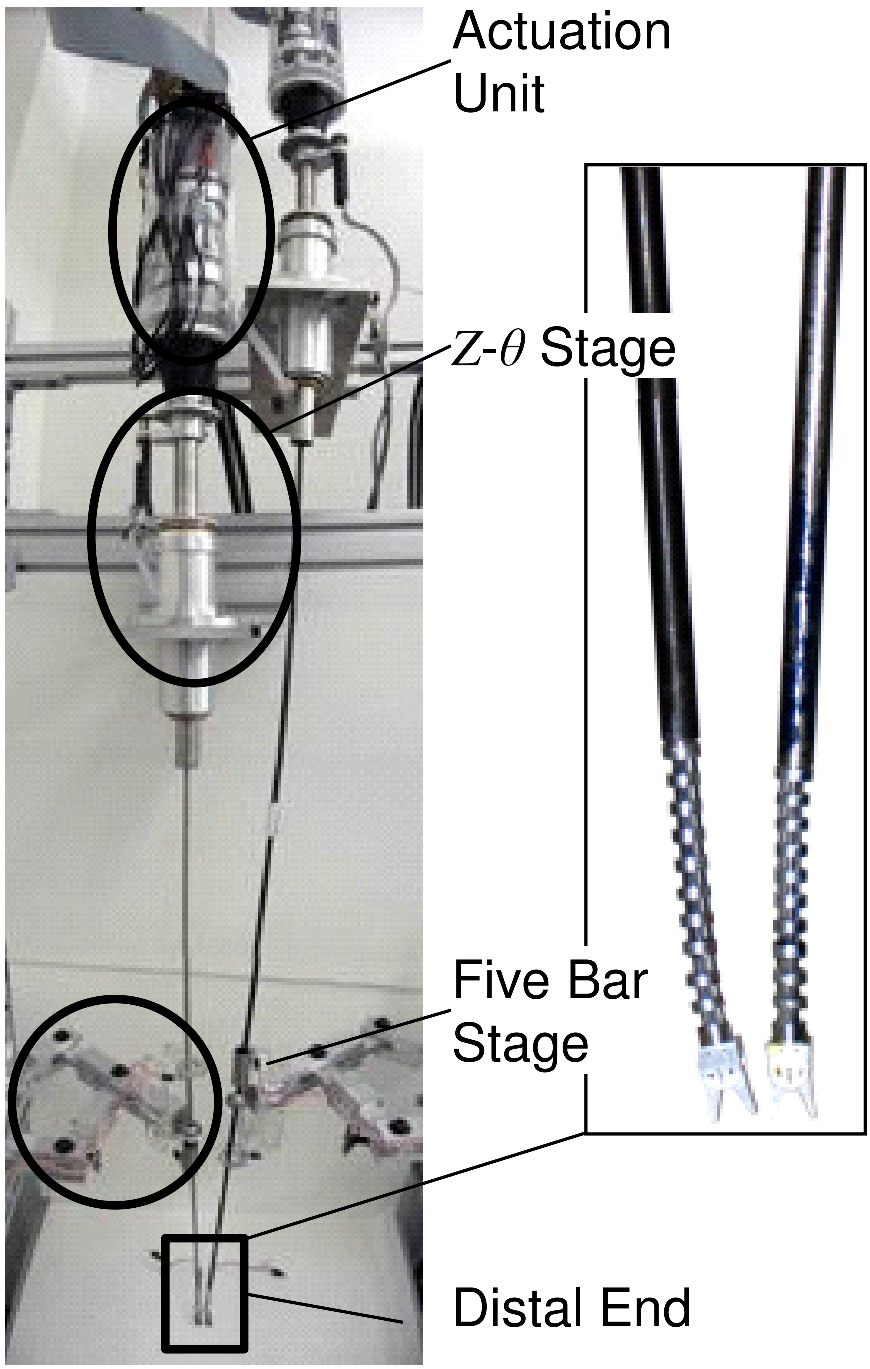


Figure A.2: Snake Robot prototype (*Credit: A. Kapoor*).

camera placement. The dexterous end effector consists of two snake-like units (SLUs) connected in series; each SLU is constructed using four superelastic nitinol tubes. The anatomy of an SLU is shown in Figure A.3. The center tube, i.e., the primary backbone, is surrounded by the three other tubes, the secondary backbones, at equally-spaced radial and angular distances. These four backbones are fixed to the end disc, but only the primary (central) backbone is fixed to the base disc; all backbones are free to glide through holes in the equally-spaced intermediate spacer discs.

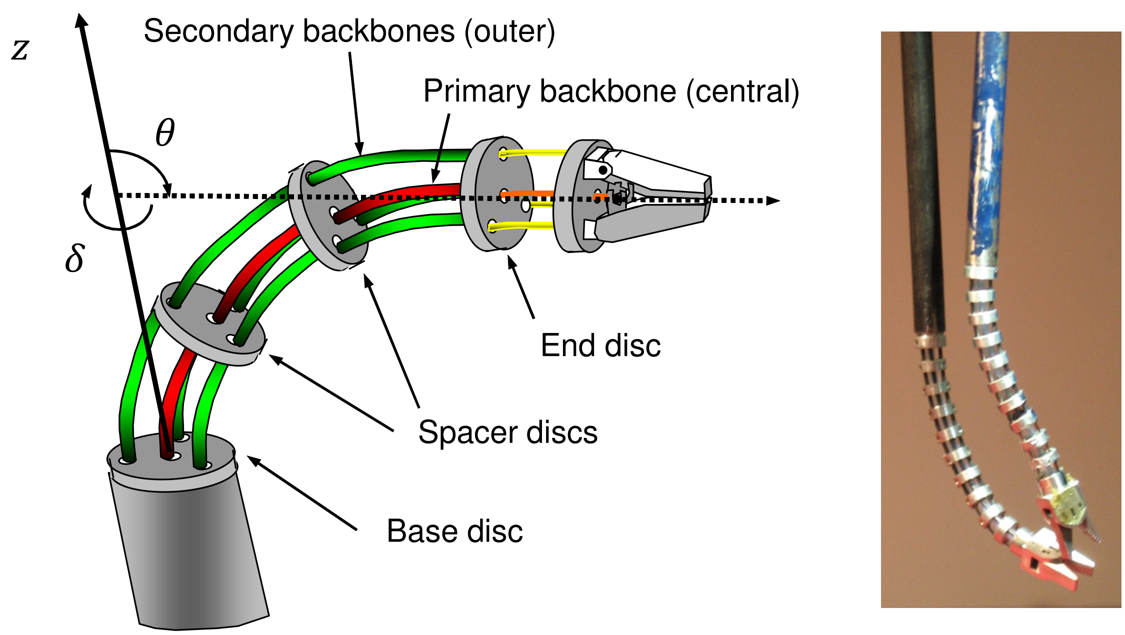


Figure A.3: Anatomy of a snake-like unit (*left*) and photo (*right*) (*Credit: A. Kapoor*).

Two degrees of freedom result from pushing and pulling the secondary (outer) backbones using three actuators located proximally. The push-pull actuation modes help prevent the backbones from buckling while satisfying structural statics [130].

The second 2-dof SLU is appended to the end of the first SLU. The backbones of this second SLU pass through the hollow backbones of the first SLU. Attached to the end of the second SLU is a gripper that is actuated via a wire passed through the hollow central backbone. This mechanical composition allows for a high payload capacity with a small size [131]. The existing SLU prototypes are 4.2 mm in diameter.

The aforementioned actuators (seven actuators for four dof and a gripper) are encased atop the shaft in a compact cylindrical actuation unit (shown in Figure A.4). As mentioned, the shaft-actuation unit assembly is also teleoperable, with four dof via four actuators, leading to eight dof total for the robot. The ZΘ stage allows for translation along and rotation about the shaft respectively. Finally, a passive universal joint mounted on a five bar mechanism gives the shaft XY mobility. This stage also stabilizes the shaft against lateral perturbations.

Each Snake Robot is actuated by 11 dc motors, accounted for by two three-axis SLUs, a gripper, a two-axis ZΘ stage, and a two-axis five bar mechanism. In our setup, two seven-axis da Vinci masters are used to command two Snake slaves for bimanual control. The pair of da Vinci masters we are using is an engineering version that did not originally include a controller.

A.1.3 Scalable Controller for the Snake Robot

All told, the low-level controller for the aforementioned prototype must be scalable to handle at least 36 axes (i.e., two 11-axis Snakes Robots and two seven-axis

Figure A.4: Snake Robot actuation unit.

da Vinci masters). A significant reduction in dimensionality and hardware complexity was achieved via the push-pull actuation of flexible wires combined with derived kinematics [1], as opposed to actuation of several precision joints. Nevertheless, the number of control axes for the Snake Robot remains considerable, as does its associated cabling. One can envision the increases in both as systems are devised for more sophisticated surgical tasks. By distributing I/O over an IEEE 1394 (FireWire) bus to nodes with low-latency field-programmable gate arrays (FPGAs), and by centralizing processing, the control system hardware proposed here aims to mitigate the potential problems in robot mobility and reliability that arise with increasing structural complexity.

The efforts presented here and in [122–124] originate in part from the need to improve the old multi-axis controller (Figure A.5) [130] of the Snake Robot and replicate the controller for other research projects. The old controller utilizes a centralized I/O arrangement, whereby command and feedback signals are transmitted in raw analog form over long cables running between the robot and the computer. The I/O devices reside on custom circuit boards, which in turn are directly attached to the computer via its ISA bus. Though the design is conceptually straightforward, the cumbersome wiring associated with it introduces complications such as noise, cable drag, reduced reliability, and greater construction effort. The debug space is vast as there are many candidates for connectivity problems, so this approach limits the ability to develop increasingly dexterous surgical robots. Figure A.6 is a photo of the Snake Robot that

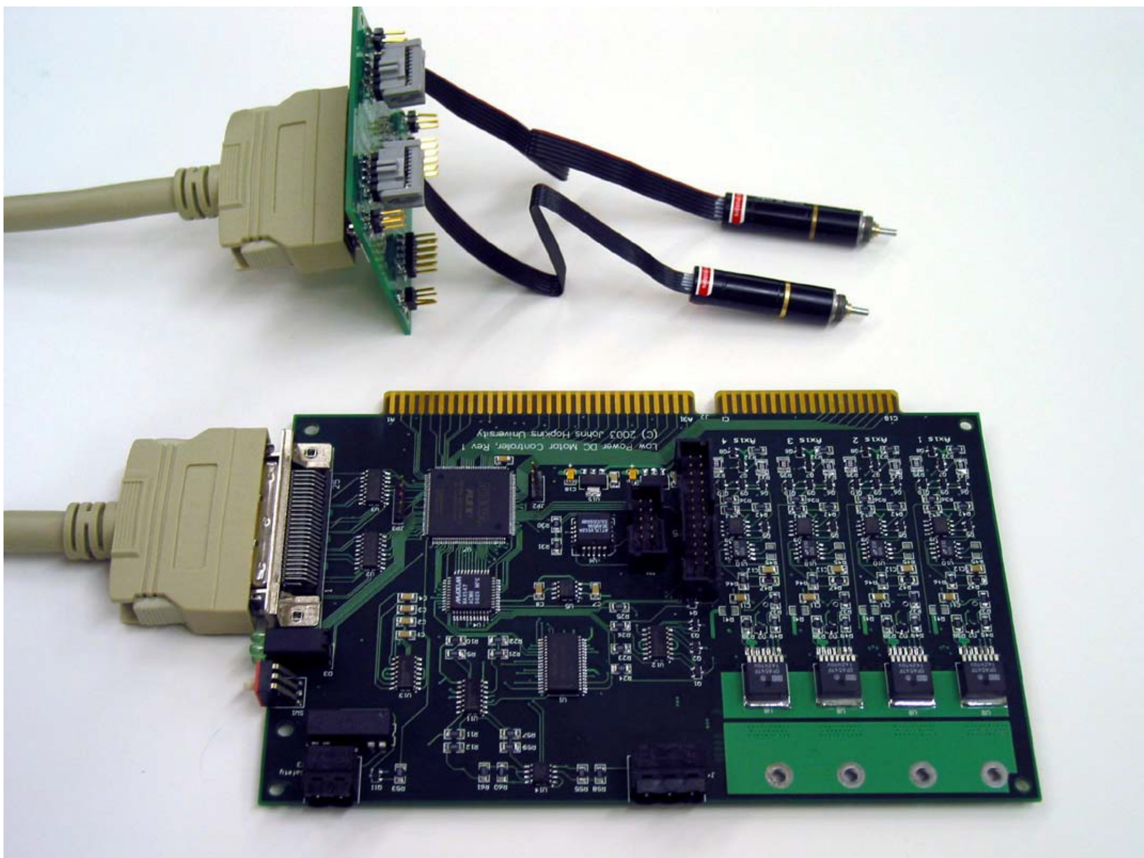


Figure A.5: Original ISA-based Snake Robot motor controller (*Credit: A. Kapoor*).

captures this scenario.

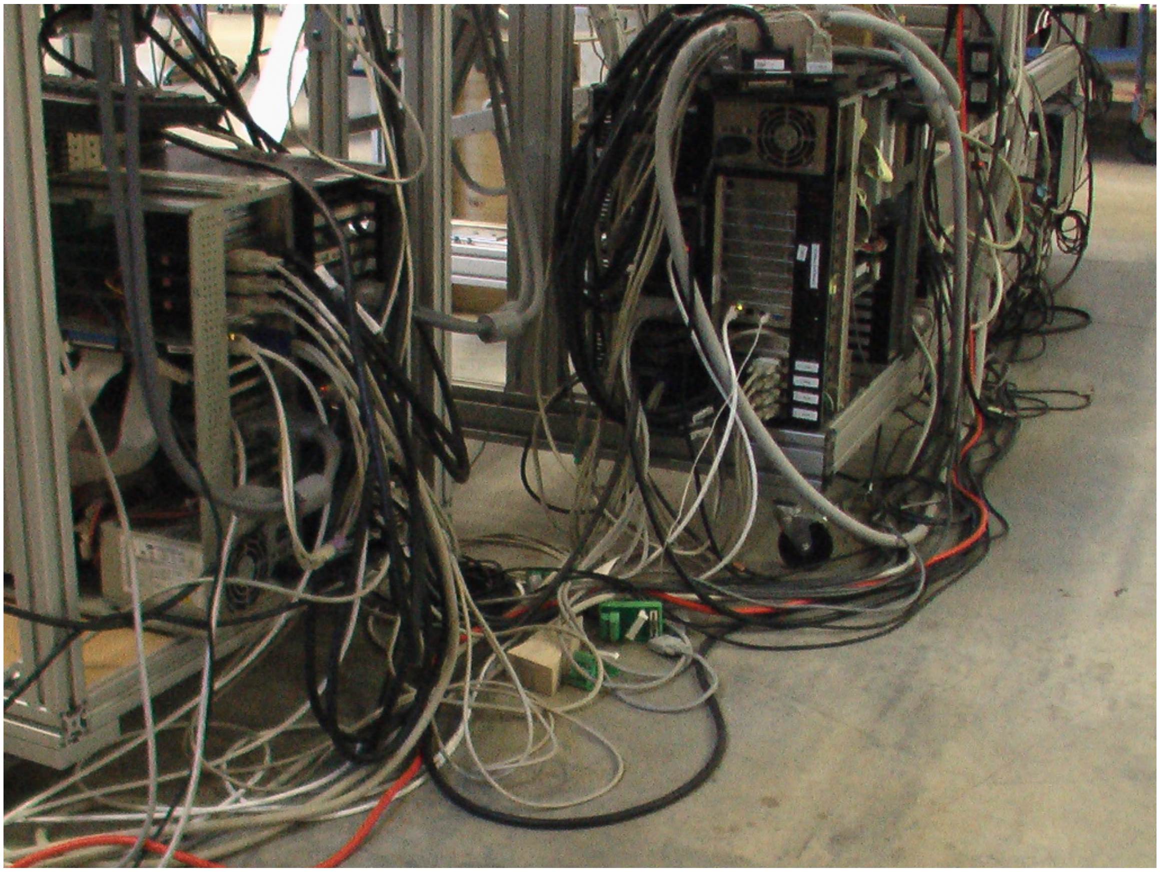


Figure A.6: Original Snake Robot cabling with centralized I/O.

A.1.4 Motivations

The long term benefits of developing a control system using IEEE 1394 are multifold. The high speed serial bus encourages the distributed I/O and centralized processing architecture. One advantage of this approach is that the I/O processing logic is simple and requires little maintenance. Signal integrity is improved because digitization occurs near the actuators, reducing the potential for noise corruption.

Cable complexity is greatly reduced because distributed I/O hardware is accessed through a serial link. This has several benefits in educational and research environments. Less effort is required in creating cables and breakout boards when new robots are developed. Since I/O hardware is replicated, standard wiring conventions are enforced. Robustness is improved overall because there is less potential for wiring problems and less cable drag affecting mobility.

Parallel buses limit the number of I/O channels that can be connected to one computer. Using the old ISA-based controller as an example, an industrial-grade computer can reliably drive four ISA cards, and the number of channels per card is constrained to four by physical size. As a result, a teleoperated dexterous robot system may need multiple computers to run.

In contrast, IEEE 1394 allows for centralized processing of a large number of multiplexed channels on a single centralized computer, so low-latency local data exchange can be used instead of network communication. The integration allows for a familiar software development environment and a standard API; this alleviates researchers and programmers from learning the idiosyncrasies of individual embedded microcontrollers, so they can instead focus on higher-level tasks. Furthermore, this architecture can more readily harness the power of high performance computers.

This work is also intended to facilitate further research by providing a generic interface and scalable mechanism for fine-grain real-time control. Such a custom solution was necessary for servoing the low power dc motors of the Snake Robot [130] because an adequate commercial solution was not available. The solution allows for flexible customization of parameters, investigation of complex control laws, and high-density distributed I/O. It is designed to ease the development of dexterous surgical robots from both hardware and software perspectives.

A.1.5 Proposed Solution

Historically, the architecture of robot controllers has been dictated by technology constraints. When computers and networks were slow, it was necessary to logically distribute the computation but to physically centralize both the computation and I/O. This is represented by a controller comprised of a large rack of embedded processors, with cables to all robot sensors and actuators. As technology improved, it became feasible to distribute both computation and I/O, as illustrated by systems that use a field bus to connect a central controller to low-level processors embedded near or within the robot.

A traditional solution to robot control comprises multiple joint-level control boards housed in a central computer through a high-performance parallel bus, such as ISA, Q-Bus, Multibus, or VME. Under this approach, however, the requisite cabling and control processing for surgical robots can become unwieldy as dexterity is increased, due to the increasing degrees of freedom. This can be a prohibitive factor for medical robot research, and furthermore does not extend well to increases in the number of joints.

Motivated by the dexterous Snake Robot, we have developed a real-time low-level control system that takes advantage of a high-speed serial bus, IEEE 1394, and capitalizes on the processing power of contemporary computers. Using a low-latency field-programmable gate array as the link between the computer and a possibly large quantity of I/O devices, all processing tasks are performed on a single computer while I/O tasks are distributed. Centralized processing simplifies software development, and a standard API is developed to augment this feature. Meanwhile, distributed I/O helps cleanly confine most wires to local joint sites thereby substantially reducing complex cabling, a high-risk source of failures. This architecture enables the exploration of complex surgical systems and technologies by allowing for more agility, reliability, and scalability.

We have thus far equated the dexterity of a robot with its degrees of freedom. Though this is not always the case, the level of scalability being sought ultimately requires a proportional increase in degrees of freedom. From the perspective of the low-level controller, there is little distinction between the concepts of degrees of freedom, joints, actuators, axes, and I/O channels because they can all be decomposed into I/O signals. Thus in this paper we sometimes use these terms interchangeably.

A.2 Related Work

Though based on IEEE 1394, the works of [159] and [160] focus on real-time control bandwidth, and not on the physical benefits of distributed I/O and centralized processing. The differences manifest in their use of IEEE 1394 as a link to an onboard computer, contrasting with our use of compact custom electronics. Our work is most similar to that of [161], where custom FPGA-based I/O boards communicate with the computer over IEEE 1394. The bandwidth was sufficient for at least six (possibly 12) dof to be updated at 1 kHz, with unit delay latency. On the other hand, this study emphasizes the physical benefits of the architecture, performance, and scalability. Ref [162] notes that using IEEE 1394 for high bandwidth PET scan data acquisition is viable due to the availability of powerful commodity computers. We agree in principle, though our respective applications are fundamentally different.

A.2.1 Ethernet-Based Alternatives to IEEE 1394

Fair bus access is incorporated into IEEE 1394 hardware; bus arbitration in Ethernet is nondeterministic, but kilohertz-range motor control is achievable on isolated networks with software modifications [163, 164]. Several Ethernet variations have been developed that make the medium very promising. Powerlink [165] employs a bus manager that schedules 200-μs cycles of isochronous and asynchronous phases. SERCOS approached a communication bottleneck in [166] with increasing axes and cycle rates, but its recent combination with Ethernet (SERCOS-III) has endowed it with the ability to update 70 axes every 250 μs.

A relative newcomer, EtherCAT [167] is an attractive protocol in which the nodes forward and append packets on-the-fly using dedicated hardware and software, resulting in the ability to communicate with 100 axes in 100 μs; [168] is an example showing its potential.

A.2.2 Other Alternatives to IEEE 1394

Many of the themes highlighted in this paper, including distributed I/O, centralized computing, scalability, and form factor, echo those of [169], which documents the MIRO surgical robot developed by the German Aerospace Center (DLR). Scalability in the MIRO robot is aided by the use of SpaceWire, a 1 GB/s full duplex serial link with latency less than 20 μs. Whereas SpaceWire has been developed by major international space agencies for space-borne systems, we prefer IEEE 1394 as it is a more accessible protocol for research, and its performance is more than adequate for demonstrating our claims. We are particularly more interested in the software-induced latency and overcoming this latency to enhance scalability.

PCI Express is a fairly new serial interface designed to replace computer expansion buses; a cable-based standard was not fully established at the time of the designs presented in this paper. PCI Express supports real-time applications such as the industrial control example in [170]. High data rates are readily available with USB, but its reliance on the host processor for bus level tasks compromises its scalability in real-time control. Conversely, IEEE 1394 self-manages the bus at the physical layer. The Controller Area Network (CAN) [171] bus is well-suited for real-time control and has been widely used, but its bandwidth is limited to 1 Mbps. Though not a serial bus, CompactPCI is an industrial backplane interface capable of 132 MB/s throughputs, used notably in space systems by NASA in transitioning from VMEbus [172].

A.3 Selection of IEEE 1394

The desire to perform real-time robot control, at frequencies of 1–10 kHz, over a serial network leads to several key requirements. First, we note that data packets are relatively small. For example, closed-loop control of a robot joint can be accomplished with as little as one feedback position (e.g., from a pot or encoder) and one control signal (e.g., voltage or current to apply to the motor). A more generous setup may contain a few feedback signals (e.g., position, velocity, motor current, amplifier status) and a few control signals (e.g., voltage and current limit). Even if 32-bit values are used for many of these, a typical data packet (read or write) would be on the order of 100 bytes. Thus, a six-joint robot would require about 1200 bytes (6×200); at a control frequency of 10 kHz, this would require a bus bandwidth of 12 MBps, or approximately 100 Mbps. This is not difficult to achieve with modern high-speed serial networks, such as IEEE 1394 (up to 400 or 800 Mbps for 1394a or 1394b, respectively), USB 2.0 (up to 480 Mbps) or Ethernet (10, 100, or 1000 Mbps).

A more critical performance metric is the latency of the data transfers because it introduces a time delay in the control computations, which compromises performance and can lead to instability. Latency is primarily determined by overhead in the protocol and the software drivers. Based on our review of specifications and published reports, we concluded that IEEE 1394 should provide the lowest latency, especially when used with a real-time operating system. The protocol supports real-time communication with guaranteed 8 kHz (125 μs) bus cycles in isochronous mode, though asynchronous mode is used instead as it allows for even faster access rates. It is an effective solution for real-time control, as shown in [159, 160], and by its use in fly-by-wire systems [173]. In the present work, the minimum requirement is to support all of the Snake Robot I/O lines with a system bandwidth of one kilohertz.

Another important requirement is the ability to daisy-chain nodes. For example, if a multi-axis robot contains embedded I/O boards, daisy-chaining allows just a single network cable to be connected to the robot—this cable connects to the first I/O board, which then connects (daisy-chains) to the second board, and so on. This allows a significant cabling reduction compared to connecting a separate network cable to each board (i.e., a star topology). Physically, all of the considered serial networks (except the outdated 10 Mbit Ethernet with coaxial cable) utilize point-to-point links but provide daisy-chaining solutions. IEEE 1394 provides an especially attractive and inexpensive solution by providing repeaters at the physical layer. In contrast, USB requires a hub (with associated cost and complexity) and Ethernet typically uses high-speed switches.

A potential drawback of IEEE 1394 is the lack of high-flexibility cables for installation within the moving structure of a robot arm. In our application, this is not a serious limitation because medical robots move slowly and infrequently compared to industrial robots. Currently, the Ethernet variations described above provide better cabling options. Ethernet, as well as USB, also have the advantage of market dominance. Although we concluded that standard Ethernet was not ideal for real-time control, we did not consider the many "real-time" Ethernet variations that have been created. Many of these Ethernet variations also support daisy-chaining without the use of switches. Given the aforementioned considerations, we ultimately selected IEEE 1394 while noting that it is not necessarily the single best choice. We furthermore chose 1394a, rather than the faster 1394b, because it provided ample bandwidth and the lower signal frequencies simplified the hardware design.

At the time of our initial evaluation (ca. 2006), we did not consider PCI Express because it was limited to backplanes and circuit boards. With the recent introduction of a cable-based standard, PCI Express appears to be an attractive alternative because it would not require protocol conversion between the motherboard and peripheral devices.

A.4 Centralized Processing, Distributed I/O

Robot systems are concurrent by nature: multiple joints must be controlled simultaneously, and there is often a hierarchy of control strategies. A typical robot controller (Figure A.7) contains "loops" for servo control, supervisory (e.g., trajectory) control, and the application. In many cases, the servo loop is distributed among multiple joint-level microprocessors.

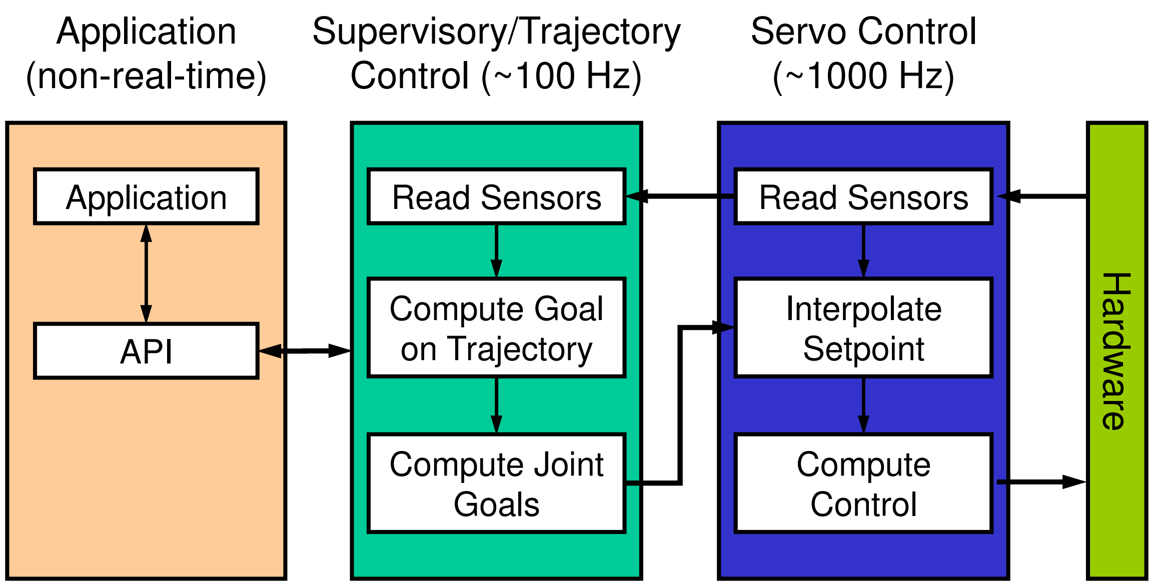


Figure A.7: Robot control hierarchy (*Credit: A. Kapoor, P. Kazanzides*).

In the early days of robotics, controllers consisted of a central computer with multiple joint-level control boards on a parallel bus, such as ISA, Q-Bus, Multibus, or VME; this was necessary for performance reasons alone. Although the central computer usually contained multiple processors (e.g., joint-level control boards), this architecture can be characterized as centralized processing and I/O.

With the emergence of high-speed serial networks, such as CAN, Ethernet, USB, and IEEE 1394, it became possible to physically distribute the joint controller boards and associated power amplifiers. By placing these components inside the robot arm, or at its base, significant reductions in cabling could be achieved. In particular, the thick cables containing multiple wires for motor power and sensor feedback could be replaced by thin network and power cables. These types of systems can be characterized as distributed processing and I/O.

Given the recent advances in processor performance, especially the move to multi-core architectures, coupled with the extraordinarily high data rates of modern serial networks, we advocate a new approach for robot control: centralized processing and distributed I/O (Figure A.8). This can be achieved by replacing the microprocessors with FPGAs that provide direct, low-latency, interfaces between the high-speed serial network and the I/O hardware. As discussed earlier, this preserves the advantages of reduced cabling, while allowing all software to be implemented on a single high-performance computer that contains a familiar software development environment, freeing developers from having to learn the idiosyncrasies of the various embedded microprocessors. Another key factor in realizing real-time centralized processing is the availability of low-cost real-time operating systems, such as those based on Linux.

A key motivation for building a novel control system is to ease the process of developing multi-axis robots in terms of both hardware and software construction.

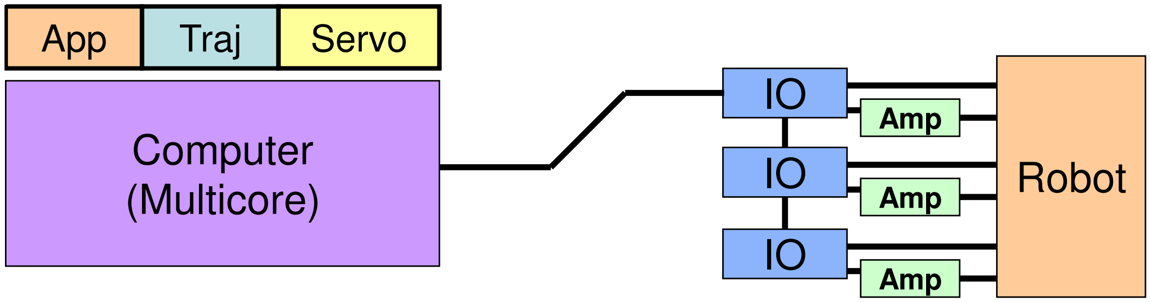


Figure A.8: Centralized processing, distributed I/O architecture.

The hardware provides fine-grain real-time control over a large number of motors, with I/O conversion tasks delegated to the actuator sites. Using IEEE 1394, the hardware confines raw analog signals to those sites and multiplexes the digital data for all channels over a high speed serial connection to a single computer.

A.5 Hardware Design

A.5.1 System Overview

Figure A.9 provides an overview of the control system, with I/O conversion distributed away from the computer to the actuator sites. Each node on the bus contains multiple channels (or axes of control, and described as the amplifier section below). Nodes can be added to the system by daisy-chaining or by direct connection to the computer—each node contains two IEEE 1394 ports for this reason. The bus is attached to a real-time computer that reads feedback signals from the channels, generates actuation commands, and writes them to their respective channels. The completed controller hardware for the Snake Robot is shown in the photo (Figure A.10). Note that this particular hardware has been designed to physically integrate on the top of the Snake Robot actuation unit; other form factors have been developed for general use.

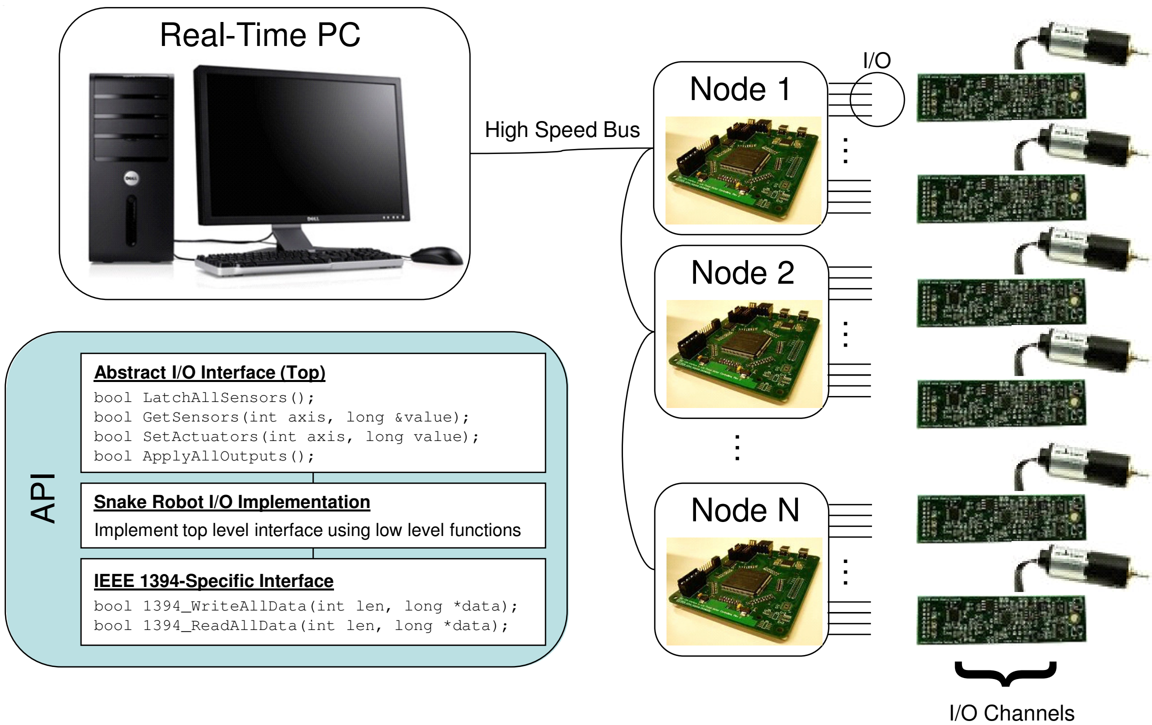


Figure A.9: Conceptual overview of the control system.

A.5.2 Amplifier Section

Depicted in Figures A.11 and A.12, the amplifier section contains the power amplification and I/O components required to control one dc motor. It is referred to as an I/O channel from the perspective of the digital section, which is detailed in the

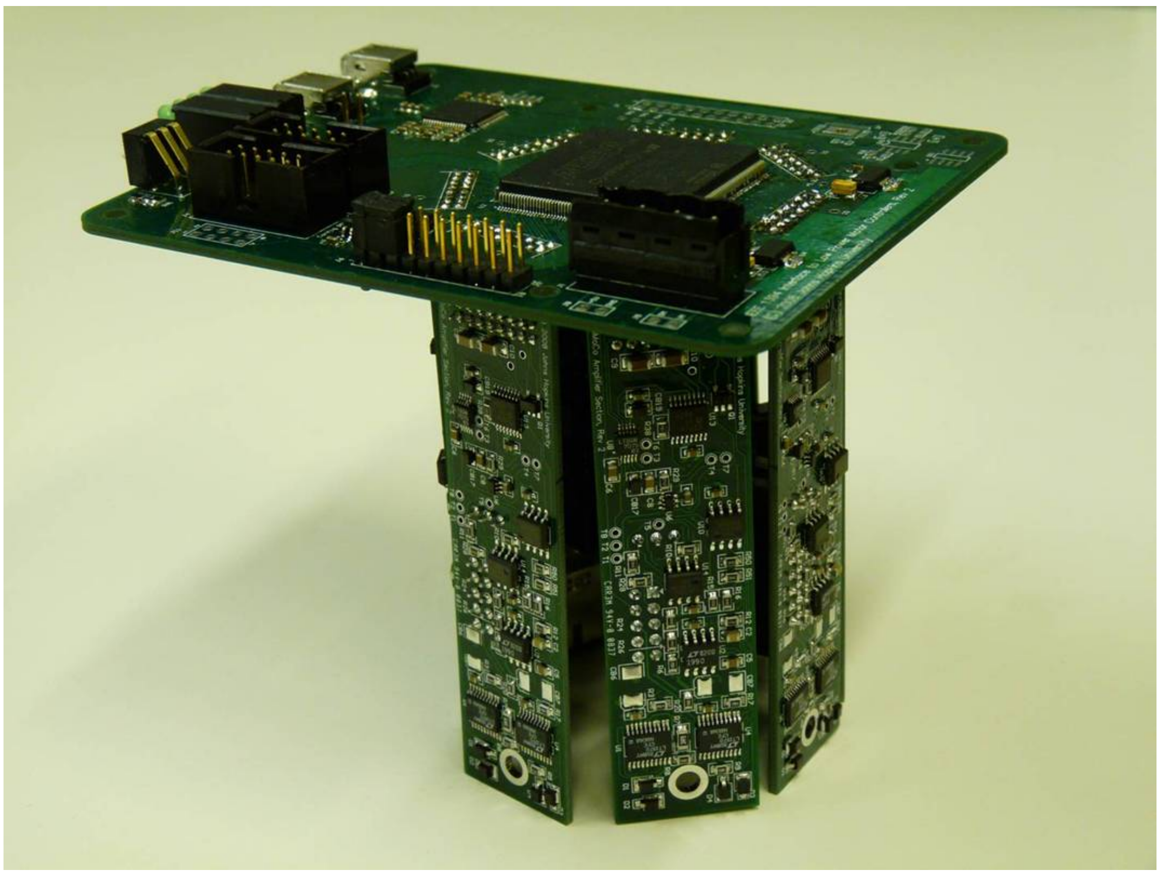


Figure A.10: Photo of the completed controller hardware (*Credit: Z. Sun*).

next subsection. Among the various I/O signals, one is used to enable or disable the motor, and one is used to signal an op amp fault, e.g., due to overheating. The power amplifier is implemented as bridge amplifier [174]; two op amp devices are configured in such a way as to require only one motor power supply. The midpoint voltage of this supply, denoted VM/2, is used as the zero level of the motor. That is, to stop the motor, both the positive and negative terminals of the motor are set to VM/2. Depending on the desired magnitude and direction of rotation, the motor command sets the master op amp voltage to some value above or below VM/2. In turn, the slave op amp voltage is driven an equal magnitude but opposite direction away from VM/2, simulating the presence of a positive and negative voltage source. Extensive tests have shown some asymmetric behavior and saturation at the supply voltage rails [175], but we expect to overcome this issue in the closed PID loop.

Other fundamental I/O components include an analog-to-digital converter (ADC), a digital-to-analog converter (DAC), and a digital potentiometer. The dual-channel ADC digitizes analog feedback, namely the potentiometer voltage that measures absolute actuator position, and the motor current, which can be used to determine torque. The DAC provides voltage commands for the motor speed, as well as the motor current limit signal. The board outputs incremental encoder pulses, properly level translated to ensure that the FPGA input voltage limit (3.3 V) is satisfied. The FPGA decodes the quadrature encoder pulses to compute motor displacement and velocity measurements.

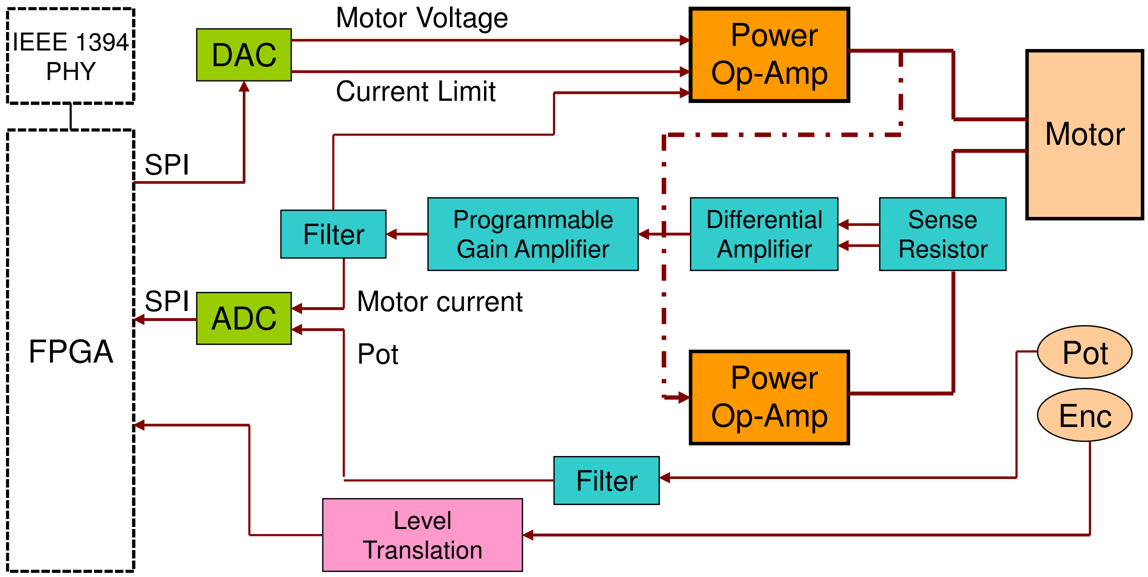


Figure A.11: Block diagram of the amplifier section (FPGA and PHY chip also shown) (*Credit: P. Kazanzides*).

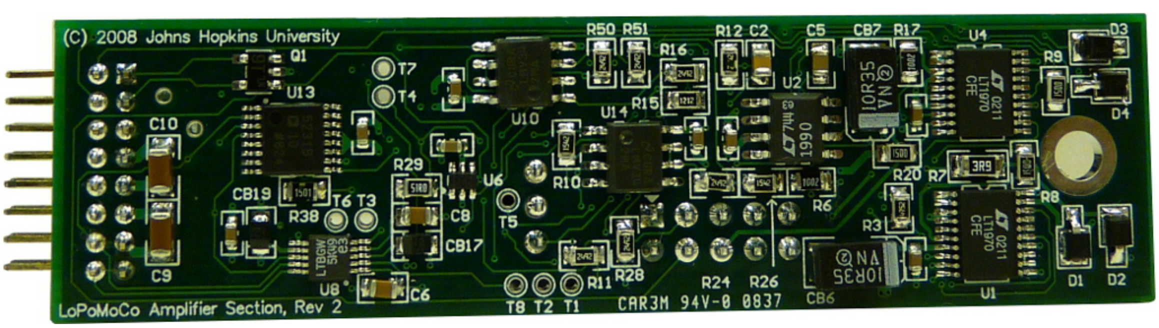


Figure A.12: Photo of the completed amplifier section.

The digital potentiometer is used to program the gain on the motor current feedback from the power amplifier. The intent is to pretune the gain for a specific motor and store that setting on its nonvolatile, 14-word, 16-bit EEPROM for operational use. Calibration procedures and measurement results are documented in [175]. In addition to a programmable wiper and EEPROM, the digital potentiometer also contains two programmable digital outputs. One of these, labeled O1, is used to select between control modes. Setting this output low selects torque control, in which case current feedback into the motor amplifier is shut off in order to maintain constant motor torque. A high setting selects speed control; motor current is fed back into the amplifier to maintain constant speed.

The ability to program the amplifier settings allows for adaptation to other robots with low-power motors. The hardware modifications required, e.g., changing the value of the motor current sense resistor, are of minimal difficulty and cost. This programmability, combined with the current and anticipated need for many such boards, serves as motivation towards a custom solution. The distributed I/O architecture encourages modular one-board-per-motor design, making the assembly compact and the boards readily replaceable.

A.5.3 Digital Section

The digital section (Figure A.13), which we loosely refer to as a node following IEEE 1394 parlance, contains circuitry for accessing the I/O channels (amplifier sec-

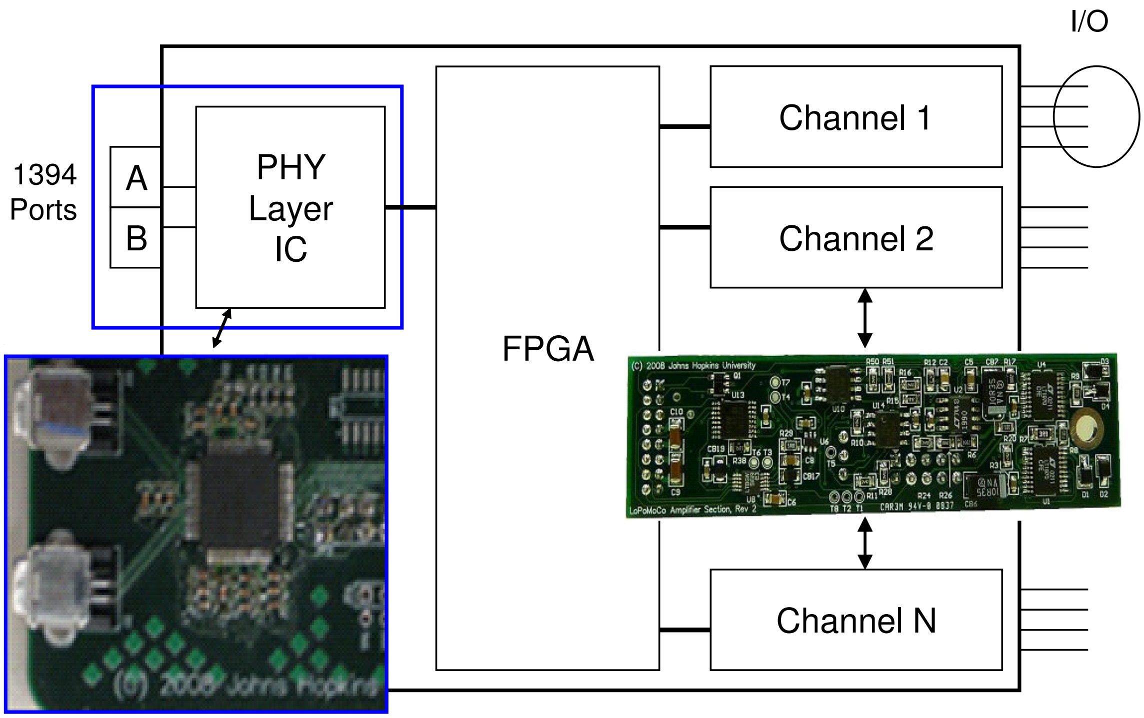


Figure A.13: Block diagram of a node.

tions) and handling bus transactions. IEEE 1394 allows up to 63 nodes per bus; each Snake Robot requires two nodes, one for the actuation unit and the other for the ZΘ and five bar stages combined—a tally of four nodes for two Snake Robots. Multiple buses can be used for yet larger numbers of axes or for heterogeneous control environments. Though not necessary for the Snake Robot, improved bus topologies over a daisy-chain configuration can be achieved by including greater than two ports per node.

Most of the functionality of each node is implemented as firmware on the FPGA, which serves as a low-latency interface between the channel I/O ports and the bus.

The FPGA receives packets from the bus, responds to them, and communicates with the I/O devices. The computer accesses channel data through control and data registers on the FPGA. The maximum number of channels accessible on one node is governed in large part by the characteristics of the FPGA, such as the number of logic elements, number of general purpose I/O pins, and speed. This design uses the Altera Cyclone II EP2C8Q208C7N [176], a low-cost model with 8256 logic elements and 138 I/O pins. It is capable of clock speeds of up to 260 MHz, though this depends heavily on firmware complexity. Under our current firmware implementation, the Cyclone II is comfortably capable of handling the seven channels of a Snake Robot actuation unit at about double the required 49.152 MHz clock frequency and 60% resource utilization. Several I/O pins remained as spares and are routed out to one of several header pins or test points. Nevertheless, more powerful FPGAs are favorable, and subsequent designs utilize higher-end devices. The FPGA is programmed through a JTAG connector during development, and a separate connector is used for storing completed firmware into a configuration PROM. Specifications allow for these connectors to be merged; though we did not do this initially due to a conservative design approach, the feature is exploited in our later work.

The Texas Instruments TSB41AB2 is a two-port IEEE 1394a physical layer integrated circuit that converts between digital data and the analog signals used to convey them at high speed (400 Mbps) over the bus. The FPGA sends and receives data over the bus by interacting with this device, specifically via control lines and an eight-bit data bus. The device also generates a 49.152 MHz clock, derived from a 24.576 MHz crystal, which the FPGA uses as its system clock. In this way all communications between the various I/O devices and the physical layer chip are synchronized. Due to the relatively high speeds that the device is capable of, careful attention was paid to its layout on the PCB, particularly around the impedance-controlled differential analog lines, termination resistors, clock I/Os, and bypass capacitors.

Also included in the digital section is a four-bit rotary switch used to uniquely identify a node to control software. Although nodes are uniquely identified via the IEEE 1394 protocol, the address assignment is dynamic. An array of LEDs is used to indicate the status of each of the seven amplifiers; a lit LED signifies that the corresponding amplifier is enabled, while an unlit LED means that the amplifier is disabled, possibly due to assertion of its fault line.

For noise isolation, and to facilitate emergency shutdowns, the motor and digital voltages are drawn from separate regulated supplies. Power from the bus is not used for these reasons, as well as to simplify the development effort, but future designs may draw bus power.

A.5.4 Implementation

We used an FPGA development board (Altera UP3, Figure A.14) for the first prototype to reduce our development risk. Realistically, we expected that our system would not work the first time we connected a computer to the prototype node. The development board eliminated the possibility that we did not properly design or fabricate the FPGA portion of the prototype. Thus, we could focus our debugging effort on the FPGA code and on the custom daughterboard (Figure A.15) that contained the IEEE 1394 physical layer chip.

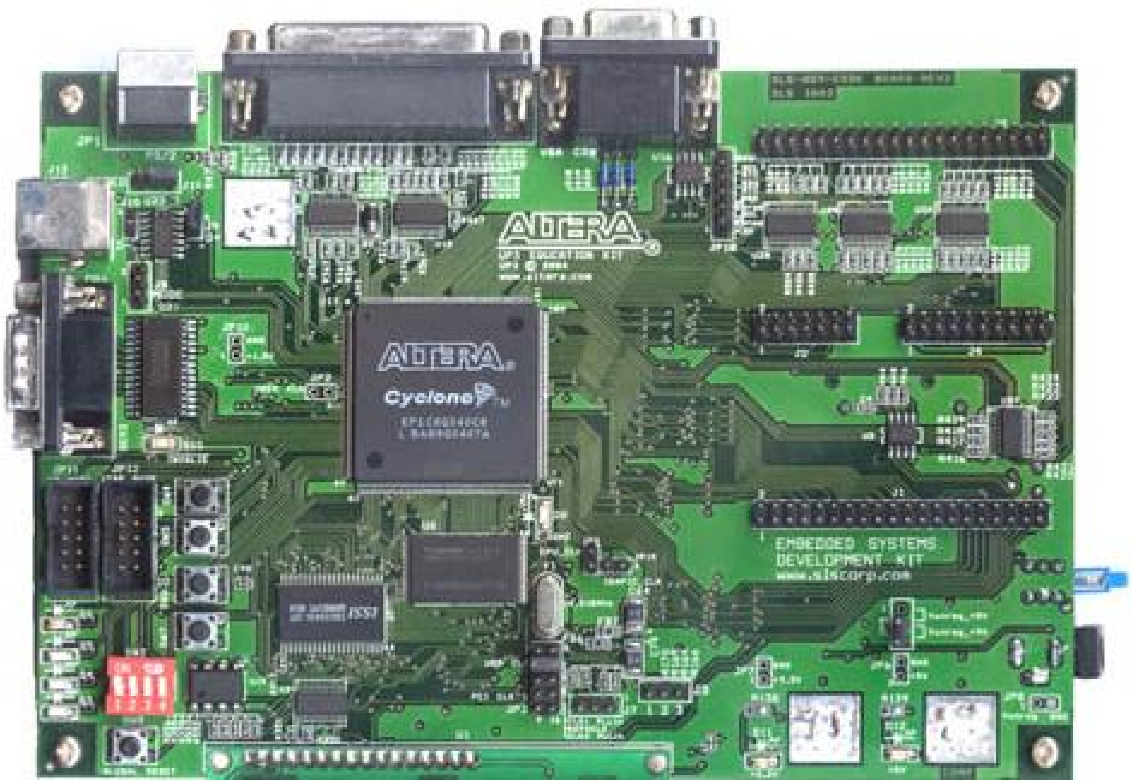


Figure A.14: Altera UP3 FPGA development board (`http://users.ece.gatech.edu/hamblen/UP3/`).

The design files for this initial phase are designated as Revision 1 ("-rev1"). This set consists of the daughterboard, the firmware for the FPGA development board, and the first revision of the amplifier section. The final revisions discussed in this paper are labeled as Revision 2 and include the complete digital section, its firmware, and

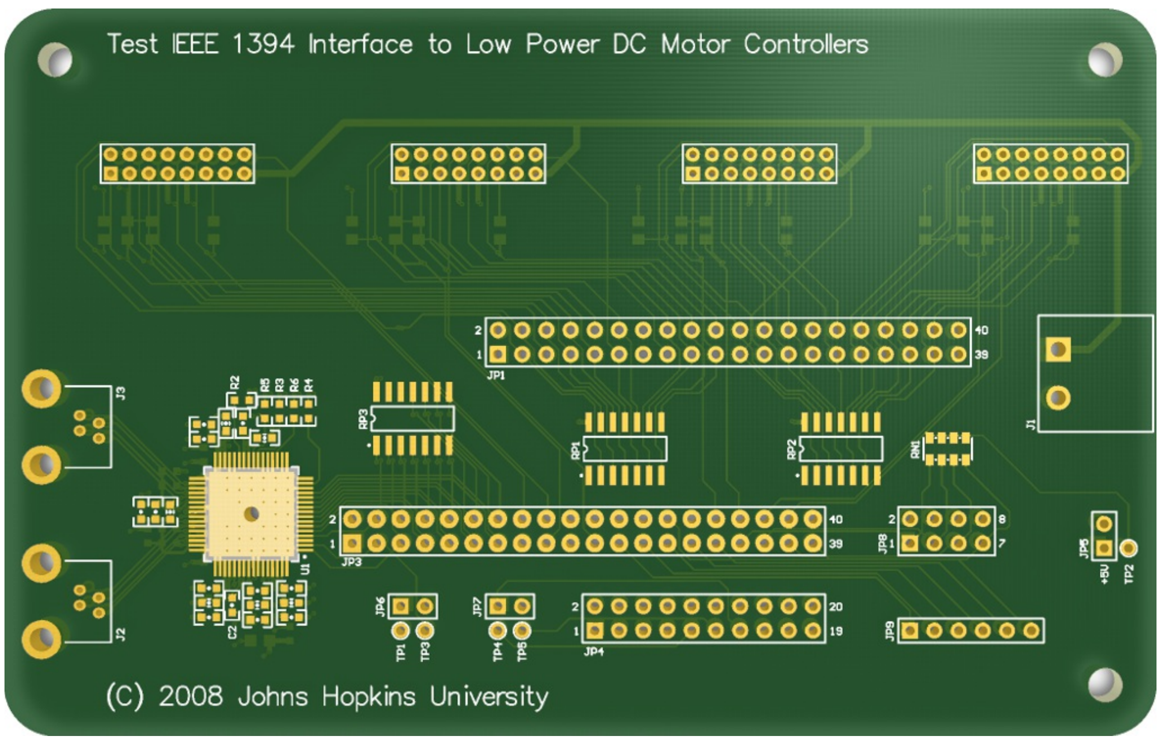


Figure A.15: Custom daughterboard containing IEEE 1394 physical layer chip.

a revised amplifier section. Revision 2 of the amplifier section represents extensive repairs and improvements over its predecessor.

A.6 Firmware Section (FPGA)

A.6.1 IEEE 1394 State Machine

Figure A.16 shows the IEEE 1394 state machine, illustrating the top level of the FPGA operation. It describes how the FPGA handles bus transactions and triggers activity on the rest of the board based on such events. As the FPGA plays a passive role in the system, action is initiated when the computer requests a data transaction, for instance a sensor read or an actuator write. Once the FPGA on the addressed node receives the request, it responds immediately with an acknowledgment, as required by the IEEE 1394 protocol. In the case of read requests, the FPGA then fetches data from an intermediate buffer and sends them to the computer with a timestamp, indicating the number of 49.152 MHz clock cycles that have elapsed since the previous read request. As described in further detail below, the contents of the intermediate read buffer are refreshed continuously to preserve real-time performance. For write requests, the FPGA loads the appropriate buffers and triggers the corresponding channel I/O devices, which in turn access the buffers for their data.

In accordance with the IEEE 1394 protocol, the FPGA firmware verifies incoming CRC values and appends CRC values to outgoing packets. Checksum failure is

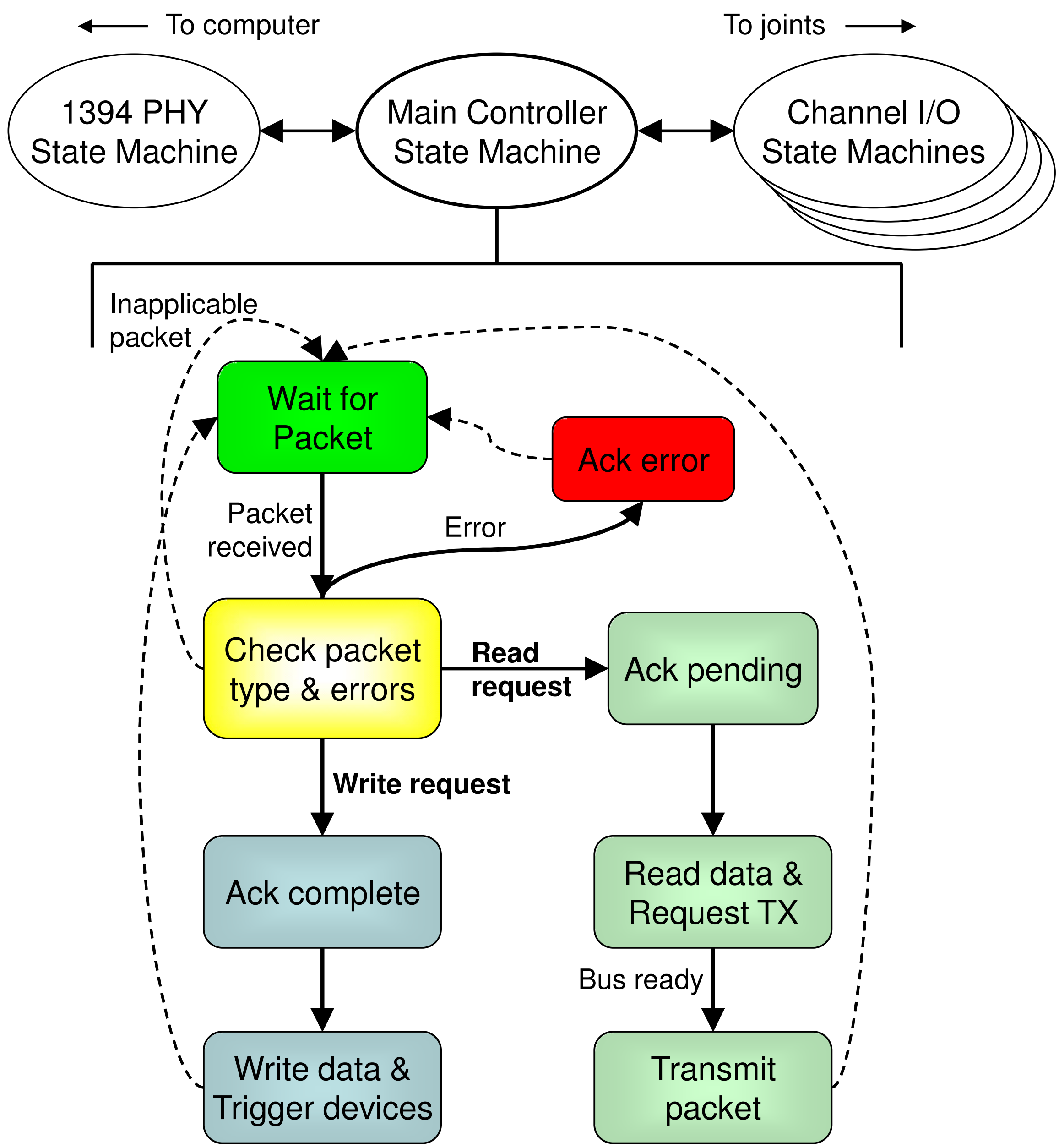


Figure A.16: FPGA structure and operation.

expected to be an extremely rare occurrence, so given the real-time nature of the system, the defined behavior is to silently drop these packets. For simplicity of implementation, and to minimize FPGA resource consumption, only a subset of the IEEE 1394 protocol is supported. Recognized transaction types as of this writing include quadlet read, quadlet write, block read, and block write. In particular, reads are of the concatenated variety, meaning that once a node receives a read request, it responds immediately without relinquishing control of the bus. This is in contrast to split read transactions, where a node may release the bus and send the response at a later time. Writes are unified; that is, the FPGA acknowledgment to a write request also indicates that the transaction is complete. This contrasts with a protocol provision that allows for a delayed response, in case a checksum failure is detected, for example. Finally, attempts have been made in the firmware implementation to allow the FPGA to accept block transactions of any size. While such an effort may be helpful for adapting the firmware to other boards, for any specific board the behavior is ultimately interwoven with the number of I/O signals per channel and number of channels on the board. Thus only specific block sizes should be requested for proper functionality.

A.6.2 Channel Modules

As detailed in the description of the amplifier section, each channel contains one control axis worth of I/O, namely an ADC, DAC, digital potentiometer, encoder lines, amplifier enable line, and amplifier status line. The FPGA interacts with the ADC, DAC, and digital potentiometer via the Serial Peripheral Interface (SPI). The encoder signals originate as an analog voltage, but arrive at the FPGA in digital form after level translation in the amplifier section. Counter logic on the FPGA converts the incremental encoder pulses into motor position and velocity measurements. The enable and status lines are simple digital signals.

Communication with these I/O devices is performed in parallel with the IEEE 1394 state machine described above. The read (input) devices, namely the ADC and encoder, run continuously and transfer their data into FPGA buffers as soon as conversion or computation results are complete. The ADC is utilized in this manner because its conversion time, roughly 2 μs, is long compared to the bus transaction time, so it is not advisable to start a new conversion and wait for the result whenever a read request is received. This implies that an ADC conversion result can be up to 2 μs stale, but this is negligible relative to the Snake Robot servo frequency of 1 kHz (1 ms). The write (output) devices, namely the DAC and digital potentiometer, wait idly for write requests, transfer write data from FPGA buffers into scratch registers when a request arrives, and perform the operation. The digital potentiometer functions as both a read and a write device, though it is not read continuously.

While each SPI device is run in separate submodules, all of the I/O devices for a channel are collectively managed by a higher level state machine. This larger module is replicated for each amplifier section and runs independently of any others. It is responsible for arbitrating the shared SPI lines of each channel's respective ADC, DAC, and digital potentiometer. The ADC and DAC accesses can run simultaneously because even though they share some SPI signals, their data move in different directions (the former on the SDO line and the latter on SDI) and thus do not conflict. This is a valuable property because both devices are involved in real-time operation. Since the digital potentiometer is both a read and a write device, however, its accesses must be handled as a special case. A read request from the computer targeting the digital potentiometer will instruct the FPGA to pause ADC conversions, read the digital potentiometer data into an FPGA register, and then resume ADC conversions. The computer then sends a read request for the stored value. A similar procedure is followed for digital potentiometer write requests by pausing the DAC temporarily. This extra complexity does not affect real-time performance because digital potentiometer accesses should be a relatively rare event.

Another special case is establishing the motor current limit through the DAC. Because the same dual-channel DAC is also used for setting the motor command, both channels cannot be written simultaneously, and thus must be performed in separate transactions. While such a limitation would seem to compromise real-time performance, setting of the current limit is expected to be a seldom act, for example, as an initialization step. Under normal operation only the motor command is set.

A.6.3 Global I/O Module

The amplifier enable lines and fault status flags were previously described in conjunction with their associated channels. However, in order to combine these signals with global events, they are passed on to higher level firmware for management. Introducing an enable mask allows different combinations of amplifiers to be enabled (or disabled) arbitrarily, so that it can be done in software using a single write transaction. Absent of this feature, separate transactions would be required for different amplifier enable requests. Managing the enable lines globally allows us to disable them automatically based on a watchdog timeout, discussed below. It also facilitates convenient grouping of the enable lines, status lines, watchdog timeout flag, and board identifier in order to report them as a single status register. Finally, a logic combination of enable and fault signals are used to set the LED array on the board.

The global I/O module includes an optional watchdog resource, which is activated via software by setting the desired timeout period. The value is in terms of a 5.2083333 μs (49.152/256 MHz) counter. As the counter is an unsigned 16-bit integer, the longest timeout period that can be specified is approximately 0.34 seconds. A timeout occurs when the value of the free running counter matches or exceeds that of the timeout period, at which point all amplifiers are disabled. The counter is cleared whenever a write transaction is received, so timeouts are avoided so long as write transactions are issued at a rate faster than the timeout period. Watchdog functionality is disabled by setting the timeout period to zero, the default setting.

Two seldom used features include the ability to access the registers of the IEEE 1394 physical layer chip, and a read-only version number register that is meant to be set during firmware synthesis. This latter function was quickly deprecated in favor of using software to store version numbers in the digital potentiometer EEPROM.

A.6.4 Address Maps

All of the I/O resources described thus far are directly accessible by computer software via quadlet transfers, with the desired device specified by an appropriate value in the address field of the packet header. In all cases, the FPGA checks the destination ID field in the packet header and only acts if the value matches the local node ID, which is automatically assigned by the physical layer. Furthermore, it only acts if the specified transaction code is supported. Otherwise packets are considered inapplicable to the node and are silently ignored. When active, the FPGA reads data from or writes data to intermediate buffers as described previously.

The address field for quadlet transfers is interpreted as follows: bits 7–4 address a channel, while bits 3–0 address a device on that channel. The address map is listed in Table A.1. One detail of note is that channel devices begin at channel address 1, as channel address 0 specifies the global I/O module, whose address map is also shown in the table. The address segmentation reveals our assumptions on the number of devices per channel and the number of channels per node.

Table A.1: Device map of quadlet transaction address field

Bits 7–4	**Bits 3–0**	**Description**
0 (Global Registers)	0	Status Register
	1	IEEE 1394 PHY IC Control Register
	2	IEEE 1394 PHY IC Data Register
	3	Watchdog Timeout Period Register
	4	Version Number Register (deprecated)
1–7 (Channel Select)	0	ADC Data Register
	1	DAC Control Register
	2	Digital Potentiometer Control Register
	3	Digital Potentiometer Data Register
	4	Quadrature Encoder Preload Register
	5	Quadrature Encoder Data Register
	6	Encoder Period Data Register
	7	Encoder Frequency Data Register (planned)

In contrast, the devices involved in block reads and writes are fixed according to its position in the data block. Also, only certain devices are accessible through block transfers, i.e., those necessary for real-time operation. Figure A.17 depicts the respective packet formats expected by the FPGA, as well as how it interprets the data fields. The FPGA firmware is meant to handle write blocks of a fixed size due to the simplicity and reliability of such an approach. Thus the MSB of each data quadlet is interpreted as a valid bit, which is set by software to indicate to the FPGA whether or not the corresponding device is to be written.

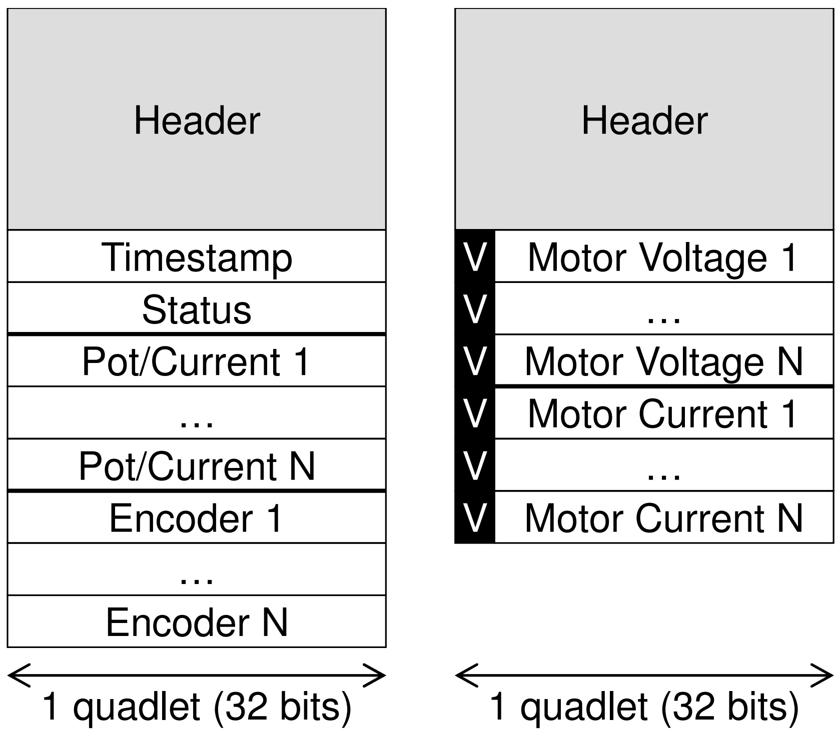


Figure A.17: Block read (*left*) and write (*right*) packet formats as interpreted by the FPGA.

A.7 Software API

A.7.1 Block Diagram

We developed a generic, intuitive API for the control hardware described above. The interface has the three-layer hierarchy shown in Figure A.18. The top layer consists of abstract I/O operations that can be used for different robots. It includes commands to latch all sensors and to apply all outputs, which are often supported by the hardware (e.g., simultaneously sampled ADCs and double-buffered DACs). All read and write operations are performed for a single axis at a time, requiring only primitive C data types (e.g., int, long).

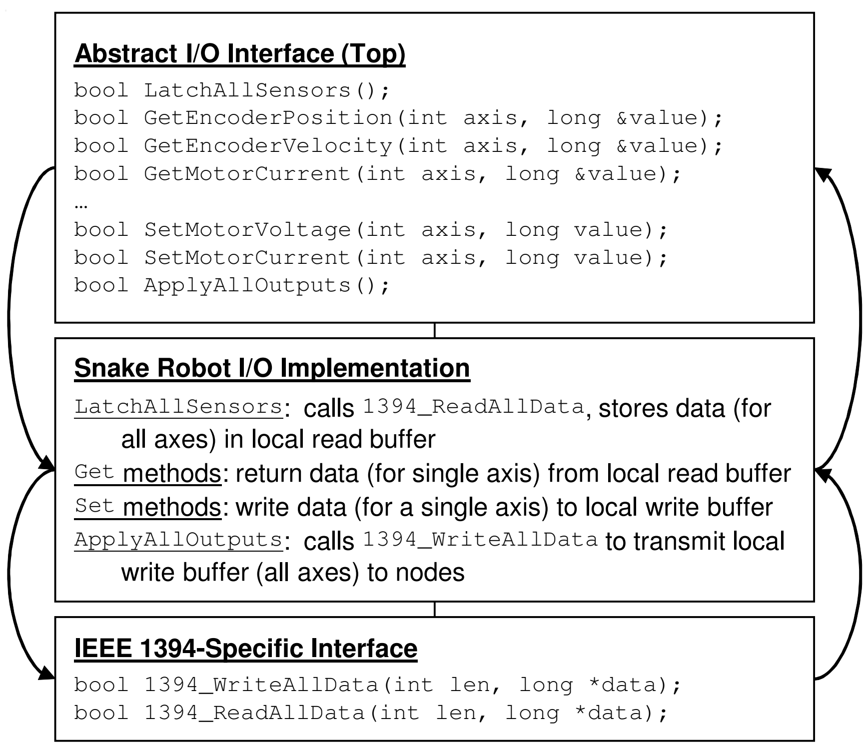


Figure A.18: General robot control API.

The second layer, which implements the abstract interface for the Snake Robot, is customized to work with data blocks, since for efficiency reasons the data for the multiple axes of a node are bundled into a single bus transaction. This layer provides access to individual axes via local buffers that are filled by `LatchAllSensors` and emptied by `ApplyAllOutputs`. So that a fixed block size can be used to simplify the FPGA implementation, the second layer maintains a valid bit for each axis that indicates to the FPGA whether or not to write the corresponding axis. The bottom layer contains function calls to the IEEE 1394 API library (libraw1394), though for real-time performance considerations RT-FireWire [160] is an alternative of interest. By carefully designing a general robot control API, the developed software can be easily maintained, and the system can be used in other surgical robots.

A.7.2 Application Programming Interface

The methods of the API are listed below. The data formats produced or expected by the hardware are diverse, as they are specific to each device. In order to maintain uniformity in the data types of the API arguments and return values, the library manipulates the data in ways that the calling code must handle properly. Some of the less straightforward considerations are detailed as follows.

The ADC, through which analog potentiometer and motor current measurements are obtained, reads voltages in the range of 0 to 2.5 V. However, it reports them as signed 14-bit integers relative to 1.25 V. To match the API convention, the library

```
// read data from given channel (channels combined for status)
unsigned long GetStatus();
unsigned long GetTimestamp();
unsigned long GetMotorCurrent(    unsigned int );
unsigned long GetAnalogPosition(  unsigned int );
unsigned long GetEncoderPosition( unsigned int );
unsigned long GetEncoderVelocity( unsigned int );
unsigned long GetDigitalPotWiper( unsigned int );
unsigned long GetAnalogSwitches(  unsigned int );
unsigned long GetEepromDataRaw(   unsigned int );
float         GetEepromDataFloat( unsigned int );

// write data for given channel (channels combined for power)
bool SetPowerControl(                  const unsigned long );
bool SetMotorVoltage(    unsigned int, const unsigned long );
bool SetMotorCurrent(    unsigned int, const unsigned long );
bool SetEncoderPreload(  unsigned int, const unsigned long );
bool SetDigitalPotWiper( unsigned int, const unsigned long );
bool SetAnalogSwitches(  unsigned int, const unsigned long );
bool SetEepromDataRaw(   unsigned int, const unsigned long );
bool SetEepromDataFloat( unsigned int, const float         );

// actual operations on buffer
bool ApplyAllOutputs();
bool LatchAllInputs();
```

converts these to unsigned 14-bit integers by adding to it the midpoint value of the 14-bit range. Stated explicitly, the ADC views the range 0 to 2.5 V as -1.25 to $+1.25$ V, and reports a corresponding value in the range -8192 to $+8191$. The library adds 8192, resulting in an API return value in the range 0 to 16383, an unsigned 14-bit integer. The caller must treat this accordingly, as the sign is related to the direction of travel.

The DAC, through which motor commands and motor current limits are programmed, outputs voltages in the range 0 to 2.5 V. It accepts as its inputs 16-bit unsigned integers, making the mapping intuitive. Note however that the direction of the motor depends on whether the DAC output is above or below the midpoint (1.25 V), and its speed depends on the absolute difference from the midpoint. In this sense the motor command can be viewed as a signed value centered about this nonzero midpoint.

The controller provides two types of encoder counts. The main type counts the number of pulses, which can be used to measure incremental movement. Because an overflow flag is asserted when the count drops below 0 or increases beyond the 24-bit counter maximum, it is important to preload the counter with the midpoint value during initialization. The secondary measurement is a 16-bit counter representing the number of 1.302083333 μs (768 kHz) clocks elapsed between encoder pulses. This is potentially useful for measuring relatively slow motor velocities, as in this regime there is a resolution problem in simply dividing the number of encoder pulses by the elapsed time. This count should be interpreted as a signed value as it starts from zero and increments or decrements depending on the motor direction.

The digital potentiometer wiper value is an unsigned 10-bit value, where 0 sets the wiper to the A terminal and the maximum value sets it to the B terminal. Given the circuit connections, this means that the motor current feedback gain increases with the wiper value according to

$$G = \frac{12.1\,k\Omega + 10\,k\Omega \times (wiper/1023)}{1.5\,k\Omega + 10\,k\Omega \times ((1023 - wiper)/1023)}. \tag{A.1}$$

Resistors designated R2 and R38 on the amplifier board have values $12.1\,k\Omega$ and $1.5\,k\Omega$ respectively. The $10\,k\Omega$ factor that appears in this formula represents the nominal resistance of the digital potentiometer. Through our tests we have found this value to vary widely by device, so gains should be tuned per board-motor pair. The gain factor for this stage is nominally 2, so a reasonable initial wiper value for the Snake Robot would be 375.

An EEPROM is embedded within each channel, so accessing it requires two tiers of addressing, one to specify the channel and the other to specify the EEPROM location within that channel. Thus the index argument for EEPROM access is interpreted slightly differently. Bits 7–4 of the index now select the channel, and bits 3–0 select the EEPROM word within that channel. In contrast, bits 3–0 select the channel for the other devices. The API also allows higher level code to store single-precision floating-point values in EEPROM. It does so by splitting the 32-bit data type into two 16-bit words at the library level, using two EEPROM locations for their storage. Note that BCD and floating-point representations occupy the same memory space.

Each block read via `LatchAllSensors` fetches, in addition to ADC and encoder data, a status and a timestamp. The status (Table A.2) is a 32-bit value that is a combination of amplifier enable bits and mask, amplifier status flags, watchdog timeout flag, and board ID (set by the onboard rotary switch). The timestamp, meanwhile, is a 32-bit value indicating the number of 49.152 MHz clock cycles elapsed since the previous `LatchAllSensors` call.

Table A.2: Status code definition

Bits 31–28	**27–24**	**23**	**22–16**	**15–8**	**7–0**
Unused(4)	Board ID(4)	WD Timeout(1)	Status(7)	EN Mask(8)	Amp EN(8)

As of this writing the API has not been tested for CISST compatibility, but integration is in progress.

A.7.3 Demonstration

A Qt-based graphical user interface program (Figure A.19) is being developed to enable testing and configuration of the various features of the board. It allows the user to control the motor voltage and current limit for each axis, as well as view their respective analog potentiometer, motor current, and encoder measurements. On the same screen the user can adjust the digital potentiometers and view the resulting

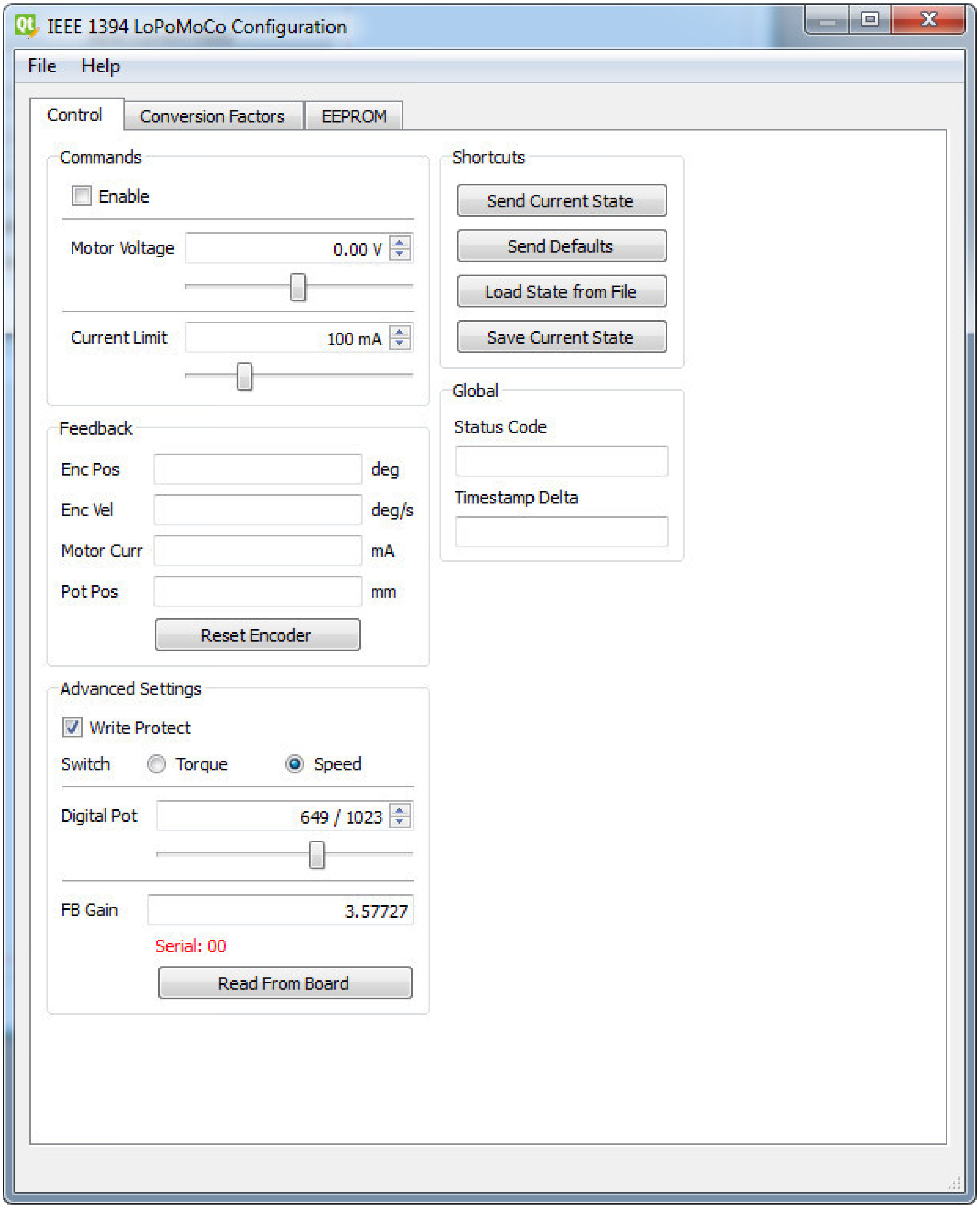


Figure A.19: Test and configuration program.

feedback gains. The GUI offers a convenient mechanism for switching between speed and torque control as well. A dedicated EEPROM tab contains controls that allow for reading and writing the EEPROM on any axis, in both binary and floating-point representations.

This demonstration program is also intended to test the API in order to expose and address any remaining issues. Finally, the program serves as a reference implementation for usage of the API. At present the code is not compatible with the CISST libraries, but integration is expected to take place in the near future. All source code, design files, and documentation referred to in this paper are stored in the LCSR Electronics repository [177].

A.8 Experiments

A.8.1 Setup and Verification

A conventional Linux desktop computer is used for the initial experiments reported below, while a real-time version of Linux, such as the Real-Time Application Interface (RTAI) and QNX, will be used to run the robot control software. The tests were run at 400 Mbps with one node connected to a 2 GHz Pentium 4 computer by a 6-foot IEEE 1394 cable.

As explained previously, the FPGA performs a read transaction as a concatenated read, which entails a request from the computer, an acknowledgment from the node, and a data response from the node (there is also a final acknowledgment from the computer to the node). There is no protocol delay (i.e., no busy wait) between receiving a read request and generating a response. Similarly, write transactions are unified, so there is no delay between the receipt of a write request and the execution of the request. Because I/O device access times are negligible relative to the system bandwidth, loopback tests of the DAC-to-ADC pair and digital I/O consistently return the appropriate values. Bus contention is not expected because (1) the computer is implemented as the bus master and the nodes as slaves; and (2) no devices unrelated to the control system will be attached to the bus.

The amount of various component-wise tests performed is voluminous and are not repeated here. Rather, the findings of these tests have been integrated into this report, while test code and their noteworthy results have been stored in the LCSR Electronics repository. The demonstration GUI presented in the previous section represents the culmination of all verification of the hardware components, API, and their integration.

A.8.2 Quadlet Transfers

Figure A.20 shows the timing results of 9,000 iterations of quadlet reads and writes. The read times appear in three principal bands (30, 42, 49 μs) and the write times in two (27, 42 μs), possibly due to variability in discrete transaction sequences (e.g., request-ack-response); the average read and write times are 34.5 and 30.2 μs respectively—each respective low band is most common. We plan to investigate these regular distribution patterns should they arise under a real-time operating system.

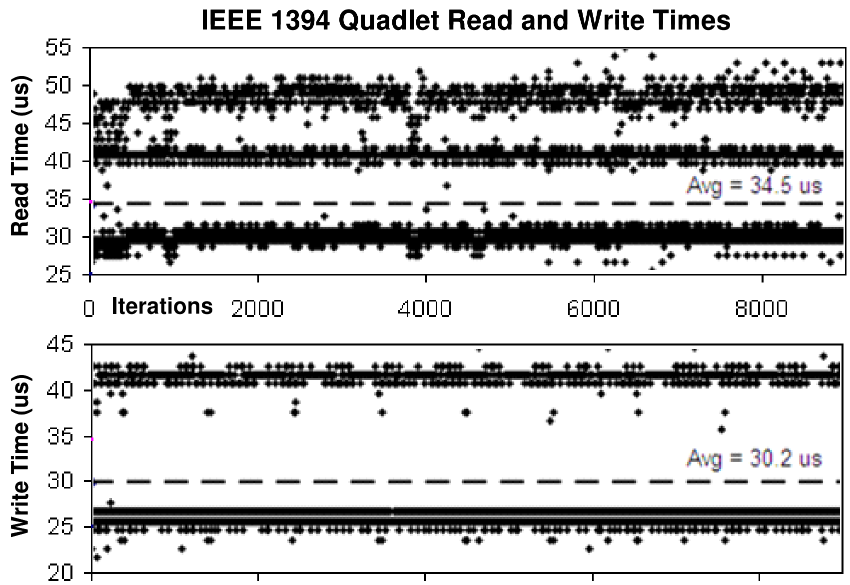


Figure A.20: IEEE 1394 quadlet read and write transaction times.

We note that there are significant latencies in the bus transactions [122–124]. In a standard servo control implementation (read-control-write) for a seven-axis robot, a combined read/write time of about 453 μs leaves only 547 μs for control computations at 1 kHz, and is not even feasible at 8 kHz. Given that the bus speed is 400 Mbps, and that the IEEE 1394 physical layer itself should not impose appreciable delays, we conclude that software overheads are a predominant factor in the latency, as in [178]. As we concluded in our previous work, it is necessary to bundle the data for multiple axes into blocks in order to overcome this limitation.

A.8.3 Block Transfers

Figure A.21 shows a raw sampling of read and write times over the full range of allowable block sizes, up to the maximum of 512 quadlets for the 400 Mbps mode. Blocks sizes are measured in quadlets, as IEEE 1394 requires packet sizes to be in multiples of 32 bits. The plots indicate base latencies of about 33.2 μs for reads and 30.7 μs for writes, which closely matches our previous findings.

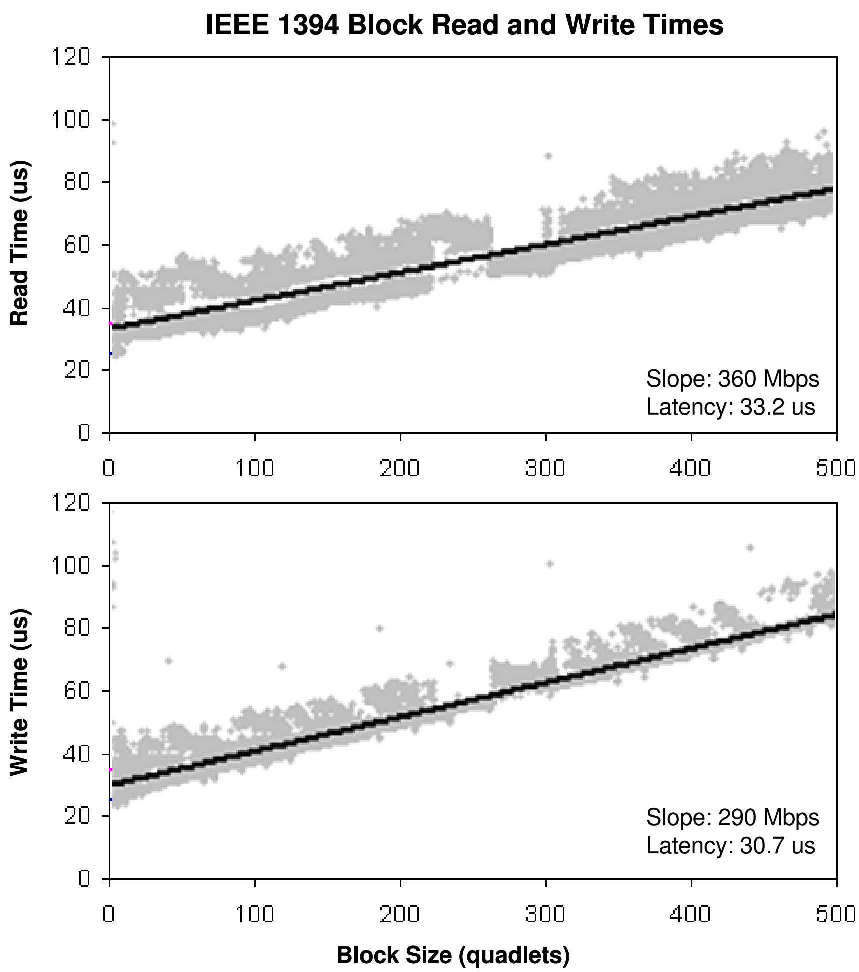


Figure A.21: IEEE 1394 transaction times vs. block size.

Based on the slopes of transaction time vs. block size, we compute average speeds of roughly 360 Mbps for block reads and 290 Mbps for block writes. Neither value reaches the nominal 400 Mbps rate due to protocol handshaking and other overheads, but it defies intuition that reads are faster than writes since the former is a slightly more complicated transaction type and incurs greater latency. A possible explanation may involve differences between how the computer and FPGA acquire bus access. We intend to resolve this anomaly in future work. At any rate, the results suggest that the number of axes can be scaled significantly with negligible incremental time consumption.

A.8.4 Discussion

Variability in the transaction times (about 20 μs, based on the noisiness of the plots) and sporadic but high-value outliers may be caused by underlying operating system activity, as we use conventional Linux for development purposes. The former observation may also be explained by variability in obtaining bus access. A real-time operating system will be used to run robot control software, and we would like to obtain performance measurements under such an environment as well.

Overall, the transaction times are well above theoretical maxima. For example, a quadlet read, 296 total bits, should only require a fraction of a microsecond to complete at 400 Mbps. We believe that software overhead is the primary source of concern in our system, and hope to reduce this in the future. Nevertheless, because software overhead should be relatively independent of data size, these results indicate that the most efficient approach is to use one block read to obtain all feedback data (from all joints serviced by the node), and one block write to send all command signals. In a standard read-control-write implementation, a combined block read/write time of about 65 μs would leave a generous 935 μs for control computations at 1 kHz, but only 60 μs at 8 kHz, and 35 μs at 10 kHz. IEEE 1394 supports isochronous transfers, which occur at a frequency of 8 kHz and may therefore be more efficient for control at this frequency.

A.9 Deployment

Note: This section presents work performed and published [126] after the original writing of this chapter.

A.9.1 Installation

The controller was ultimately applied to a newer Snake Robot that is the subject of Section 3.4.2. From the perspective of the motion controller, this third unit is identical to the two of Section A.1.2. To complete the installation, an aluminum attachment was created to both mount the controller atop the actuation unit and act as a heatsink for the amplifiers. Additionally, an extra circuit board was created to widen the amplifier board ring, facilitating assembly. Figure A.22 shows the controller from Figure A.10 installed on the robot.

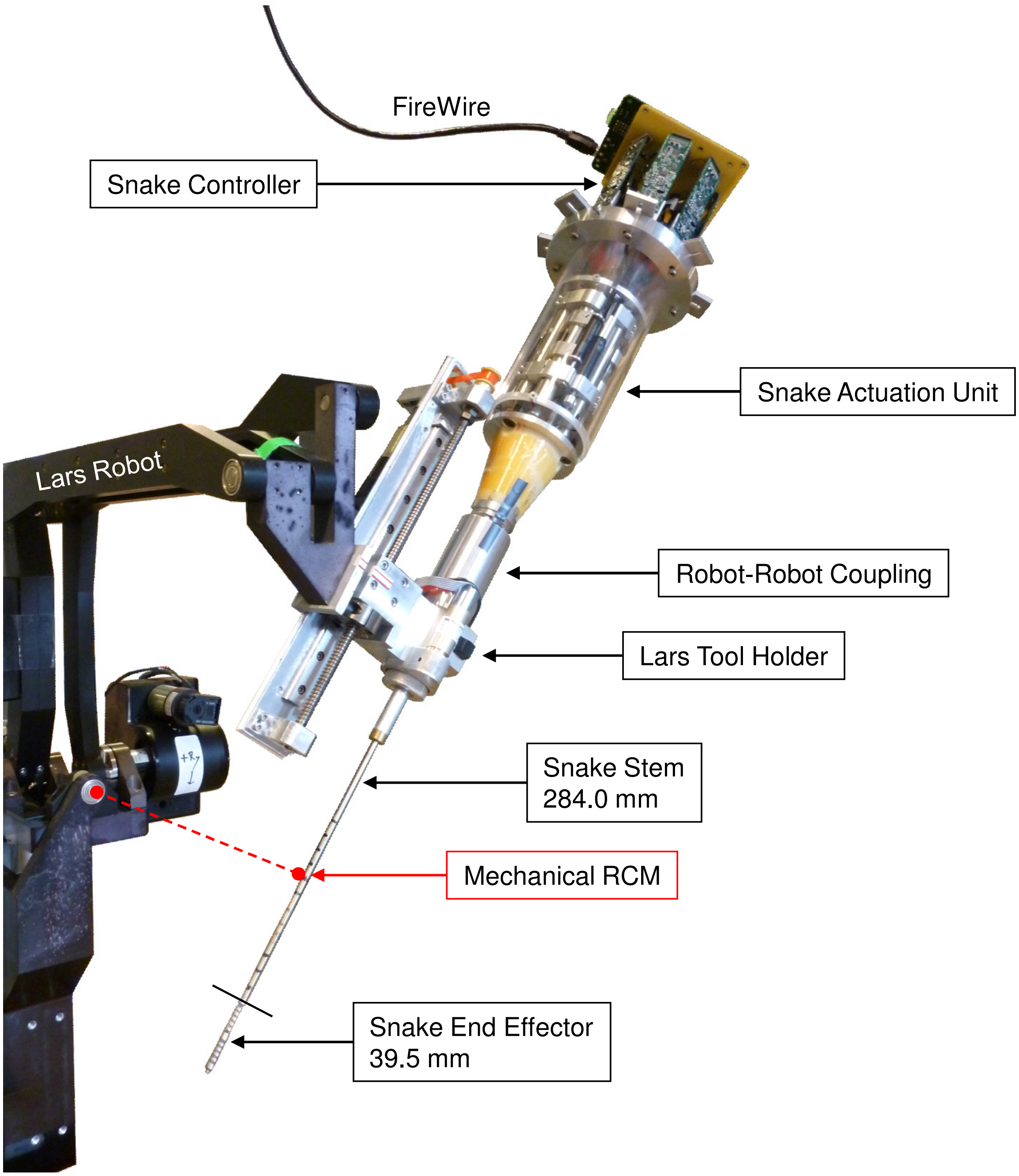


Figure A.22: Completed controller mounted on the Snake Robot actuation unit (Figure 3.7, reproduced here for clarity).

A.9.2 Experiments and Results

Robot control software, based on the CISST libraries, runs on an RTAI-patched Linux PC with a 2 GHz Pentium 4 processor and 512 MB of RAM. (RTAI was replaced with Xenomai in subsequent uses.) At the trajectory level, the software accepts six-dof Cartesian configurations and computes the inverse kinematics in 125-Hz cycles. The resulting joint values are transferred to the servo level, which updates the actuator positions at 1 kHz (nominal). The algorithm follows the constrained optimization approach described in [120, 131].

The optimization framework implemented as part of the robot control software allows us to run simplified experiments without having to reduce the complexity of the original algorithm. Given that the end effector can move with only four degrees of freedom, orientations within the dexterous workspace can be ensured at *some* position by adjusting the weights of the least squares minimization problem. The tip orientation is given by the forward kinematics in Equation (A.2) [1].

$$R_{tip} = \begin{bmatrix} c^2\delta(s\theta - 1) + 1 & -s\delta c\delta(s\theta - 1) & c\theta c\delta \\ -s\delta c\delta(s\theta - 1) & -c^2\delta(s\theta - 1) + s\theta & -c\theta s\delta \\ -c\theta c\delta & c\theta s\delta & s\theta \end{bmatrix} \tag{A.2}$$

Here, θ and δ are the angles illustrated in Figure A.23 and s and c are shorthand for *sin* and *cos*. By equating Equation (A.2) to the yaw-pitch-roll representation of a rotation matrix, we find an expression for the orientation in terms of yaw (γ), pitch

(β), and roll (α) angles. Then solving for α, we obtain

$$\alpha = \tan^{-1}\left(\frac{\sin\beta\sin\gamma}{\cos\beta + \cos\gamma}\right) . \tag{A.3}$$

In one test we measure the robot path corresponding to an input path, a composition of yaw and pitch varying sinusoidally (±45°, 0.1 Hz, in phase). Because the roll angle is a function of yaw and pitch and varies slowly, only α needs to be shown. As a reference, γ and β correspond to rotations about the robot y- and x-axes respectively, and the z-axis is aligned with the stem as mentioned above. Also listed are servo loop duty cycles (busy periods), measured by pulsing the parallel port and capturing the signals on an oscilloscope.

A.9.2.1 Number of Nodes

To determine the scalability of the system, the test input path is applied with one, two, and three nodes connected. Though the extra nodes do not drive mechanical components, the behavior of an n-node, $7n$-axis system is faithfully represented from an I/O latency standpoint. The resulting paths are plotted in Figures A.24a and b. There appears to be no effect from increasing nodes, likely because the extra overhead does not lead to violated timing requirements.

The times listed in Table A.3 show that the duty cycle of the servo loop increases with the number of nodes. The computer can write all nodes with a single broadcast

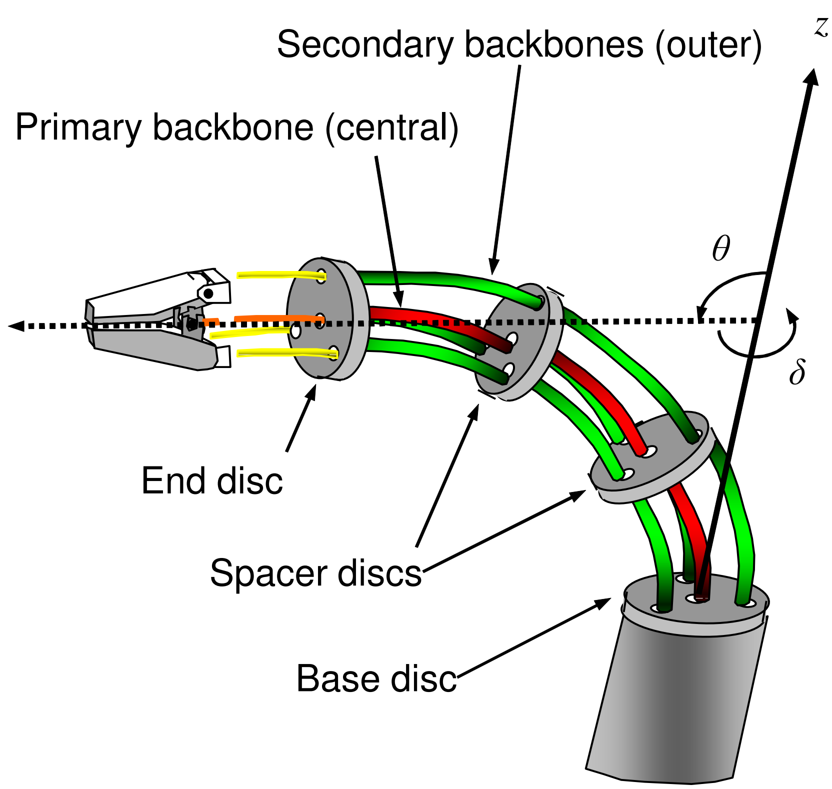


Figure A.23: Anatomy and kinematics of a snake-like unit; bend angle θ and bend plane δ comprise the degrees of freedom (Figure 3.8, reproduced here for clarity).

packet, so the time increase is due primarily to extra read transactions (computation and transfer times for additional axes are negligible, on the order of a few microseconds total [124]). A rough extrapolation based on an additional 40 μs of latency per node suggests that this specific system can accommodate up to six nodes while running at a 50% duty cycle. This figure may be increased by improving upon the naive implementation of read transactions.

To illustrate the dominance of the robot's mechanical bandwidth over the controller latency, the result of a faster sinusoidal command (0.4 Hz) is shown in Figure A.24c. There is a delay of about 60 ms, whereas no delay is discernible in the plots above it.

Table A.3: Servo loop duty cycles

# of Nodes[1]	Duty Cycle	Servo Rate[2]	Duty Cycle
1	298 μs (30%)	800 Hz	288 μs (23%)
2	334 μs (33%)	1000 Hz	298 μs (30%)
3	356 μs (36%)	1250 Hz	290 μs (36%)
		2000 Hz	294 μs (59%)

[1] Servo rate = 1 kHz   [2] # of nodes = 1

A.9.2.2 Servo Rates

We test the performance of the system at various servo rates, using one node, to determine its configurability in this scenario. For servo rates of 800, 1000, 1250, and 2000 Hz, the maximum error of the robot position from the command is 6.9×10^{-3} degrees in each case under the test input. The paths are similar to those shown in Figures A.24a and b and are thus omitted. These results suggest that the controller

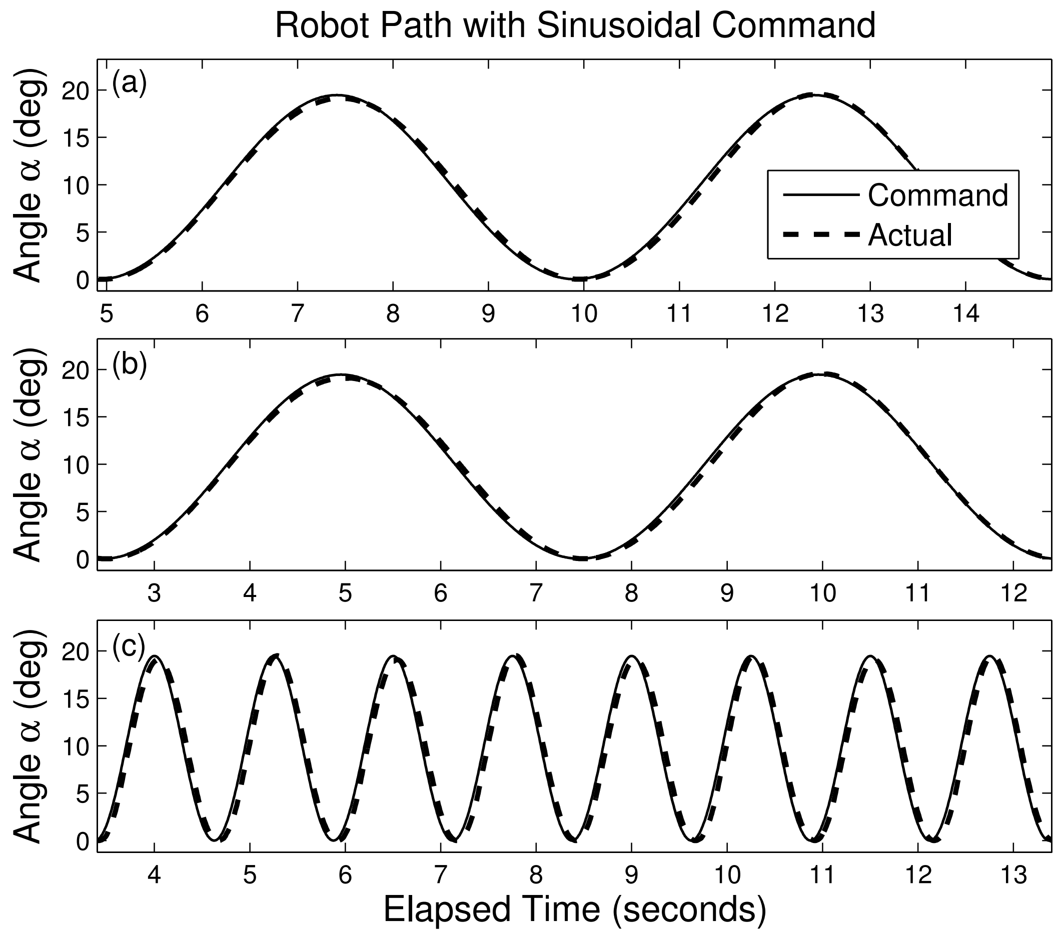


Figure A.24: Path of the robot with test input and 1 kHz servo rate. (a) With one node ($n = 1$). (b) With $n = 3$. (c) With $n = 1$ and a 4x faster input. There are no noticeable differences between 1–3 nodes, while a slight lag in (c) is visible. Time axes have been shifted and truncated (from 80 s) for comparison.

is able to run at different rates without affecting functionality. The servo loop duty cycles are listed in Table A.3. Bus transaction latencies prescribe an upper limit on the servo rate. In keeping the duty cycle at a safe margin (roughly 50%), the robot appears to work as intended.

A.9.2.3 Teleoperation

A teleoperation setup is used to evaluate the performance of the control system under realistic conditions. Position commands from a PHANTOM Omni haptic device are sent over a local area network to the robot computer. The robot of Figure A.22 is used. To expand this test, two nodes (14 control axes) are used; the servo rate is set at 1 kHz.

Figure A.25 displays the path taken by the robot in response to arbitrary motion commands from the Omni over a period of 60 seconds. The system appears to work as intended, except where abrupt changes in the path are commanded; we suspect these are due to the limitations of the robot itself.

A.10 Future Work

As of this writing, the presented controller is complete and being used on a 3D ultrasound-guided Snake Robot for minimally invasive cardiac surgery research. A heatsink-mount was fabricated to attach the controller to the actuation unit. Adapters for the motor connectors were made due to a change in pinout that oc-

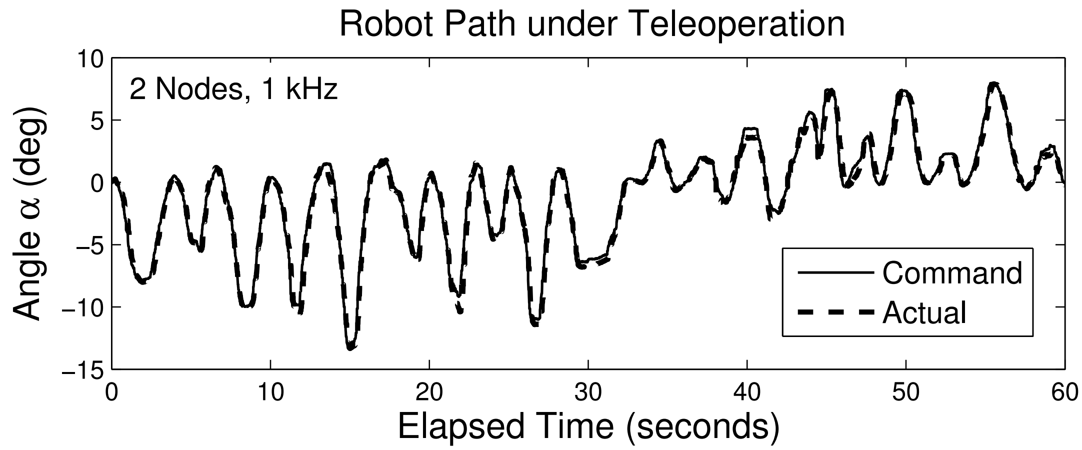


Figure A.25: Path of the robot teleoperated by a PHANTOM Omni.

curred after the circuit boards had been printed. With these adapters, the motor cabling could no longer fit inside the ring of amplifier boards as previously planned, so an interface board was created to widen the ring.

The Snake Robot control software has been modified to use the new API and controller. Our careful efforts in crafting the API helped ease the transition. We would like to experiment with QNX, a real-time Linux-based operating system that is reportedly being used by several robotics groups. It would be interesting to see whether usage of a real-time operating system has a significant impact on bus transaction times.

We are able to achieve real-time performance with one node, but it is uncertain how this will hold when multiple nodes are connected. One way to alleviate this concern is to employ single transactions that encompass all nodes. This is largely if not entirely an FPGA implementation. The computer can send write commands by broadcasting large packets and allowing each FPGA to extract its own segment of data. Similarly-spirited read transactions appear to be somewhat more challenging to perform due to the nature of the architecture. The feasibility of using isochronous transactions for this purpose should be explored.

Ultimately the goal is to further surgical robotics research. An immediate project enabled by this work is the convenient deployment of additional Snake Robots for tasks such as camera manipulation. Other potential applications include dexterous ultrasound imaging and ablation. A similar API for a new IEEE 1394-based controller [155,158] has been incorporated into the Surgical Assistant Workstation (SAW) [179], a standard medical robotics software framework.

A.11 Conclusions

Though parallel buses such as ISA, Q-Bus, Multibus, and VME have become tried-and-true interfaces for robot control, they are increasingly deprecated with the emergence of IEEE 1394, PCI Express, and Ethernet-based protocols, which feature greatly simplified cabling. These high speed serial networks provide higher performance than traditional field buses, such as CAN, SERCOS, and RS-485, which have also been used for real-time control.

The ability of the IEEE 1394 bus to multiplex a large number of data channels helps reduce wiring complexity, making systems more robust and scalable to many axes of control. Centralized processing eases intra-robot (e.g., master-slave) communication and allows systems to utilize ever-advancing computing power. Such a setup simplifies the software development environment, which is especially advantageous in areas such as research and education, where typical users are not proficient with software development using embedded microprocessors and associated tools.

This paper presents a scalable controller architecture based on IEEE 1394 for communication between the computer and actuated joints. A key element is the use of programmable logic, such as an FPGA, to provide link layer services for the network by routing read and write requests to the appropriate hardware device. The advantages of centralizing processing and distributing I/O via a high-speed serial network are discussed. The concept was demonstrated by a custom controller for the Snake Robot. As the design lends itself towards modularity, the theme served as an influence throughout development. With the potential need for many axes of control in surgical robotics research, modular hardware can help increase availability and flexibility while making systems easier to troubleshoot.

Our first prototype consisted of three boards: a commercial FPGA development board, a custom daughterboard with an IEEE 1394 physical layer chip, and custom motor power amplifier boards. After running a number of tests on this setup, we then constructed a second prototype that combines the first two boards (with FPGA and physical layer chip) into a single custom board. We also fabricated a new set of amplifier boards implementing several repairs and design changes.

Preliminary performance data, obtained under a conventional operating system

(Linux), suggest that this approach is feasible for real-time control with rates up to several kHz; higher rates, such as 10 kHz, appear to be challenging with the current setup based on the measured latencies.

Though the described control system is not necessarily a novel design given existing technologies, we contend that it will ease the development of dexterous robots and allow researchers to experiment with more robot-assisted surgical tasks.

Acknowledgments

Special thanks go to Peter Kazanzides for his comprehensive advisement on this project. I would also like to thank Hamid Wasti of Regan Designs, Inc. (Coeur d'Alene, Idaho) for his expertise on the design and layout of the controller boards, and Ankur Kapoor for sharing his knowledge and experience. Finally I thank Russell Taylor for his overall support of my research and research-related efforts. This work was supported in part by the National Science Foundation (NSF) under Engineering Research Center grant #EEC9731748, NSF grant #MRI0722943, and by Johns Hopkins University internal funds.

Addendum: For all of their help with the Snake Robot I thank Nabil Simaan, Kai Xu, Wei Wei, Walt Krug, and Mike Franckowiak. Author was supported by a Philips Research North America fellowship through portions of this work.

Appendix B

An Approach to Robotic Guidance of an Uncalibrated Endoscope in Beating Heart Surgery

Preamble

This section, a modified reproduction of [151], describes a collaborative effort led by A. Popovic. Section B.4 has been summarized as the experiments on a beating heart phantom were conducted by A. Popovic exclusively. This work highlights some of the difficulties of direct pursuit of a moving target from the perspective of a different surgical application.

Abstract

Minimally invasive cardiac surgery is performed under direct vision provided by an endoscope. During cardiac bypass surgery, counterintuitive control and small field-of-view endoscopic visualization is not sufficient to localize all relevant structures of the heart. This work presents a first step towards enhanced endoscopic visualization in cardiac surgery by allowing image-based steering of the endoscope without the need of camera calibration. In the envisioned clinical scenario, a surgeon or an endoscopy assistant selects a point or an area in the live endoscope video stream and the robotic system steers the endoscope towards the selected target on a beating heart.

Experiments in a simulated setup have shown that the algorithm successfully performs various feature tracking tasks within the requirements. The results from these experiment led to an improvement of the algorithm to enhance the final stage of feature capturing. With the lessons learned in the simulated setup, a subsequent series of experiments were performed in a clinical-like setting on a beating heart phantom. The experiments show that all reasonable target features can be reached within 1.2 s and 7 pixels of accuracy at a safe and conservative robot speed.

B.1 Introduction

B.1.1 Motivation

Coronary arteries are the major supplier of blood to the heart muscle. Coronary artery disease (CAD) is a result of plaque buildup on an artery wall causing narrowing of the artery and disruption of the free blood flow. CAD caused one out of every five deaths in the United States in 2005 [180]. Coronary artery bypass graft (CABG) surgery is a procedure done to bypass atherosclerotic narrowing in coronary arteries relieving symptoms of the CAD. As of today, CABG remains the most effective therapy for CAD [181–183].

Introduction of surgical endoscopes has revolutionized (cardiac) surgery allowing minimally invasive approaches under a direct vision. In many cases, reduced trauma and improved recovery come at the cost of prolonged surgery due to the difficulty of handling long tools fixed at a fulcrum point and unnatural requirements for hand-eye coordination.

Apart from the widely accepted advantage of minimally invasive surgery—reduced trauma to the tissue—another objective in minimally invasive heart surgery is to eliminate the need for cardiopulmonary bypass (CPB). CPB is associated with significant morbidity and mortality through complications such as hemolysis, air embolism, and clotting [184].

The two most reported technical difficulties during minimally invasive procedures are difficult instrument handling during anastomosis on moving tissue and inadequate visualization [185–188]. Several solutions have been proposed to tackle the instrumentation challenge, which include robotic systems (e.g., da Vinci®, Intuitive Surgical, Inc.), mechanical heart stabilizers, and automatic anastomosis tools (e.g., PAS-port®, Cardica, Inc.). This work therefore focuses on the visualization issues. It is widely accepted in the clinical community that small field-of-view counterintuitive endoscopic visualization is not sufficient to localize all relevant structures of the heart during a cardiac bypass surgery [185, 186].

The counterintuitive operation is caused by an unknown mapping between the surgeon's hands and the endoscope view [189]. This mapping is learned during the surgery. Reports show that an intraoperative change in camera position or orientation may cause significant increases in time needed to learn the mapping [189,190], thereby increasing the duration of the surgery. In CABG, the endoscope is usually inserted into the chest cavity from the right side of the patient or from the posterior-anterior direction. The instruments are inserted from the left side of the patient, and the surgeon stands left of the patient [191]. As an additional difficulty, control of the endoscope and surgical tools is usually distributed over at least two people: a surgeon holding the instruments and an assisting surgeon holding the endoscope. Therefore there are three relevant coordinate systems, encompassing the endoscopic view, the surgeon, and the assistant.

The small field-of-view of standard endoscopes used in cardiac surgery are usually inadequate to visualize important landmarks on the heart [192] while at the same time providing poor image resolution [185]. A usual tradeoff is to keep the endoscope as close as possible to the target area to improve spatial resolution. The assisting surgeon performs free-hand endoscope manipulation to detect the position of the current field of view relative to prominent anatomic landmarks such as the apex of the heart, the groove between the medial aspect of the left atrial appendage and pulmonary artery, and the ventricular septum [187]. As these landmarks are identified, the location of the target artery can be predicted and the endoscope can be steered towards it. This process is hindered by the above described hand-eye coordination problems. Visualization issues are reported as one of the most common causes of conversions from minimally invasive to open surgery [187, 188].

This work presents a first step towards enhanced endoscopic visualization in cardiac surgery by allowing image-based steering of the endoscope bypassing uncertainties arising from counterintuitive intraoperative setup. In the envisioned clinical scenario, a surgeon or an endoscopy assistant selects a point or an area in the live endoscope video stream and the robotic system steers the endoscope so that the selected area appears in the middle of the endoscope image. The system is aimed to allow a simpler manipulation of the endoscope using a direct mapping from the endoscope view as seen by the surgeon and the endoscope. This will bypass the difficult mapping required between the view, the endoscope, the surgeon, and the assistant controlling the endoscope and as a result, a solo surgery can be performed. This will also allow the surgeon to explore the heart surface in a more structured fashion.

The platform presented here will serve in the future as a basis for image-based steering using other imaging modalities, such as intraoperative ultrasound (3D trans-esophageal echocardiography, or TEE) and preoperative computed tomography (CT).

B.1.2 Background and System Requirements

This work employs visual servoing, a robot control method using visual input to set robot movement parameters. In image-based approaches the control loop of visual servoing is closed through a transformation between the image plane and the robot joint velocities established through a calibration process [193, 194]. The transformation matrix is referred to as an *image Jacobian* or *interaction matrix*. An alternative to image-based methods are position-based methods performing explicit 3D reconstruction of feature points from the image [195]. The performance of both methods is highly sensitive to the accuracy of the intrinsic parameters of the camera, e.g., focal length(s) and image optical center.

Clinical environments are not well-suited for accurate camera calibration which is a tedious, error-prone process. The optical system may show a significant drift in its intrinsic characteristics due to heat, humidity, and other environmental disturbances. Extrinsic camera parameters, i.e., the transformation between the camera and robot joints, can be made reproducible through a unique mounting of the endoscope shaft. Finally, the robot forward kinematics represented through a joint Jacobian—joint velocities as a function of end effector velocity—is significantly less sensitive to interferences in the operating room, though a drift in the joint calibration can be induced through incautious handling of the system.

Considering the clinical and technical insights described above, the following requirements for an image guided robotic endoscopy system were defined:

- The system should impose a fixed fulcrum point to avoid injuries to the patient's skin and underlying tissue.
- The system should not require a camera calibration. It should work with any kind of endoscope and allow intraoperative replacement of the working endoscope.

Different approaches for solving visual servoing in medical robotics and endoscopic surgery have been proposed. Zhang *et al.* [196] propose a method involving camera calibration and tool tracking in the endoscope view based on three optical markers. The optical markers can provide depth information given the known distance and thickness. Thus a full image Jacobian can be computed. In another system, proposed by Krupa *et al.* [197], a laser pointer with four laser dots is attached to the endoscope to provide depth information relative to the organ. The image Jacobian is obtained similarly as in Zhang *et al.* using three markers on the tool. However, the use of artificial markers on the heart surface may not be clinically acceptable. It would require the surgeon to place the markers using endoscopic instruments which might be a time consuming process. Therefore, the time expected to be saved in the learning of mapping might be partially or fully lost due to the time required for the placement of the markers. Also, the markers might cover some important structures, for example arteries hidden by a layer of fibrous-fatty tissue.

Hynes *et al.* [198] propose a method for visual servoing without the need for camera calibration used in minimally invasive surgery based on the previous work by Jagersand *et al.* [199]. This method performs an online estimation of the image Jacobian using a correction formula. The control loop starts with an initial estimate of the matrix and corrects the values using motion observed by a camera decoupled from the robot. The method requires a random initial estimate of the image Jacobian, which can lead to unpredictable erroneous movements [199] unsuitable in sensitive surgical settings.

Wei *et al.* [200] propose a proportional control scheme for real-time visual servoing of the AESOP robot for laparoscopic surgery. The robot is moved proportionally to the distance between the tracked point and the target location. No optimization of the tracking speed is proposed which might induce oscillations in periodically moving organs such as the heart.

In this work, a practical method for visual servoing with an uncalibrated endoscope is presented. The method simplifies preoperative workflow by avoiding camera calibration and removing uncertainties that might arise from a suboptimal calibration. No artificial markers attached to the heart are used in order to make the proposed setup more clinically acceptable.

B.2 Visual Servoing

B.2.1 Kinematic Considerations

The robot used in this system is the IBM/JHU LARS robot (Section 3.4.1) holding a 7-mm forward-view endoscope. The particular kinematics of the distal four DOFs, constructed in a parallelogram-like structure imposes a hardware restriction to possible movements of the mounted end effector. This structure is referred to as orthogonally decoupled remote center-of-motion (RCM) [128]. The hardware RCM can be positioned so that the motion center coincides with the point where the endoscope enters the patient's body (fulcrum) avoiding potentially traumatic injuries to the tissue and ribs. If a hardware RCM is not at the fulcrum, a virtual RCM can be implemented in the kinematic model of the robot.

The RCM is linearly translated as the endoscope is introduced deeper into the body. This motion is analogous to zooming and we consider it convenient for it to be controlled by the user rather than the robot for safety reasons. Considering the imposed constraints described, the two end effector DOFs (θ and ϕ) are controlled by six robotic joints. The roll of the endoscope is also controlled by the user.

B.2.2 Velocity Control

The Jacobian J of the LARS robot has been previously calibrated and is used as is. The forward and inverse kinematics, along with the optimization algorithm for the virtual RCM, are described in [140]. The implementation available from the *cisst* software package [127] was used.

The inverse kinematics problem of the LARS robot is solved using a constrained optimization framework, with the joint positions acting as variables to be optimized for a given end effector configuration. The virtual RCM is modeled as a virtual fixture and implemented as a constraint to the optimization problem. The relation between velocity in robot configuration space, $v_r = [v_\theta, v_\phi]^T$, and a point velocity in the image coordinate system, $v_i = [v_x, v_y]^T$, can be expressed as

$$\begin{bmatrix} v_\theta \\ v_\phi \end{bmatrix} = \begin{bmatrix} \frac{\frac{b_x^2}{x^2}}{\sqrt{1-\frac{b_x^2}{x^2}}} & 0 \\ 0 & -\frac{\frac{b_y^2}{y^2}}{\sqrt{1-\frac{b_y^2}{y^2}}} \end{bmatrix} \begin{bmatrix} v_x \\ v_y \end{bmatrix}, b_x = \frac{d}{lf_x}, b_y = \frac{d}{lf_y} \tag{B.1}$$

as long as the optical and geometric centers of the image are colocated, where l is the length of endoscope from the fulcrum point to the tip, d is distance from the endoscope tip to the heart surface, and f_x and f_y are focal lengths of the endoscopic system in the x- and y-axes respectively. None of these values are known without the calibration. We propose an optimization technique to control v_θ and v_ϕ from v_x and v_y without explicitly measuring or estimating the image Jacobian.

The first step in the optimization scheme is to map the spherical cap parametrized with θ and ϕ to a 2D plane using a spherical projection. This projection is 1-1 unscaled warping. The resulting rectangle represents the full configuration space of the end effector (endoscope) with respect to two parameters of the end effector. In the next step, the 2D rectangle of point velocities (image) can be mapped to the 2D rectangle of the configuration space.

The first assumption introduced is a linear mapping between angular velocities of the robot and point displacement in the image. The motivation for mapping the position but not velocity resides in the fact that the expected goal of visual servoing is to track a user-selected feature point so that, after the robot task is completed, the point is in the geometrical center of image (reference point).

The velocity setting loop is shown in Figure B.1. As the feature point moves away from the center of the image it produces a 2D velocity vector $\vec{v}$. Since the scale between the robot and the image is not known, $\vec{v}$ can be normalized and transferred to the θ-ϕ plane resulting in vector $\vec{v_r}$. The projected values on the θ- and ϕ-axes define a unit robot displacement in a given direction ($\vec{v_r}$). Since $\vec{v_r}$ is normalized, velocities in the θ and ϕ directions would be ≤ 1 rad/s. Depending on other unknown parameters (depth, focal length, speed of the point) that speed would usually be either insufficient to reach the point, or too fast such that it overshoots the point. Therefore, a scale has to be adopted to account for these uncertainties.

Adaptation of the mapping scale can be considered an optimization problem with

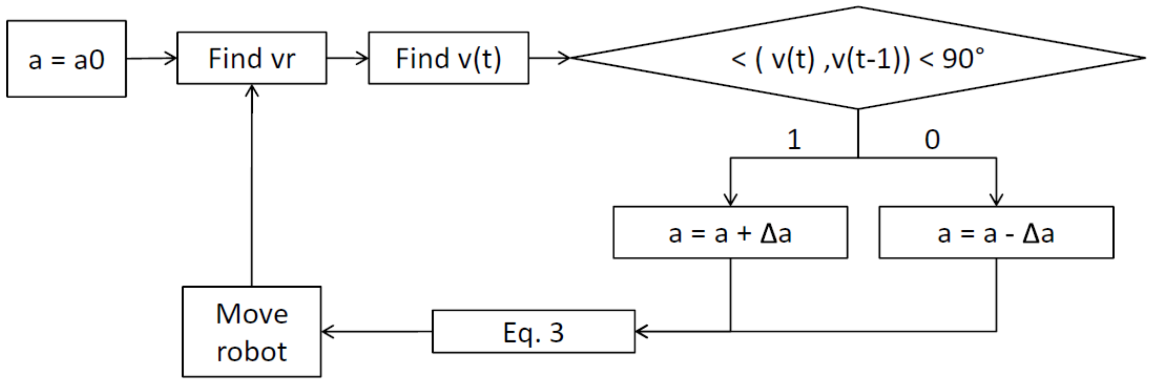


Figure B.1: Velocity setting loop.

the objective to control the robot at a high velocity far from the feature to assure high tracking speed and to control the robot at a low velocity in the proximity of the feature to assure stability and prohibit overshoot. Four numerical components have been used for solving this optimization problem:

- A proportional factor p representing the Euclidean distance of the feature point to the center of the image.
- A PD controller implemented with two numerical factors K_p and K_d.
- A dynamic scale factor a.

These factors are combined in a control loop as

$$\begin{bmatrix} v_\theta(t) \\ v_\phi(t) \end{bmatrix} = a \cdot (K_p \cdot p(t) \cdot \vec{v_r}(t) + K_d \cdot \frac{dp(t)}{dt} \cdot \vec{v_r}(t)), \tag{B.2}$$

where $[v_\theta(t), v_\phi(t)]^T$ is a velocity vector at time t. Therefore, the velocity vector, assuming a unit of time, is

$$\begin{bmatrix} v_\theta(t) \\ v_\phi(t) \end{bmatrix} = a \begin{bmatrix} K_p \cdot p_x(t) + K_d \cdot (p_x(t) - p_x(t-1)) \\ K_p \cdot p_y(t) + K_d \cdot (p_y(t) - p_y(t-1)) \end{bmatrix} \tag{B.3}$$

The scale factor a is a dynamic optimization factor used to facilitate learning of the current setup and compensate for changes in the setup and target velocity. The value is recomputed in each optimization step, i.e., in each frame of the endoscopy stream. The following section describes the setting of the scale factor in more detail.

B.2.3 Scale Factor

This section describes the heuristics used to set the scale factor for the control loop (Figure B.1). As the robot moves along a defined direction, the position of the tracked point is updated due to two reasons: (1) the point is moving in the endoscope coordinate system due to the endoscope movement, and (2) the point is moving in the world coordinate system (e.g., through the beating of the heart or breathing) and thus in the endoscope coordinate system as well. A new position of the point updated at every frame is evaluated against an accuracy threshold. If the point displacement is smaller than a given threshold, the robot is not moved and the loop continues waiting for larger displacements; this helps maintain a stable image in the presence of noise or tremor. If the point has moved between two given frames, the new unit direction vector is evaluated against the previous one. There are two possible scenarios: the new vector is in the same general direction as the previous one (the vectors are in the same quadrants) or the new vector is pointing to a new direction. In the first case, it can be assumed that the robot is moving in the correct direction and that the robot did not overshoot the point. Thus, the direction is updated, the value of a is increased by predefined amount Δa, and tracking continues. If the direction is changed by more than $\pi/2$ it is assumed that the endoscope overshot the target and a is decreased by Δa to facilitate convergence. This method performs well in the presence of feature detection noise because the direction is evaluated against a reference point (e.g., the geometrical center of the image) and it does not change if the feature point is oscillating. An advantage of this control loop is that the robot learns an optimal speed factor a in real time. An additional constraint in the system is a maximum speed set in each axis for safety reasons ($v_{\theta,max} = v_{\phi,max} = v_{max}$).

B.3 Experiments on a Virtual Phantom

B.3.1 Setup

This section describes a series of experiments on a virtual phantom to demonstrate validity of the proposed approach. The setup is shown in Figure B.2. A rigid forward-view monocular endoscope (Richard Wolf, Inc.) with 170 mm length, 10 mm diameter and 1/32 mm focal length, attached to a 30 Hz CCD camera operating at NTSC image resolution is mounted on the LARS robot's end effector. An RCM point is set at 100 mm off the tip of the endoscope. The inertia of the robot was measured at 0.250 s, representing the latency of motion after a non-zero velocity is set.

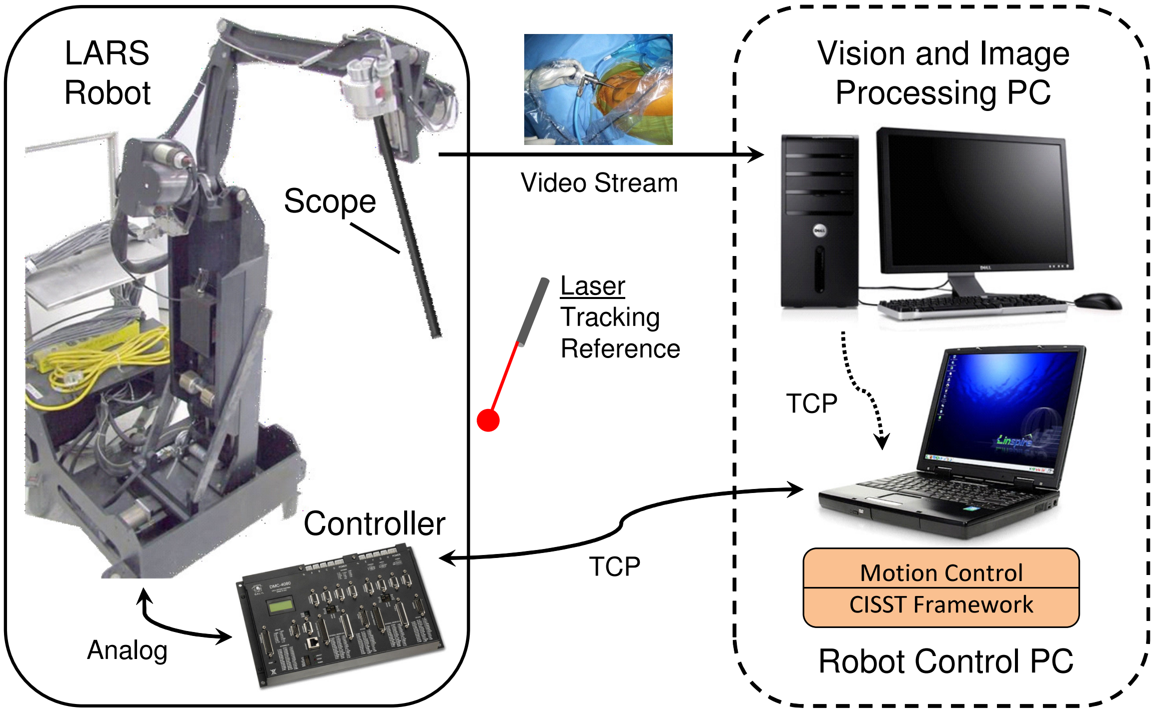


Figure B.2: Robotic endoscope system used in visual tracking experiments.

A red laser dot was projected onto a blue screen used as a background. The dot is segmented using an RGB thresholding algorithm and the dot center is computed from principal shape moments of the dot contour. The laser dot serves as a measurement reference. The robot can be guided so that the laser dot appears to move towards virtually defined targets. This particular imaging setup was used to decouple the effects of image processing computations from the performance of the robot control algorithms.

Datasets of virtual targets moving on a flat surface are used as inputs to the visual servoing system. This approach allows the use of a flat blue screen with virtual functions implementing a 3D-like movement. For example, 1.5 periods of a sinusoidal speed profile on a flat surface are analogous to a constant velocity movement of a point along a hemisphere. This setup allows for flexible testing in a simple environment without oversimplifying the problem under consideration. Further details will be given in later sections of the text.

B.3.2 Experiments

B.3.2.1 Straight Line

This set of experiments is performed to evaluate tracking of a point moving along a straight line at different speed patterns. Parameters are set empirically to $a = 1$ with an increment/decrement step of $\Delta a = \pm 0.1$, $K_p = 0.1$, and $K_d = 0$. The maximal speed of the robot is set at 0.2 rad/s. The system was operated at 30 Hz. The first experiments are done for a constant speed, in two segments. In the first segment, the robot tracks a constantly moving point. In the second, the robot is stopped while the target continues to move away. After 5 s, the robot is restarted (Figure B.3).

After issuing the restart command, movement of the robot is observed after a period of about 0.3 s. That is in accordance with the latency of the system. The robot reaches the target after approximately 1 s, overshoots it (Figure B.3), and stabilizes

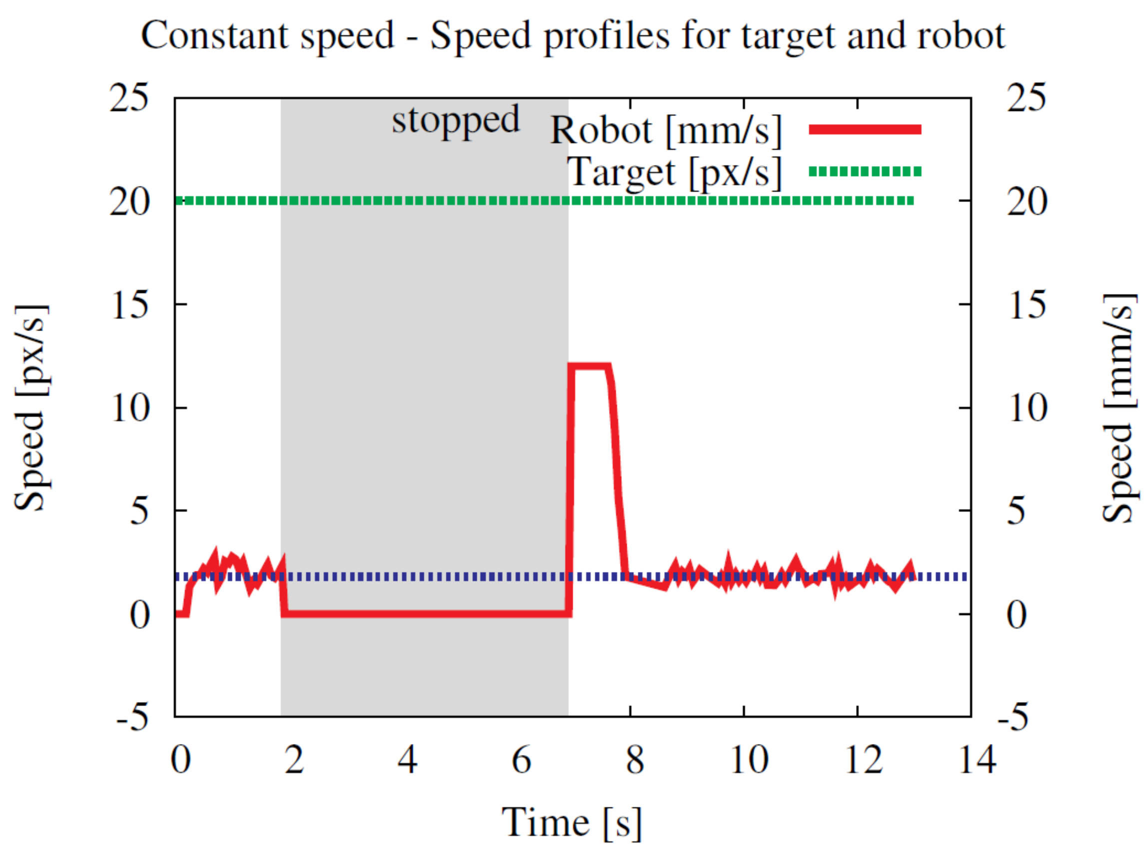


Figure B.3: Speed profiles for tracking a constant speed target in 1D. The dashed blue line signifies the expected robot speed.

at approximately 5 px from the target. The speed stabilizes at approximately ±1 mm/s around the expected speed. This can be filtered out using two methods: (1) selecting a larger accuracy circle, and (2) imposing the minimal speed to prevent the robot from slowing down too much in the proximity of the target. These insights will be used while setting up the heart phantom experiment.

The second experiment is done for a varying target speed. The same motion pattern was used. Resulting profiles are similar to those of the constant speed experiments (Figure B.4).

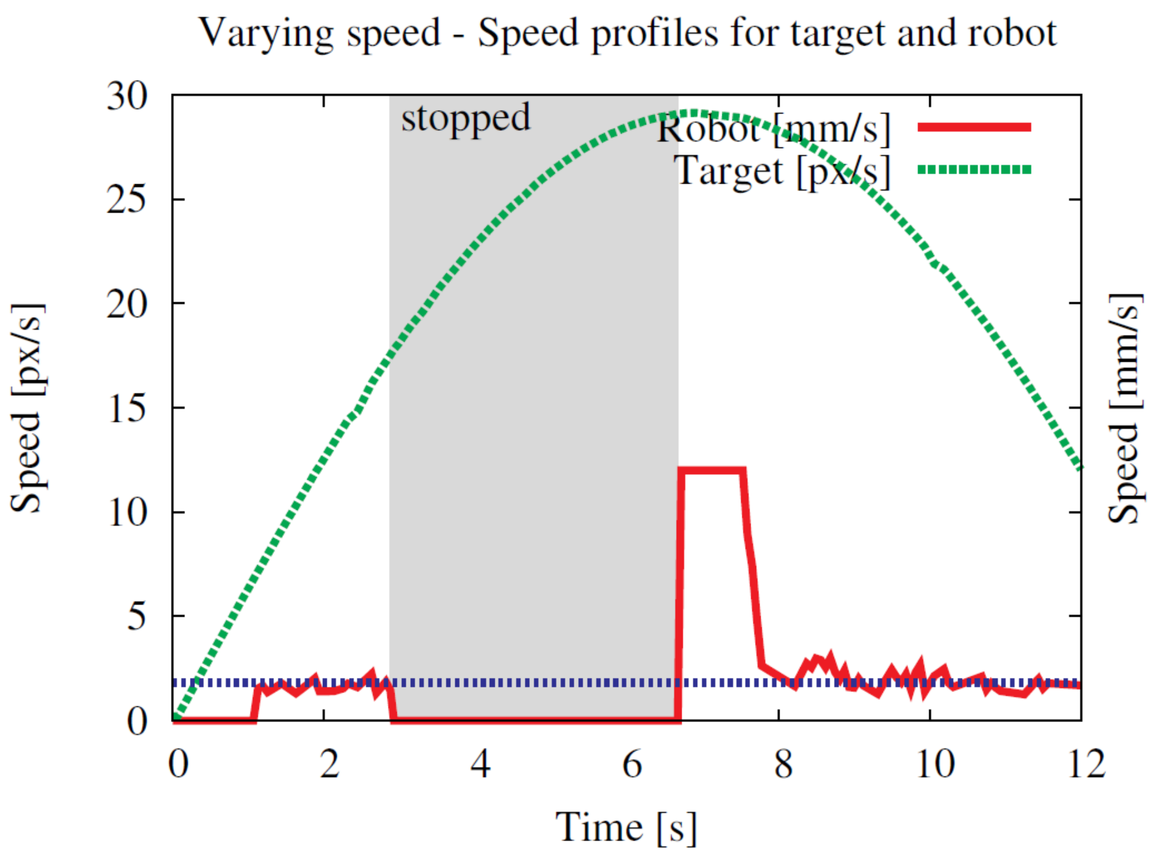


Figure B.4: Speed profiles for tracking a changing speed target in 1D.

B.3.2.2 Circular Motion

In the second set of experiments the target moves in a circular pattern with a constant angular velocity (Figure B.5). Referring back to the algorithm implementation, two axes are controlled separately. Therefore, this movement is analogous to a combined movement on a damped sinusoidal path with a damped speed (Figure B.6).

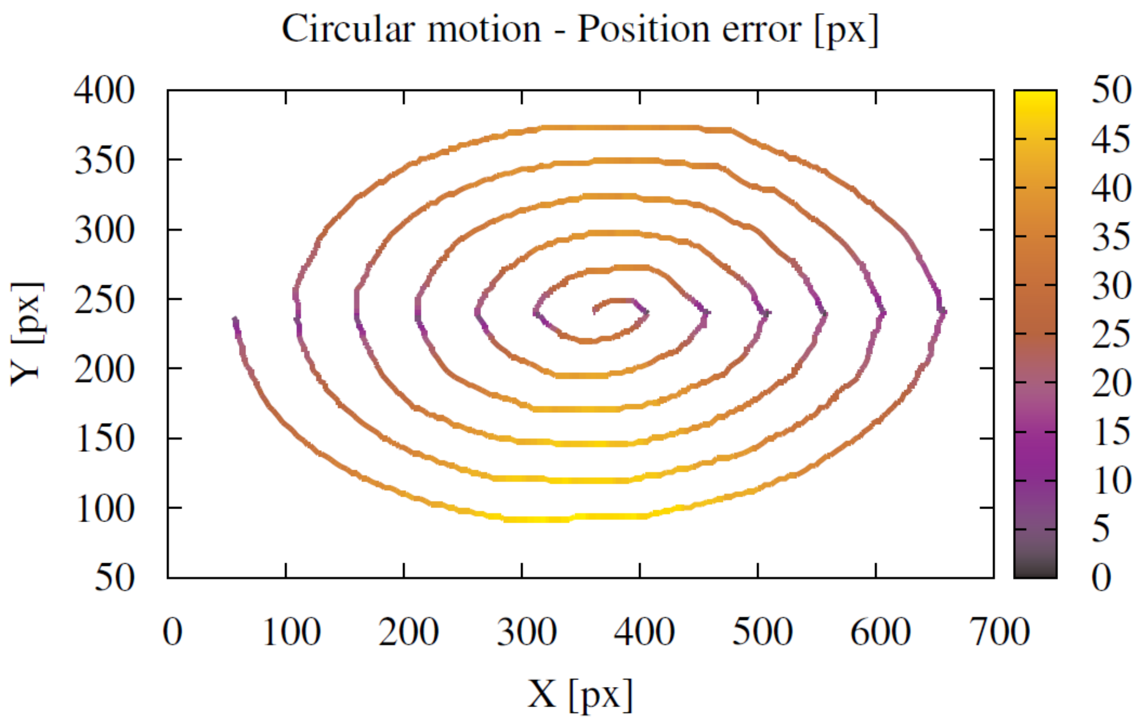


Figure B.5: Positions of target in circular motion experiment, with errors from robot positions coded in color.

The first apparent irregularity that can be observed is the fact that errors are more prominent along the x-axis. That axis is dominant because that is where the main motion is occurring, due to asymmetry of the spiral. A second important insight arises from the periodicity of error in time. The period of the error signal is changing

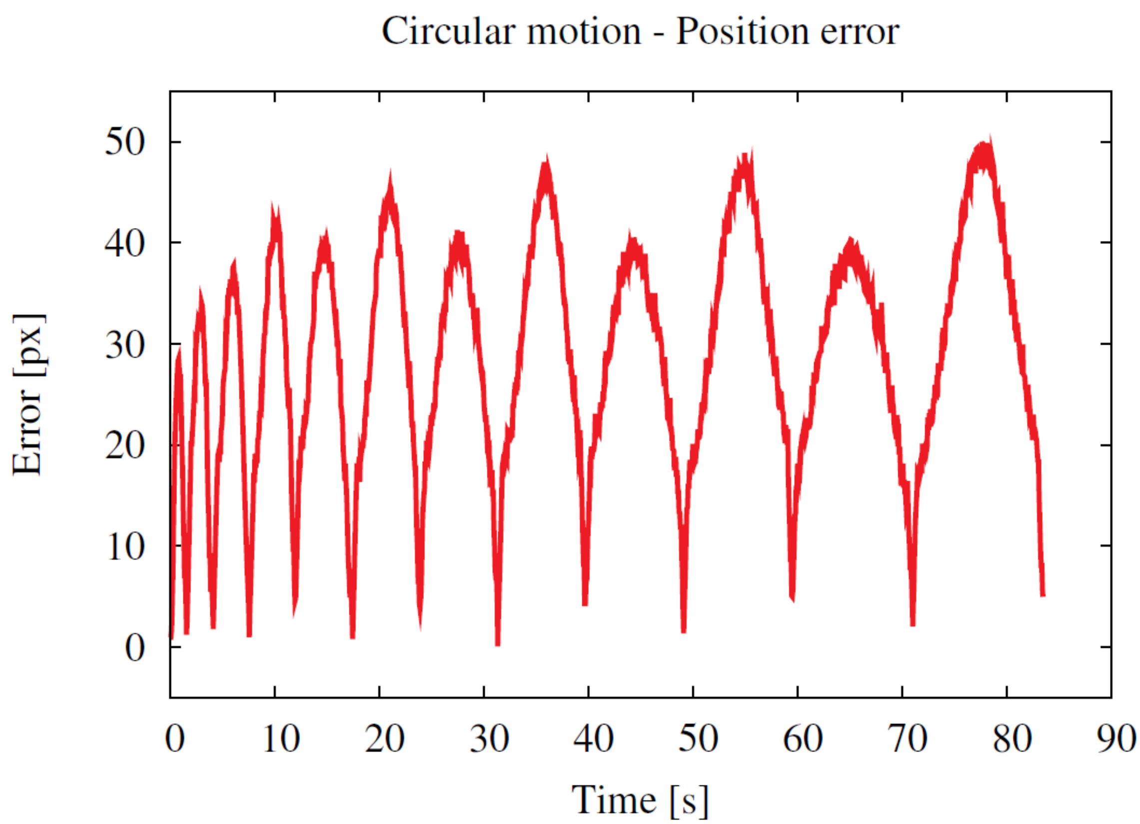


Figure B.6: Position errors from Figure B.5 as a function of time.

indicating that catching up with the varying speed of the feature is more critical than its change in direction. This is an important realization that will lead to improvement of the algorithm, described later. Finally, top-bottom asymmetry can be explained though asymmetry in the spiral (the bottom part is longer) so when the point gets to that part, it is the first time the algorithm is encountering that speed and it requires more time to adjust.

B.3.2.3 Pulsed Motion

The final experiments are done for a pulse speed profile (Figure B.7) of a changing period. This test was done to evaluate for abrupt changes in direction. The results show stabilization after 1 s and a stable speed at ±1 mm/s.

B.3.3 Insights Obtained from Virtual Target Experiments

A number of previously unforeseen lessons were gleaned as a result of performing experiments in which the robot chases a virtual target traveling under various speed profiles. These insights, enumerated below, are used to design the beating heart phantom experiments.

1. The results demonstrate a stable operation in the presence of both varying speed and changing direction in linear and non-linear target trajectories.

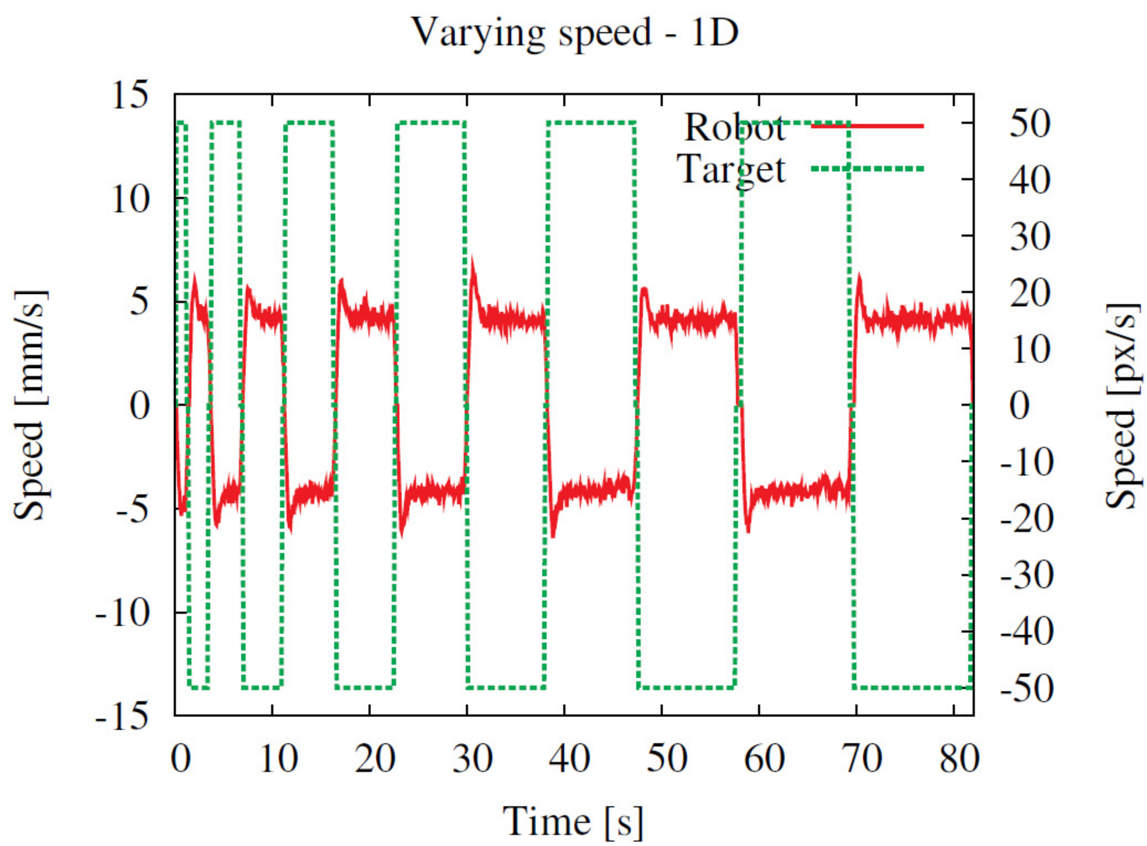


Figure B.7: Speed profiles for tracking a pulsed speed target in 1D.

2. The change in target speed appears to be more prone to errors in the robotic position than the target's change in direction. Since direction setting is more responsive, oscillatory motion with direction setting might be acceptable.

3. Very rare occasions of a large overshoot indicate that $K_p = 0.1$ might be too conservative.

4. To improve stability, a minimal speed should be imposed in a similar manner to the maximal speed.

5. The algorithm tends to lag behind the target.

The most important change introduced as a consequence of these insights is improvement in the target catching phase. As explained before, as the distance between the robot point and the target point decreases, the speed decreases accordingly. An improvement is introduced in the endgame segment. When p drops below a threshold, K_d is set to zero and a is put into increase mode. Therefore, in the last few frames, the robot speeds up again to catch the point. Insight 2 is in favor of this improvement, since it relaxes the restriction to not overshoot.

B.4 Experiments on a Beating Heart Phantom

A complete account of the visual tracking and servoing experiments on a beating heart phantom can be found in [151]. The setup consists of the LARS robot holding a forward viewing endoscope, as in the virtual phantom experiments. Instead of a neutral background, the Chamberlain Group Robotic Beating Heart Trainer model with a highly detailed exterior and movement of a live human heart was used. The control approach was designed to remove the need for camera calibration.

In each trial, an anatomical landmark was selected manually in the endoscopy video, triggering the visual servoing loop, whereby the robot moved the endoscope to center the feature in the view. Feature tracking was performed at 30 frames per second using a two-step process including pyramidal Lucas-Kanade optical flow followed by adaptive tracking modified to address both the periodic motion of the heart and robot motion relative to the heart [201]. Out of 23 runs, 20 were marked as a success (87%); tracking was lost before the robot could reach the remaining three targets.

All tested features were reached within 1.2 s. When the robot speed was limited to 2 mm/s, the time-to-reach increased to about 6 s as expected, but the progression was not uniform due to the periodic motion of the tracked feature.

B.5 Discussion

A practical approach to visual servoing with an uncalibrated image sensor and a virtual RCM is presented. Robot control using an image-based velocity compensation loop allows guidance without knowing the intrinsic parameters of the camera, relative distance from the robot to the tracked point, or absolute distance the tracked point has traveled. The endoscope can be mounted at any depth relative to the robot adapter.

As a result, the closed loop approach reduces accuracy requirements of both robot kinematics and transformation between the endoscope and the robot. The system can accommodate to clinically relevant intraoperative changes in the setup—adjustment of the endoscope zoom, exchange of the endoscope (often done in CABG surgery to switch between oblique and forward viewing endoscopes [187]), and modifications in the endoscope mounting.

The algorithm requires preoperative setting of various control parameters (a, K_p, and K_d) which might not be desired. However, the optimizing nature of the algorithm allows a significantly larger flexibility in the parameters as compared to camera calibration in the conventional image-based visual servoing. An additional impractical constraint imposed in this system is a fixed orientation between the camera coordinate system relative to the robot coordinate system (similar as in [200]) allowing a direct translation of the direction vector. This problem can be solved either through a unique mechanical coupling of the camera, automatic image reorientation using robot encoder values from the rotational joints, or the use of an additional orientation sensor such as an Inertial Measurement Unit [202].

Advanced techniques of conventional/traditional visual servoing do not translate well to real-time systems used in dynamic surgical environments due to system setup and calibration requirements. A design of a robotic control system for automatic servoing of the endoscope towards landmarks selected in an endoscopy video stream is presented. The purpose of the system is to improve the handling of the endoscope by bridging hand-eye coordination problems with automatic operation for endoscopic cardiac surgery. The results of experiments using simulated targets yielded valuable insights on the practical application of visual servoing to complex surgical environments. The results of phantom experiments indicate the promise of the approach proposed for use in live minimally invasive cardiac surgery. The system will serve as a platform for future integration of other imaging modalities.

Bibliography

[1] N. Simaan, R. H. Taylor, and P. Flint, "A dexterous system for laryngeal surgery," in *IEEE Int. Conf. on Robotics and Automation (ICRA).* IEEE, 2004, pp. 351–357.

[2] Borden Institute, "Ch. 16: Thoracic Injuries," in *Emergency War Surgery: Third United States Revision*, A. C. Szul, L. B. Davis, B. G. Maston, D. Wise, L. R. Sparacino, and J. Shull, Eds. United States Dept. of Defense, 2004, ch. 16, pp. 1–16.

[3] J. C. Williams and W. C. Elkington, "Slow progressing cardiac complications—a case report," *J. Chiropractic Medicine*, vol. 7, no. 1, pp. 28–33, Mar. 2008.

[4] A. J. Marshall, N. J. Ring, and P. L. Newman, "An unexplained foreign body in the myocardium," *J. Royal Society of Medicine*, vol. 95, no. 5, pp. 250–251, May 2002.

[5] P. N. Symbas, A. L. Picone, C. R. Hatcher, and S. E. Vlasis-Hale, "Cardiac

missiles. A review of the literature and personal experience," *Annals of Surgery*, vol. 211, no. 5, pp. 639–47; discussion 647–8, May 1990.

[6] G. M. Actis Dato, A. Arslanian, P. Di Marzio, P. L. Filosso, and E. Ruffini, "Posttraumatic and iatrogenic foreign bodies in the heart: report of fourteen cases and review of the literature," *J. Thoracic and Cardiovascular Surgery*, vol. 126, no. 2, pp. 408–14, Aug. 2003.

[7] J. Evans, L. A. Gray, A. Rayner, and R. L. Fulton, "Principles for the management of penetrating cardiac wounds," *Annals of Surgery*, vol. 189, no. 6, pp. 777–84, Jun. 1979.

[8] Borden Institute, "Ch. 4: Thoracic Trauma," in *War Surgery in Afghanistan and Iraq: A Series of Cases, 2003-2007*, S. C. Nessen, D. E. Lounsbury, and S. P. Hetz, Eds. United States Dept. of Defense, 2008, ch. 4, pp. 118–156.

[9] J. P. Steinhoff, C. Pattavina, and R. Renzi, "Puncture wound during CPR from sternotomy wires: case report and discussion of periresuscitation infection risks," *Heart & Lung*, vol. 30, no. 2, pp. 159–160, 2001.

[10] J. T. Moore, M. W. A. Chu, B. Kiaii, D. Bainbridge, G. Guiraudon, C. Wedlake, M. Currie, M. Rajchl, R. V. Patel, and T. M. Peters, "A navigation platform for guidance of beating heart transapical mitral valve repair," *IEEE Transactions on Biomedical Engineering*, vol. 60, no. 4, pp. 1034–40, Apr. 2013.

[11] A. H. Gosline, N. V. Vasilyev, E. J. Butler, C. Folk, A. Cohen, R. Chen, N. Lang, P. J. del Nido, and P. E. Dupont, "Percutaneous intracardiac beating-heart surgery using metal MEMS tissue approximation tools," *Int. J. Robotics Research*, pp. 1081–1093, May 2012.

[12] S. G. Yuen, D. T. Kettler, P. M. Novotny, R. D. Plowes, and R. D. Howe, "Robotic motion compensation for beating heart intracardiac surgery," *Int. J. Robotics Research*, vol. 28, no. 10, pp. 1355–1372, May 2009.

[13] F. T. Henley and J. W. Ballard, "Percutaneous removal of flexible foreign body from the heart," *Radiology*, vol. 92, no. 1, p. 176, 1969.

[14] A. B. Weisse, C. B. Moschos, G. E. Levinson, and T. J. Regan, "Snaring of a broken Sommer pacemaker electrode. Intracardiac catheter fragments and nonsurgical methods of removal," *Annals of Internal Medicine*, vol. 72, no. 5, pp. 695–697, May 1970.

[15] I. Enge and A. Flatmark, "Percutaneous removal of intravascular foreign bodies by the snare technique," *Acta Radiologica Diagnosis (Sweden)*, vol. 14, no. 6, pp. 747–752, 1973.

[16] A. R. Geraci and M. W. Selman, "Pulmonary artery catheter emboli: Successful nonsurgical removal," *Annals of Internal Medicine*, vol. 78, no. 3, pp. 353–356, 1973.

[17] D. J. Fergusson, “Removal of an intracardiac foreign body without thoracotomy,” *RN*, vol. 37, no. 8, p. OR14, Aug. 1974.

[18] F. Khaja and J. Lakier, “Foreign body retrieval from the heart by two catheter technique,” *Catheterization and Cardiovascular Diagnosis*, vol. 5, no. 3, pp. 263–268, 1979.

[19] A. Lijoi, L. Barberis, G. C. Passerone, and G. Venere, “Removal of iatrogenic foreign bodies within cardiac chambers with a new snare catheter,” *Texas Heart Institute Journal*, vol. 9, no. 2, pp. 153–156, Jun. 1982.

[20] R. Uflacker, S. Lima, and A. C. Melichar, “Intravascular foreign bodies: percutaneous retrieval,” *Radiology*, vol. 160, no. 3, pp. 731–735, 1986.

[21] G. G. Hartnell, “Homemade snare for removal of foreign bodies,” *Radiology*, vol. 181, no. 3, pp. 903–904, 1991.

[22] J. C. Liu, H. S. Tseng, C. Y. Chen, M. S. Chern, S. C. Ko, J. H. Chiang, and C. Y. Chang, “Percutaneous retrieval of intravascular foreign bodies: experience with 19 cases,” *Kaohsiung Journal of Medical Sciences*, vol. 18, no. 10, pp. 492–499, Oct. 2002.

[23] M. Murphy, A. E. Healey, J. Harper, S. Saltissi, and R. G. McWilliams, “Percutaneous removal of a right atrial catheter fragment: the value of the En Snare,” *Nephrology, Dialysis, Transplantation*, vol. 19, no. 10, pp. 2686–2687, Oct. 2004.

[24] R. F. Bonvini, A. Rastan, S. Sixt, E. Noory, U. Beschorner, O. Leppanen, F. Mach, U. Schwarzwälder, K. Bürgelin, and T. Zeller, "Percutaneous retrieval of intravascular and intracardiac foreign bodies with a dedicated three-dimensional snare: A 3-year single center experience," *Catheterization and Cardiovascular Interventions*, vol. 74, no. 6, pp. 939–945, 2009.

[25] K. Pathak and D. Yadav, "Percutaneous removal of intracardiac and intravascular foreign bodies," *Medical Journal Armed Forces India*, vol. 67, no. 4, pp. 367–369, 2011.

[26] N. P. Smyth, M. R. Boivin, and J. M. Bacos, "Transjugular removal of foreign body from the right atrium by endoscopic forceps," *J. Thoracic and Cardiovascular Surgery*, vol. 55, no. 4, p. 594, 1968.

[27] T. R. Shaw, "Removal of embolised catheters using flexible endoscopy forceps," *British Heart Journal*, vol. 48, no. 5, pp. 497–500, 1982.

[28] M. Drabinsky, "Retrieval of embolized central venous catheters by a dormia ureteral stone dislodger with straight filiform tip," *Chest*, vol. 69, no. 3, pp. 435–437, 1976.

[29] J. B. Selby, C. J. Tegtmeyer, and G. M. Bittner, "Experience with new retrieval forceps for foreign body removal in the vascular, urinary, and biliary systems," *Radiology*, vol. 176, no. 2, pp. 535–538, 1990.

[30] L. H. Harrison Jr., J. A. Kisslo Jr., and D. C. Sabiston Jr., "Extraction of intramyocardial foreign body utilizing operative ultrasonography," *J. Thoracic and Cardiovascular Surgery*, vol. 82, no. 3, pp. 345–349, Sep. 1981.

[31] K. Sakai, S. Hoshino, and M. Osawa, "Needle in the heart: two-dimensional echocardiographic findings," *American Journal of Cardiology*, vol. 53, no. 10, p. 1482, May 1984.

[32] D. A. Fyfe, J. R. Edgerton, A. Chaikhouni, and C. H. Kline, "Preoperative localization of an intracardiac foreign body by two-dimensional echocardiography," *American Heart Journal*, vol. 113, no. 1, pp. 210–212, Jan. 1987.

[33] K. Shaikh, J. Cilley, W. O'Connor, and A. J. DelRossi, "Intra-operative echocardiography: a useful tool in the localization of small intracardiac foreign bodies," *J. Cardiovascular Surgery*, vol. 30, no. 1, pp. 42–43, 1989.

[34] V. E. Font, C. C. Gill, and D. E. Lammermeier, "Echocardiographically guided removal of an intracardiac foreign body," *Cleveland Clinic Journal of Medicine*, vol. 61, no. 3, pp. 228–231, 1994.

[35] F. Robicsek, J. P. Rielly, and M. C. Marroum, "The use of cyanoacrylate adhesive (Krazy Glue) in cardiac surgery," *J. Cardiac Surgery*, vol. 9, no. 3, pp. 353–356, May 1994.

[36] S. J. Fry, M. H. Picard, J. F. Tseng, S. M. Briggs, and E. M. Isselbacher,

"The echocardiographic diagnosis, characterization, and extraction guidance of cardiac foreign bodies," *J. American Society of Echocardiography*, vol. 13, no. 3, pp. 232–239, Mar. 2000.

[37] M. A. Aguirre, D. Trousdale, A. John, and P. E. Greilich, "The use of transesophageal echocardiography in determining the structural and functional impact of traumatic intracardiac foreign bodies," *Anesthesia and Analgesia*, vol. 107, no. 4, pp. 1155–1157, Oct. 2008.

[38] F. W. Mohr, V. Falk, A. Diegeler, T. Walther, J. F. Gummert, J. Bucerius, S. Jacobs, and R. Autschbach, "Computer-enhanced "robotic" cardiac surgery: experience in 148 patients," *J. Thoracic and Cardiovascular Surgery*, vol. 121, no. 5, pp. 842–53, May 2001.

[39] Y. Nakamura, K. Kishi, and H. Kawakami, "Heartbeat synchronization for robotic cardiac surgery," in *IEEE Int. Conf. on Robotics and Automation (ICRA)*, vol. 2. IEEE, 2001, pp. 2014–2019.

[40] N. A. Patronik, M. A. Zenati, and C. N. Riviere, "Crawling on the heart: a mobile robotic device for minimally invasive cardiac interventions," in *Int. Conf. on Medical Image Computing and Computer-Assisted Intervention (MICCAI)*, ser. Lecture Notes in Computer Science, C. Barillot, D. R. Haynor, and P. Hellier, Eds., vol. 3217. Springer Berlin Heidelberg, 2004, pp. 9–16.

[41] A. Degani, H. Choset, A. Wolf, T. Ota, and M. A. Zenati, "Percutaneous

intrapericardial interventions using a highly articulated robotic probe," in *IEEE RAS/EMBS Int. Conf. on Biomedical Robotics and Biomechatronics (BioRob)*, 2006, pp. 7–12.

[42] A. Degani, H. Choset, A. Wolf, and M. A. Zenati, "Highly articulated robotic probe for minimally invasive surgery," in *IEEE Int. Conf. on Robotics and Automation (ICRA)*, 2006, pp. 4167–4172.

[43] A. Degani, H. Choset, B. Zubiate, T. Ota, and M. Zenati, "Highly articulated robotic probe for minimally invasive surgery," in *IEEE Int. Conf. of the Engineering in Medicine and Biology Society (EMBS)*, 2008, pp. 3273–3276.

[44] 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 *IEEE Int. Conf. of the Engineering in Medicine and Biology Society (EMBS)*, 2008, pp. 250–253.

[45] T. Ota, A. Degani, B. Zubiate, A. Wolf, H. Choset, D. Schwartzman, and M. A. Zenati, "Epicardial atrial ablation using a novel articulated robotic medical probe via a percutaneous subxiphoid approach," *Innovations (Philadelphia, Pa.)*, vol. 1, no. 6, p. 335, 2006.

[46] T. Ota, A. Degani, D. Schwartzman, B. Zubiate, J. McGarvey, H. Choset, and M. A. Zenati, "A highly articulated robotic surgical system for minimally

invasive surgery," *Annals of Thoracic Surgery*, vol. 87, no. 4, pp. 1253–1256, 2009.

[47] S. G. Yuen, S. B. Kesner, N. V. Vasilyev, P. J. Nido, and R. D. Howe, "3D ultrasound-guided motion compensation system for beating heart mitral valve repair," in *Int. Conf. on Medical Image Computing and Computer-Assisted Intervention (MICCAI)*, ser. Lecture Notes in Computer Science, D. Metaxas, L. Axel, G. Fichtinger, and G. Székely, Eds., vol. 5241. Springer Berlin Heidelberg, 2008, pp. 711–719.

[48] S. G. Yuen, N. V. Vasilyev, P. J. del Nido, and R. D. Howe, "Robotic tissue tracking for beating heart mitral valve surgery," *Medical Image Analysis*, vol. 17, no. 8, pp. 1236–1242, 2013.

[49] S. B. Kesner and R. D. Howe, "Design and control of motion compensation cardiac catheters," in *IEEE Int. Conf. on Robotics and Automation (ICRA)*, 2010, pp. 1059–1065.

[50] S. Kesner, S. Yuen, and R. Howe, "Ultrasound servoing of catheters for beating heart valve repair," in *Information Processing in Computer-Assisted Interventions (IPCAI)*, N. Navab and P. Jannin, Eds. Geneva, Switzerland: Springer Berlin Heidelberg, 2010, pp. 168–178.

[51] S. B. Kesner and R. D. Howe, "Design of a motion compensated tissue resection catheter for beating heart cardiac surgery," in *ASME Design of*

Medical Devices Conference, 2011. [Online]. Available: https://biorobotics.harvard.edu/pubs/2011/ref_conf/skesner_DMD2011_TissueResection.pdf

[52] ——, "Force control of flexible catheter robots for beating heart surgery," in *IEEE Int. Conf. on Robotics and Automation (ICRA)*, 2011, pp. 1589–1594.

[53] ——, "Position control of motion compensation cardiac catheters," *IEEE Transactions on Robotics*, vol. 27, no. 6, pp. 1045–1055, 2011.

[54] H. Ren, N. V. Vasilyev, and P. E. Dupont, "Detection of curved robots using 3D ultrasound," in *IEEE/RSJ Int. Conf. on Intelligent Robots and Systems (IROS)*, Sep. 2011, pp. 2083–2089.

[55] P. Abolmaesumi, M. R. Sirouspour, and S. E. Salcudean, "Real-time extraction of carotid artery contours from ultrasound images," in *Computer-Based Medical Systems*, 2000, pp. 181–186.

[56] P. Abolmaesumi, M. R. Sirouspour, S. E. Salcudean, and W. H. Zhu, "Adaptive image servo controller for robot-assisted diagnostic ultrasound," in *IEEE/ASME Int. Conf. on Advanced Intelligent Mechatronics*, vol. 2. IEEE, 2001, pp. 1199–1204.

[57] P. Abolmaesumi, S. E. Salcudean, W. H. Zhu, M. R. Sirouspour, and S. P. DiMaio, "Image-guided control of a robot for medical ultrasound," *IEEE Transactions on Robotics and Automation*, vol. 18, no. 1, pp. 11–23, 2002.

[58] J. Stoll, P. Dupont, and R. D. Howe, "Ultrasound-based servoing of manipulators for telesurgery," in *SPIE Telemanipulator and Telepresence Technologies*, M. R. Stein, Ed., vol. 4570, no. 1. SPIE, Feb. 2002, pp. 78–85.

[59] K. Masuda, E. Kimura, N. Tateishi, and K. Ishihara, "Three dimensional motion mechanism of ultrasound probe and its application for tele-echography system," in *IEEE/RSJ Int. Conf. on Intelligent Robots and Systems (IROS)*. IEEE, 2001, pp. 1112–1116.

[60] A. Vilchis, J. Troccaz, P. Cinquin, K. Masuda, and F. Pellissier, "A new robot architecture for tele-echography," *IEEE Transactions on Robotics and Automation*, vol. 19, no. 5, pp. 922–926, 2003.

[61] J. Hong, T. Dohi, M. Hashizume, K. Konishi, and N. Hata, "An ultrasound-driven needle-insertion robot for percutaneous cholecystostomy," *Physics in Medicine and Biology*, vol. 49, pp. 441–455, 2004.

[62] E. M. Boctor, G. Fischer, M. A. Choti, G. Fichtinger, and R. H. Taylor, "A dual-armed robotic system for intraoperative ultrasound guided hepatic ablative therapy: a prospective study," in *IEEE Int. Conf. on Robotics and Automation (ICRA)*, vol. 3. IEEE, 2004, pp. 2517–2522.

[63] D. Lee, N. Koizumi, K. Ota, S. Yoshizawa, A. Ito, Y. Kaneko, Y. Matsumoto, and M. Mitsuishi, "Ultrasound-based visual servoing system for lithotripsy,"

in *IEEE/RSJ Int. Conf. on Intelligent Robots and Systems (IROS)*, 2007, pp. 877–882.

[64] W. Bachta and A. Krupa, "Towards ultrasound image-based visual servoing," in *IEEE Int. Conf. on Robotics and Automation (ICRA)*, May 2006, pp. 4112–4117.

[65] T. Ortmaier, M.-A. Vitrani, G. Morel, and S. Pinault, "Robust real-time instrument tracking in ultrasound images," in *SPIE Medical Imaging*, W. F. Walker and S. Y. Emelianov, Eds., vol. 5750, no. 1. SPIE, Apr. 2005, pp. 170–177.

[66] ——, "Robust real-time instrument tracking in ultrasound images for visual servoing," in *IEEE Int. Conf. on Robotics and Automation (ICRA)*, 2005.

[67] M.-A. Vitrani, G. Morel, N. Bonnet, and M. Karouia, "A robust ultrasound-based visual servoing approach for automatic guidance of a surgical instrument with in vivo experiments," in *IEEE RAS/EMBS Int. Conf. on Biomedical Robotics and Biomechatronics (BioRob)*. IEEE, 2006, pp. 35–40.

[68] M.-A. Vitrani, H. Mitterhofer, N. Bonnet, and G. Morel, "Robust ultrasound-based visual servoing for beating heart intracardiac surgery," in *IEEE Int. Conf. on Robotics and Automation (ICRA)*, vol. 13, no. 5. IEEE, 2007, pp. 3021–3027.

[69] M. Sauvee, P. Poignet, and E. Dombre, "Ultrasound image-based visual servoing

of a surgical instrument through nonlinear model predictive control," *Int. J. Robotics Research*, vol. 27, no. 1, pp. 25–40, Jan. 2008.

[70] R. Mebarki, A. Krupa, and F. Chaumette, "Image moments-based ultrasound visual servoing," in *IEEE Int. Conf. on Robotics and Automation (ICRA)*, vol. 8, 2008, pp. 113–119.

[71] R. Mebarki, A. Krupa, and C. Collewet, "Automatic guidance of an ultrasound probe by visual servoing based on B-mode image moments," in *Int. Conf. on Medical Image Computing and Computer-Assisted Intervention (MICCAI)*. Springer, 2008, pp. 339–346.

[72] R. Mebarki, A. Krupa, and F. Chaumette, "Modeling and 3D local estimation for in-plane and out-of-plane motion guidance by 2D ultrasound-based visual servoing," in *IEEE Int. Conf. on Robotics and Automation (ICRA)*. IEEE, 2009, pp. 319–325.

[73] ——, "2-D ultrasound probe complete guidance by visual servoing using image moments," *IEEE Transactions on Robotics*, vol. 26, no. 2, pp. 296–306, 2010.

[74] 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 three-dimensional ultrasound for guiding surgical tasks," *Computer Aided Surgery*, vol. 8, no. 2, pp. 82–90, 2003.

[75] M. P. Fronheiser, S. F. Idriss, P. D. Wolf, and S. W. Smith, "Vibrating interven-

tional device detection using real-time 3-D color Doppler," *IEEE Transactions on Ultrasonics, Ferroelectrics, and Frequency Control*, vol. 55, no. 6, pp. 1355–62, 2008.

[76] A. J. Rogers, E. D. Light, and S. W. Smith, "3-D ultrasound guidance of autonomous robot for location of ferrous shrapnel," *IEEE Transactions on Ultrasonics, Ferroelectrics, and Frequency Control*, vol. 56, no. 7, pp. 1301–3, 2009.

[77] J. Mung, F. Vignon, and A. Jain, "A non-disruptive technology for robust 3D tool tracking for ultrasound-guided interventions," in *Int. Conf. on Medical Image Computing and Computer-Assisted Intervention (MICCAI)*, ser. Lecture Notes in Computer Science, G. Fichtinger, A. Martel, and T. Peters, Eds., vol. 6891. Springer Berlin Heidelberg, 2011, pp. 153–160.

[78] P. M. Novotny, J. W. Cannon, and R. D. Howe, "Tool localization in 3D ultrasound images," in *Int. Conf. on Medical Image Computing and Computer-Assisted Intervention (MICCAI)*, ser. Lecture Notes in Computer Science, R. E. Ellis and T. M. Peters, Eds., vol. 2879. Springer Berlin Heidelberg, 2003, pp. 969–970.

[79] M. G. Linguraru, P. M. Novotny, and R. D. Howe, "Enhancement of instrument appearance in ultrasound images by distribution and spatial analysis," in *Int. Conf. on Medical Image Computing and Computer-Assisted Intervention (MICCAI)*, 2005.

[80] M. G. Linguraru and R. D. Howe, “Texture-based instrument segmentation in 3D ultrasound images,” in *SPIE Medical Imaging*, vol. 6144, 2006, p. 61443J.

[81] M. G. Linguraru, N. V. Vasilyev, P. J. D. Nido, and R. D. Howe, “Statistical segmentation of surgical instruments in 3-D ultrasound images,” *Ultrasound in Medicine & Biology*, vol. 33, no. 9, pp. 1428–37, 2007.

[82] M. Uhercík, J. Kybic, H. Liebgott, and C. Cachard, “Model fitting using RANSAC for surgical tool localization in 3-D ultrasound images,” *IEEE Transactions on Biomedical Engineering*, vol. 57, no. 8, pp. 1907–16, Aug. 2010.

[83] R. J. Schneider, D. P. Perrin, N. V. Vasilyev, G. R. Marx, P. J. del Nido, and R. D. Howe, “Mitral annulus segmentation from three-dimensional ultrasound,” in *IEEE Int. Symposium on Biomedical Imaging (ISBI)*, 2009, pp. 779–782.

[84] ——, “Mitral annulus segmentation from 3D ultrasound using graph cuts,” *IEEE Transactions on Medical Imaging*, vol. 29, no. 9, pp. 1676–1687, 2010.

[85] R. J. Schneider, W. C. Burke, G. R. Marx, P. J. Nido, and R. D. Howe, “Modeling mitral valve leaflets from three-dimensional ultrasound,” in *Functional Imaging and Modeling of the Heart*, ser. Lecture Notes in Computer Science, D. N. Metaxas and L. Axel, Eds., vol. 6666. Springer Berlin Heidelberg, 2011, pp. 215–222.

[86] R. J. Schneider, N. A. Tenenholtz, D. P. Perrin, G. R. Marx, P. J. Nido, and

R. D. Howe, "Patient-specific mitral leaflet segmentation from 4D ultrasound," in *Int. Conf. on Medical Image Computing and Computer-Assisted Intervention (MICCAI)*, ser. Lecture Notes in Computer Science, G. Fichtinger, A. Martel, and T. Peters, Eds., vol. 6893. Springer Berlin Heidelberg, 2011, pp. 520–527.

[87] M. G. Linguraru, N. V. Vasilyev, P. J. Nido, and R. D. Howe, "Atrial septal defect tracking in 3D cardiac ultrasound," in *Int. Conf. on Medical Image Computing and Computer-Assisted Intervention (MICCAI)*, ser. Lecture Notes in Computer Science, R. Larsen, M. Nielsen, and J. Sporring, Eds., vol. 4190. Springer Berlin Heidelberg, 2006, pp. 596–603.

[88] M. G. Linguraru, A. Kabla, N. V. Vasilyev, P. J. del Nido, and R. D. Howe, "Real-time block flow tracking of atrial septal defect motion in 4D cardiac ultrasound," in *IEEE Int. Symposium on Biomedical Imaging (ISBI)*, 2007, pp. 356–359.

[89] M. G. Linguraru, A. Kabla, G. R. Marx, P. J. del Nido, and R. D. Howe, "Real-time tracking and shape analysis of atrial septal defects in 3D echocardiography," *Academic Radiology*, vol. 14, no. 11, pp. 1298–1309, 2007.

[90] M. G. Linguraru, N. V. Vasilyev, G. R. Marx, W. Tworetzky, P. J. D. Nido, and R. D. Howe, "Fast block flow tracking of atrial septal defects in 4D echocardiography," *Medical Image Analysis*, vol. 12, no. 4, pp. 397–412, 2008.

[91] S. Xu, J. Kruecker, S. Settlemier, and B. J. Wood, “Real-time motion tracking using 3D ultrasound,” in *SPIE Medical Imaging*, vol. 6509, 2007, p. 65090X.

[92] G. D. Hager and P. N. Belhumeur, “Efficient region tracking with parametric models of geometry and illumination,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 20, no. 10, pp. 1025–1039, 1998.

[93] F. Maffessanti, R. M. Lang, C. Corsi, V. Mor-Avi, and E. G. Caiani, “Feasibility of left ventricular shape analysis from transthoracic real-time 3-D echocardiographic images,” *Ultrasound in Medicine & Biology*, vol. 35, no. 12, pp. 1953–62, 2009.

[94] E. C. Pua, M. P. Fronheiser, J. R. Noble, E. D. Light, P. D. Wolf, D. von Allmen, and S. W. Smith, “3-D ultrasound guidance of surgical robotics: a feasibility study,” *IEEE Transactions on Ultrasonics, Ferroelectrics, and Frequency Control*, vol. 53, no. 11, pp. 1999–2008, 2006.

[95] S. W. Smith, J. Pavy H. G., and O. T. Von Ramm, “High-speed ultrasound volumetric imaging system—Part I: Transducer design and beam steering,” *IEEE Transactions on Ultrasonics, Ferroelectrics, and Frequency Control*, vol. 38, no. 2, pp. 100–108, 1991.

[96] O. T. Von Ramm, S. W. Smith, and J. Pavy H. G., “High-speed ultrasound volumetric imaging system—Part II: Parallel processing and image display,” *IEEE*

Transactions on Ultrasonics, Ferroelectrics, and Frequency Control, vol. 38, no. 2, pp. 109–115, 1991.

[97] J. Whitman, M. P. Fronheiser, N. M. Ivancevich, and S. W. Smith, "Autonomous surgical robotics using 3-D ultrasound guidance: feasibility study," *Ultrasonic Imaging*, vol. 29, no. 4, pp. 213–9, Oct. 2007.

[98] J. Whitman, M. P. Fronheiser, and S. W. Smith, "3-D ultrasound guidance of surgical robotics using catheter transducers: feasibility study," *IEEE Transactions on Ultrasonics, Ferroelectrics, and Frequency Control*, vol. 55, no. 5, pp. 1143–5, 2008.

[99] E. D. Light, J. F. Angle, and S. W. Smith, "Real-time 3-D ultrasound guidance of interventional devices," *IEEE Transactions on Ultrasonics, Ferroelectrics, and Frequency Control*, vol. 55, no. 9, pp. 2066–78, 2008.

[100] M. P. Fronheiser, J. Whitman, N. M. Ivancevich, and S. W. Smith, "3-D ultrasound guidance of surgical robotics: autonomous guidance and catheter transducers," in *IEEE Ultrasonics Symposium*, 2007, pp. 2527–2530.

[101] K. Liang, E. D. Light, A. J. Rogers, D. Von Allmen, and S. W. Smith, "3-D ultrasound guidance of autonomous surgical robotics: Feasibility studies," in *IEEE Int. Ultrasonics Symposium.* IEEE, Sep. 2009, pp. 582–585.

[102] K. Liang, D. V. Allmen, A. J. Rogers, E. D. Light, and S. W. Smith, "Three-

dimensional ultrasound guidance of autonomous robotic breast biopsy: feasibility study," *Ultrasound in Medicine & Biology*, vol. 36, no. 1, pp. 173–7, 2010.

[103] A. Krupa and F. Chaumette, "Control of an ultrasound probe by adaptive visual servoing," in *IEEE/RSJ Int. Conf. on Intelligent Robots and Systems (IROS)*, 2005, pp. 2681–2686.

[104] A. Krupa, "Automatic calibration of a robotized 3D ultrasound imaging system by visual servoing," in *IEEE Int. Conf. on Robotics and Automation (ICRA)*, 2006, pp. 4136–4141.

[105] A. Krupa and F. Chaumette, "Guidance of an ultrasound probe by visual servoing," *Advanced Robotics*, vol. 20, no. 11, pp. 1203–1218, 2006.

[106] A. Krupa, G. Fichtinger, and G. D. Hager, "Full motion tracking in ultrasound using image speckle information and visual servoing," in *IEEE Int. Conf. on Robotics and Automation (ICRA)*. IEEE, 2007, pp. 2458–2464.

[107] J. Stoll, P. Novotny, R. Howe, and P. Dupont, "Real-time 3D ultrasound-based servoing of a surgical instrument," in *IEEE Int. Conf. on Robotics and Automation (ICRA)*. IEEE, 2006, pp. 613–618.

[108] P. M. Novotny, J. A. Stoll, N. V. Vasilyev, P. J. Nido, P. E. Dupont, and R. D. Howe, "GPU based real-time instrument tracking with three dimensional ultrasound," in *Int. Conf. on Medical Image Computing and Computer-Assisted*

Intervention (MICCAI), ser. Lecture Notes in Computer Science, R. Larsen, M. Nielsen, and J. Sporring, Eds., vol. 4190. Springer Berlin Heidelberg, 2006, pp. 58–65.

[109] 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 *IEEE Int. Conf. on Robotics and Automation (ICRA)*. IEEE, 2007, pp. 2655–2660.

[110] P. M. Novotny, J. A. Stoll, N. V. Vasilyev, P. J. Del Nido, P. E. Dupont, T. E. Zickler, and R. D. Howe, "GPU based real-time instrument tracking with three-dimensional ultrasound," *Medical Image Analysis*, vol. 11, no. 5, pp. 458–464, 2007.

[111] P. M. Novotny, D. T. Kettler, P. Jordan, P. E. Dupont, P. J. del Nido, and R. D. Howe, "Stereo display of 3D ultrasound images for surgical robot guidance," in *IEEE Int. Conf. of the Engineering in Medicine and Biology Society (EMBS)*, 2006, pp. 1509–1512.

[112] P. M. Novotny, S. K. Jacobsen, N. V. Vasilyev, D. T. Kettler, I. S. Salgo, P. E. Dupont, P. J. Del Nido, and R. D. Howe, "3D ultrasound in robotic surgery: performance evaluation with stereo displays," *Int. J. Medical Robotics and Computer Assisted Surgery (MRCAS)*, vol. 2, no. 3, pp. 279–85, Sep. 2006.

[113] A. Jain, L. Gutierrez, and D. Stanton, "3D TEE registration with X-ray fluoroscopy for interventional cardiac applications," in *Functional Imaging and*

Modeling of the Heart, N. Ayache, H. Delingette, and M. Sermesant, Eds. Springer Berlin Heidelberg, 2009, pp. 321–329.

[114] R. J. Housden, A. H. Gee, G. M. Treece, and R. W. Prager, "3-D ultrasonic strain imaging using freehand scanning and a mechanically-swept probe," *IEEE Transactions on Ultrasonics, Ferroelectrics, and Frequency Control*, vol. 57, no. 2, pp. 501–6, 2010.

[115] L. Nouaille, P. Vieyres, and G. Poisson, "Process of optimisation for a 4 DOF tele-echography robot," *Robotica*, vol. 30, no. 07, pp. 1131–1145, Dec. 2012.

[116] J. B. Kostis, E. Mavrogeorgis, A. Slater, and S. Bellet, "Use of a range-gated, pulsed ultrasonic Doppler technique for continuous measurement of velocity of the posterior heart wall," *Chest*, vol. 62, no. 5, p. 597, Nov. 1972.

[117] C. E. Rackley, H. T. Dodge, Y. D. Coble, and R. E. Hay, "A method for determining left ventricular mass in man," *Circulation*, vol. 29, no. 5, pp. 666–671, 1964.

[118] K. L. Ryan, W. H. Cooke, C. A. Rickards, K. G. Lurie, and V. A. Convertino, "Breathing through an inspiratory threshold device improves stroke volume during central hypovolemia in humans," *J. Applied Physiology (Bethesda, Md. 1985)*, vol. 104, no. 5, pp. 1402–9, May 2008.

[119] R. Chan, R. Manzke, D. A. Stanton, and G. Schechter, "Anatomically and

functionally accurate soft tissue phantoms and method for generating same," World Patent WO2 008 075 303, Jun. 26, 2008.

[120] J. Funda, R. H. Taylor, B. Eldridge, S. Gomory, and K. G. Gruben, "Constrained Cartesian motion control for teleoperated surgical robots," *IEEE Transactions on Robotics and Automation*, vol. 12, no. 3, pp. 453–465, Jun. 1996.

[121] H. T. Sen, N. Deshmukh, R. Goldman, P. Kazanzides, R. H. Taylor, E. Boctor, and N. Simaan, "Enabling technologies for natural orifice transluminal endoscopic surgery (N.O.T.E.S) using robotically guided elasticity imaging," in *SPIE Medical Imaging*, vol. 8316, Feb. 2012, p. 83161Y.

[122] P. Kazanzides and P. Thienphrapa, "Centralized processing and distributed I/O for robot control," in *IEEE Int. Conf. on Technologies for Practical Robot Applications (TePRA)*, 2008, pp. 84–88.

[123] P. Thienphrapa and P. Kazanzides, "A distributed I/O low-level controller for highly-dexterous snake robots," in *IEEE Biomedical Circuits and Systems Conference (BioCAS)*, 2008, pp. 9–12.

[124] ——, "A scalable system for real-time control of dexterous surgical robots," in *IEEE Int. Conf. on Technologies for Practical Robot Applications (TePRA)*, 2009, pp. 16–22.

[125] P. Thienphrapa, "A centralized processing, distributed I/O motor controller based on IEEE 1394 for the Snake Robot," Ph.D. qualifying project report, Johns Hopkins Univ., May 2010. [Online]. Available: http://cs.jhu.edu/~pault/wiki/images/3/39/Qual2010.pdf

[126] P. Thienphrapa and P. Kazanzides, "Design of a scalable real-time robot controller and application to a dexterous manipulator," in *IEEE Int. Conf. on Robotics and Biomimetics (ROBIO)*, 2011, pp. 2295–2300.

[127] "The *cisst* Libraries and Surgical Assistant Workstation (SAW)." [Online]. Available: https://trac.lcsr.jhu.edu/cisst

[128] R. H. Taylor, J. Funda, B. Eldridge, S. Gomory, K. Gruben, D. LaRose, M. Talamini, L. Kavoussi, and J. Anderson, "A telerobotic assistant for laparoscopic surgery," *IEEE Engineering in Medicine and Biology Magazine*, vol. 14, no. 3, pp. 279–288, 1995.

[129] N. Simaan, R. H. Taylor, and P. Flint, "High dexterity snake-like robotic slaves for minimally invasive telesurgery of the upper airway," in *Int. Conf. on Medical Image Computing and Computer-Assisted Intervention (MICCAI)*, ser. Lecture Notes in Computer Science, C. Barillot, D. R. Haynor, and P. Hellier, Eds., vol. 3217. Springer Berlin Heidelberg, 2004, pp. 17–24.

[130] A. Kapoor, N. Simaan, and P. Kazanzides, "A system for speed and torque

control of DC motors with application to small snake robots," in *IEEE Int. Conf. on Mechatronics and Robotics (MechRob)*, 2004.

[131] A. Kapoor, "Motion constrained control of robots for dexterous surgical tasks," Ph.D. dissertation, Johns Hopkins Univ., Sep. 2007.

[132] N. Simaan, Kai Xu, Wei Wei, A. Kapoor, P. Kazanzides, R. Taylor, and P. Flint, "Design and integration of a telerobotic system for minimally invasive surgery of the throat," *Int. J. Robotics Research*, vol. 28, no. 9, pp. 1134–1153, 2009.

[133] A. Kapoor, N. Simaan, and R. H. Taylor, "Suturing in confined spaces: Constrained motion control of a hybrid 8-DoF robot," in *IEEE Int. Conf. on Advanced Robotics (ICAR)*, Jul. 2005, pp. 452–459.

[134] M. Li and R. H. Taylor, "Optimum robot control for 3D virtual fixture in constrained ENT surgery," in *Int. Conf. on Medical Image Computing and Computer-Assisted Intervention (MICCAI)*, ser. Lecture Notes in Computer Science, R. E. Ellis and T. M. Peters, Eds., vol. 2878. Springer Berlin Heidelberg, 2003, pp. 165–172.

[135] ——, "Spatial motion constraints in medical robot using virtual fixtures generated by anatomy," in *IEEE Int. Conf. on Robotics and Automation (ICRA)*, vol. 2, Apr. 2004, pp. 1270–1275.

[136] ——, "Performance of surgical robots with automatically generated spatial vir-

tual fixtures," in *IEEE Int. Conf. on Robotics and Automation (ICRA)*, Apr. 2005, pp. 217–222.

[137] M. Li, A. Kapoor, and R. H. Taylor, "A constrained optimization approach to virtual fixtures," in *IEEE/RSJ Int. Conf. on Intelligent Robots and Systems (IROS)*, Aug. 2005, pp. 1408–1413.

[138] M. Li, M. Ishii, and R. H. Taylor, "Spatial motion constraints using virtual fixtures generated by anatomy," *IEEE Transactions on Robotics*, vol. 23, no. 1, pp. 4–19, Feb. 2007.

[139] A. Kapoor, M. Li, and R. H. Taylor, "Spatial motion constraints for robot assisted suturing using virtual fixtures," in *Int. Conf. on Medical Image Computing and Computer-Assisted Intervention (MICCAI)*, ser. Lecture Notes in Computer Science, J. S. Duncan and G. Gerig, Eds., vol. 3750. Springer Berlin Heidelberg, 2005, pp. 89–96.

[140] ——, "Constrained control for surgical assistant robots," in *IEEE Int. Conf. on Robotics and Automation (ICRA)*, 2006, pp. 231–236.

[141] A. Kapoor and R. H. Taylor, "A constrained optimization approach to virtual fixtures for multi-handed tasks," in *IEEE Int. Conf. on Robotics and Automation (ICRA)*, May 2008, pp. 3401–3406.

[142] T. Xia, A. Kapoor, P. Kazanzides, and R. H. Taylor, "A constrained optimiza-

tion approach to virtual fixtures for multi-robot collaborative teleoperation," in *IEEE/RSJ Int. Conf. on Intelligent Robots and Systems (IROS)*, Sep. 2011, pp. 639–644.

[143] P. Thienphrapa, B. Ramachandran, and A. Popovic, "Automatic online registration between a robot and images," World Patent WO2 013 038 301, Mar. 21, 2013.

[144] S. Voros, J.-A. Long, and P. Cinquin, "Automatic detection of instruments in laparoscopic images: A first step towards high-level command of robotic endoscopic holders," *Int. J. Robotics Research*, vol. 26, no. 11–12, pp. 1173–1190, 2007.

[145] P. Thienphrapa, H. Elhawary, B. Ramachandran, D. Stanton, and A. Popovic, "Tracking and characterization of fragments in a beating heart using 3D ultrasound for interventional guidance," in *Int. Conf. on Medical Image Computing and Computer-Assisted Intervention (MICCAI)*, Sep. 2011, pp. 211–218.

[146] B. Ramachandran, P. Thienphrapa, A. Jain, and A. Popovic, "Tracking using 3D ultrasound for guiding cardiac interventions," in *Int. Conf. on Biomedical Engineering*, 2011.

[147] R. C. Gonzalez and R. E. Woods, *Digital Image Processing.* Prentice Hall, 2008.

[148] J. P. Lewis, “Fast normalized cross-correlation,” in *Vision Interface*, vol. 10, no. 1, 1995, pp. 120–123.

[149] D. Eaton, “Fast normalized cross-correlation in 3D.” [Online]. Available: http://www.cs.ubc.ca/~deaton/tut/normxcorr3.html

[150] P. Thienphrapa, B. Ramachandran, R. H. Taylor, and A. Popovic, “A system for 3D ultrasound-guided robotic retrieval of foreign bodies from a beating heart,” in *IEEE RAS/EMBS Int. Conf. on Biomedical Robotics and Biomechatronics (BioRob)*, 2012, pp. 743–748.

[151] A. Popovic and P. Thienphrapa, “An approach to robotic guidance of an uncalibrated endoscope in beating heart surgery,” in *IEEE RAS/EMBS Int. Conf. on Biomedical Robotics and Biomechatronics (BioRob)*, 2010, pp. 106–113.

[152] P. Thienphrapa, B. Ramachandran, H. Elhawary, R. H. Taylor, and A. Popovic, “Multiple capture locations for 3D ultrasound-guided robotic retrieval of moving bodies from a beating heart,” in *SPIE Medical Imaging*, vol. 8316, Feb. 2012, p. 831619.

[153] R. L. J. Tawes, “Intraoperative autotransfusion: advantages, disadvantages, and contraindications,” *Seminars in Vascular Surgery*, vol. 7, no. 2, pp. 95–97, Jun. 1994.

[154] P. Thienphrapa, B. Ramachandran, H. Elhawary, A. Popovic, and R. H. Taylor,

"Intraoperative analysis of locations for 3D ultrasound-guided capture of foreign bodies from a beating heart," in *Hamlyn Symposium on Medical Robotics*, 2012.

[155] "JHU Open Source Mechatronics." [Online]. Available: http://jhu-cisst.github.io/mechatronics/

[156] Z. Chen, A. Deguet, R. H. Taylor, S. DiMaio, G. Fischer, and P. Kazanzides, "An open-source hardware and software platform for telesurgical robot research," *MICCAI Workshop on Systems and Architecture for Computer Assisted Interventions*, Sep. 2013. [Online]. Available: http://hdl.handle.net/10380/3419

[157] P. Kazanzides, Z. Chen, A. Deguet, G. S. Fischer, R. H. Taylor, and S. DiMaio, "An open-source research kit for the da Vinci® surgical robot," in *IEEE Int. Conf. on Robotics and Automation (ICRA)*, May 2014.

[158] "Intuitive Research Kit." [Online]. Available: https://trac.lcsr.jhu.edu/cisst/wiki/sawIntuitiveResearchKitTutorial

[159] M. O. F. Sarker, C. H. Kim, S. Baek, and B.-J. You, "An IEEE-1394 based real-time robot control system for efficient controlling of humanoids," in *IEEE/RSJ Int. Conf. on Intelligent Robots and Systems (IROS)*, Oct. 2006, pp. 1416–1421.

[160] Y. Zhang, B. Orlic, P. M. Visser, and J. F. Broenink, "Hard real-time network-

ing on FireWire," in *Real-Time Linux Workshop*. Eindhoven, The Netherlands: IOS Press, Nov. 2005, pp. 1–8.

[161] G. A. Pratt, P. Willisson, C. Bolton, and A. Hofman, "Late motor processing in low-impedance robots: Impedance control of series-elastic actuators," in *American Control Conference*, vol. 4, Jun. 2004, pp. 3245–3251.

[162] T. K. Lewellen, C. M. Laymon, R. S. Miyaoka, K. S. Lee, and P. E. Kinahan, "Design of a FireWire based data acquisition system for use in animal PET scanners," in *IEEE Nuclear Science Symposium Conference Record*, vol. 4, Nov. 2001, pp. 1974–1978.

[163] S. Schneider, "Making Ethernet work in real time," *Sensors Magazine*, vol. 17, no. 11, Nov. 2000.

[164] J. Kerkes, "Real-time Ethernet," *Embedded Systems Design*, vol. 14, no. 1, Jan. 2001.

[165] "Ethernet Powerlink." [Online]. Available: http://www.ethernet-powerlink.org/

[166] S.-Y. Lin, C.-Y. Ho, and Y.-Y. Tzou, "Distributed motion control using real-time network communication techniques," in *Int. Power Electronics and Motion Control Conference (IPEMC)*, vol. 2, Aug. 2000, pp. 843–847.

[167] "EtherCAT Technology Group." [Online]. Available: http://www.ethercat.org/

[168] S. G. Robertz, K. Nilsson, R. Henriksson, and A. Blomdell, "Industrial robot motion control with real-time Java and EtherCAT," in *IEEE Conf. on Emerging Technologies and Factory Automation (ETFA)*, Sep. 2007, pp. 1453–1456.

[169] U. Hagn, M. Nickl, S. Jörg, G. Passig, T. Bahls, A. Nothhelfer, F. Hacker, L. Le-Tien, A. Albu-Schäffer, R. Konietschke, M. Grebenstein, R. Warpup, R. Haslinger, M. Frommberger, and G. Hirzinger, "The DLR MIRO: a versatile lightweight robot for surgical applications," *Industrial Robot: An International Journal*, vol. 35, no. 4, pp. 324–336, 2008.

[170] C. Szydlowski, "Implementing PCI Express for industrial control," *RTC Magazine*, vol. 13, Sep. 2004.

[171] "Controller Area Network." [Online]. Available: http://www.can-cia.org/

[172] B. Walls, M. McClelland, S. Persyn, and D. M. Werner, "Leveraging flight heritage to new CompactPCI space systems: a fusion of architectures," in *Digital Avionics Systems Conference (DASC)*, vol. 2, Oct. 2001, pp. 8C4/1–8C4/7.

[173] G. Baltazar and G. P. Chapelle, "FireWire in modern integrated military avionics," *IEEE Aerospace and Electronic Systems Magazine*, vol. 16, no. 11, pp. 12–16, Nov. 2001.

[174] P. Kazanzides, "Bridge motor amplifier with speed control," Internal report, Johns Hopkins Univ., Dec. 2006.

[175] R. Huang, P. Thienphrapa, and P. Kazanzides, “Calibration of the Snake amplifier board,” Internal report, Johns Hopkins Univ., May 2009.

[176] *Cyclone II Device Handbook.* Altera Corporation, 2008.

[177] “LCSR Electronics Repository (Protected).” [Online]. Available: https://trac.lcsr.jhu.edu/electronics/

[178] M. O. F. Sarker, C. H. Kim, J.-S. Cho, and B.-J. You, “Development of a network-based real-time robot control system over IEEE 1394: Using open source software platform,” in *IEEE Int. Conf. on Mechatronics*, Jul. 2006, pp. 563–568.

[179] B. Vagvolgyi, S. DiMaio, A. Deguet, P. Kazanzides, R. Kumar, C. Hasser, and R. H. Taylor, “The Surgical Assistant Workstation: A software framework for telesurgical robotics research,” *MICCAI Workshop on Systems and Architecture for Computer Assisted Interventions*, Jul. 2008. [Online]. Available: http://hdl.handle.net/10380/1466

[180] American Heart Association, “Heart disease and stroke statistics: 2005 update. Dallas, Tex.” *American Heart Association*, 2005.

[181] P. W. Serruys, M.-C. Morice, A. P. Kappetein, A. Colombo, D. R. Holmes, M. J. Mack, E. Stå hle, T. E. Feldman, M. van den Brand, E. J. Bass, N. Van Dyck, K. Leadley, K. D. Dawkins, and F. W. Mohr, “Percutaneous coronary

intervention versus coronary-artery bypass grafting for severe coronary artery disease," *New England Journal of Medicine*, vol. 360, no. 10, pp. 961–972, 2009.

[182] M. A. Hlatky, D. B. Boothroyd, K. A. Melsop, M. M. Brooks, D. B. Mark, B. Pitt, G. S. Reeder, W. J. Rogers, T. J. Ryan, P. L. Whitlow, and R. D. Wiens, "Medical costs and quality of life 10 to 12 years after randomization to angioplasty or bypass surgery for multivessel coronary artery disease," *Circulation*, vol. 110, no. 14, pp. 1960–1966, 2004.

[183] K. Shirai, A. J. Lansky, R. Mehran, G. D. Dangas, C. O. Costantini, M. Fahy, S. Slack, G. S. Mintz, G. W. Stone, and M. B. Leon, "Minimally invasive coronary artery bypass grafting versus stenting for patients with proximal left anterior descending coronary artery disease," *American Journal of Cardiology*, vol. 93, no. 8, pp. 959–962, 2004.

[184] A. T. Cheung, G. E. Cruz-Shiavone, Q. C. Meng, A. Pochettino, J. A. Augoustides, J. E. Bavaria, and E. A. Ochroch, "Cardiopulmonary bypass, hemolysis, and nitroprusside-induced cyanide production," *Anesthesia & Analgesia*, vol. 105, no. 1, pp. 29–33, 2007.

[185] H. Shennib, A. Bastawisy, M. J. Mack, and F. H. Moll, "Computer-assisted telemanipulation: an enabling technology for endoscopic coronary artery bypass," *Annals of Thoracic Surgery*, vol. 66, no. 3, pp. 1060–1063, 1998.

[186] M. J. Mack, "Minimally invasive cardiac surgery," *Surgical Endoscopy And Other Interventional Techniques*, vol. 20, no. 2, pp. S488–S492, 2006.

[187] V. Falk, S. Jacobs, and F.-W. W. Mohr, "Closed-chest, robotically assisted CABG," *Multimedia Manual of Cardio-Thoracic Surgery*, vol. 2006, no. 0315, 2006.

[188] V. Falk, S. Jacobs, J. F. Gummert, T. Walther, and F. W. Mohr, "Computer-enhanced endoscopic coronary artery bypass grafting: the da Vinci experience," *Seminars in Thoracic and Cardiovascular Surgery*, vol. 15, no. 2, pp. 104–111, 2003.

[189] J. G. Holden, J. M. Flach, and Y. Donchin, "Perceptual-motor coordination in an endoscopic surgery simulation," *Surgical Endoscopy*, vol. 13, no. 2, pp. 127–132, 1999.

[190] R. H. M. Goossens and M. A. van Veelen, "Assessment of ergonomics in laparoscopic surgery," *Minimally Invasive Therapy & Allied Technologies*, vol. 10, no. 3, pp. 175–179, 2001.

[191] H. A. Tabaie, J. A. Reinbolt, W. P. Graper, T. F. Kelly, and M. A. Connor, "Endoscopic coronary artery bypass graft (E-CABG) procedure with robotic assistance," in *Heart Surgery Forum*, vol. 2, no. 4, 1999, pp. 310–317.

[192] R. P. J. Budde, T. C. Dessing, R. Meijer, P. F. A. Bakker, C. Borst, and

P. F. Gründeman, “Robot-assisted 13 MHz epicardial ultrasound for endoscopic quality assessment of coronary anastomoses,” *Interactive Cardiovascular and Thoracic Surgery*, vol. 3, no. 4, pp. 616–620, 2004.

[193] Y.-H. Liu, H. Wang, C. Wang, and K. K. Lam, “Uncalibrated visual servoing of robots using a depth-independent interaction matrix,” *IEEE Transactions on Robotics*, vol. 22, no. 4, pp. 804–817, 2006.

[194] E. Malis, “Visual servoing invariant to changes in camera-intrinsic parameters,” *IEEE Transactions on Robotics and Automation*, vol. 20, no. 1, pp. 72–81, 2004.

[195] W. J. Wilson, C. C. Williams Hulls, and G. S. Bell, “Relative end-effector control using Cartesian position based visual servoing,” *IEEE Transactions on Robotics and Automation*, vol. 12, no. 5, pp. 684–696, 1996.

[196] X. Zhang and S. Payandeh, “Application of visual tracking for robot-assisted laparoscopic surgery,” *J. Robotic Systems*, vol. 19, no. 7, pp. 315–328, 2002.

[197] A. Krupa, J. Gangloff, C. Doignon, M. F. de Mathelin, G. Morel, J. Leroy, L. Soler, and J. Marescaux, “Autonomous 3-D positioning of surgical instruments in robotized laparoscopic surgery using visual servoing,” *IEEE Transactions on Robotics and Automation*, vol. 19, no. 5, pp. 842–853, 2003.

[198] P. Hynes, G. I. Dodds, and A. J. Wilkinson, “Uncalibrated visual-servoing

of a dual-arm robot for MIS suturing," in *IEEE RAS/EMBS Int. Conf. on Biomedical Robotics and Biomechatronics (BioRob)*, 2006, pp. 420–425.

[199] M. Jagersand, O. Fuentes, and R. Nelson, "Experimental evaluation of uncalibrated visual servoing for precision manipulation," in *IEEE Int. Conf. on Robotics and Automation (ICRA)*, vol. 4, 1997, pp. 2874–2880.

[200] G.-Q. Wei, K. Arbter, and G. Hirzinger, "Real-time visual servoing for laparoscopic surgery. Controlling robot motion with color image segmentation," *IEEE Engineering in Medicine and Biology Magazine*, vol. 16, no. 1, pp. 40–45, 1997.

[201] A. Popovic, "Real-time feature tracking for image-guided robotics in cardiac endoscopy," *Int. J. of Computer Assisted Radiology and Surgery*, vol. 5, pp. 142–143, 2010.

[202] K. Höller, J. Penne, A. Schneider, J. Jahn, J. Guttiérrez Boronat, T. Wittenberg, H. Feuß ner, and J. Hornegger, "Endoscopic orientation correction," in *Int. Conf. on Medical Image Computing and Computer-Assisted Intervention (MICCAI)*, ser. Lecture Notes in Computer Science, G.-Z. Yang, D. Hawkes, D. Rueckert, A. Noble, and C. Taylor, Eds., vol. 5761. Springer Berlin Heidelberg, 2009, pp. 459–466.

Vita

Paul Thienphrapa was born and raised in Los Angeles, California, with 4.5 pre-high school years spent growing up in Chiang Mai, Thailand. He received the B.S. degree in electrical and computer engineering from the California Institute of Technology. While enrolled in the MSEE program at California State University, Los Angeles, Paul was a research assistant in its Structures, Pointing, and Control Engineering (SPACE) Laboratory. He phased into a software engineering position at Sierra Monolithics, Inc. (Redondo Beach, California), then over a year later he entered the Ph.D. program in electrical engineering at Johns Hopkins University. After a transfer of departments, he received the MSE degree in computer science. Over his academic career, Paul has received a number of merit-based scholarships.

Paul is now a research scientist with Philips Research North America (Briarcliff Manor, New York), simultaneously completing the Ph.D. degree in computer science at Johns Hopkins University under the advisement of Prof. Russell H. Taylor. In his previous role he was a research assistant in the Department of Computer Science, the Engineering Research Center for Computer Integrated Surgical Systems and Technologies (ERC CISST), and the Laboratory for Computational Sensing and Robotics (LCSR) at Johns Hopkins University. As a Ph.D. candidate he has worked annually as a summer intern at Philips Research on image guided surgical robotics projects; he has also served as teaching assistant and grader for similarly-themed courses. His research interests include robot assisted minimally invasive surgery, image guided surgery, and the engineering of electronics, firmware, drivers, and application software for real-time robotic and surgical systems. He is the author of several peer-reviewed technical papers and has served as a reviewer for multiple publications. He is co-inventor of three patented technologies, and has been a member of the Institute of Electrical and Electronics Engineers (IEEE) for the past 10 years. Paul is a regular volunteer in academic and technical outreach activities such as youth robotic competitions and tutoring.

An avid participant in athletic activities, Paul served as captain for his high school varsity soccer and cross country teams, while earning all-conference honors and reaching the Los Angeles city finals in track and field's triple jump event. As an undergraduate he was a four-year letterman in soccer and track and field, competing at the Division III level of the National Collegiate Athletic Association (NCAA). Paul has also won several intramural titles, incurring a number of hospital visits en route.