

University of Nebraska - Lincoln

DigitalCommons@University of Nebraska - Lincoln

Mechanical (and Materials) Engineering --
Dissertations, Theses, and Student Research

Mechanical & Materials Engineering,
Department of

10-2009

Towards Supervised Autonomous Task Completion Using an *in vivo* Surgical Robot

Jason J. Dumpert

University of Nebraska at Lincoln, jasondumpert@gmail.com

Follow this and additional works at: <https://digitalcommons.unl.edu/mechengdiss>



Part of the [Biomedical Commons](#), [Biomedical Devices and Instrumentation Commons](#), [Mechanical Engineering Commons](#), and the [Robotics Commons](#)

Dumpert, Jason J., "Towards Supervised Autonomous Task Completion Using an *in vivo* Surgical Robot" (2009). *Mechanical (and Materials) Engineering – Dissertations, Theses, and Student Research*. 4.
<https://digitalcommons.unl.edu/mechengdiss/4>

This Article is brought to you for free and open access by the Mechanical & Materials Engineering, Department of at DigitalCommons@University of Nebraska - Lincoln. It has been accepted for inclusion in Mechanical (and Materials) Engineering – Dissertations, Theses, and Student Research by an authorized administrator of DigitalCommons@University of Nebraska - Lincoln.

TOWARDS SUPERVISED AUTONOMOUS TASK COMPLETION USING AN *IN*
VIVO SURGICAL ROBOT

by

Jason James Dumpert

A DISSERTATION

Presented to the Faculty of
The Graduate College at the University of Nebraska
In Partial Fulfillment of Requirements
For the Degree of Doctor of Philosophy

Major: Engineering

Under the Supervision of Professor Shane M. Farritor

Lincoln, Nebraska

December, 2009

TOWARDS SUPERVISED AUTONOMOUS TASK COMPLETION USING AN *IN VIVO* SURGICAL ROBOT

Jason James Dumpert, Ph.D.
University of Nebraska, 2009

Advisor: Shane M. Farritor

Laparoscopy is a minimally invasive alternative to traditional abdominal surgery. Unlike traditional surgery, a laparoscopic procedure can be completed using small incisions. The use of these small incision results in reduced pain to the patient, shorter recovery times, and less trauma to skin, muscle and other tissues. However, these benefits to the patient are offset by the increased difficulty to the surgeon performing the procedure. These difficulties include reduced dexterity, reduced perception, and longer procedure times. The use of small *in vivo* robotic devices in minimally invasive surgery is one possible solution to these problems. The movement of these devices is not constrained by the position of the entry incision, because the devices would be completely intracorporeal. In addition to improving the quality of minimally invasive surgery, devices such as these could be used to perform supervised autonomous surgical tasks over a high latency communications channel. This dissertation discusses the contributions of the author towards the goal of creating surgical robots that can perform supervised autonomous surgical tasks. First, the design and testing of several *in vivo* robotic devices is described. Next, experimental results using visual quality metrics comparing *in vivo* cameras to laparoscopes are presented. Next, experiments conducted with the cooperation of NASA during the NEEMO 9 mission are discussed. These experiments compared the usefulness of *in vivo* robots to laparoscopes in simulated surgical tasks. Next, a sterilizable camera device was designed, and then tested in three survivable pig surgeries. The device was shown to cause no tissue damage or infection, and was used as the sole visual feedback device for a laparoscopic cholecystectomy. Finally, a prototype system was developed to demonstrate that a dexterous manipulator device could be used to perform supervised autonomous surgical tasks. A closed loop controller using visual feedback was implemented to control the robot. Benchtop tests demonstrating supervised autonomous task completion are presented. The author believes this work represents some work in using *in vivo* surgical robots to automate surgical tasks.

ACKNOWLEDGEMENTS

I would like to thank my family for all their love and support through the years. I could not have made it to this point without you. I'd also like to thank Professor Shane Farritor, my advisor for both my master's and PhD research, for his help and advice throughout my time at UNL. I'm very grateful for the opportunity to participate in this exciting research. Finally, I'd like to thank Renee for her patience, encouragement, love and support over the past few years.

Contents

| | | |
|----------|--|-----------|
| 1 | Introduction | 1 |
| 1.1 | Literature Review | 2 |
| 1.1.1 | Minimally Invasive Surgery | 2 |
| 1.1.2 | Robotic Surgery | 3 |
| 1.1.3 | Robotic Endoscopes | 6 |
| 1.1.4 | NOTES | 8 |
| 1.1.5 | Visual Servoing | 9 |
| 1.1.6 | Tracking | 10 |
| 1.2 | Dissertation Objective and Contributions | 11 |
| 1.3 | Dissertation Outline | 13 |
| 2 | In-Vivo Robots | 14 |
| 2.1 | Manipulator (M3-20a) | 14 |
| 2.1.1 | Author's Contributions on M3-20a | 16 |
| 2.2 | Crawler (MR-15a-c) | 19 |
| 2.2.1 | Author's Contributions on MR-15a-c | 21 |
| 2.3 | Pan and Tilt (PT-15) | 23 |
| 2.3.1 | Author's Contributions on PT-15a-b | 26 |
| 2.4 | Wireless Camera (WC-15a) | 26 |
| 2.4.1 | Author's Contributions on WC-15a | 27 |
| 2.5 | Adjustable Focus (AF-20a) | 28 |
| 2.5.1 | Author's Contributions on AF-20a | 29 |
| 2.6 | Crawler (MR-20a) | 32 |
| 2.6.1 | Author's Contributions on MR-20a | 33 |
| 2.7 | Crawler (MR-15d) | 33 |
| 2.7.1 | Author's Contributions on MR-15d | 35 |
| 2.8 | Crawler (MR-20b) | 39 |
| 2.8.1 | Author's Contributions on MR-20b | 39 |
| 2.9 | Tilt Camera(T-15a) | 40 |
| 2.9.1 | Author's Contributions on T-15a | 41 |
| 2.10 | Crawler (MR-20c) | 42 |
| 2.10.1 | Author's Contributions on MR-20c | 43 |
| 2.11 | Tilt Camera(T-15b) | 47 |

| | | |
|----------|---|------------|
| 2.11.1 | Author's Contributions on T-15b | 48 |
| 2.12 | Biopsy Crawler (BMR-22a) | 49 |
| 2.12.1 | Author's Contributions on BMR-22a | 50 |
| 2.13 | Wireless Crawler (WMR-22a) | 51 |
| 2.13.1 | Author's Contributions on WMR-22a | 52 |
| 2.14 | Crawler (MR-12a) | 53 |
| 2.14.1 | Author's Contributions on MR-12a | 54 |
| 2.15 | Mobile Camera (MRC-100) | 54 |
| 2.15.1 | Author's Contributions on MRC-100 | 57 |
| 2.16 | Ceiling Pan-Tilt (CPT-100) | 58 |
| 2.16.1 | Author's Contributions on CPT-100 | 60 |
| 2.17 | CPT-MRC Software | 61 |
| 2.18 | Dexterous Manipulator (AB1) | 63 |
| 2.18.1 | Author's Contributions on AB1 | 66 |
| 2.19 | Dexterous Manipulator (AB2) | 67 |
| 2.19.1 | Author's Contributions on AB2 | 69 |
| 2.20 | Dexterous Manipulator (AB3) | 70 |
| 2.20.1 | Author's Contributions on AB3 | 72 |
| 2.21 | AB Robot Software | 73 |
| 2.22 | Summary | 79 |
| 3 | Camera Comparisons | 82 |
| 3.1 | Background and Motivation | 82 |
| 3.2 | Methods and Materials | 82 |
| 3.3 | Experimental Results | 86 |
| 3.3.1 | MTF Testing | 86 |
| 3.3.2 | Color Accuracy | 89 |
| 3.3.3 | Distortion | 89 |
| 3.4 | Summary | 92 |
| 4 | NEEMO Experiments | 94 |
| 4.1 | NEEMO Mission Background | 94 |
| 4.2 | Experiments | 96 |
| 4.3 | Results and Analysis | 100 |
| 4.4 | Summary | 106 |
| 5 | Survivable Pig Studies | 108 |
| 5.1 | Ceiling Pan-tilt (CPT) Camera | 109 |
| 5.2 | Pre-Operative Procedure | 109 |
| 5.3 | Operative Procedure | 112 |
| 5.4 | Post-Operative Observations | 114 |
| 5.4.1 | Animal A | 115 |
| 5.4.2 | Animal B | 115 |

| | | |
|----------|--|------------|
| 5.4.3 | Animal C | 116 |
| 5.5 | Summary | 116 |
| 6 | Surgical Tasks | 118 |
| 6.1 | A Dexterous <i>In vivo</i> Manipulator Robot | 119 |
| 6.2 | Supervised Autonomous Tasks | 120 |
| 6.2.1 | Tracker | 121 |
| 6.2.2 | Controller | 135 |
| 6.2.3 | Stereo Vision | 146 |
| 6.2.4 | Software | 150 |
| 6.3 | Results | 155 |
| 6.4 | Summary | 166 |
| 7 | Conclusions and Future Work | 168 |
| 7.1 | Statistical Comparisons | 169 |
| 7.2 | Supervised Autonomous Task Completion | 170 |
| A | Electronics | 173 |
| | Bibliography | 189 |

List of Figures

| | | |
|------|--|----|
| 2.1 | M3-20a Robot | 15 |
| 2.2 | M3-20a System Flowchart | 15 |
| 2.3 | M3-20a Control Software | 17 |
| 2.4 | M3-20a Class Diagram | 18 |
| 2.5 | Crawler (MR-15b) | 19 |
| 2.6 | MR-15 System Flowchart | 20 |
| 2.7 | MR-15b Electronics | 21 |
| 2.8 | MR-15c Electronics | 22 |
| 2.9 | MR-15c Software | 22 |
| 2.10 | Pan and Tilt Devices | 24 |
| 2.11 | PT-15 Block Diagram | 25 |
| 2.12 | Wireless Camera (WC-15a) | 27 |
| 2.13 | Adjustable Focus (AF-20a) | 28 |
| 2.14 | AF-20a Block Diagram | 29 |
| 2.15 | AF-20a <i>In vivo</i> Gall Bladder Image | 30 |
| 2.16 | AF-20a User Interface | 31 |
| 2.17 | AF-20a Software | 31 |
| 2.18 | Crawler (MR-20a) | 32 |
| 2.19 | MR-20a Block Diagram | 33 |
| 2.20 | Crawler (MR-15d) | 34 |
| 2.21 | Joystick Mapping GUI | 35 |
| 2.22 | MR-15d Drive Electronics | 35 |
| 2.23 | MR-15d Collaboration Diagram | 36 |
| 2.24 | Crawler Control: Main window | 38 |
| 2.25 | MR-20b Drive Electronics | 39 |
| 2.26 | T-15a Fixed Base Camera | 40 |
| 2.27 | T-15a Drive Electronics | 41 |
| 2.28 | Crawler (MR-20c) | 42 |
| 2.29 | MR-20c Block Diagram | 43 |
| 2.30 | MR-20c (NEEMO variant) | 44 |
| 2.31 | MR-20c/T-15b Control Box | 45 |
| 2.32 | MR-20c/T-15b Control Box Block Diagram | 46 |
| 2.33 | Control Box Software | 47 |

| | | |
|------|---------------------------------------|----|
| 2.34 | Tilt Camera (T-15b) | 48 |
| 2.35 | T-15b Block Diagram | 48 |
| 2.36 | T-15b NEEMO variant | 49 |
| 2.37 | BMR-22a Biopsy Crawler | 50 |
| 2.38 | BMR-22a Block Diagram | 50 |
| 2.39 | WMR-22a Wireless Crawler | 52 |
| 2.40 | WMR-22a Block Diagram | 52 |
| 2.41 | MR-12a Endoluminal Crawler | 53 |
| 2.42 | MRC-100 Version A | 55 |
| 2.43 | MRC-100 Version B | 56 |
| 2.44 | MRC-100/CPT-100 Block Diagram | 57 |
| 2.45 | MRC-100 PCB | 58 |
| 2.46 | CPT-100 Magnetic Attachment | 59 |
| 2.47 | CPT-100 Version A | 59 |
| 2.48 | CPT-100 Version B | 60 |
| 2.49 | CPT-100 PCB | 61 |
| 2.50 | CPT-MRC Software | 62 |
| 2.51 | CPT-MRC User Interface | 63 |
| 2.52 | AB1 Robot | 64 |
| 2.53 | AB1 Block Diagram | 65 |
| 2.54 | Virtual Incision Controller | 66 |
| 2.55 | AB1 <i>In vivo</i> | 67 |
| 2.56 | AB2 | 68 |
| 2.57 | AB2 Block Diagram | 69 |
| 2.58 | Virtual Incision Controller 2.0 | 69 |
| 2.59 | AB3 | 70 |
| 2.60 | AB3 Block Diagram | 71 |
| 2.61 | AB3 PCB | 72 |
| 2.62 | AB3 <i>In vivo</i> view | 72 |
| 2.63 | AB Robot Control Main Window | 74 |
| 2.64 | AB Robot Joint Control Dialog | 75 |
| 2.65 | AB Robot Joystick Control Dialog | 76 |
| 2.66 | AB Robot Open Loop Dialog | 77 |
| 2.67 | AB Software Collaboration Diagram | 78 |
| 3.1 | Vision Targets | 84 |
| 3.2 | Experimental Setup | 85 |
| 3.3 | Laparoscope Control and Video Capture | 85 |
| 3.4 | MTF Target | 87 |
| 3.5 | MTF Curves for Both Systems | 88 |
| 3.6 | MTF Comparison | 88 |
| 3.7 | Micron Camera Color Error | 90 |
| 3.8 | Laparoscope Color Error | 90 |

| | | |
|------|--|-----|
| 3.9 | Color Error for Both Systems | 91 |
| 3.10 | Lens Distortion | 91 |
| 4.1 | External View of Aquarius Habitat | 95 |
| 4.2 | Abdominal Cavity Simulator | 97 |
| 4.3 | Rope Passing Setup | 98 |
| 4.4 | Stretch and Dissect Task Setup | 99 |
| 4.5 | Appendectomy Task Setup | 100 |
| 4.6 | Rope Passing Task | 101 |
| 4.7 | Time Difference Graphs | 102 |
| 4.8 | Stretch-Dissect Task | 103 |
| 4.9 | Appendectomy Task | 104 |
| 5.1 | CPT-100 Robotic Camera | 109 |
| 5.2 | OR Layout for Survivable Surgeries | 111 |
| 5.3 | <i>In vivo</i> Images from CPT | 114 |
| 6.1 | AB3 Robot | 119 |
| 6.2 | System Block Diagram | 121 |
| 6.3 | HSV Color Space | 122 |
| 6.4 | Large Bowel | 124 |
| 6.5 | Liver and Abdominal Wall | 125 |
| 6.6 | Tracker Marker Obstruction | 126 |
| 6.7 | Tracker Operation | 127 |
| 6.8 | Hue-Saturation Masks | 128 |
| 6.9 | Image Coordinate Conversion | 129 |
| 6.10 | AB Robot Measurements | 130 |
| 6.11 | Tracker Performance | 134 |
| 6.12 | Joint Control System | 137 |
| 6.13 | PID Controller Implementation | 138 |
| 6.14 | Position and PID Internal Values | 141 |
| 6.15 | Gripper Shoulder Models | 142 |
| 6.16 | Cautery Shoulder Models | 143 |
| 6.17 | Gripper Wrist Models | 144 |
| 6.18 | Cautery Wrist Models | 145 |
| 6.19 | Single Camera | 147 |
| 6.20 | Dual Cameras | 148 |
| 6.21 | Stereo Vertical Adjustment | 150 |
| 6.22 | Supervised Autonomous Task User Interface | 153 |
| 6.23 | <i>digitalController</i> Collaboration Diagram | 154 |
| 6.24 | Supervised Autonomous Task Setup | 155 |
| 6.25 | Task Start | 158 |
| 6.26 | Task Step 1 | 159 |
| 6.27 | Task Step 2 | 160 |

| | | |
|------|---|-----|
| 6.28 | Task Step 3 | 161 |
| 6.29 | Task Step 4 | 162 |
| 6.30 | Task Step 5 | 163 |
| 6.31 | Task Step 6 | 164 |
| 6.32 | End Effector Position | 165 |
| 6.33 | Positional Error | 165 |
| | | |
| A.1 | M3-20a Analog to PWM PCB | 174 |
| A.2 | M3-20a Motor Driver PCB | 175 |
| A.3 | PT-15b Imager PCB | 176 |
| A.4 | AF-20a PCB | 177 |
| A.5 | MR-15d Controller PCB | 178 |
| A.6 | OV7930 PCB | 179 |
| A.7 | MR-20c Control PCB | 180 |
| A.8 | MR-20c and T-15b Imager PCB | 181 |
| A.9 | NEEMO Controller Box PCB-Main | 182 |
| A.10 | NEEMO Controller Box PCB-Imager and Motor Drive | 183 |
| A.11 | BMR-22a PCB | 184 |
| A.12 | WMR-22a PCB | 185 |
| A.13 | MRC-100/CPT-100 PCB | 186 |
| A.14 | AB3 PCB | 187 |
| A.15 | AB3 Interface Box | 188 |

List of Tables

| | | |
|-----|--|-----|
| 2.1 | MR-15 Wheel Types | 19 |
| 2.2 | <i>In vivo</i> Robots | 80 |
| 2.3 | <i>In vivo</i> Robots, cont. | 81 |
| 3.1 | Color Error | 89 |
| 3.2 | Radial Distortion | 92 |
| 4.1 | Rope Passing Analysis | 101 |
| 4.2 | Rope Passing Time Analysis | 101 |
| 4.3 | Stretch and Dissect Analysis | 103 |
| 4.4 | Survey Results | 104 |
| 4.5 | <i>Post-hoc</i> Power Analysis | 106 |
| 5.1 | Preoperative medications and Anesthetics | 110 |
| 6.1 | Distance Table | 129 |
| 6.2 | Transfer Function Decision Table | 136 |
| 6.3 | PID constants | 139 |

Chapter 1

Introduction

Laparoscopy is a minimally invasive alternative to traditional abdominal surgery. Unlike traditional surgery, a laparoscopic procedure can be completed using small incisions. The use of these small incisions results in reduced pain to the patient, shorter recovery times, and less trauma to skin, muscle and other tissues. However, these benefits to the patient are offset by the increased difficulty for the surgeon performing the procedure. Because the incisions are small (1.06 cm-1.4 cm using a 10 mm trocar [1]), the surgeon has no direct access to the surgical environment, and therefore must use specialized cameras and instruments to perform the procedure. This means minimally invasive surgery is minimal access surgery from the surgeon's point of view [2]. The difficulties introduced include reduced dexterity, reduced perception, and longer procedure times [2]. One possible solution to these problems is the use of small *in vivo* robotic devices for minimally invasive surgery. Traditional laparoscopic instruments have their movement constrained to pivot about the entry incision. Small robotic devices could be contained entirely in the abdominal cavity, and therefore the movement of these devices would not be constrained by the size or position of the incisions.

1.1 Literature Review

1.1.1 Minimally Invasive Surgery

The roots of minimally invasive surgery can be traced back to the 1800s with visualization devices created by Bozzini in 1805 and Desormeaux in 1853 [3]. These devices combined mirrors with candles or alcohol lamps to enable non-direct visualization of internal anatomy. Following the invention of the electric light bulb in 1879 were the first rigid endoscopes, first by Nitze and Leiter and then by von Mikulicz and Leiter. These devices were used for urologic procedures as well as viewing the esophagus. The first true laparoscopic procedure is credited to George Kelling in 1902. Kelling developed a method for pneumoperitoneum to examine the abdominal cavity of a dog using a Nitze device. In 1912, Hans Christian Jacobaeus, an internist from Stockholm, reported on 115 patients who had laparoscopy used for various diagnoses. These early endoscopes and laparoscopes had a high risk of thermal injury to abdominal tissues due to the fact that the light source was placed at the distal end of the device.

Improvements to laparoscopy continued over the next several decades. Sharp tipped trocars were introduced in 1920 by Orndoff. A foot pump operated insufflator was created by Goetze in 1921. The Veress needle was introduced in 1938 by Hungarian physician Janos Veress. This needle was a good method of introducing insufflation with a minimal risk of damage to abdominal organs and is still in use today.

Laparoscopy began moving from diagnosis to therapeutic use in the 1930s. Ferrers, a gynecologist, used electrocautery to perform laparoscopic adhesiolysis in 1933. In 1936, Boesch performed an endoscopic tubal ligation using electrocoagulation. Ruddock performed 500 liver biopsies using a peritoneoscope, which increased the diagnostic accuracy from 63.9% to 91.7%.

Improvements to light sources and optics came in the 1950s. Fourestier introduced a quartz light rod in 1952, and Harold Hopkins developed a system using flexible glass fibers to transmit the image. Later, in 1957, Basil Hirschowitz developed a fiber optic gastroscope that was less fragile than the rigid quartz light rod. These devices not only improved color, brightness, and clarity but were also much safer than previous devices, because the high temperature light source was not at the distal end of the device.

Laparoscopy did not truly transition from a diagnostic tool to a therapeutic tool until 1983, when German gynecologist Kurt Semm performed the first laparoscopic appendectomy. Erich Muhe of Germany performed the first laparoscopic cholecystectomy in 1985. In the United States, the first laparoscopic cholecystectomy was performed in 1988 by Barry McKernan and William Saye. After this, laparoscopic procedures quickly become an accepted method of performing surgery. Laparoscopic surgery is now used for many different procedures including: antireflux surgery; hernia repairs; extraction of the spleen, adrenal glands, kidney, and liver lobes; and colectomy for cancer.

1.1.2 Robotic Surgery

Although laparoscopic surgery is now widespread in use and has several advantages over conventional open surgery, there are some disadvantages to its use. Because laparoscopy is essentially “surgery from a distance”, the surgeon cannot directly touch or manipulate the patient’s anatomy, which results in a lack of tactile feedback. Also, the surgeon cannot directly observe the work he or she is doing, but must rely upon a projection from a camera inserted through an incision. This results in a loss of depth perception, because the three-dimensional operating field is projected to a two-dimensional image on a monitor. The postures a surgeon must assume to perform

this type of surgery can result in muscle fatigue and chronic injury [4]. Because of these problems, several systems have been developed for aiding surgeons in performing minimally invasive surgery.

In 1984, AESOP (Automated Endoscopic System for Optimal Positioning), the first surgical robot to receive FDA approval for use in abdominal procedures, was introduced [5]. This robot is used to hold and position a laparoscope, and is controlled by the surgeon by means of voice commands. Studies done by urologists at Johns Hopkins showed procedures done using the AESOP system had no increase in operating time and that AESOP provided a significantly steadier camera platform as compared to a human camera holder [6, 7]. The AESOP system can also be used for solo surgeon procedures, where the primary surgeon is the only participant in the operating field. Geis, et al. performed 24 laparoscopic hernia repairs, cholecystectomies, and Nissen funduplications in this manner with statistically similar procedure times as with standard laparoscopic surgery [8]. These studies show that AESOP can reliably replace a human camera holder and provide a stable camera platform during laparoscopic procedures.

The Zeus robot (Computer Motion Inc., now owned by Intuitive Surgical) is a surgical robotic system that was created primarily for endoscopic microsurgery [9]. It consists of two main components: a surgeon side system and a patient side robotic system. The surgeon sits at a console and controls the robotic arms of the patient side system using handles while viewing the endoscopic images on a screen. The robotic system consists of two arms for instruments and a third (AESOP) for positioning of the endoscopic camera. The Zeus system features hand tremor reduction and filtering, motion indexing, and motion scaling to increase the precision and accuracy of movement. In one study by Ghodoussi, et al., the Zeus system was used to complete a porcine laparoscopic cholecystectomy with the patient in France and the surgeon

in New York [9]. In another study, the Zeus system was used successfully perform endoscopic coronary artery bypasses on six dogs, and later on two human patients [10].

The da Vinci surgical system from Intuitive Surgical is a tele-robotic surgical system similar to Zeus, which consists of three parts. The first part is a control console with two hand controllers and a 3-D display similar to a pair of binoculars. The console also contains a computer for the entire system. The second component is a tower that contains video equipment and the insufflation unit. The third component is the robot itself. The robot has three arms, two of which hold laparoscopic instruments and the third which holds the stereo laparoscope. Like Zeus, da Vinci enhances dexterity with motion scaling, hand tremor filtering, and motion indexing. The cable driven instruments on da Vinci move with seven degrees of freedom, with two degrees of rotation. [11]. The imaging system, with two 5 mm scopes placed in a 12 mm housing [5], can be used in 3-D mode. In this mode, the surgeon's left and right eyes view a separate video stream in the binocular display. This 3-D vision mode has been shown to reduce errors by 93% and increase dexterity by 10-15%, as compared with using the 2-D mode [12].

Da Vinci received FDA approval for use in general abdominal surgery in July 2000 and is the only system still commercially available. As of June 2008, there were 946 da Vinci systems installed worldwide [13]. The system does have several limitations, cost being the greatest one. Each da Vinci system costs between \$1.0 million to \$1.7 million, with annual service costs of \$100,000 to \$150,000, which makes them unavailable to most hospitals. Also, the system can be cumbersome, requiring difficult tool changes [14,15], significant set-up time, and significant operational space [16]. In spite of these limitations, the system has gained significant use for certain procedures such as radical prostatectomy.

1.1.3 Robotic Endoscopes

While some research has focused on developing large robotic systems that remain outside the body (e.g., AESOP, Zeus, da Vinci), other groups have focused on surgical robots and video cameras that are completely *in vivo*. These devices are being developed for both diagnostic and therapeutic purposes. The PillCam SB, developed by Given Imaging Ltd., is a commercially available wireless video capsule that can be swallowed by the patient. The device measures 11 mm x 26 mm and weighs less than 4 grams [17]. The device contains a light source and imager and can transmit 2 images per second to an external sensor array worn by the patient. This device was FDA approved in 2001 and is used to diagnose Crohn's disease, small bowel tumors, celiac disease, and gastrointestinal bleeding. A similar device, the Sayaka endoscope, was developed by RF System Lab and entered clinical trials in the United States in 2008 [18]. This device captures 30 two-megapixel images per second and features LEDs for illumination. The camera draws power through induction charging from a vest worn by the patient. This vest also contains a wireless receiver that stores the images for later analysis.

Researchers at the University of Washington are developing a tethered capsule endoscope (TCE) system using a resonant fiber optic laser scanner, which fits into a 6.4 mm swallowable capsule [19]. This device uses a piezoelectric tube to vibrate the tip of a single mode optical fiber in a spiral motion while light is passed through the fiber, illuminating the area to be imaged [20]. The reflected light is detected by an external base station through separate multimode optical fibers. The base station then reconstructs the final images at a 15 Hz frame rate. The device is being developed as a low cost method of detecting esophageal cancer [21].

Hu et al. have developed an *in vivo* endoscope with pan/tilt capabilities [22]. This device has a built-in LED light source, a CCD camera head, and can fit through a 12

mm trocar. The device has been tested in a porcine model.

Other research has focused on developing devices for colonoscopy. Dario et al., have developed an actuation system based on shape memory alloy (SMA) pneumatic microvalves capable of propelling a microrobot along the colon [23]. The device moves by inchworm locomotion, achieved by vacuum suction clamping to the colon wall [24]. The device achieved speeds of up to 10 cm/min in a reconstructed colon model.

Breedveld describes an endoscope that uses a ‘rolling donut’ to traverse the colon [25]. This device is constructed from three stents whose diameter and shape can be changed to move in the colon. Kim et al. described a ‘smart colonoscope’ that includes locomotive functions, auto-camera steering functions, and a human friendly user interface [26]. This device, like the devices described by Dario et al., uses inchworm motion to traverse the colon. This device’s user interface consists of a head mounted display, eye tracking system with a micro-camera, and a joystick to manipulate the colonoscope.

Menciassi et al. investigated both inchworm and sliding claspers for locomotion [27]. These devices also used shape memory alloy springs to drive a steering mechanism. Slatkin and Burdick developed a robotic endoscope to access the small intestine [28]. This device uses a series of grippers and extensors to propel itself. The grippers provide traction against the lumen wall while the extensors extend and retract the space between the grippers. By using a computer to coordinate the motions of several segments, the device can move along the inside of the gastrointestinal tract. Prototype devices tested in anesthetized pigs have shown that this type of device could move through the small intestine.

Other research has focused on developing miniature robotic cameras for other areas of surgery, such as heart surgery. Researchers at the Robotics Institute at Carnegie Mellon are working on a robot that can move on the surface of a beating

heart. The HeartLander consists of two body sections that can independently adhere to the epicardial surface and translate during motion [29]. The device adheres to the heart using suction, which is provided by an external pump via a tether. Visual feedback from the device is sent via a 1.6 mm fiber optic endoscope running through the tether. The device has been tested in several animal trials with no significant epicardial damage.

1.1.4 NOTES

Laparoscopic surgery has been highly successful in reducing patient recovery time and patient trauma. Natural orifice transluminal endoscopic surgery (NOTES) has the potential to take these advantages a step further. In NOTES, external incisions are completely eliminated, as the surgeon accesses the peritoneal cavity through a natural orifice. The transition from laparoscopic surgery to NOTES has the potential to further reduce patient pain and trauma as the transition from open surgery to laparoscopic surgery did. Some of the first NOTES procedures were performed by Kaloo et al. using a transgastric approach in animal models [30]. Five survival experiments were performed on pigs in which the peritoneal cavity was accessed through an incision in the gastric wall. Wagh et al. successfully performed transgastric oophorectomy and tubectomy in porcine models in survival studies in 2006 [31]. Lima et al. performed transvesical endoscopic thoracoscopy on six anesthetized pigs in survivable surgeries [32].

Recently, there have been NOTES procedures successfully performed on humans. These include a laparoscopically assisted transvaginal cholecystectomy [33], a transvaginal cholecystectomy [34], and a transgastric cholecystectomy [35]. These studies have demonstrated the feasibility of NOTES, but there are significant limitations. NOTES is limited by the size of the natural orifice and the need for the instruments to be

highly flexible in order to traverse these natural lumen. Also, according to Kalloo et al., there are difficulties in determining orientation, applying off-axis forces, and passing multiple tools through a single entrance point [36]. New instruments have been developed to address these limitations, such as the TransPort system from USGI Medical [37]. Completely *in vivo* devices have also been developed. These systems are being developed as transitions to NOTES, enabling abdominal surgery using a single incision. In these systems, multiple devices are introduced through a single external incision. The Magnetic Anchoring and Guidance System (MAGS) is a system that includes an intra-abdominal camera and instruments such as retractors and cautery [38]. These instruments are held to the interior abdominal wall by external magnets. Prototype NOTES devices have been developed at the University of Nebraska as a part of this dissertation, and will be discussed further in Sections 2.18, 2.19, and 2.20, as well as in Chapter 6.

1.1.5 Visual Servoing

There have been some efforts to develop surgical robots capable of performing some tasks autonomously. Much of this research involves visual servoing, which combines visual tracking and control theory. Some of the earliest work in visual servoing was by Hill and Park, who used feedback information from a camera for real time control of a robot [39]. Casals et al. were some of the first to apply visual servoing to surgical robotics. They created a system to automatically position a laparoscope during a surgical procedure [40]. This system consisted of a robot to hold a laparoscope, a vision system, and a controller to generate robot trajectories. The vision system uses multiple processors for edge extraction, straight line extraction, tool detection, and measurement of tool position. The tools are marked with straight lines and rings to facilitate detection. The controller uses the current tool and laparoscope positions to

calculate the robot trajectory for optimal placement of the laparoscope. Wei et al. also designed a system to control the position of a laparoscope [41]. This system uses color segmentation, rather than a shape detection implementation the Casals system used, to track the position of surgical instruments. The controller uses the calculated three-dimensional positions of the instruments to calculate commands for the robot holding the laparoscope. A different method was developed by Krupa et al., who used small LEDs attached to the tips of instruments for tracking position [42]. The system also utilizes a custom laser pointer instrument holder to project laser dots onto the surgical field to enable tracking even when the instrument tips are not in the field of view of the camera. The controller uses the tracker data to command an AESOP surgical robot to automatically position the laparoscopic instruments. Hynes et al. are researching using a visually servoed robotic system to perform autonomous surgical tasks such as suturing [43]. This system uses a stereo vision system with two seven degree of freedom manipulators and has been demonstrated to be capable of manipulating a length of thread using two grippers. Also, Zong et al. are working on a visually servoed suturing system for keratoplasty (corneal grafting) [44]. This system has successfully demonstrated guiding a needle to an objective point autonomously.

1.1.6 Tracking

In order to implement a surgical robotic system using visual servoing, it is necessary to first develop a visual system for tracking joint position. Visual tracking involves extracting motion information from image sequences. A simple method of motion detection is image subtraction, where the difference of pixel values from two sequential images are examined. If an object is moving relative to a stable background in the image, this will show up as values significantly different from zero [45]. Gilbert et al. describe an early video tracking system developed for the U.S. Army White

Sands Missile Range [46]. This system used pixel intensity to classify each pixel as target, plume, or background. The target is then separated from the background, and location and orientation are calculated. These features are used to control the zoom, azimuth, and elevation of the optics to keep the target in the field of view. Tao et al. describe a method of tracking an unknown number of moving objects in video using hidden Markov models [47]. This algorithm is able to track multiple objects with the presence of clutter, while accounting for appearance/disappearance of objects due to occlusion. Maccormick and Blake developed a method called *probabalistic exclusion principal*, which solves the problem of multiple identical object tracking, in which the trackers coalesce onto the same target [48]. Other methods of multiple object tracking have been developed which use feature correspondence [49] and multiple hypothesis tracking [50]. Uemura, et. al. describe a method of tracking using two-dimensional color histograms in the hue-saturation plane of HSV color space [51,52]. This method allows for a changeable color range for tracked objects. Stern and Efros describe a similar method, which uses multiple color space models for tracking [53]. This method improves tracker performance by dynamically switching between color space models to adapt to changing illumination.

1.2 Dissertation Objective and Contributions

The objective of this dissertation is to demonstrate that *in vivo* surgical robots can be used in minimally invasive surgery for visual feedback, and eventually for supervised autonomous surgical tasks. Summaries of the animal surgeries where these devices were used to both augment and replace conventional laparoscopic visual systems are given. Experimental results showing that the *in vivo* cameras described here have visual quality comparable to standard laparoscopes are presented. These experiments

include quantitative comparisons using standard visual quality metrics and comparisons in completing simulated surgical tasks. Results from survivable animal surgeries are presented which show that these devices are sterilizable, and can be used in surgeries without causing any tissue damage or infection. Finally, preliminary experiments are discussed which suggest that these devices could be used to perform supervised autonomous surgical tasks in situations where the patient is a great distance from a trained surgeon. These experiments involve the design and testing of a system using a visual tracker as positional feedback for a closed loop visual servoing system to enable a prototype NOTES robot to complete supervised autonomous tasks.

The work of the author described in this dissertation contributes has two major contributions to the area of research concerning the use of *in vivo* robots in minimally invasive surgery.

1. Electrical design, software design, and testing of at least 19 different prototype *in vivo* surgical robots.
2. Some of the first research in using *in vivo* robots to automate low level surgical tasks.

The *in vivo* devices described in Chapter 2 all were a combination of mechanical design, electrical design, software design. All of the electronics design for these devices was the work of the author. This includes design of motor drivers, imager printed circuit boards and all interface hardware. In addition this, all software design and implementation for these devices was the work of the author. This includes firmware design for Cypress PSoC and Microchip PIC microcontrollers and development of control software implemented on a Windows based PC.

The research described in Chapter 6 is, to the author's knowledge, some of the first research done in the use of small *in vivo* surgical robots to automate low level

tasks. The author designed the electronics which control the robot used in these experiments. In addition, the author developed the software used to complete these tasks. This involved developing a real-time visual tracking system, integrating an existing stereo vision algorithm, and designing a PID controller to enable closed loop position control of the surgical robot.

1.3 Dissertation Outline

The remainder of this dissertation consists of the following chapters. An overview of all the *in vivo* robotic devices that were designed and tested as part of this dissertation are described in Chapter 2. Electronic design and software are discussed, and a summary of *in vivo* test results for each robot is given. Chapter 3 describes experiments conducted to compare image quality of the *in vivo* robots described here to image quality of a standard laparoscope. Several image quality metrics are used, including MTF, color accuracy, and distortion. The NEEMO experiments are described in Chapter 4. This project, which was a collaboration with NASA, investigated the use of *in vivo* robots for long term space missions. Data were also gathered to perform statistical comparisons of standard laparoscopes and *in vivo* robots. Chapter 5 discusses the use of an *in vivo* robot used as sole visual feedback in multiple survivable animal surgeries. Chapter 6 discusses using an *in vivo* dexterous manipulator to perform supervised autonomous surgical tasks. This system uses a real-time visual tracker and a positional controller to show the feasibility of these tasks. Finally, Chapter 7 discusses suggestions for improvements on the *in vivo* devices described in this dissertation, including improvements to the visual tracker and stereo vision components of the task completion system, and suggestions for improvements on the NEEMO experiments to better prove the efficacy of these *in vivo* robots.

Chapter 2

In-Vivo Robots

This chapter discusses all the *in vivo* robots that were developed as a part of this dissertation. A short description of each robot is given, with images and system block diagrams provided. *In vivo* tests of each device was used in are also discussed. Tables 2.2 and 2.3 on pages 80-81 summarize all the robots described here. It should be noted that the design and testing of these devices was a team project, and that the work of the author relates to the electronics and software aspects of the design of these robots, as well as systems integration and testing. Details of the mechanical designs can be found in the work of Rentschler [54], Lehman [55], and Wood [56].

2.1 Manipulator (M3-20a)

The M3-20a robot is a three degree of freedom manipulator arm designed for the manipulation of tissue during laparoscopic surgery. The device, shown in Figure 2.1, has three 20 mm diameter resin joints powered by brushed DC motors with quadrature encoders. A diagram of the control electronics is shown in Figure 2.2.

A PC containing a Motion Engineering (MEI) PCI-DSP [57] runs a MFC based control program. MFC, or Microsoft Foundation Classes, is a library that wraps

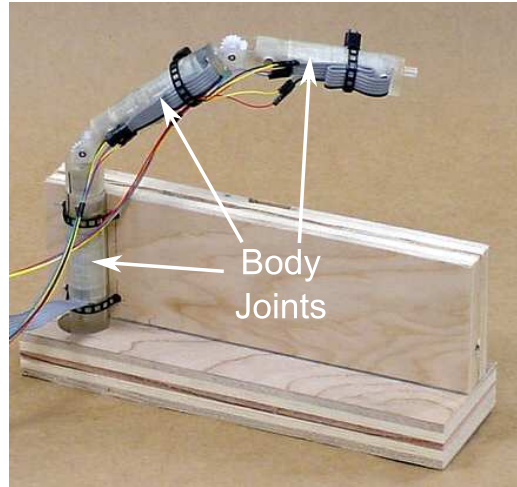


Figure 2.1: M3-20a Robot

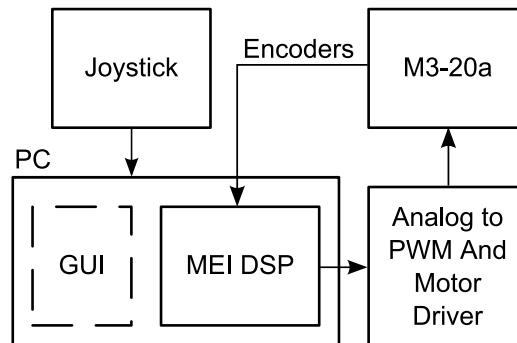


Figure 2.2: M3-20a System Flowchart

the Microsoft Windows API in C++ classes to simplify application design. All the Windows-based control programs subsequently discussed in this dissertation have been developed using MFC. The program reads data from a standard universal serial bus (USB) joystick, and sends commands to the MEI card. The PCI-DSP is a PCI (Peripheral Component Interconnect, a standard personal computer expansion bus) digital signal processor (DSP) based motion controller card, which can drive up to 4 axes in open or closed loop configurations. In this setup, a simple velocity controller was configured, with feedback coming from each motor's integrated quadrature encoder. The PCI-DSP generates an bipolar analog voltage output for driving

motors, which is converted to pulse-width modulated (PWM, a modulation scheme that involves varying the duty cycle of a square wave) and direction signals by a custom motor driver printed circuit board, or PCB. This PCB features a Cypress Programmable System on a Chip (PSoC) [58], which is configured to convert the bipolar analog signal from the PCI-DSP to a PWM and direction output. The PWM and direction signals drive an H-Bridge (a device consisting of 4 transistor switches, which allows driving of the load in two directions) integrated circuit motor driver, which in turn drives each motor on the robot.

The M3-20a was tested *in vivo* in an anesthetized pig on June 9, 2003. The device successfully retracted a small section of liver, which showed that it could produce enough force at the end effector to move tissue. However, the device proved too large to operate effectively in the insufflated peritoneal cavity. Also, the lack of a stable mounting platform hindered the device's performance.

2.1.1 Author's Contributions on M3-20a

The author's contributions in the design and testing of the M3-20a were software development, electronics hardware design, and *in vivo* testing. The details on these contributions follow.

The Windows based PC control program mentioned in the previous section was developed and tested by the author. The program was developed in Microsoft Visual Studio using the previously mentioned MFC class library. The program was developed as a dialog based application. A screen capture of the dialog window is shown in Figure 2.3. The *Position Status* section displays the current velocity, command position, actual position, and error position, all read from the PCI-DSP motion controller card. The *Axis Status* displays any error conditions reported by the PCI-DSP. It contains buttons to reset error conditions and stop each axis individually. The

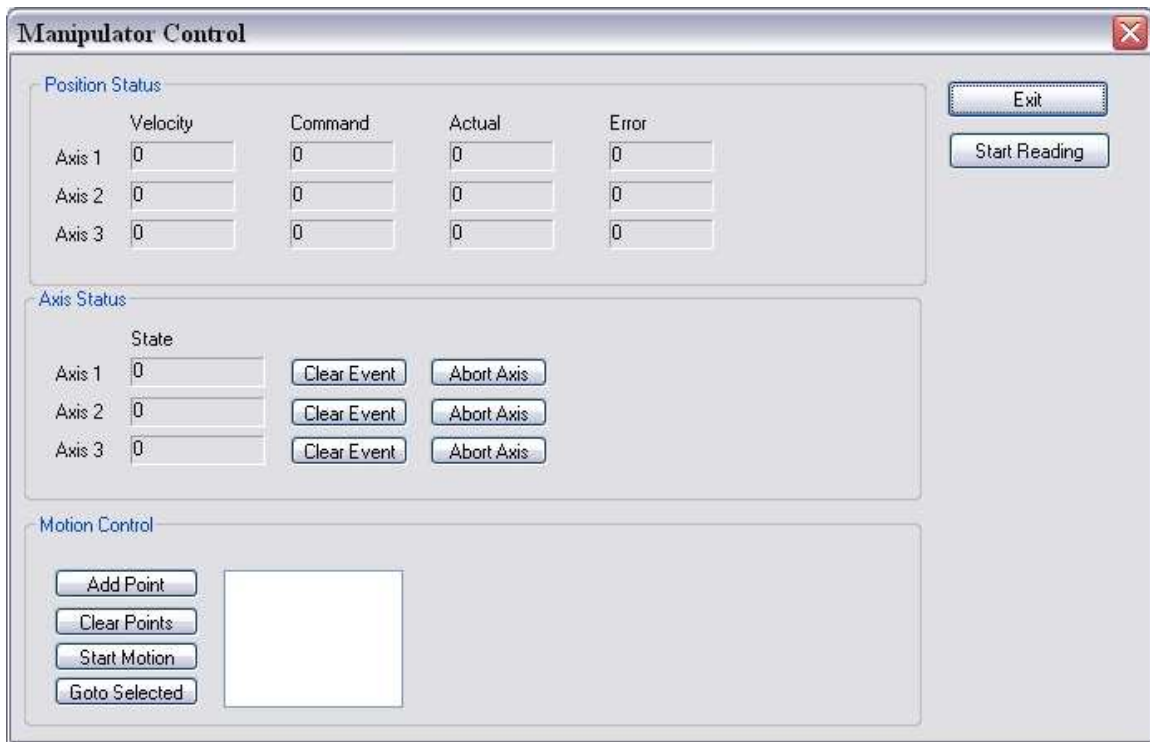


Figure 2.3: M3-20a Control Software

Motion Control section allows for limited coordinated motion. The user enters individual axis positions and these motions are executed by the motion controller. This feature only allows for specifying absolute axis positions, not the Cartesian position of the end effector.

A collaboration diagram of the C++ classes comprising the software is shown in Figure 2.4. The *CMotionControl* class is a C++ wrapper for the PCI-DSP software library, which was written in C. It also provides data members and functions for storing multi-point motion profiles generated by the user from the GUI. The *CJoystick* class is used to read data from a USB joystick connected to the PC. The joystick interface is through the DirectInput library contained in the Microsoft DirectX multimedia library. These two classes are data members of the main dialog class, *CManipulator-JoystickDlg*. This class implements a timer thread, updated at 33 Hz, which reads

data from the joystick, updates the dialog indicators, and sends velocity movement commands to the PCI-DSP.

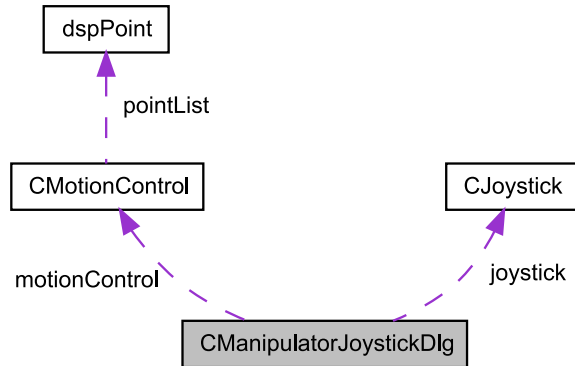


Figure 2.4: M3-20a Class Diagram

The author also designed and tested printed circuit boards (PCBs) for interfacing the analog output signal from the PCI-DSP to the small DC motors driving the M3-20a robot. The first board, analog to PWM conversion (see Figure A.1 in Appendix A), features two operational amplifier (op-amp) circuits for each of the three channels. The first circuit scales the ± 10 volt output signal to a ± 2.5 volt signal. The second circuit shifts this ± 2.5 volt signal to a 0 to 5 volt signal. At the output of these op-amp circuits is an 8-bit Cypress PSoC microcontroller. The firmware on this microcontroller, also developed in assembly language by the author, reads in all three analog signals using on-board analog to digital converters (with an input voltage range of 0 to 5V), and converts each to PWM and direction signals. The second PCB (see Figure A.2 in Appendix A) consists of FAN8200 dual motor driver integrated circuits (ICs), along with several connectors for interface with the previous PCB and with the motors on the device.

The author also set up the PC and all support hardware prior to the *in vivo* test of this device. During the testing, the author controlled the device under direction of the surgeon.

2.2 Crawler (MR-15a-c)

The MR-15a, MR-15b (Figure 2.5) and MR-15c are 15 mm diameter, two wheel crawlers used to investigate *in vivo* mobility. The bodies are resin from a rapid

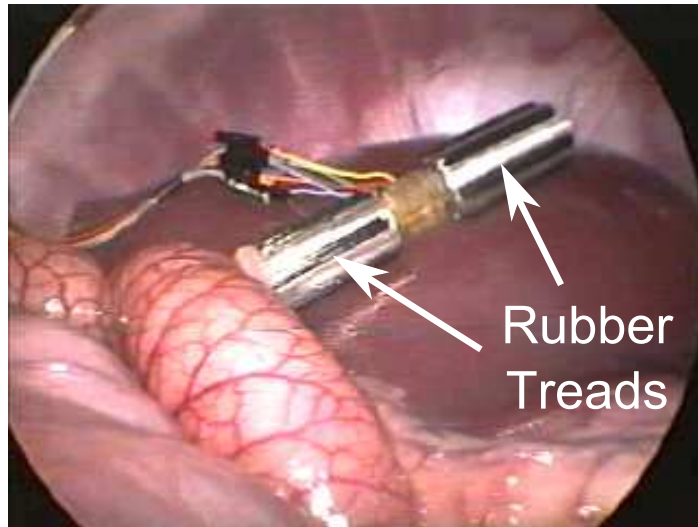


Figure 2.5: Crawler (MR-15b)

prototyping process, with machined aluminum wheels. The wheel types are detailed in Table 2.1. The wheels are independently driven by brushless DC (BLDC) motors.

| Device | Wheel Type |
|--------|--------------------------------|
| MR-15a | Ball Screw Helical |
| MR-15b | Rubber Windshield Wiper Blades |
| MR-15c | Deep Helical Groove |

Table 2.1: MR-15 Wheel Types

A block diagram of the drive electronics can be seen in Figure 2.6. The joysticks used here are 2-axis miniature analog joysticks taken from a console game controller. The analog conversion PCB contains a PSoC microcontroller that converts the joystick signals to magnitude and direction signals for the BLDC driver PCBs. This enables

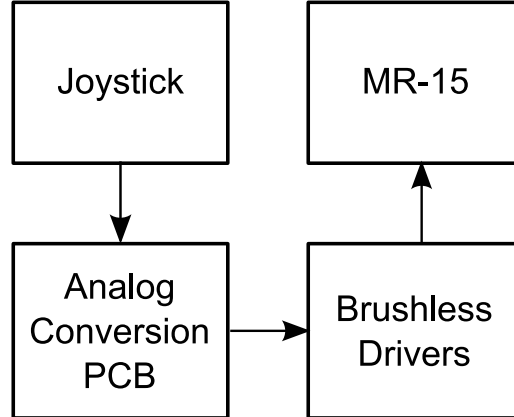


Figure 2.6: MR-15 System Flowchart

open loop velocity control of each crawler wheel.

The MR-15b was tested *in vivo* on June 9, 2003, in an anesthetized pig. The crawler moved well on the liver, but easily wrapped around the soft tissue on the bowels. The rubber treads were too adhesive for this tissue. Another problem was the wire tether on the crawler, which limited its mobility.

The MR-15c was tested *in vivo* on April 9, 2004, in an anesthetized pig. A load cell (an electronic device that converts force to a voltage signal) was also inserted and connected to the MR-15c to measure the amount of drawbar, or pulling, force generated by the robot on the liver, spleen, and bowel. The drawbar force tests worked well until the right motor failed in the device. The MR-15c was also tested on October 27, 2004, in the abdominal cavity of a human cadaver. The crawler functioned properly, but there were significant movement problems. The rigid wheels on the crawler could not achieve any traction on the slick, rigid organs. Another major problem was high centering on the exposed middle body section of the robot. In certain situations, the center of the robot became stuck on a section of tissue and the robot was unable to generate enough torque to continue moving.

2.2.1 Author's Contributions on MR-15a-c

The author's contributions in the design and testing of the MR-15b and MR-15c robots were electronics hardware design, software development, and *in vivo* testing. The details on these contributions follow.

An overview of the electronics hardware developed by the author for the MR-15b is shown in Figure 2.7. The analog joysticks were taken from a console game controller.

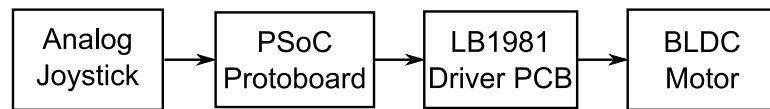


Figure 2.7: MR-15b Electronics

They consist of two orthogonal potentiometers, of only one of which is used in this application, which return to center when no force is applied. One joystick is used for each wheel, to enable independent movement of each wheel. A 5 volt supply is connected to each joystick, resulting in a 0 to 5 volt output. The joysticks are connected to a PSoC microcontroller mounted on a prototyping breadboard. The microcontroller converts each analog joystick signal to the magnitude and direction signals required by the motor driver. The motor driver is a LB1981 brushless DC motor driver. The driver is supplied on a PCB with all support circuitry needed to drive a brushless DC motor, with an analog voltage input for speed and a single digital input for direction. The motors on the MR-15b device is connected via a tether to the output connector of the motor driver.

The software design for the MR-15b consisted of the firmware running on the PSoC microcontroller, written in assembly language. The program ran in an endless loop, continually reading the analog voltages and converting each to magnitude and direction signals for driving the LB1981 boards.

The MR-15c shares a majority of the hardware design of the MR-15b, but with

some differences, shown in Figure 2.8. The PSoC microcontroller and LB1981 BLDC

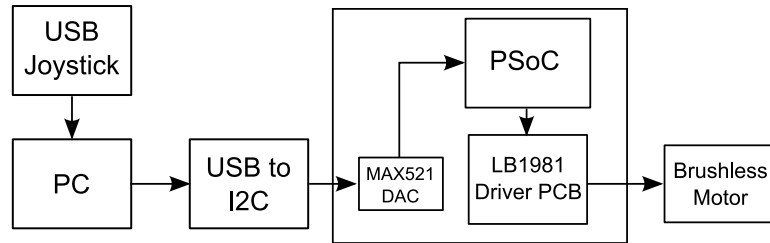


Figure 2.8: MR-15c Electronics

drivers remain the same, but in place of the analog joysticks is a USB joystick connected to a PC running a control program. This program sends commands to a USB to I2C (Inter-Integrated Circuit, a two-wire serial bus standard) converter. The USB to I2C converter sends I2C commands to a MAX521 digital to analog converter (DAC). The PSoC reads the analog signals from the DAC as with the joystick in the MR-15b device.

The Windows control software was developed as a command line interface program. A class diagram is shown in Figure 2.9 The *i2cControl* class is a simple wrapper

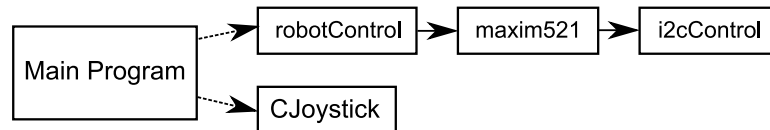


Figure 2.9: MR-15c Software

for the USB to I2C converter driver, which is provided as a dynamically loaded library (dll). The class dynamically loads the library and loads all needed functions. The *maxim521* class inherits *i2cControl*, and is the driver for communicating with the MAX521 DAC. The *robotControl* class inherits *maxim521* and provides higher level functions for controlling the MR-15c robot. The *CJoystick* class, used in the M3-20a software (Sec 2.1.1), is again used as a wrapper for the DirectInput joystick interface. The main program creates an instance of both *robotControl* and *CJoystick*

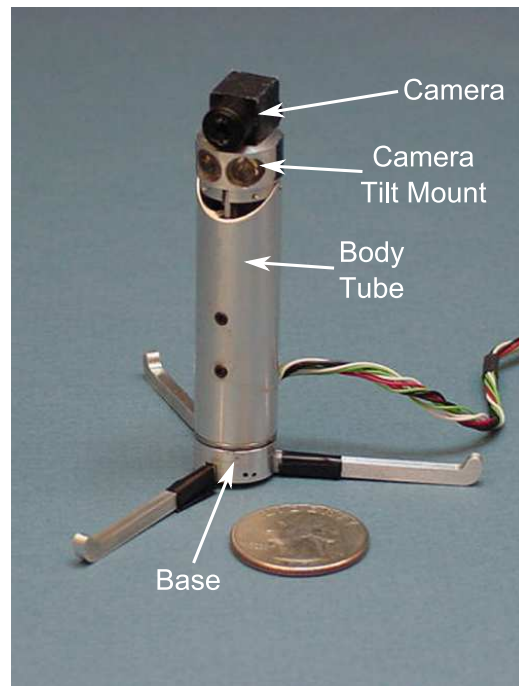
for accessing the robot and Joystick. The program includes modes for joystick control of the MR-15c, PT-15b (see section 2.3), and MR-15c ramp motor speed generation for measuring drawbar force.

In addition to the motor control hardware, the author implemented a circuit to measure drawbar force generated by the MR-15c. A load cell was placed *in vivo* to measure the drawbar force, and an external PCB was designed to amplify the load cell signal. An LT1167 instrumentation op-amp was used to amplify the load cell signal, which is digitized by a PC with a National Instruments data acquisition card installed. A National Instruments Labview program was developed to acquire and save the load cell data, which was analyzed after the *in vivo* test.

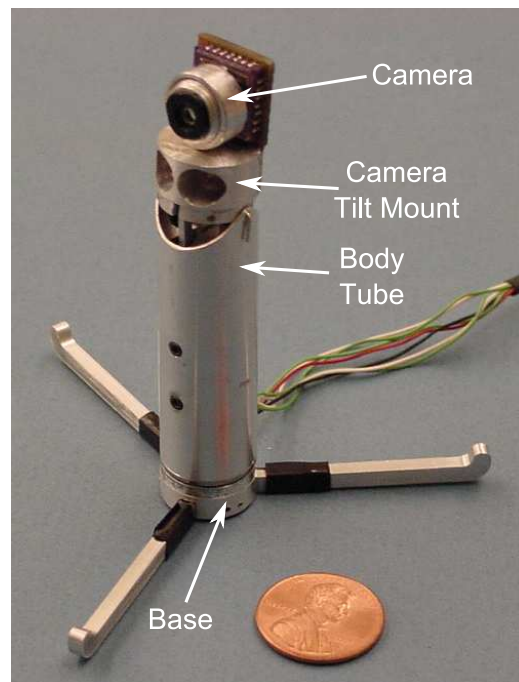
The author also set up all support hardware prior to *in vivo* testing of these devices, with the exception of the MR-15a. During the testing, the author controlled each device under direction of the surgeon, with the exception of the MR-15a.

2.3 Pan and Tilt (PT-15)

The PT-15 devices are 15 mm diameter, fixed base devices constructed of machined aluminum. Each device has 2 permanent magnet direct current (PMDC) motors, one of which rotates the main body cylinder axially, with the other tilting the imager/LED mount. Both the a (Figure 2.10(a)) and b (Figure 2.10(b)) revisions have 2 white LEDs (light emitting diode) mounted below the imager for illumination. The imager on the PT-15a device is an off-the-shelf black and white analog NTSC (National Television System Committee, an analog television format) format wired imager. The imager on the PT-15b is a custom PCB with an Omnivision OV7930 NTSC camera chip [59]. The lens is mounted on a threaded insert, which allows for manual changes of the focus. The focus must be set prior to being inserted in the *in vivo* environment.



(a) Pan and Tilt (PT-15a)



(b) Pan and Tilt (PT-15b)

Figure 2.10: Pan and Tilt Devices

Figure 2.11 shows a block diagram of the drive electronics for the PT-15a and PT-15b devices. A USB joystick connected to a Windows PC functions as the main user input. The PC is running a command line program which allows the user to control a PT-15 device and a MR-15c crawler from the same user interface. A keyboard command enables the user to switch joystick control between the PT-15 and MR-15 devices. Based on joystick and keyboard commands, the program sends data to a USB to I2C converter board. This board interfaces with a custom PCB with an I2C digital to analog converter (DAC). The DAC drives the analog to PWM driver and PMDC H-bridge drivers, which drive the DC motors on the robot.

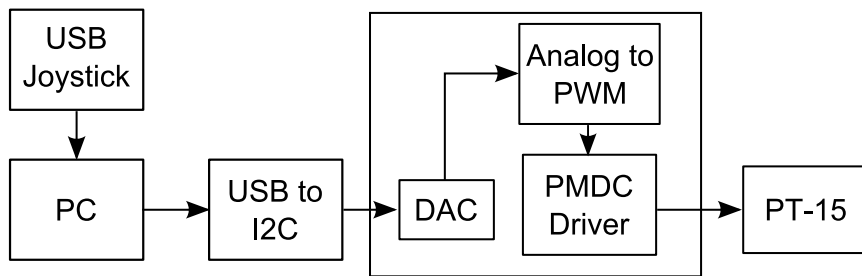


Figure 2.11: PT-15 Block Diagram

The PT-15a was tested on February 21, 2004 in an anesthetized pig. The pan-tilt functionality worked well. The camera returned a good image, but achieving the correct focus was a problem. Prior to insertion, the focus was set, but it was not correct. In order to obtain a better quality image, the researchers decided to finish the procedure as an open procedure, using one large incision rather than the small incisions a laparoscopic procedure would use. This allowed the researchers to adjust the focus of the camera to obtain a better quality image, which allowed the surgeon to complete the procedure.

The PT-15b was tested *in vivo* on April 9, 2004. The pan-tilt functionality again worked well. The new Omnivision camera returned a good image, but the focus problems were still present. Resolution and color reproduction tests were also conducted.

2.3.1 Author's Contributions on PT-15a-b

The author's contributions in the design and testing of the PT-15a and PT-15b robots were electronics hardware design, software development, and *in vivo* testing. The details on these contributions follow.

The author designed and tested the I2C DAC board and integrated it with the analog to PWM and motor driver boards. The Windows control software developed for the MR-15c (see Figure 2.9 in Section 2.2.1) also controlled the PT-15a and b devices. The author also designed the imager PCB on the PT-15b (see Figure A.3 in Appendix A), which was the first of the devices described in this dissertation with a custom designed imager. This was the first PCB with the Omnivision OV7930 imager. The PCB measures 0.502 inches by 0.807 inches and contained the OV7930 imager, as well as a 14.31818 MHz oscillator and two white light emitting diodes (LEDs) for illumination. It requires a 5 volt power supply and has analog video output.

The author also set up all support hardware prior to *in vivo* testing of these devices. During the testing, the author controlled each device under direction of the surgeon.

2.4 Wireless Camera (WC-15a)

The WC-15a is a passive, wireless camera made of machined plastic. It is attached to the inner abdominal wall by means of a wire barb, see Figure 2.12. The device is entirely passive, with no motors for movement. The main PCB contains a voltage regulator and an Omnivision OV7930 NTSC camera chip. The video output from the camera board is sent to an off-the-shelf wireless video transmitter. The device is powered by three batteries.

The WC-15a was tested *in vivo* on April 9, 2004. Attaching the device to the

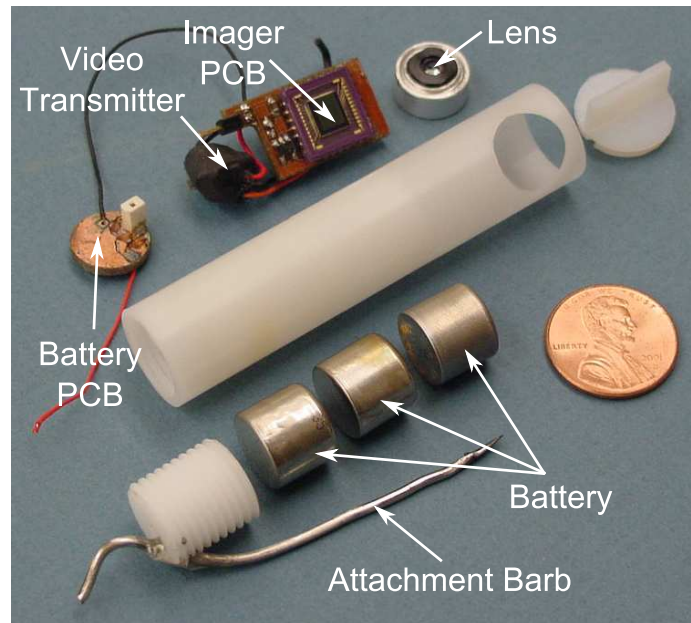


Figure 2.12: Wireless Camera (WC-15a)

inner abdominal wall using the wire barb was cumbersome. As with previous devices, the difficulty in changing focus made obtaining a good video image time consuming. However, when the camera was adjusted to the correct focus, the video quality was acceptable.

2.4.1 Author's Contributions on WC-15a

The author's contributions in the design and testing of the WC-15a robot was electronics hardware design and *in vivo* testing. The details on these contributions follow.

The wireless video transmitter used in WC-15 requires a 9V power supply. Three 3 volt batteries (see Figure 2.12) were used in series for the power supply. A round PCB was designed and fabricated which made contact with the negative terminal of the battery pack. The positive terminal was connected to the battery PCB by a short length of wire. The battery PCB has a one contact receptacle which mates with a pin on the imager PCB to supply 9 volts to the board. The ground connection is made

with a short length of wire. The wireless transmitter fits between the imager PCB and the battery PCB and connects to the imager PCB for power, ground and video. The electronics are placed in the plastic tube prior to testing.

The author also set up all support hardware prior to *in vivo* testing, including a receiving antenna, wireless receiver, and video recording and display devices.

2.5 Adjustable Focus (AF-20a)

The AF-20a device is a fixed base camera, built to test an remotely adjustable focus mechanism. The focus is adjusted by a BLDC motor which is connected to the

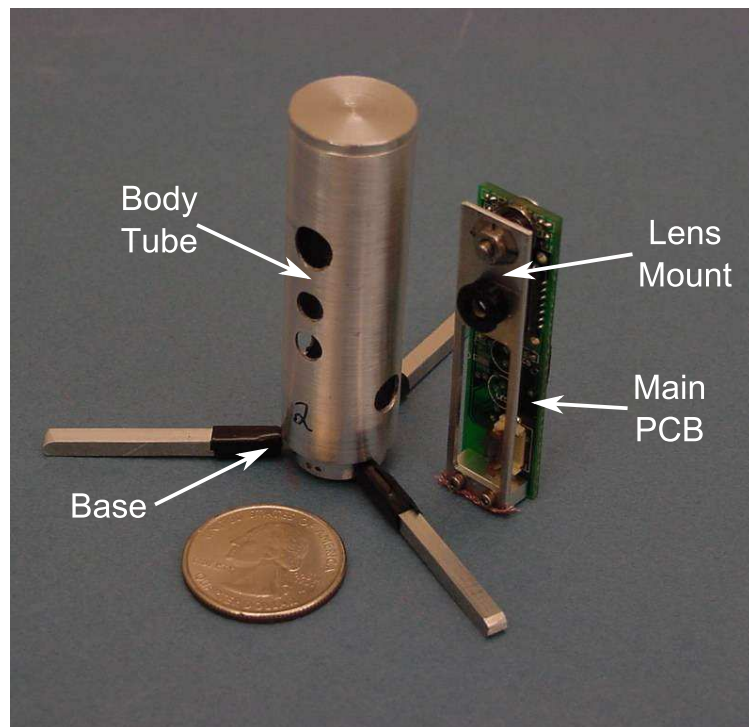


Figure 2.13: Adjustable Focus (AF-20a)

PCB in an aluminum mount. The PCB contains an OV7930 imager, LED driver and two white LEDs. The external interface connector routes power, NTSC video, serial camera control bus (SCCB), and BLDC connections. A block diagram of the

system is shown in Figure 2.14. The AF-20a device was an important step forward

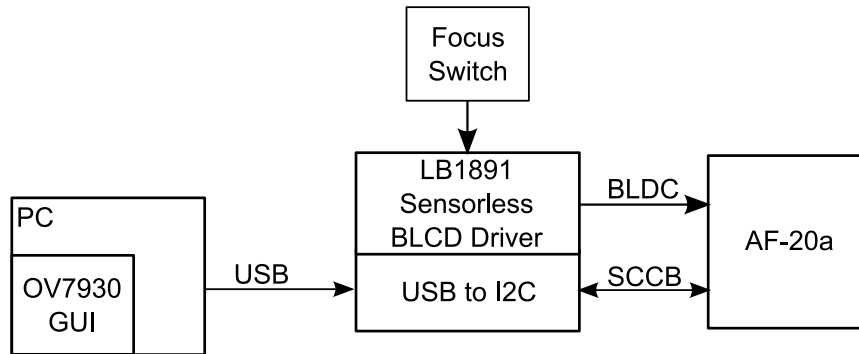


Figure 2.14: AF-20a Block Diagram

in the improvement of *in vivo* imaging in two respects. First, this device was the first with a remotely adjustable focus. Previously, the camera devices had to be placed at a fixed distance from the area of interest to achieve an in-focus image. With the AF-20a, the camera could be focused on areas of varying distance easily. The second improvement was image adjustment. The OV7930 imager has several internal registers used to adjust image characteristics such as contrast, saturation, tint, and brightness. This enabled the researchers to improve the video quality from the device. The AF-20a was tested *in vivo* on December 1, 2004. The adjustable focus mechanism worked well enough to verify the proof of concept; however, there was some sticking in the mechanism that caused some difficulty. The camera settings adjustments were very beneficial in improving image quality. An image capture from the device can be seen in Figure 2.15.

2.5.1 Author's Contributions on AF-20a

The author's contributions in the design and testing of the AF-20a robot were electronics hardware design, software design, and *in vivo* testing. The details on these contributions follow.



Figure 2.15: AF-20a *In vivo* Gall Bladder Image

Figure A.4 in Appendix A shows a schematic of the PCB designed by the author for the AF-20a robot. This board features a OV7930 imager and support components, as well as connectors for passing the brushless motor signals to the focus motor. The user interface designed for the AF-20a robot is shown in Figure 2.16. It allows the user to change gain, color balance, hue, saturation, brightness, exposure, sharpness, horizontal flip, and vertical flip. It was developed in Microsoft Visual Studio using the MFC library. The class diagram for the AF-20a control software is shown in Figure 2.17. The main dialog class, *Dlg_OV7930*, controls all GUI properties. The *ov7930* class serves as an interface to the high level imager functions, such as brightness and exposure. This is derived from *ovSccb*, which generates low level serial commands for the imager. The *i2cControl* class communicates with the USB to I2C converter board, which sends commands to the imager IC.

The author also set up all support hardware prior to *in vivo* testing of this device. During the testing, the author controlled each device under direction of the surgeon.

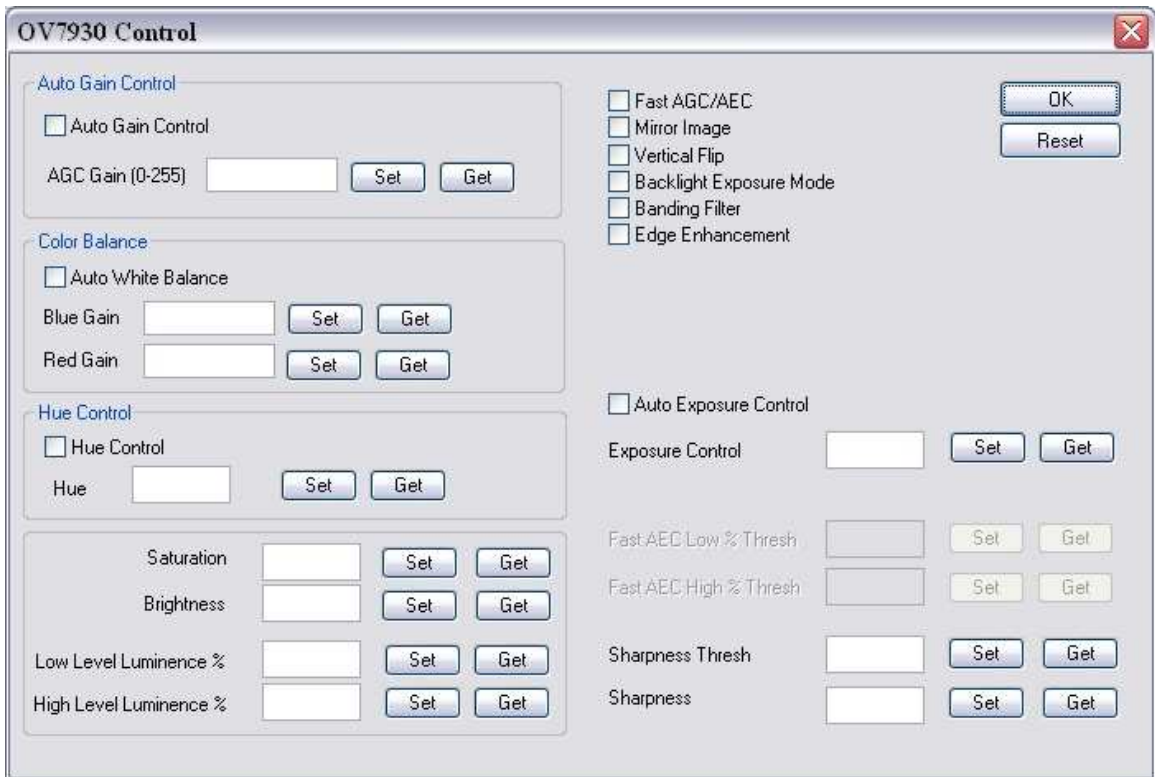


Figure 2.16: AF-20a User Interface

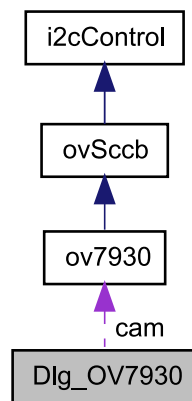
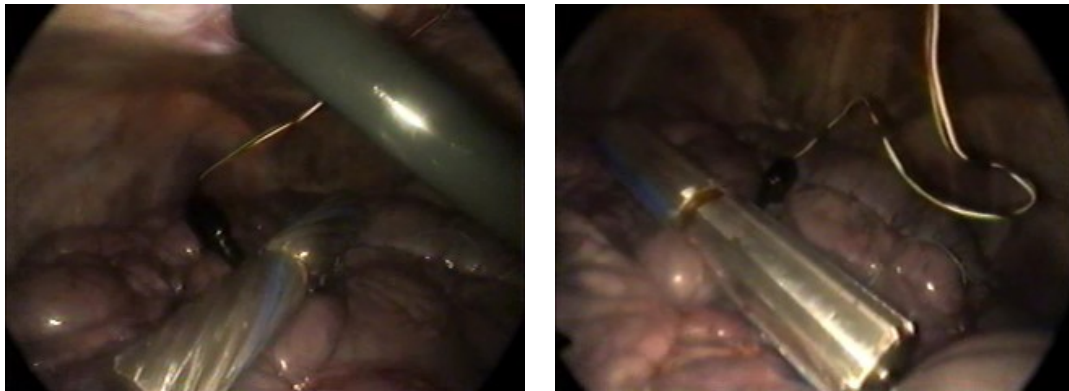


Figure 2.17: AF-20a Software

2.6 Crawler (MR-20a)

The MR-20a is a 20 mm crawler with aluminum wheels. Due to high centering problems with previous crawlers, the MR-20a was designed with a very small space between the wheels. Because the main purpose of this device was to find solutions to mobility problems, this device had no camera system. Two externally driven brushed DC motors are used to drive the wheels. The system setup is similar to the PT-15 device (Fig 2.11).



(a) Helical Wheels

(b) Male Wheels

Figure 2.18: Crawler (MR-20a)

The MR-20a was tested *in vivo* on December 17, 2004. Two different sets of wheel treads were tested, a male profile with 8 grousers (Figure 2.18(b)) and a helical profile with 10 grousers (Figure 2.18(a)). The male wheel performed poorly because its grousers were too thick, which caused it to behave like a smooth wheel with little traction. The helical wheel performed much better, and was able to traverse the bowel and other organs. The control scheme was also found to be flawed. Up to this point, the joystick value (0% to 100%) was directly mapped to the PWM duty cycle. When the joystick was actuated, the rotational speed of the wheels increased too rapidly and the wheels had a tendency to ‘spin’ while gaining no traction. This

test demonstrated that finer control of speed was needed.

2.6.1 Author's Contributions on MR-20a

The author's contributions in the design and testing of the MR-20a robot were electronics hardware design and *in vivo* testing. The details on these contributions follow.

The MR-20a control hardware consisted of a modified MR-15c setup (see Figure 2.19). This existing setup was modified by using the M3-20a analog to PWM and

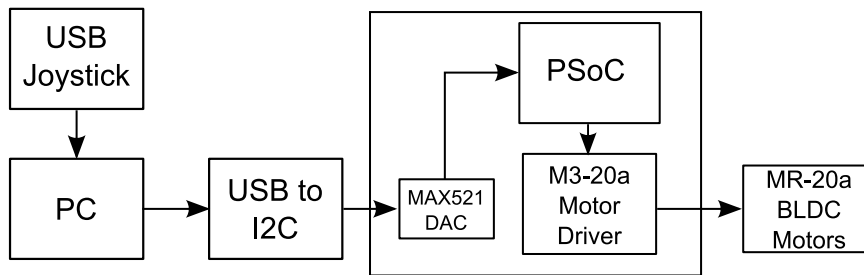


Figure 2.19: MR-20a Block Diagram

motor driver PCBs in the place of the BLDC drivers. This allowed the MR-15c software to be used for control without any modifications.

The author also set up all support hardware prior to *in vivo* testing of these devices. During the testing, the author controlled each device under direction of the surgeon.

2.7 Crawler (MR-15d)

The 4th generation of the 15 mm crawler (Figure 2.20) features aluminum helical profile wheels with a small gap between them. This device also has a new spring tail to aid in maneuverability. Like the MR-20a, this device has no camera system, only two externally driven brushed DC motors. After the control problems with the previous MR-20a device, a new joystick mapping technique was implemented by the

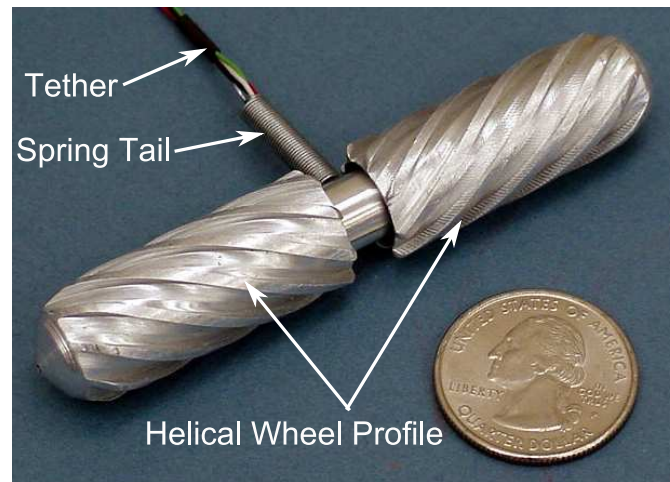


Figure 2.20: Crawler (MR-15d)

author for this crawler. A screenshot of this mapping feature can be seen in Figure 2.21. The graph in the left of the dialog box displays the current mapping, with the joystick input magnitude value on the x-axis and the output value on the y-axis. The output value is scaled by a factor of 10 which results in the PWM output duty cycle. The controls on the right side of the dialog box enable the user to define either a piecewise linear or a third order polynomial mapping. The MR-15d was tested *in vivo* on January 21, 2005. The device was able to traverse the liver, spleen, small bowel and large bowel easily and without causing tissue damage. The new joystick mapping function was a great improvement in controlling the crawler, allowing finer control of the wheel speed which reduced the chance of wheel slippage. The helical wheel shape and use of a spring tail were confirmed to facilitate the best mobility in the *in vivo* environment. The MR-15d was also tested *in vivo* on July 22, 2005. This test was used to gather drawbar force information using a load cell. A load cell was attached to the crawler, with the other end held stationary by the surgeon. A ramping PWM signal was sent to the device while the drawbar force was recorded by the computer.

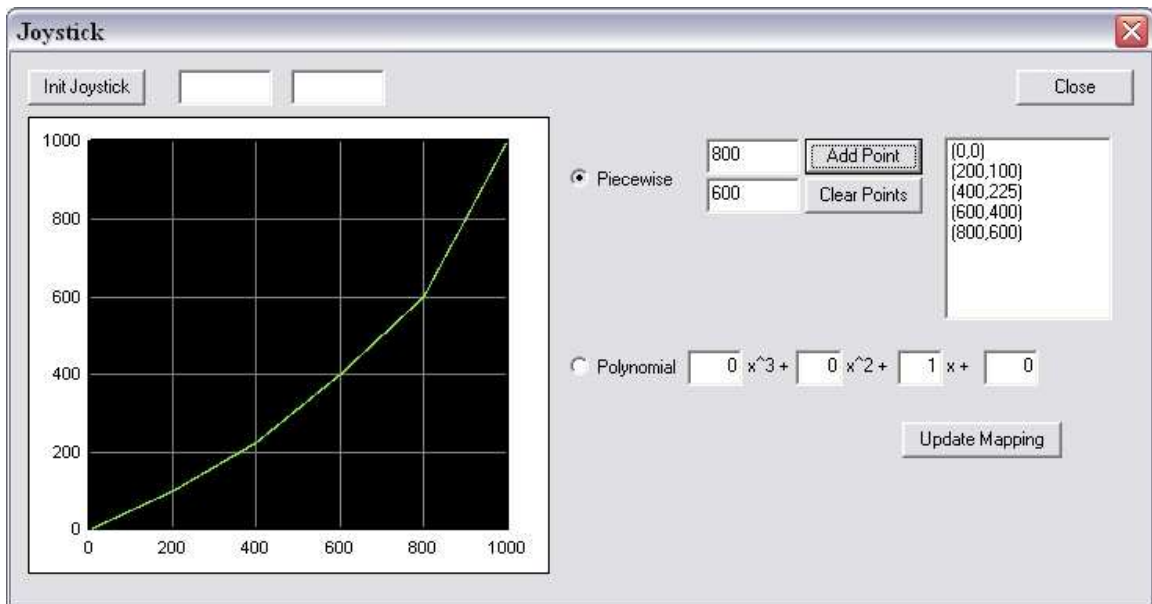


Figure 2.21: Joystick Mapping GUI

2.7.1 Author's Contributions on MR-15d

The author's contributions in the design and testing of the MR-15d robot were electronics hardware design and *in vivo* testing. The details on these contributions follow.

The electronics driving the MR-15d robot was a new setup designed and tested by the author, shown in Figure 2.22. A new PCB, the PIC driver PCB, was designed

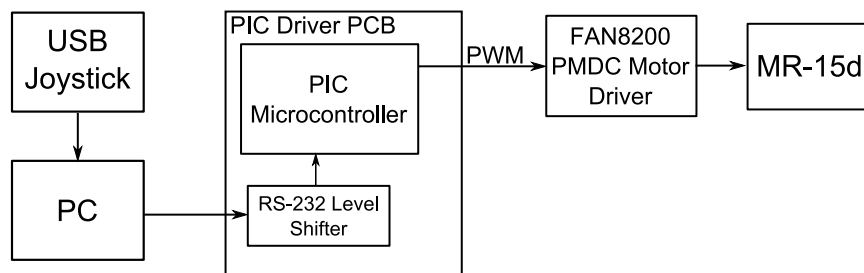


Figure 2.22: MR-15d Drive Electronics

and tested (see Figure A.5 in Appendix A). This PCB contains a RS-232 level shifter to convert RS-232 voltage levels to TTL voltage levels. A PIC microcontroller, from Microchip [60], reads serial commands from a host PC and generates PWM signals

which drive the M3-20a motor driver PCB. These drivers actuate the motors on the MR-15d robot. The PIC driver PCB was designed to drive several different types of robots, and contains an OV7930 imager interface, as well as capability for driving a camera focusing actuator. The firmware on the microcontroller generates all signals for these interfaces based on serial commands from a host PC.

The control software for the MR-15d was also an improvement over previous control software. A new Windows based GUI was developed by the author using the MFC library, and is shown in Figure 2.24. This software featured the new joystick mapping system, discussed previously, as well as the capability of generating motor ramping profiles for tests measuring drawbar force. The collaboration diagram for

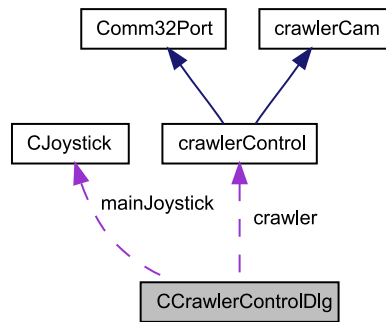


Figure 2.23: MR-15d Collaboration Diagram

this new software is shown in Figure 2.23. The previously mentioned *CJoystick* class is a wrapper for the DirectInput library and includes functions for configuring and reading data from a USB joystick. The *crawlerCam* class includes functions for reading and writing the configuration registers of a OV7930 imager. The *Comm32Port* class provides an interface to a serial port, but was not developed by the author [61]. The *crawlerControl* class inherits the *Comm32Port* and *crawlerCam* base classes and provides functions for motor control, focus actuator control, and camera control. The *CCrawlerControlDlg* class is the main dialog class. It instantiates *CJoystick* and *crawlerControl* objects to interface with a USB joystick and a MR-15d or MR-20b

robot device. This class also updates all GUI objects in the main dialog. This new control software was developed to control both the MR-15d and MR-20b robots, which were designed and built concurrently. The MR-15d, which has no imager or focusing actuators, only required the motor drive functions for operation.

The author also set up all support hardware prior to *in vivo* testing of this device. During the testing, the author controlled the device under direction of the surgeon.

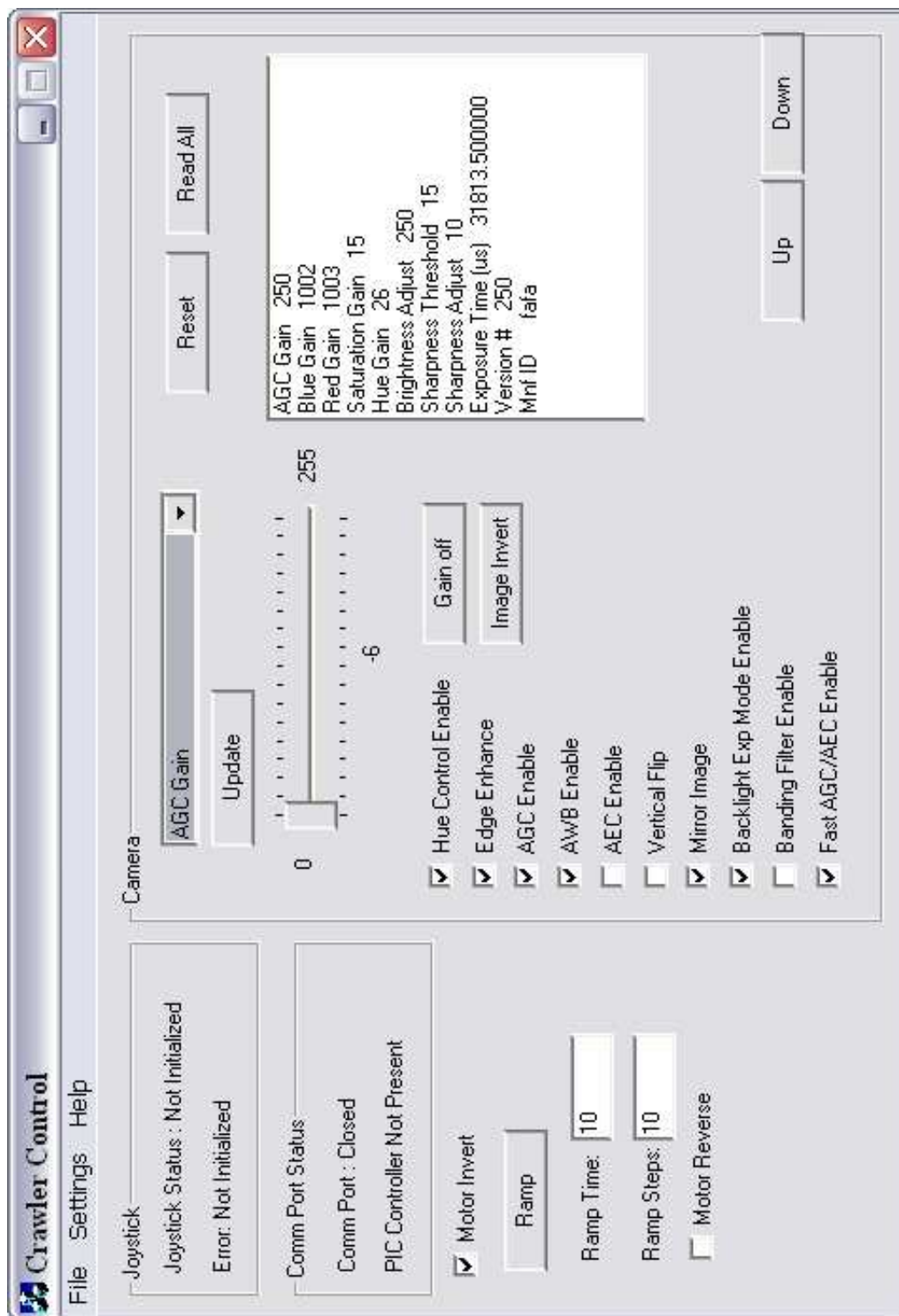


Figure 2.24: Crawler Control: Main window

2.8 Crawler (MR-20b)

The MR-20b adds a camera board to the MR-20a device. The crawler requires a 10 mm wide ribbon cable for camera signals and power, in addition to the existing motor wires. This device does not incorporate adjustable focus capabilities, due to space constraints. The software used to control the MR-15d is used to control this device.

The MR-20b was tested *in vivo* on January 21, 2005. The movement of the device was severely limited by the addition of the flat ribbon cable. The additional wires also had a tendency to elevate the tail of the device, which forced the camera to tilt down about 10°. The camera functioned returned an image that was observed to be acceptable, but the lens was smudged easily.

2.8.1 Author's Contributions on MR-20b

The author's contributions in the design and testing of the MR-20b robot were electronics hardware design and *in vivo* testing. The details on these contributions follow.

An imager PCB was designed and fabricated for the MR-20b robot. It contains a OV7930 imager and support components for the imager (see Figure A.6 in Appendix A). The drive electronics used for the MR-15d device were also used for the MR-20b device. Since the MR-20b included an imager, the OV7930 serial camera interface on the PIC driver PCB was used to control imager settings. The control software

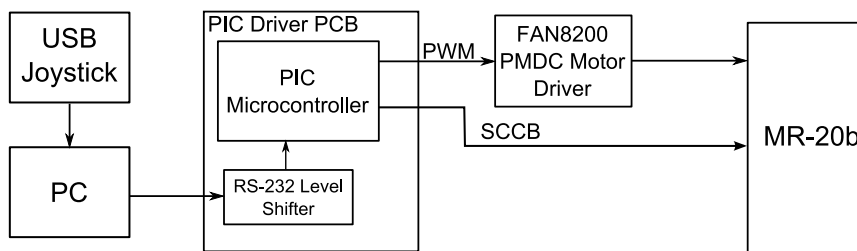


Figure 2.25: MR-20b Drive Electronics

developed for the MR-15d was also used to control the MR-20b. The MR-20b took advantage of the camera control settings to enable adjustment of the video image during testing.

The author also set up all support hardware prior to *in vivo* testing of this device. During the testing, the author controlled the device under direction of the surgeon.

2.9 Tilt Camera(T-15a)

The T-15a is a fixed base camera device with tilt capability. The device rotates about a point roughly in the middle of the main body tube to adjust the angle of the camera from 0° (Figure 2.26(a)) to about 45° (Figure 2.26(b)). The T-15a also has an adjustable focus mechanism, built in LEDs for illumination, and a small handle to grasp the device using a laparoscopic tool.

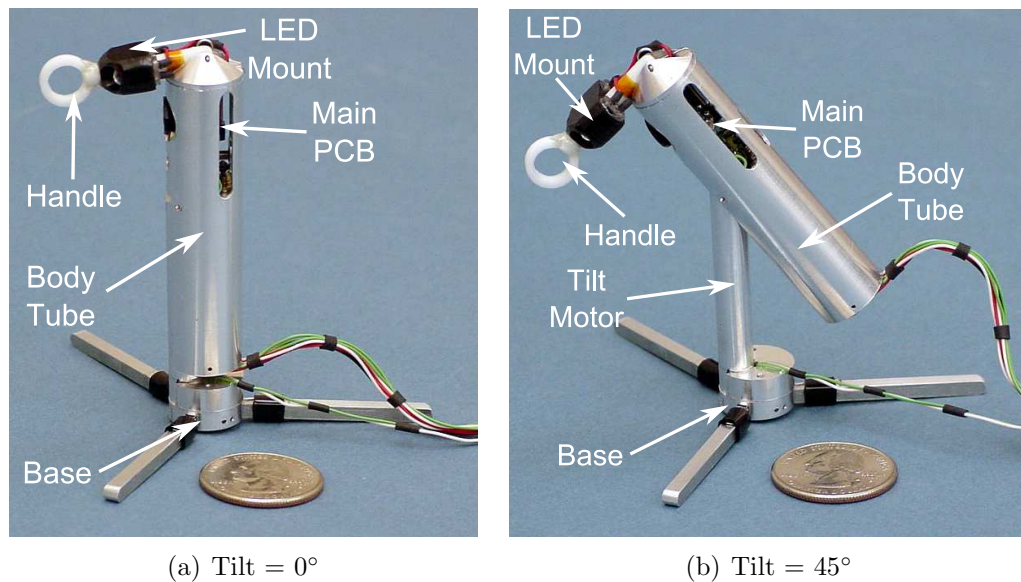


Figure 2.26: T-15a Fixed Base Camera

The T-15a was tested *in vivo* on March 18, 2005. There was difficulty in placing the device due to the top heavy design and the freely swinging tilting mechanism.

Also, the tripod base was unstable on the uneven surface. The tilting mechanism and LEDs functioned as designed. The camera functioned well initially, but quickly became cloudy. After the test, the lens was found to have fluid in it, which resulted in the cloudy image.

2.9.1 Author's Contributions on T-15a

The author's contributions in the design and testing of the T-15a robot were electronics hardware design and *in vivo* testing. The details on these contributions follow.

An imager PCB was designed and fabricated for the T-15a robot. The schematic is identical to the MR-20b imager PCB (see Figure A.6 in Appendix A), but a new circuit board layout was designed to be compatible with the different mechanical design of the T-15a device.

The T-15a used the same drive electronics (Figure 2.27) and control software as the MR-20b and MR-15d. The tilt actuation is controlled from the joystick, and the focus actuator is controlled by two pushbuttons in the GUI. The author also set up

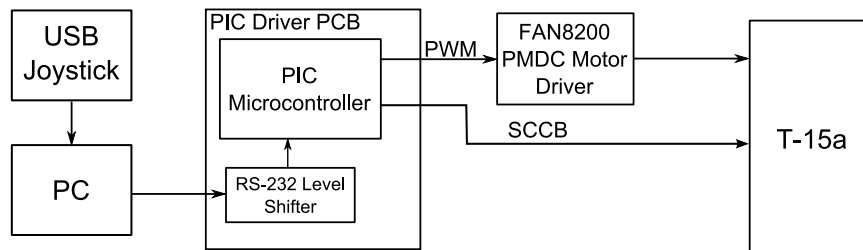


Figure 2.27: T-15a Drive Electronics

all support hardware prior to *in vivo* testing of this device. During the testing, the author controlled the device under direction of the surgeon.

2.10 Crawler (MR-20c)

The MR-20c (Figure 2.28) is a 20 mm diameter crawler. It features a Omnivision OV7930 imager, two drive motors, and focusing capability. A block diagram of the electronics can be seen in Figure 2.29. To reduce the number of external wires needed, the control and motor drive electronics are integrated into the robot using two PCBs. PCB 1 contains a PSoC microcontroller and FAN8200 H-Bridge ICs to control the three motors on the device. PCB 2 contains the OV7930 imager and was similar in design to previous OV7930 PCBs. The robot is connected externally to a control box, which provides power, control signals, and a video interface. The control box serves to provide power to the device, as well as send control signals from the PC to the device. The control box is connected via a RS232 serial interface to a PC running a control program. The crawler movement commands are input via a USB joystick attached to the PC.

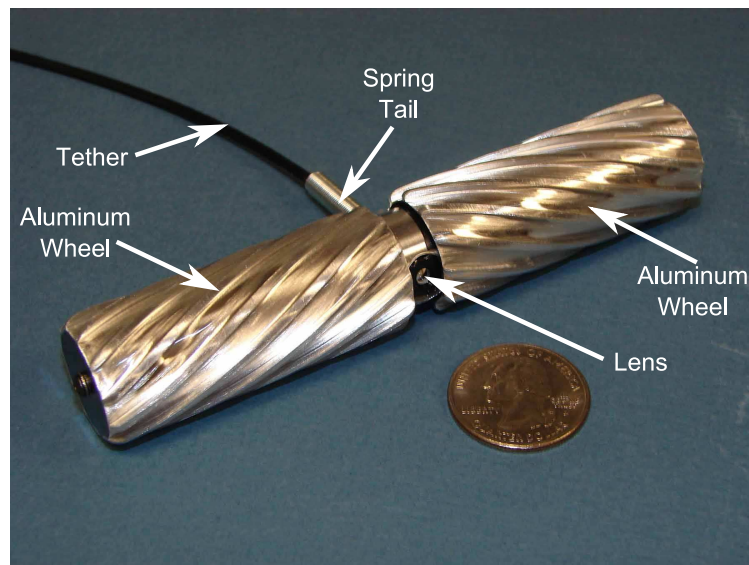


Figure 2.28: Crawler (MR-20c)

The MR-20c was tested *in vivo* on September 12, 2005 in a canine surgery conducted at the University of Rochester Medical Center. Two procedures, a prostatec-

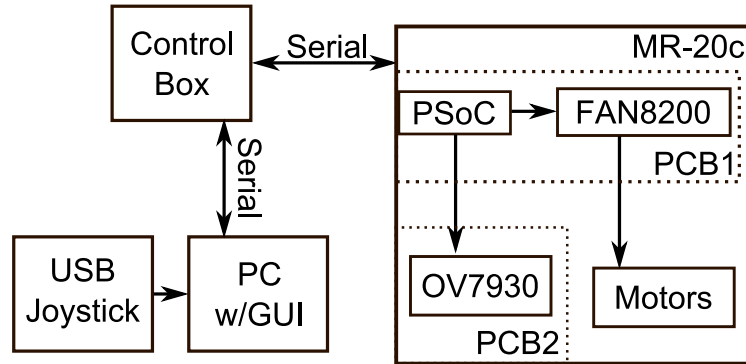


Figure 2.29: MR-20c Block Diagram

tomy and a nephrectomy, were completed using the MR-20c and the T-15b (Section 2.11) with the laparoscope for visual feedback. Two MR-20c devices were available for the procedures. The mobility of the MR-20c was still hindered by thin tissues, such as the omentum. The focusing on one device failed, and one of the wheels on the other device was not functioning properly. The devices were used to view the abdominal wall, assist with proper trocar placement, and provided the surgeon with multiple viewing angles [62]. The devices provided the surgeon with an additional frame of reference not available with a standard endoscope.

A variant of the MR-20c, seen in Figure 2.30, was also built. This device was used in the NEEMO experiments in March, 2006, that will be fully described in Chapter 4. The only difference in this variant was the wheel shape, which was grooved with rubber washers placed in the grooves to facilitate movement in the simulator box which was constructed for the NEEMO experiments.

2.10.1 Author's Contributions on MR-20c

The author's contributions in the design and testing of the MR-20c robot were electronics hardware design and *in vivo* testing. The details on these contributions follow.

As previously mentioned, the MR-20c contained two separate PCBs: one with

a microcontroller and motor drivers and one with an imager. A schematic of the microcontroller/motor driver PCB can be seen in Figure A.7 in Appendix A. The schematic of the imager PCB is identical to previous OV7930 PCBs and can be seen in Figure A.6.

An control box for the MR-20c robot was also designed and built by the author. This control box allows for control of a MR-20c and a T-15b tilt camera simultaneously. This control box is shown in Figure 2.31. The control box has a two line liquid crystal display (LCD) for displaying status information to the user, a rotary selector switch for selecting the device to control, another rotary control for LED brightness adjustment, a switch for actuating the camera focus and T-15b tilt adjustment, and a power switch. The device is powered from a 12 volt wall adapter and connects to a PC via a RS-232 serial cable. The schematic for the control box PCB can be seen in Figures A.9 and A.10, with a block diagram shown in Figure 2.32. The PC communicates with the control box via a RS-232 serial connection. The main processor of the control box is a RCM3200 core module, featuring a Rabbit 3000 microcontroller [63]. The MR-20c device is connected directly to the RCM3200 over a two-wire serial link. The RCM3200 translates joystick data and imager commands from the PC to commands for the MR-20c robot. The control box also serves as the user interface to the

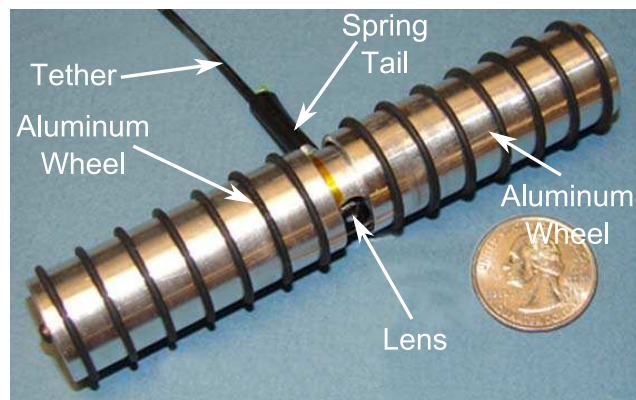


Figure 2.30: MR-20c (NEEMO variant)

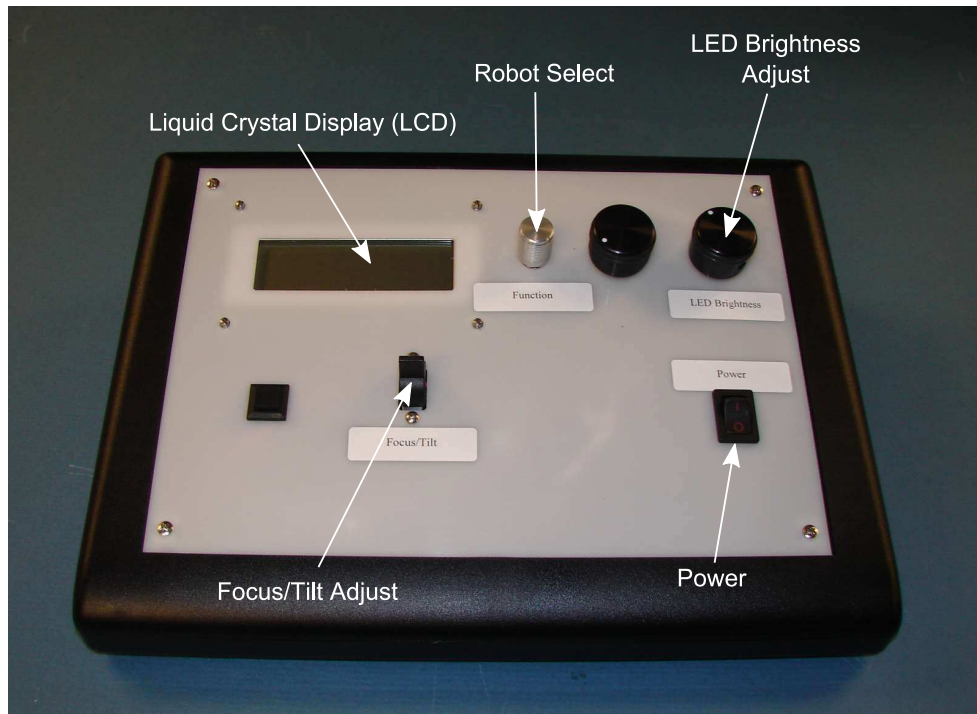


Figure 2.31: MR-20c/T-15b Control Box

T-15b robot. The RCM3200 generates PWM signals for a FAN8200 motor driver, which drive the focus and tilt motors on the T-15b. A PIC16F873 microcontroller from Microchip is connected to the RCM3200 via another two-wire serial bus. The PIC16F873 is capable of controlling up to two four separate OV7930 imagers, although only one is used in this setup. The function switch and tilt/focus switches are connected to a Maxim MAX7311 I2C port expander, which allows the RCM3200 to read in multiple digital signals over a single I2C connection. The control box is also capable of controlling an separate LED light source, used in the NEEMO experiments (see Chapter 4). A quadrature encoder is used as a brightness control, and is read directly by the RCM3200's built in quadrature decoder. Finally, a LCD is connected to the RCM3200's external data bus. The LCD is used for displaying status messages to the user.

The software on the MR-20c device was written in C and assembly language by

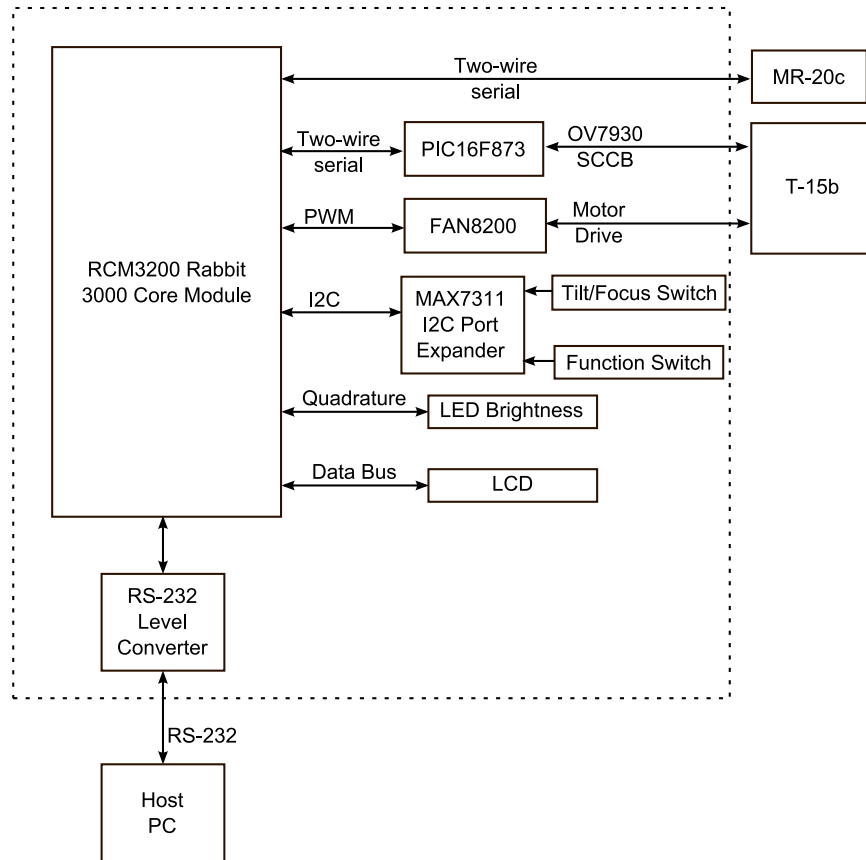


Figure 2.32: MR-20c/T-15b Control Box Block Diagram

the author. The software initializes the OV7930 imager and then starts an loop, waiting for an external command to arrive. When a command arrives, the software parses it and generates either motor signals for the drive motors or focus actuator, or a camera command for the OV7930 imager.

The software on the control box was written in C. A flowchart of the program is shown in Figure 2.33. The program first checks for a joystick command packet and processes it if one is present. It then reads in the current state of the LED brightness control, mode switch, and tilt/focus switch. The LCD is then updated with status information. The program then updates the appropriate device based on the state of the mode and tilt/focus switches.

The author also set up all support hardware prior to *in vivo* testing of the MR-20c

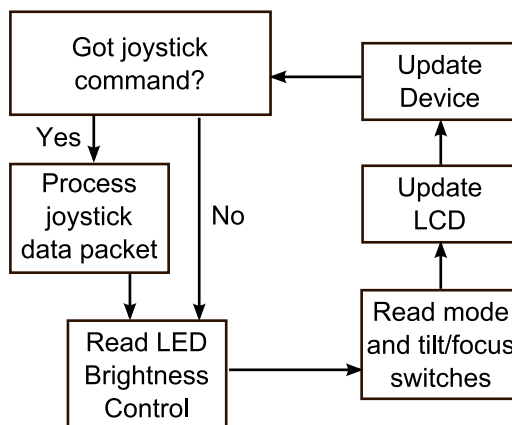


Figure 2.33: Control Box Software

device. During the *in vivo* testing, the author controlled the device under direction of the surgeon.

2.11 Tilt Camera(T-15b)

The T-15b (Figure 2.34) is a 15 mm tilt camera device. Electrically, it is identical to the T-15a device. There are some mechanical changes that deal with the tilting mechanism, handle, and sealing of the device. The T-15b is connected to the same control box as the MR-20c. Unlike the MR-20c, this device does not have an on-board controller or motor driver, due to space constraints. A block diagram of the T-15b system can be seen in Figure 2.35.

The T-15b was tested alongside the MR-20c on September 12, 2005, in the canine surgery described in Section 2.10. The video quality was sufficient to aid in trocar placement and providing additional viewing angles during the procedures, but the lens had to be changed after the first lens became cloudy due to a leak. The tilting functionality was acceptable, and focusing mechanism was functional. As with the MR-20c, this device provided the surgeon with an additional frame of reference not available with a standard endoscope.

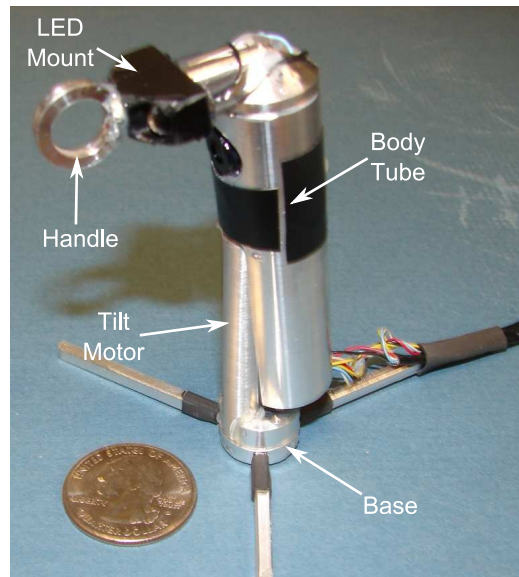


Figure 2.34: Tilt Camera (T-15b)

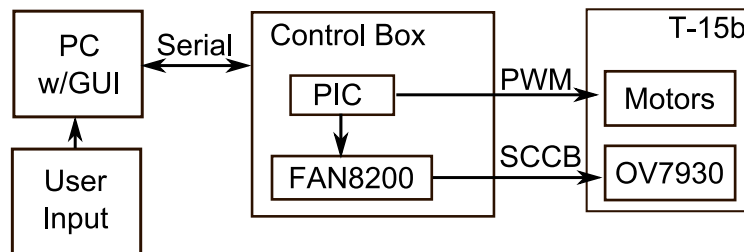


Figure 2.35: T-15b Block Diagram

A variant of the T-15b, seen Figure 2.36, was also built. The only change was the handle and LEDs on the top of the device were replaced with a flat cap.

This device, along with the MR-20c, was used in the NEEMO experiments in March, 2006, that will be fully described in Chapter 4.

2.11.1 Author's Contributions on T-15b

The author's contributions in the design and testing of the robot were electronics hardware design and *in vivo* testing. A schematic of the OV7930 imager PCB can be seen in Figure A.6. The T-15b robot shares control hardware and software with the

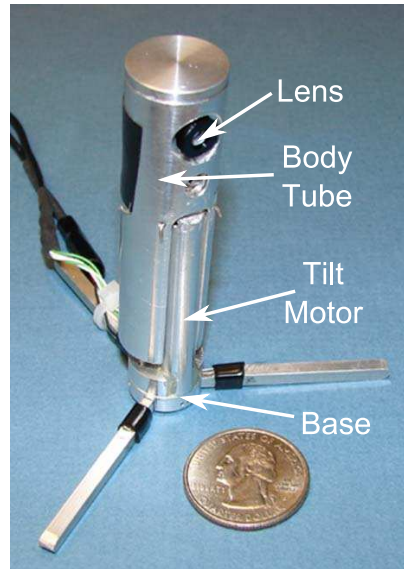


Figure 2.36: T-15b NEEMO variant

MR-20c device. Details of this hardware and software can be found in Section 2.10.1.

The author also set up all support hardware prior to *in vivo* testing of the T-15b device. During the *in vivo* testing, the author controlled the device under direction of the surgeon.

2.12 Biopsy Crawler (BMR-22a)

The BMR-22a (Figure 2.37) is a 22 mm crawler with aluminum helical profile wheels. The device is externally powered and controlled. It features a camera based on the Omnivision OV7930 imager and a biopsy grasper. This grasper is used to sample a specific piece of tissue for later analysis. A block diagram can be seen in Figure 2.38. User input is from a USB joystick and an on-screen GUI running on the host PC. The robot and PC interface over a serial link. A PSoC controller on the robot interprets commands from the PC and commands the OV7930 imager and the FAN8200 H-Bridge motor drivers, which drive the drive wheels and the biopsy arm actuator.

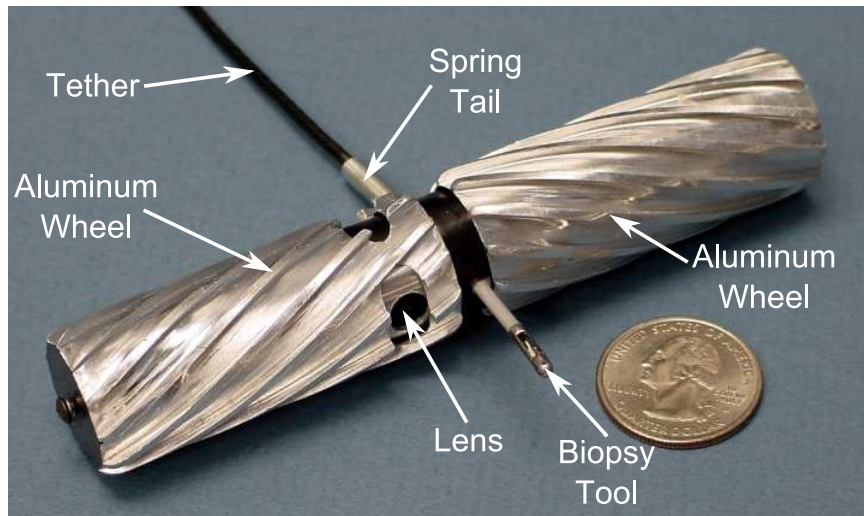


Figure 2.37: BMR-22a Biopsy Crawler

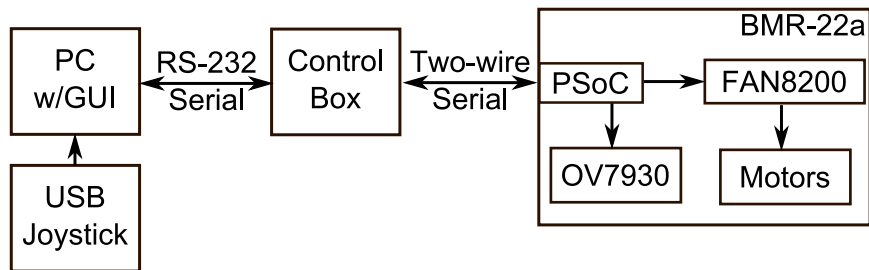


Figure 2.38: BMR-22a Block Diagram

The BMR-22a was tested *in vivo* December 14, 2005. The goal was to demonstrate tissue manipulation by taking a sample of liver tissue. This goal was accomplished, as the device successfully sampled the liver twice. The BMR-22a was also successful in traversing several tissues, including liver and small bowel.

2.12.1 Author's Contributions on BMR-22a

The author's contributions in the design and testing of the robot were electronics hardware design and *in vivo* testing. The details on these contributions follow.

The author designed and tested a PCB for the BMR-22a robot. The schematic for this board can be seen in Figure A.11 in Appendix A. The functionality of this board is similar to the two board combination found in the MR-20c robot (see Section 2.10), except the third motor driver is used to drive the biopsy actuator instead of a camera focus actuator. The control box described in Section 2.10.1 is used to control the BMR-22a robot.

The author also set up all support hardware prior to *in vivo* testing of the BMR-22a device. During the *in vivo* testing, the author controlled the device under direction of the surgeon.

2.13 Wireless Crawler (WMR-22a)

The WMR-22a (Figure 2.39) is a 22 mm crawler with aluminum helical profile wheels. The device is battery powered and receives motor commands over a wireless link from the host computer. A block diagram of the WMR-22a can be seen in Figure 2.40. The user controls the crawler from a USB joystick, which is connected to a PC running a control program. The wireless transceiver is connected to the PC via a RS-232 link. The crawler commands are received by the transceiver on the WMR-22a device. These commands are sent to the PSoC controller over a two-wire serial link. The PSoC controller converts these commands to PWM commands for the FAN8200 H-Bridge IC, which drives the motors on the robot. Due to space constraints in the robot and lack of a suitable wireless video transmitter, a camera was not included on this device.

The WMR-22a was tested *in vivo* on December 14, 2005. The goal was to test wireless transmission of motor commands. The wireless transmission of commands was intermittent, but the robot worked well when the transmission was good. The



Figure 2.39: WMR-22a Wireless Crawler

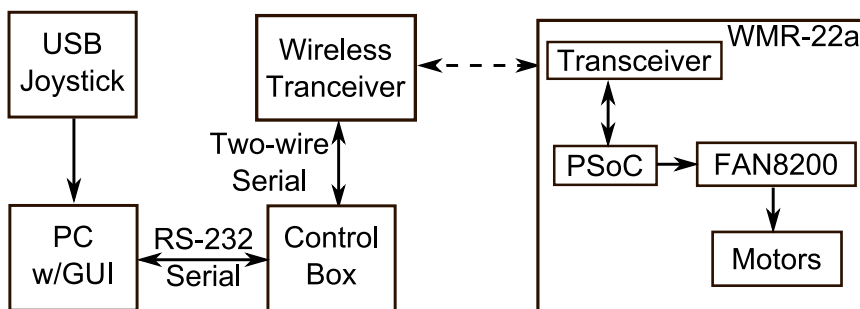


Figure 2.40: WMR-22a Block Diagram

transmission was much better when the transmitting antenna was placed inside the abdomen of the pig, indicating that the transceivers did not have enough power to transmit through the abdominal wall. The mobility of the device was hindered by the use of shorter wheels, which caused the crawler to frequently become high centered. This device did prove that wireless control of these *in vivo* robots is possible.

2.13.1 Author's Contributions on WMR-22a

The author's contributions in the design and testing of the robot were electronics hardware design and *in vivo* testing. The details on these contributions follow.

The author designed and tested a PCB for the WMR-22a robot. The schematic

for this board can be seen in Figure A.12 in Appendix A. The control box described in Section 2.10.1 is used to control the WMR-22a robot.

The author also set up all support hardware prior to *in vivo* testing of the WMR-22a device. During the *in vivo* testing, the author controlled the device under direction of the surgeon.

2.14 Crawler (MR-12a)

The MR-12a is a 12 mm crawler designed to demonstrate natural orifice surgery with *in vivo* robots. It has aluminum wheels, 2 PMDC motors, but no other electronics (See Figure 2.41).

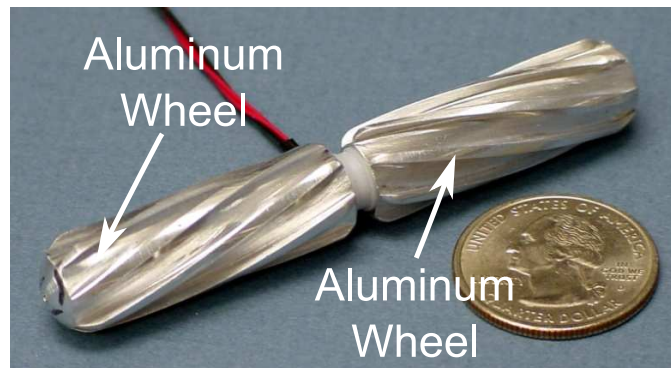


Figure 2.41: MR-12a Endoluminal Crawler

The MR-12a was tested *in vivo* on December 14, 2005. The device was introduced into the stomach through the esophagus. An incision in the stomach was created, and the device was introduced into the abdominal cavity. The crawler then traversed the abdominal cavity. The crawler was then moved back into the stomach, and retrieved back through the esophagus. The device performed well throughout the test, in spite of rigorous handling and use in very wet conditions.

2.14.1 Author's Contributions on MR-12a

The author's contributions in the design and testing of the robot were electronics hardware design and *in vivo* testing. The details on these contributions follow.

The control box described in Section 2.10.1 is used to control the MR-12a robot. The software on the control box was modified by the author to add a mode for controlling a simple mobile robot with two drive motors.

The author also set up all support hardware prior to *in vivo* testing of the MR-12a device. During the *in vivo* testing, the author controlled the device under direction of the surgeon.

2.15 Mobile Camera (MRC-100)

The Mobile Robotic Camera, or MRC-100, is a mobile camera device. These were designed to be sterilizable and easier to mass produce than previous robots. Two versions were built: Version A (Figure 2.42) has a metal body and machined plastic wheels and Version B (Figure 2.43) has an all plastic body and wheels. The drive electronics are similar in both versions (see Figure 2.44). Each has two PMDC motors for movement, a CMOS imager, focusing capability using a voice-coil motor type actuator, a Atmel AT89C5131 microcontroller [64], and a TB6557 H-Bridge chip from Toshiba [65]. The AT89C5131 is a 8051 based device with full speed USB controller on-board. It also features a UART, 5 PWM channels, and an internal oscillator. The TB6557 device contains four H-bridge drivers, used for driving the motors; and two constant current drivers, one of which drives the focusing actuator. It interfaces with the microcontroller via a three-wire serial interface and independent PWM channels.

The imager used is a MT9V125 from Micron [66]. The color CMOS sensor has a 640x480 pixel array with NTSC video output. The MT9V125 is a 1/4 inch optical

format imager housed in a 52-Ball IBGA package, measuring 9 mm x 9 mm. It features color correction, auto white balance, gamma correction, and black level conditioning. It communicates with the microcontroller via a two wire serial interface. The imager, microcontroller, and motor driver are all integrated on a custom PCB. A six conductor cable connects the device to an external breakout box (added after the April 10, 2007 test), which provides a USB port for connecting to a host PC, a RCA jack for video output, and a power input jack.

The software on the microcontroller configures the device as a virtual serial port. The operating system on the control PC identifies the device as a standard serial port via a device driver supplied by Atmel [67]. A custom control program on the PC sends user commands from a USB joystick and the GUI to the robot. A block diagram of the system can be seen in Figure 2.44.

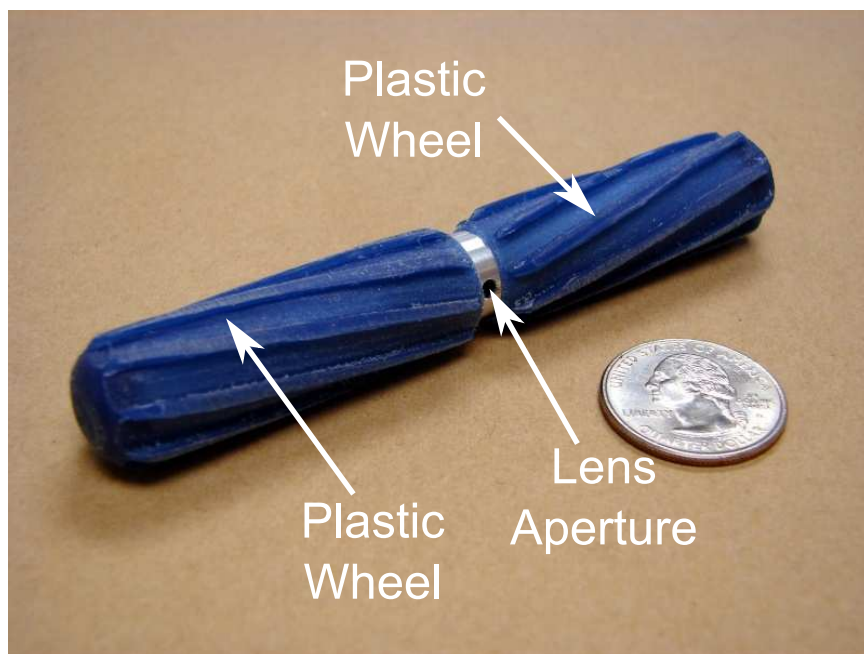


Figure 2.42: MRC-100 Version A

The MRC-100 version A was tested *in vivo* on August 11, 2006. All motors, actuators, and electronics were functional in benchtop testing prior to the *in vivo*

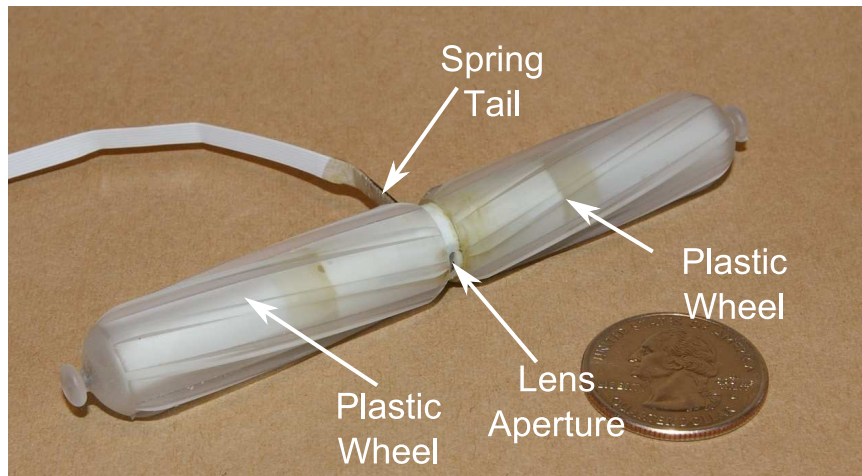


Figure 2.43: MRC-100 Version B

test. During the test, there were some problems with excessive camera noise and interference. Also, the use of an electrocautery generator caused the MRC-100 to reset or simply stop responding. It was determined that more adequate shielding was needed to prevent this kind of interference. Mobility was also difficult due to problems navigating with the on-board camera and the tendency of the device to wrap up in its own tether.

MRC-100 Version B was tested on April 10, 2007. The device was sterilized prior to the surgery. After sterilization, the device was tested on the benchtop to ensure the sterilization process did not have any detrimental effects on its operation. The motors, focusing mechanism, and camera adjustments functioned as they did prior to sterilization. The electrocautery interference was still a problem, and it was also noted that the USB power supply was noisy. This caused interference in the video output.

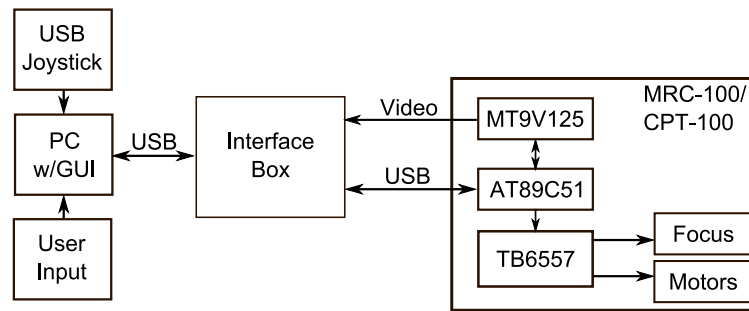


Figure 2.44: MRC-100/CPT-100 Block Diagram

2.15.1 Author's Contributions on MRC-100

The author's contributions in the design and testing of the robot were electronics hardware design and *in vivo* testing. The details on these contributions follow.

The author designed and tested a PCB for the MRC-100. A schematic for this board can be seen in Figure A.13 in Appendix A. This schematic was shared with the PCB designed for the CPT-100 (Section 2.16), but the layouts of the two boards differ slightly. The MRC-100 PCB (see Figure 2.45) was designed for two motors, where the CPT-100 was designed for one motor (see Figure 2.49). The software for the MRC-100, also developed by the author, is described in Section 2.17.

The author also set up support hardware prior to *in vivo* testing of the MRC-100 device. During the *in vivo* testing, the author controlled the device under direction of the surgeon.

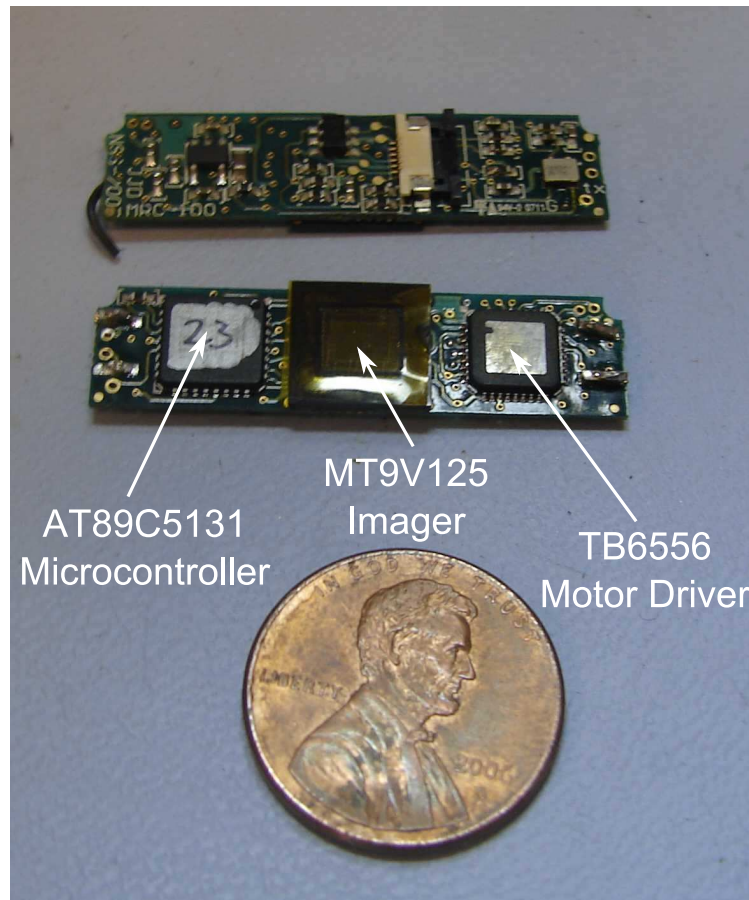


Figure 2.45: MRC-100 PCB

2.16 Ceiling Pan-Tilt (CPT-100)

The Ceiling Pan-Tilt, or CPT-100, is a fixed camera that attaches to the abdominal wall by means of an external magnetic handle. Figure 2.46 shows the CPT device and magnetic handle on the benchtop. The surgeon can move the handle, which changes the viewpoint of the robot. The CPT-100 also has a motor which allows the camera to tilt $\pm 45^\circ$. This panning and tilting allows for a complete view of the surgical environment. As with the MRC-100, two versions of the CPT-100 were developed: an all metal prototype (Figure 2.47) and a plastic body device (Figure 2.48). The drive electronics are nearly identical to the MRC-100 (Section 2.15), except in the

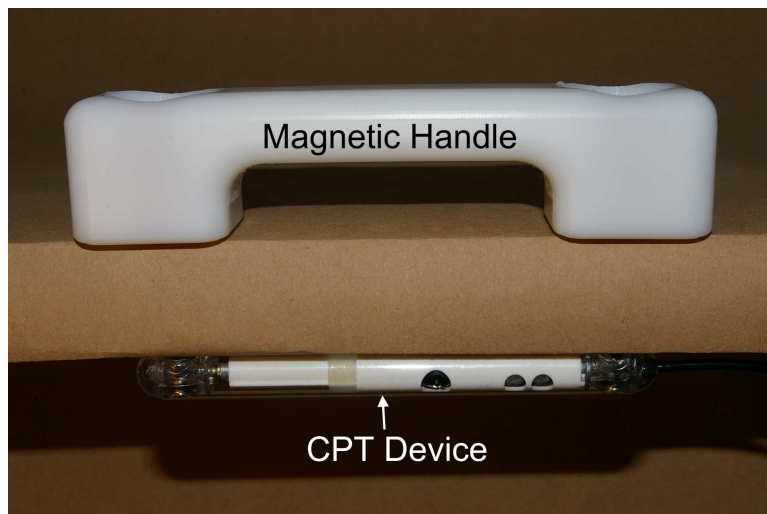


Figure 2.46: CPT-100 Magnetic Attachment

place of one motor are two white LEDs used for illumination.

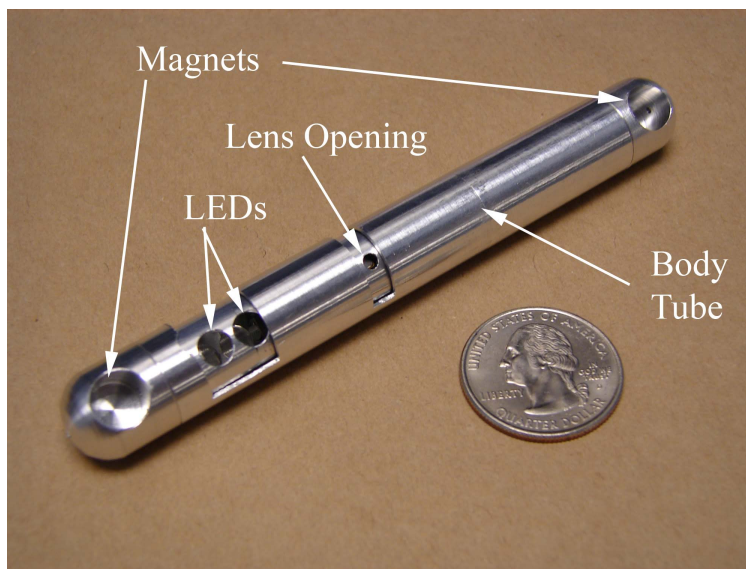


Figure 2.47: CPT-100 Version A

Version A of the CPT-100 was tested *in vivo* on August 11, 2006. The device functioned well, with no major difficulties. Version B was tested *in vivo* on April 10, 2007; May 29, 2007; and July 17, 2007 in survivable animal studies. These studies are discussed in detail in Chapter 5. The CPT-100 was also tested *in vivo* on August

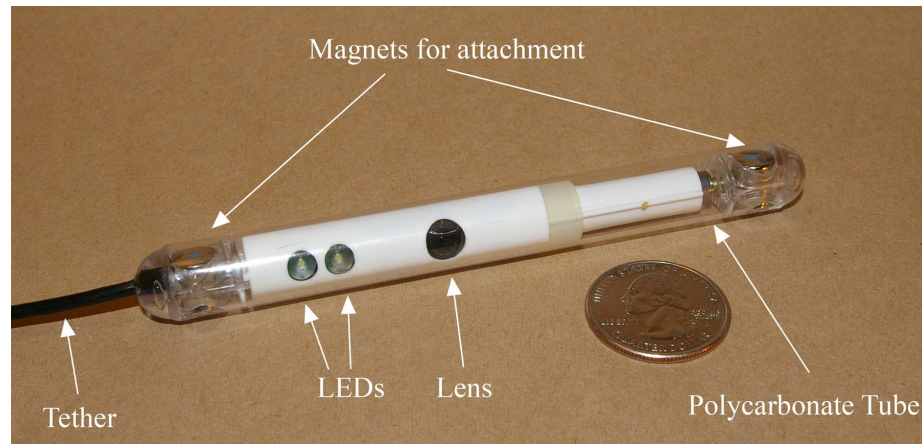


Figure 2.48: CPT-100 Version B

21, 2007. This test was a NOTES procedure in which the CPT was introduced into the abdominal cavity through an incision in the stomach.

2.16.1 Author's Contributions on CPT-100

The author's contributions in the design and testing of the CPT-100 robot were electronics hardware design and *in vivo* testing. The details on these contributions follow.

The author designed and tested a PCB for the CPT-100 (see Figure 2.49). A schematic for this board can be seen in Figure A.13 in Appendix A. As previously mentioned, this PCB is similar to the MRC-100 PCB. The software for the MRC-100, also developed by the author, is described in Section 2.17. The author also set up all support hardware prior to *in vivo* testing of the CPT-100 device. During the *in vivo* testing, the author controlled the device under direction of the surgeon.

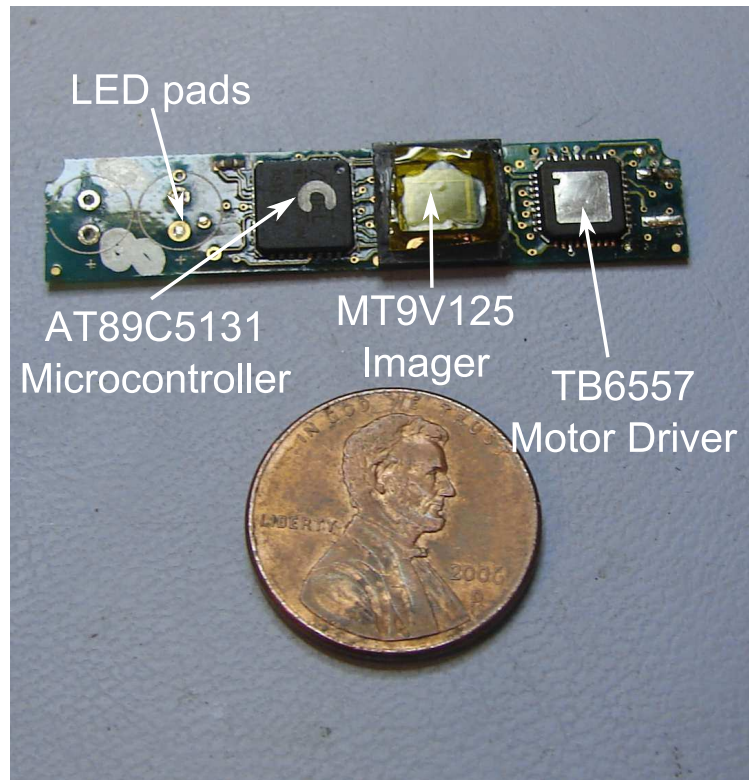


Figure 2.49: CPT-100 PCB

2.17 CPT-MRC Software

The control software for the CPT-100 and MRC-100 devices was developed and tested by the author. All Windows based software, with the exception of the USB CDC driver, was developed in Microsoft Visual Studio in the C++ language. The firmware for both devices was developed in μ Vision2, from Keil Software, in C. A block diagram of the software system is shown in Figure 2.50. The user interacts with the program through a graphical user interface, or GUI. This GUI, shown in Figure 2.51, was developed using the Microsoft Foundations Class (MFC) library, and is dialog based. The MRC-100 motors are controlled by the user through a USB joystick. The device control library, also written in C++ by the author, contains several C++ classes used to access the MRC-100 and CPT-100 devices. The device control li-

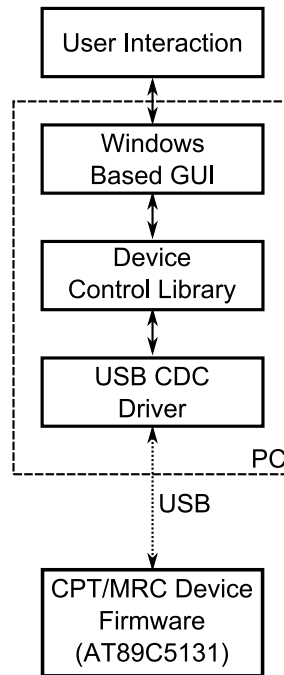


Figure 2.50: CPT-MRC Software

library communicates with the CPT/MRC device through the USB CDC driver. This driver, which is a standard Windows driver, implements the USB Communications Device Class interface. With the CPT/MRC device configured as a CDC device, the CPT/MRC device is displayed in Windows as a standard RS-232 serial port. This enables the device control library to access the device as a standard serial port, simplifying development. The CPT/MRC device firmware is based on a sample USB CDC implementation supplied with the microcontroller.

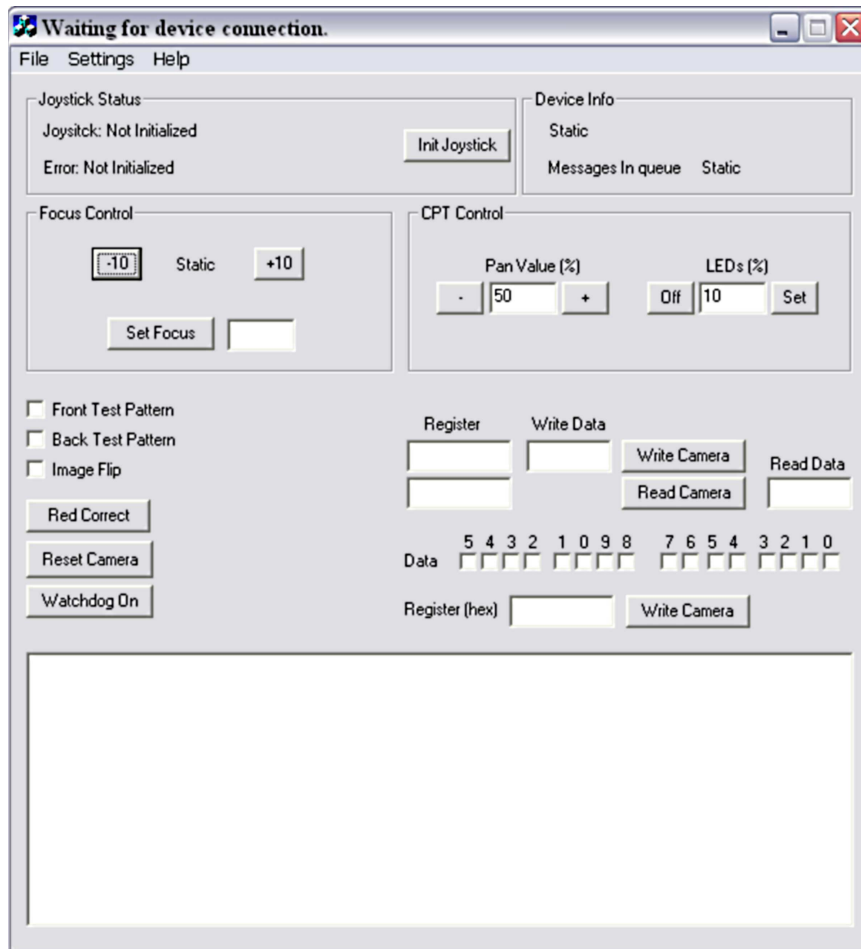


Figure 2.51: CPT-MRC User Interface

2.18 Dexterous Manipulator (AB1)

The AB1 (Figure 2.52) is the first generation of *in vivo* devices designed for NOTES procedures. It is designed to be an *in vivo* surgical platform, with cameras and instruments integrated on the device. This device has a central “body” link and two “arm” links, all made from machined aluminum. Each arm is connected to the body by a rotational shoulder joint providing yaw. The distal segment of each arm features a prismatic joint that moves each end effector. The left end effector is a gripper that is actuated by a small motor. The right end effector is a heat cautery device. A DC current is passed through the element, which then heats up enough to cut tissue.

Each end effector can also rotate about the long axis of the arm. The robot also has

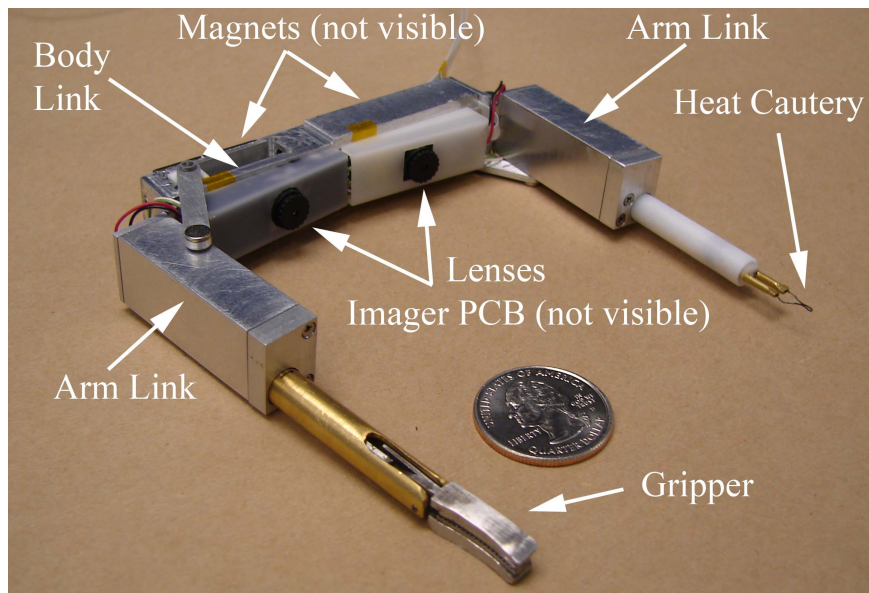


Figure 2.52: AB1 Robot

two permanent magnets attached to the back of the body. These magnets, along with an external magnetic handle, enable the device to “attach” to the internal abdominal wall during surgery (see Figure 2.55). A more detailed discussion of the mechanical design can be found in work by Lehman [55].

A block diagram of the electronics can be seen in Figure 2.53. The AB1 robot contains 2 MRC-100 PCBs (See Section 2.15). These boards are connected to an interface box with two flat flex ribbon cables. The interface box contains two breakout boards that have an RCA-type receptacle for video, and a USB connector for connection to the PC. The interface box also houses a National Instruments USB-6008 Analog/Digital I/O board. The 6008 reads signals from the main controller, the Virtual Incision (see Figure 2.54). The Virtual Incision is a controller that is based on a surgeon’s operation of standard laparoscopic tools during minimally invasive surgery. Potentiometers mounted on the device are used to sense position. Design details for this device can be found in work by Wood [56]. The GUI based program on the PC

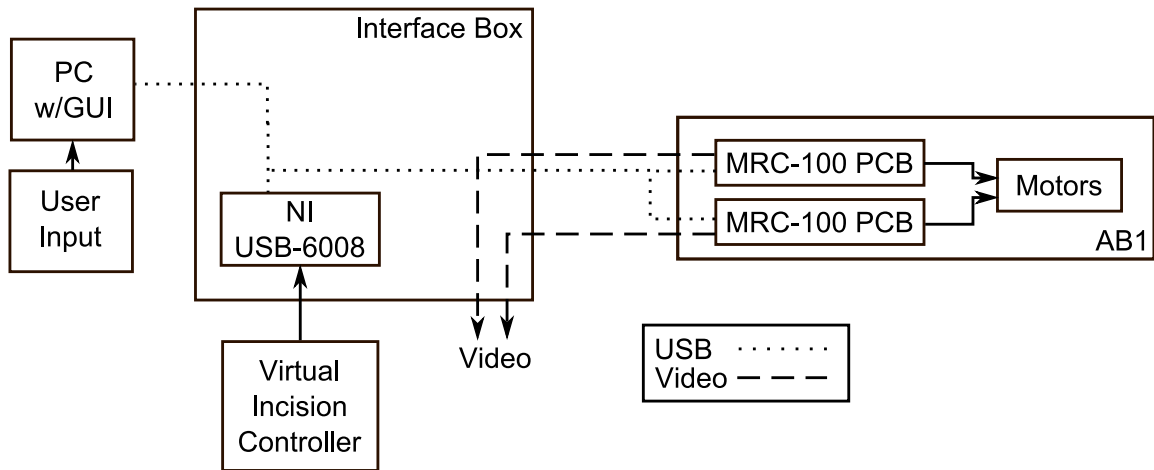


Figure 2.53: AB1 Block Diagram

receives data from the USB-6008 and through the GUI and sends motor and camera commands to the robot. The control scheme is an open-loop, velocity control setup, as no positional feedback mechanism was available to design a closed loop controller.

The AB1 was tested *in vivo* on May 15, 2007, in an anesthetized pig. An attempt was made to introduce the device into the peritoneal cavity via an incision in the stomach to demonstrate a NOTES-type procedure. The plan was to insert an overtube down the esophagus of the pig, and pass the AB1 down this tube and into the stomach. An incision was to be made in the lining of the stomach. Next, the surgeon would manually move the AB1 into the peritoneal cavity through this incision using the endoscopic tools. However, this approach failed when it was discovered that the esophagus of the pig used in the procedure was too small to allow passage of the overtube. The decision was then made to introduce the device into the peritoneal cavity via an abdominal incision. This incision was closed after the robot was inserted. The robot was controlled with the virtual incision controller and allowed the surgeon to successfully manipulate and cut tissue. In spite of the problems with insertion, this experiment demonstrated that a robot of this type could be used in NOTES procedures.

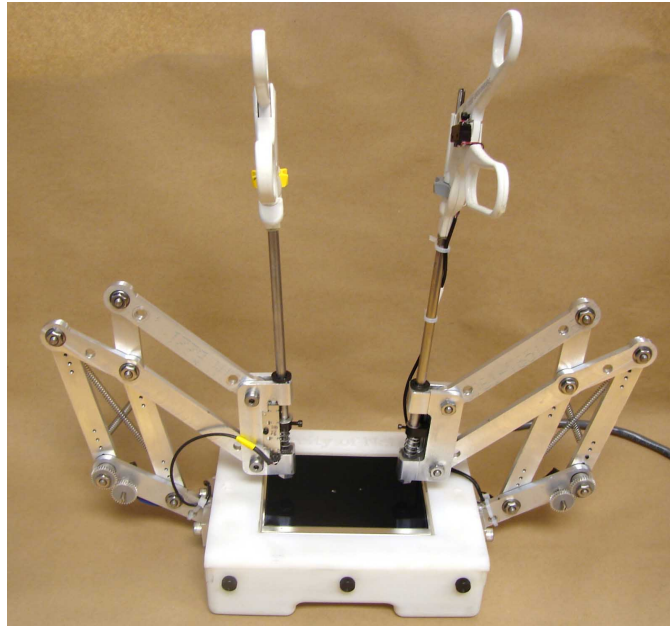


Figure 2.54: Virtual Incision Controller

2.18.1 Author's Contributions on AB1

The author's contributions in the design and testing of the AB1 robot were electronics hardware design, software development, and *in vivo* testing. Since the electronics hardware on the AB1 used existing PCBs from the MRC-100, the author's work was integration and design of the interface box described in the previous section. The AB software developed by the author will be discussed in Sections 2.21 and 6.2.4. During *in vivo* testing, the author set up all support hardware for the device, and monitored the device software interface while the surgeon controlled the robot.

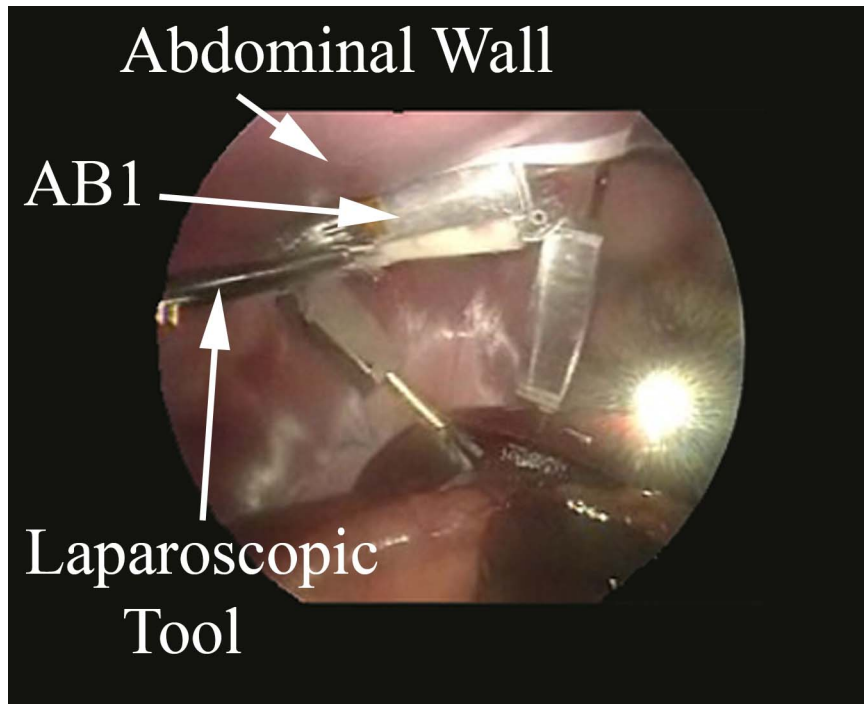


Figure 2.55: AB1 *In vivo*

2.19 Dexterous Manipulator (AB2)

The next revision of the AB robot series is the AB2, seen in Figure 2.56. This device differs from the AB1 in several ways:

1. Body built by a rapid-prototyping process rather than machined aluminum.
2. Stereo camera pair built on a separate module, attached to body with magnets.
3. Motor drivers are external.

The system block diagram for this device can be seen in Figure 2.57, and is very similar to the AB1 robot, except for the addition of two more MRC-100 PCBs in the interface box for driving the robot's motors. This change was made to allow a higher voltage supply for the motors. In the AB1, the motor supply voltage was limited to the supply voltage of the USB, which limited the total power available for driving

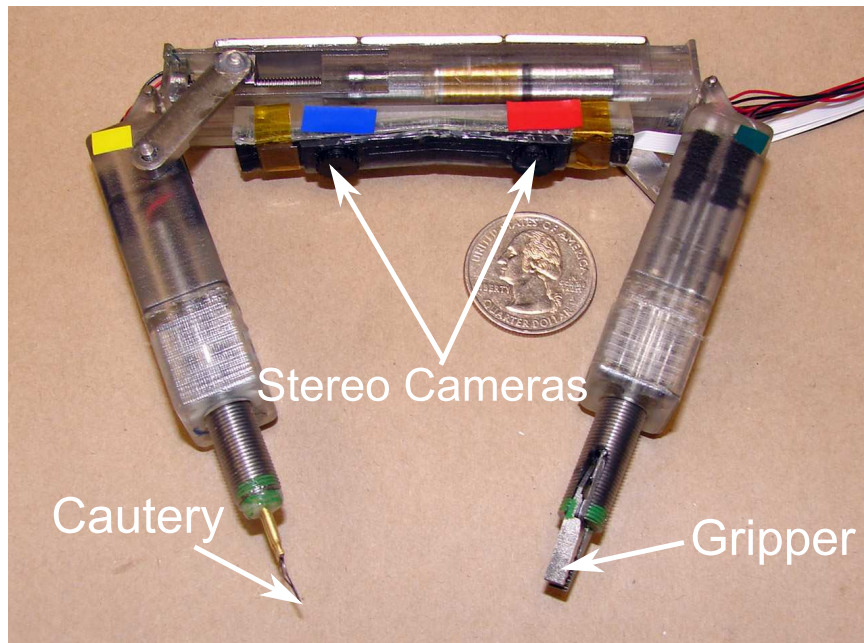


Figure 2.56: AB2

the motors. A new virtual incision controller (see Figure 2.58) is used to control the device. This device features two, three-axis analog joysticks. The USB-6008 is used to digitize the analog joystick values for use by the control program. As with the AB1, open-loop velocity control is used on AB2. The AB2 was tested *in vivo* on December 18, 2007 in an anesthetized pig. The plan was to perform a NOTES cholecystectomy, advancing the AB2 through an overtube into the stomach and then introducing the AB2 to the abdominal cavity through an incision in the stomach. At the beginning of the procedure, it was decided that the esophagus of the animal was too small to allow the AB2 to move through. The AB2 was then introduced to the abdominal cavity through an incision in the abdomen. Ultimately, mechanical problems forced the team to perform the operation as an open procedure.

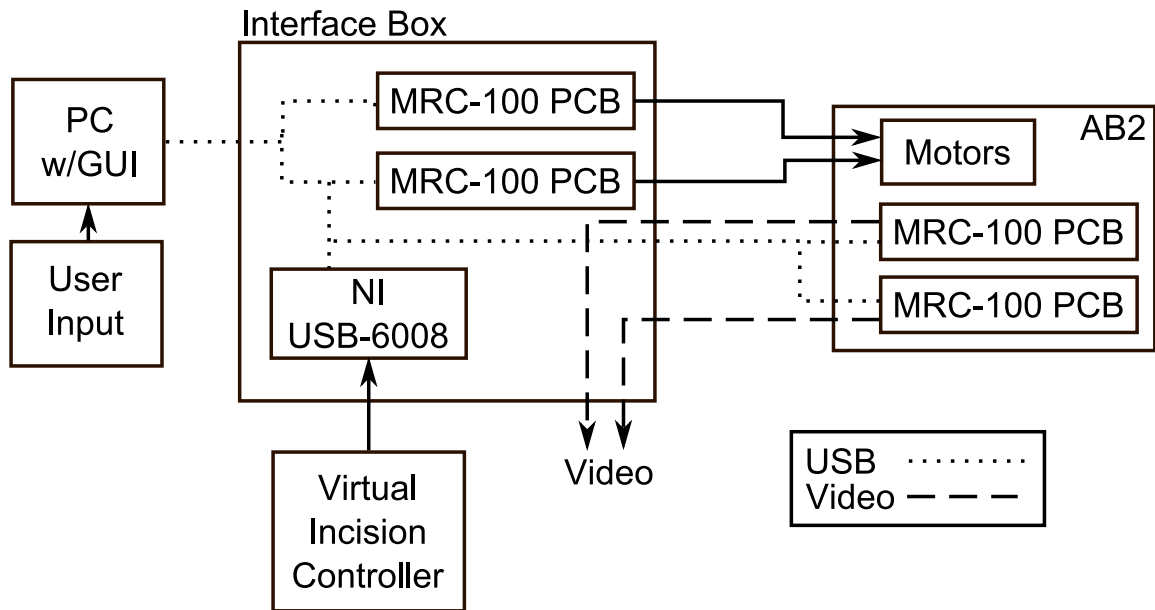


Figure 2.57: AB2 Block Diagram



Figure 2.58: Virtual Incision Controller 2.0

2.19.1 Author's Contributions on AB2

The author's contributions in the design and testing of the AB2 robot were electronics hardware design, software development, and *in vivo* testing. As with the AB1, the electronics hardware on the AB2 used existing PCBs from the MRC-100. The author's work was integration and design of an upgraded interface box described in the previous section. The AB software developed by the author will be discussed in Sections 2.21 and 6.2.4. During *in vivo* testing, the author set up all support hardware for the device, and monitored the device software interface while the surgeon controlled the

robot.

2.20 Dexterous Manipulator (AB3)

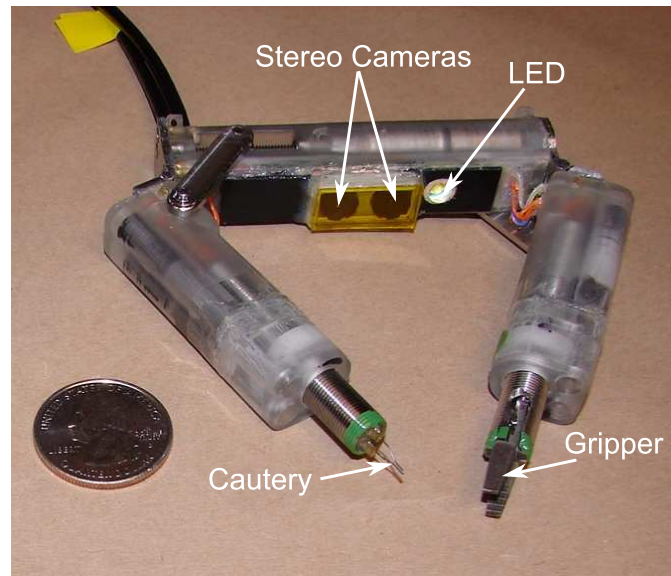


Figure 2.59: AB3

The third and final iteration of the AB robots is the AB3, seen in Figure 2.59. This device has a similar mechanical design as the previous AB robots, but the electronics are somewhat different. A block diagram of the AB3 system can be seen in Figure 2.60. The major change to the AB3 is the design of a dedicated dual imager PCB (see Figure 2.61). This PCB features two Micron MT9V135 imagers. The board also contains a ultra bright white XR-E series LED from Cree [68]. The XR-E series can produce up to 100 lm of luminous flux at 400 mA of current. This is the first AB robot to have a light source built in. Two MRC-100 PCBs are added to the interface box to communicate with the Micron imagers on the AB3. These are modified MRC boards with the imager chips removed and wires patched into the camera control lines. The other two MRC boards function as before on the AB2 device, driving the

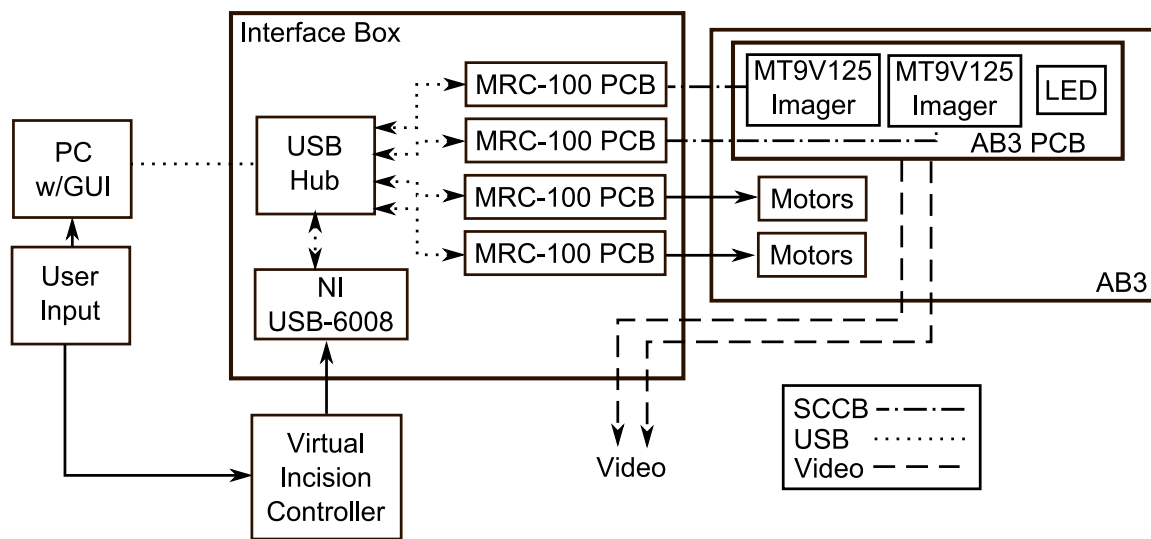


Figure 2.60: AB3 Block Diagram

motors on the robot. An LED illumination control was also added to the interface box (not shown in the drawing).

The AB3 was tested *in vivo* on March 25, 2008, in an anesthetized pig. Unfortunately, the cautery arm of the device was damaged during insertion and was only partially functional. The imagers worked well, and the video quality was sufficient for the surgeon to perform the procedure using only the AB3 video for visual feedback. Also, the gripper arm was fully functional, and was able to manipulate tissue well enough to perform a partial cholecystectomy with the partially functional cautery arm. The second *in vivo* test using AB3 was performed on May 30, 2008, in an anesthetized pig. In this test there were no problems, and the surgeon was able to use the device to perform a partial cholecystectomy. A video frame grab showing the view from both imagers can be seen in Figure 2.62. This successful test did, however, show some limitations of the AB3 platform. Because the rotation of the arms was constrained to one plane, no “up-and-down” motion was possible. This motion is essential to performing surgical procedures. The heat cautery tool was also noted to be inferior to the standard electrosurgical instruments typically used in laparoscopic

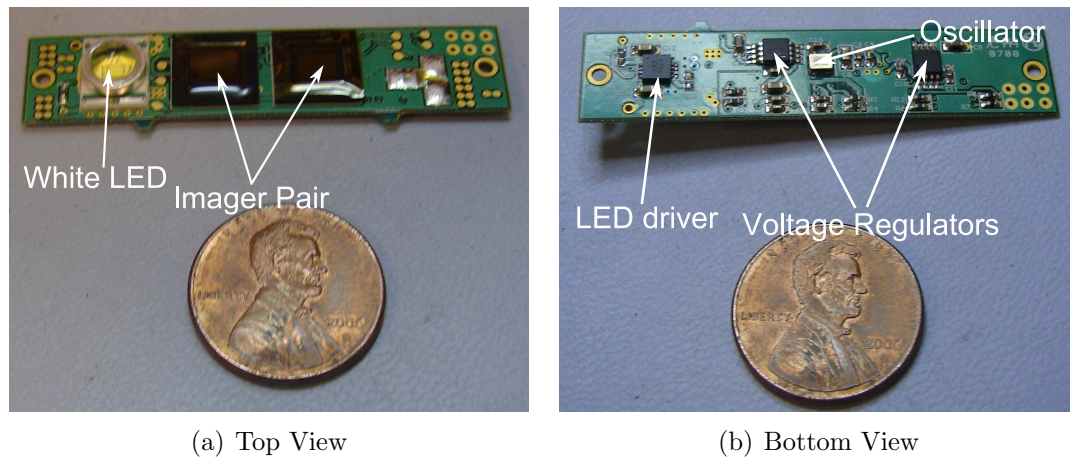
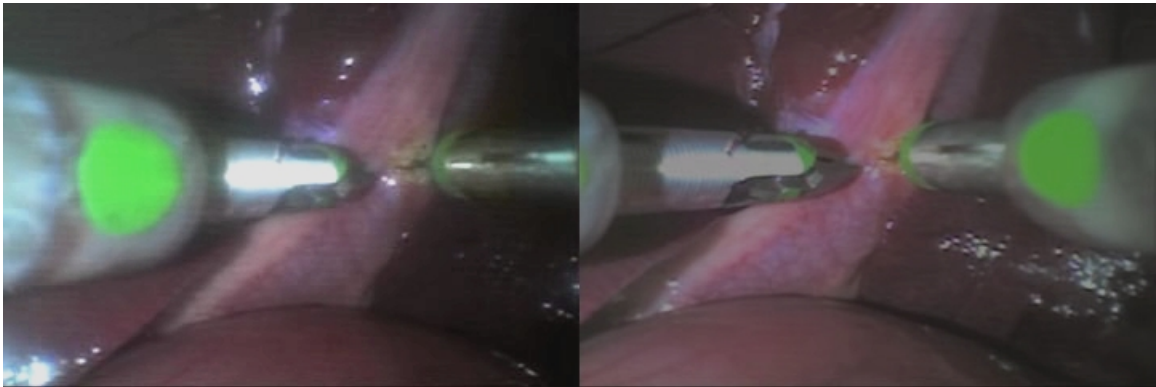


Figure 2.61: AB3 PCB

Figure 2.62: AB3 *In vivo* view

surgery. These tests did show that it is feasible to perform laparoscopic procedures using a completely *in vivo* surgical platform. The AB3 device was also used in extensive benchtop tests to show feasibility of using these types of robots for supervised autonomous surgical tasks. These experiments will be discussed in Chapter 6.

2.20.1 Author's Contributions on AB3

The author's contributions in the design and testing of the AB3 robot were electronics hardware design, software development, and *in vivo* testing. The details on these contributions follow.

The previously mentioned AB3 PCB was designed and tested by the author. A schematic of this board can be seen in Figure A.14 in Appendix A. The interface box for the AB3 was also designed by the author. An internal wiring diagram of the interface box can be seen in Figure A.15. The box contains four separate MRC-100 PCBs. Two of these boards were modified by removing the imager chips and routing the serial camera control signals to the AB3 PCB. This allowed the existing camera control software to be used with the AB3 robot. The other two MRC-100 PCBs were used for motor control by routing the motor driver outputs to the AB3 PCB. All four MRC-100 boards were connected to a built in USB hub in the interface box. A USB-6008 I/O board read signals from the Virtual Incision controller and also sent commands to the gripper motor driver, which was a single axis motor driver PCB. The AB3 device connected to the interface box via a 25-pin connector.

The AB software developed by the author will be discussed in Sections 2.21 and 6.2.4. During *in vivo* testing, the author set up all support hardware for the device, and monitored the device software interface while the surgeon controlled the robot.

2.21 AB Robot Software

The AB robot control software was developed by the author in C++ using Microsoft Visual Studio and is a dialog based graphical user interface. The main window is shown in Figure 2.63. This window monitors for MRC-100 boards connected to the computer and displays these to the user. The Control menu allows the user to access the other modules of the program: connect with AB device, Open Loop Control, Joystick Control, Joint Control, and Closed Loop Control. The software loads device settings from an external text file *settings.ini*. These settings include device comm port mapping and joystick axis to device joint mapping.

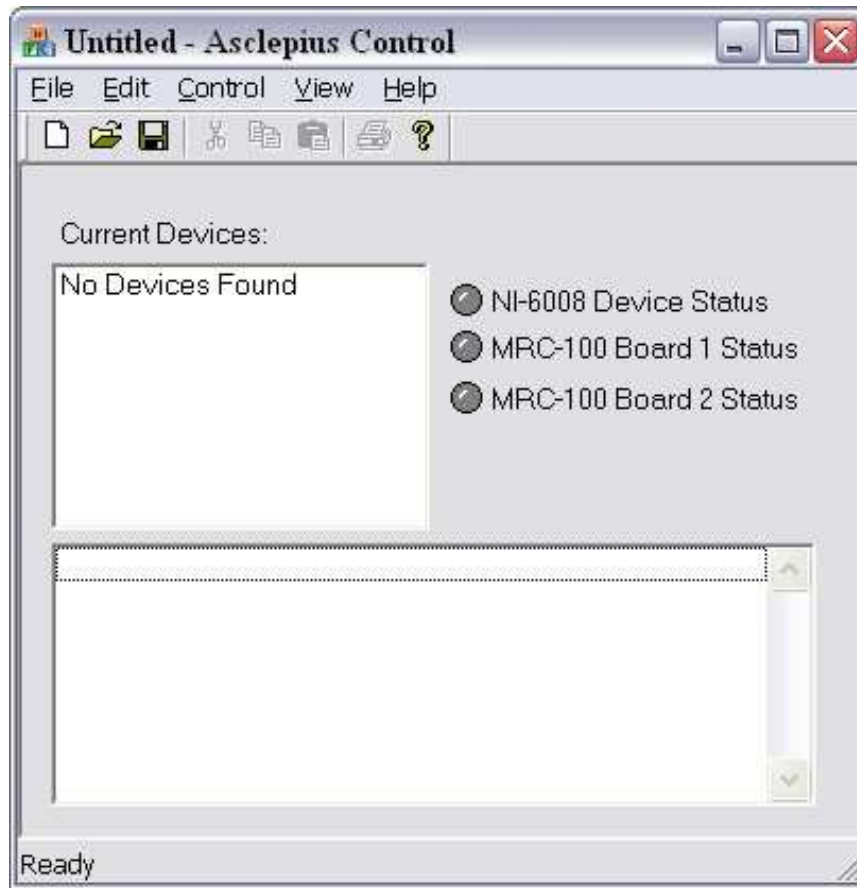


Figure 2.63: AB Robot Control Main Window

The joint control dialog, shown in Figure 2.64, allows the user to move each joint individually for testing purposes. The joint mapping data is loaded from the settings file.

The joystick control dialog, shown in Figure 2.65, allows the user to control the AB robot using the virtual incision controller. The dialog displays the current position of each joint on the virtual incision controller. The controller to robot joint mapping is loaded from the settings file.

The open loop dialog, shown in Figure 2.66, allows the user to run open loop tests on individual robot joints. It is integrated with frame grabbers and the tracking software developed for the surgical task completion experiments, which are discussed

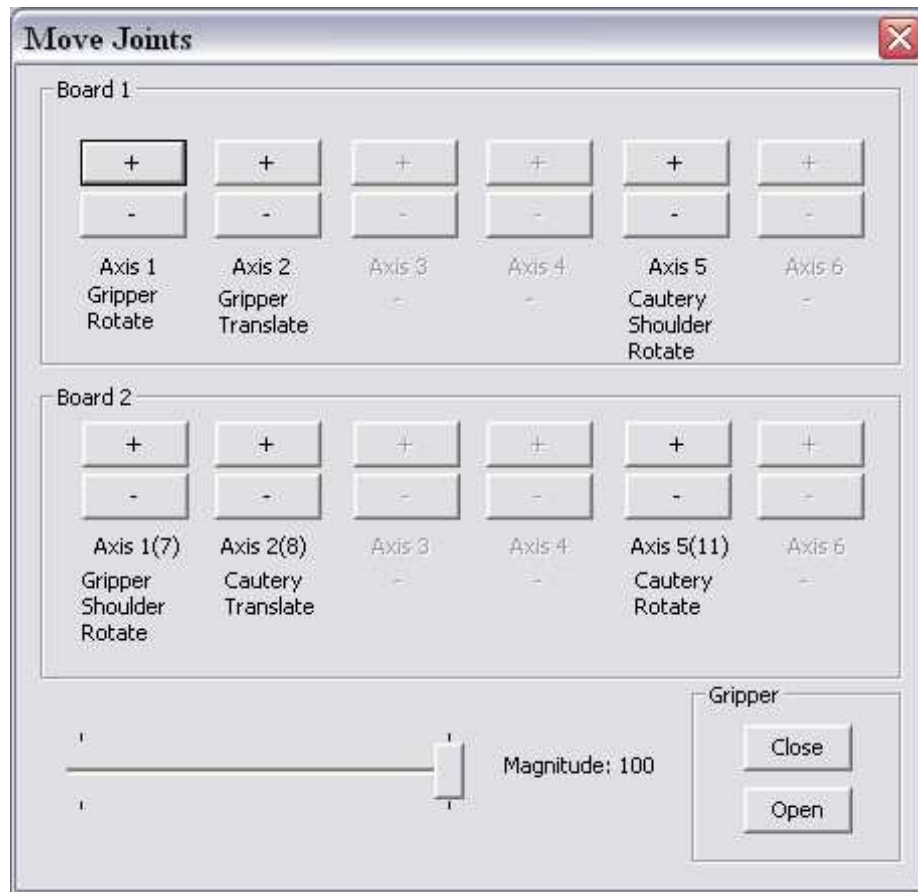


Figure 2.64: AB Robot Joint Control Dialog

in Chapter 6. The user creates a text file with a series of joint commands and time steps, for example:

```
4
.1
0 0 0 0 0 0 0 0
0 0 0 0 0 0 25 0
0 0 0 0 0 0 25 0
0 0 0 0 0 0 0 0
```

In this example, the user specifies 4 data points, spaced at 0.1 second intervals. Each column represents a specific joint on the robot. The values in each row are commands

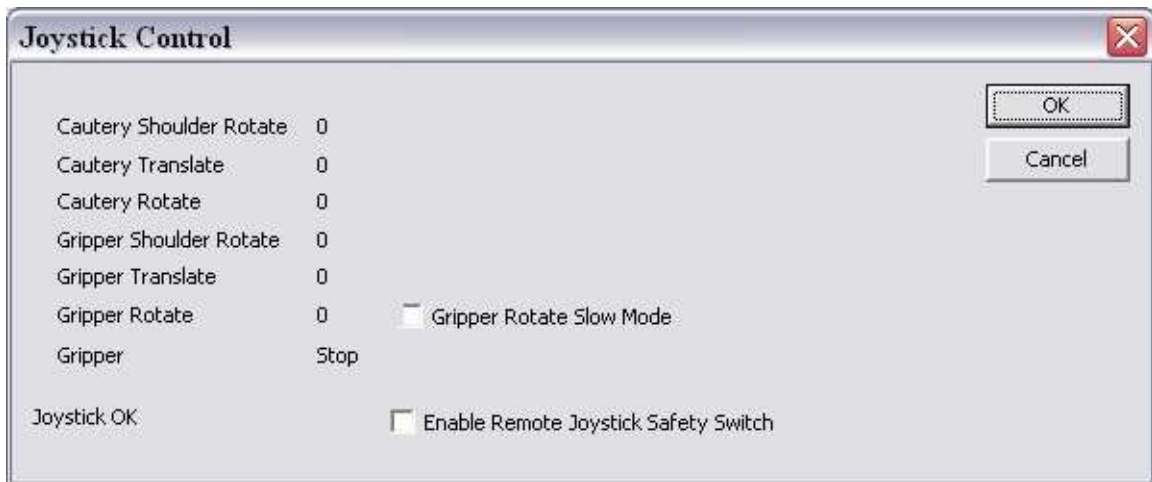


Figure 2.65: AB Robot Joystick Control Dialog

sent to each motor, ranging from -100% to $+100\%$. This file would command the cautery shoulder joint to move at 25% speed at 0.1 sec, and stop the joint at 0.3 sec. This allows the user to set up any motion profile needed. The dialog provides the option to: save frame grabs at each time step, save frame grabs and tracking data, or perform only the movement.

The last module of the AB robot user interface software is the closed loop control dialog. This was developed by the author to perform the task completion experiments, and is discussed in detail in Section 6.2.4.

A class diagram of the AB robot driver software, also developed by the author, is shown in Figure 2.67. The class for interfacing with the AB3 robot, *AB3_Device*, is derived from the *DualBoardDevice* class. *DualBoardDevice* contains two instances of the *NSSDevice* class, corresponding to the two MRC-100 boards used to drive the motors on the AB3 robot. The *NSSDevice* class was originally developed for the CPT-100 and MRC-100 robots. The *NSSDevice* class contains an instance of the *NSS_serialUSB_comms* which handles communications with individual CPT and MRC boards. The *AB3_Device* also contains an instance of the *overheadTracker*

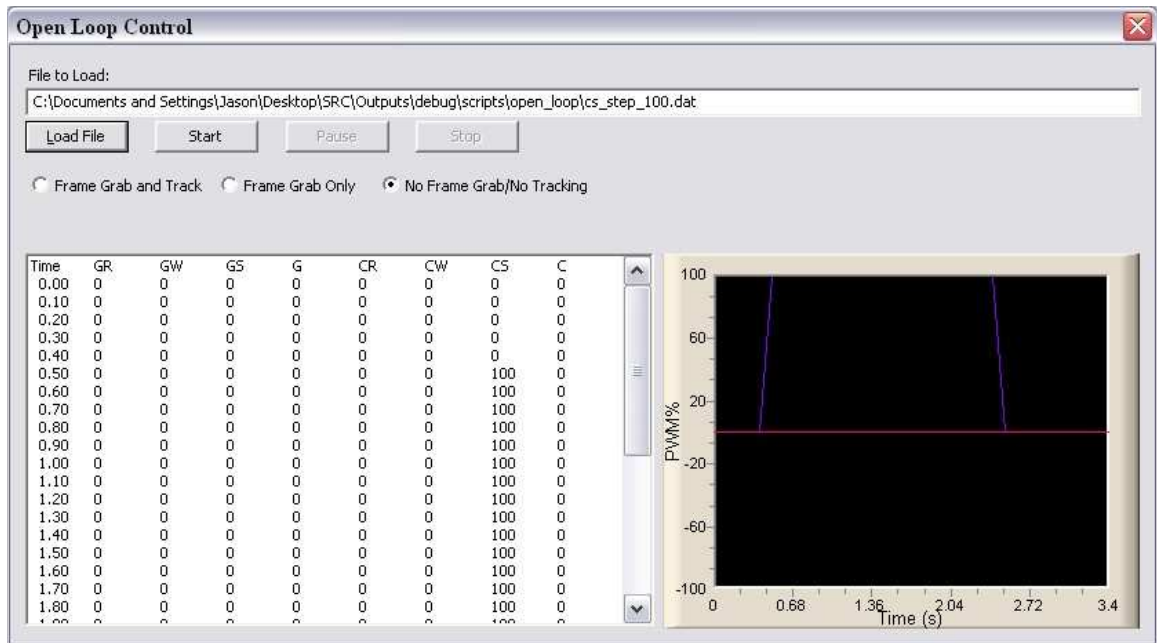


Figure 2.66: AB Robot Open Loop Dialog

class, which is used in the task completion experiments and will be discussed in Section 6.2.4.

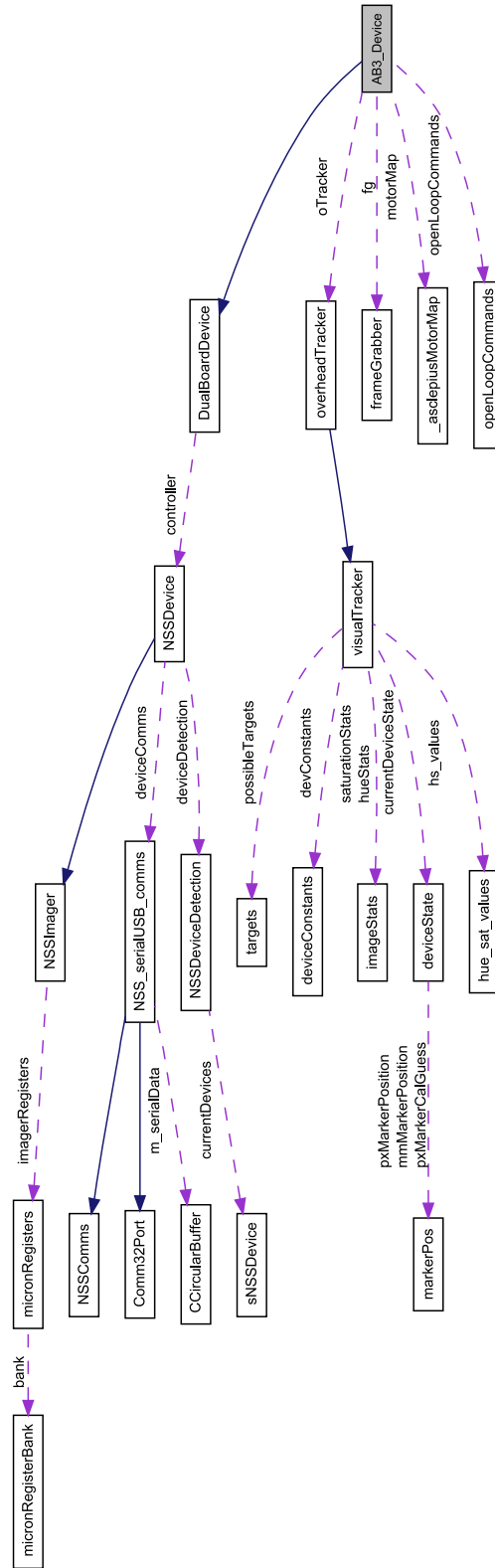


Figure 2.67: AB Software Collaboration Diagram

2.22 Summary

The robots described in this chapter were designed and tested as a part of this dissertation. These robots included mobile robots, adjustable focus cameras, fixed base cameras, a mobile robot biopsy platform, and dexterous manipulator devices. The author was the sole designer of the electronics and software aspects of all the devices discussed here. Tables 2.2 and 2.3 summarize the *in vivo* robots and list dates of *in vivo* tests for each device.

| Robot | Type | Test Date(s) | Notes |
|--------------|-------------------------|--------------------------|---|
| M3-20a | Manipulator | 06-09-2003 | 3 degree of freedom manipulator arm |
| MR-15b | Crawler | 06-09-2003 | 15 mm crawler, no imager |
| MR-15c | Crawler | 04-09-2004 10-27-2004 | 15 mm crawler, no imager |
| PT-15a | Fixed Base | 02-21-2004 | Pan-tilt camera, wireless |
| PT-15b | Fixed Base | 04-09-2004 | Pan-tilt camera, wired |
| WC-15a | Wireless Camera | 04-09-2004 | Passive wireless camera |
| AF-20a | Adjustable Focus Camera | 12-01-2004 | Fixed base, adjustable focus camera |
| MR-20a | Crawler | 12-17-2004 | 20 mm crawler, no imager |
| MR-15d | Crawler | 01-21-2005 07-22-2005 | 15 mm crawler, no imager, 1st successful mobility platform |
| MR-20b | Crawler | 01-21-2005 | 20 mm crawler with camera |
| MR-20c | Crawler | 09-12-2005 | 20mm crawler, used in NEEMO experiments and canine surgeries |
| T-15a | Fixed Base | 03-18-2005 | Tilt camera with adjustable focus |
| T-15b | Fixed Base | 09-12-2005 | Tilt camera with adjustable focus, used in NEEMO experiments and canine surgeries |

Table 2.2: *In vivo* Robots

| Robot | Type | Test Date(s) | Notes |
|--------------|-----------------------|---|--|
| MR-12a | Crawler | 12-14-2005 | 12 mm crawler, no camera, used to demonstrate NOTES |
| BMR-22a | Crawler | 12-14-2005 | 22 mm crawler with camera and biopsy tool |
| WMR-22a | Crawler | 12-14-2005 | 22 mm crawler, no camera, wireless |
| MRC-100 | Crawler | 08-11-2006(a) 04-10-2007(b) | 15 mm crawler with camera, used in survivable surgeries |
| CPT-100 | Ceiling Camera | 08-11-2006(a) 04-10-2007(b) 05-29-2007(b) 07-17-2007(b) 08-21-2007(b) | 15 mm tilt camera, magnetic attachment, used in survivable surgeries |
| AB1 | Dexterous Manipulator | 05-15-2007 | Two arm dexterous device with stereo camera |
| AB2 | Dexterous Manipulator | 12-18-2007 | Two arm dexterous device with stereo camera |
| AB3 | Dexterous Manipulator | 03-25-2008 05-30-2008 | Two arm dexterous device with stereo camera, used in task completion experiments |

Table 2.3: *In vivo* Robots, cont.

Chapter 3

Camera Comparisons

3.1 Background and Motivation

The majority of *in vivo* robots described in Chapter 2 have featured an imaging system to provide visual feedback to the surgeon; either as the sole feedback device or to augment the laparoscope. In order for these devices to be usable in this function, the visual quality must be comparable to or better than imaging systems currently in use. To compare visual quality in a quantitative way, tests were performed to measure visual quality. The metrics used were modulation transfer function (MTF), color accuracy, and distortion. MTF measures the overall resolution of a imaging system, and is expressed in a certain number of line pairs per millimeter. Color accuracy measures the system's ability to reproduce colors accurately. Distortion is an effect that causes straight lines to appear curved.

3.2 Methods and Materials

The imager device used in these experiments is the MT9V125 color digital CMOS image sensor from Micron [66]. The sensor has a 640x480 active pixel array with

NTSC analog composite video output. The MT9V125 is a 1/4 inch optical format imager housed in a 52-Ball IBGA package, measuring 9x9 mm. It features color correction, auto white balance, gamma correction, and black level conditioning, all user-controllable over a two-wire serial interface. The imager is mounted to a custom designed PCB. Two 10000 mcd white LEDs were used for illumination. The lens is a Sunex DSL758 [69], an all plastic lens module for 1/4 inch optical format imagers. It has a focal length of 3.8 mm and a diagonal field of view of 60°. This imager/lens combination was used in the following previously discussed *in vivo* devices: MRC-100 (pg 54), CPT-100 (pg 58), AB1 (pg 63), AB2 (pg 67), and AB3 (pg 70).

The laparoscope used in these experiments is a Karl Storz 0° device connected to a Storz Tricam SL NTSC camera unit. The Tricam SL contains 3-CCD chips for processing three primary colors [70]. The light source is a Storz Xenon 175, which features a 175 watt xenon lamp with a color temperature of 6000 K.

The vision target used for MTF testing is a ISO 12233 Resolution chart printed on Kodak photo paper, measuring 7.75" x 4.75". This chart is a standard reference for measuring resolution of electronic still imaging systems. The chart used for color error measurements is a Mini ColorChecker from GretagMagbeth, LLC. The ColorChecker is a standard Macbeth color chart, measuring 3.25" x 2.25". The resolution chart and Macbeth chart can be seen in Figure 3.1. For distortion tests, a square grid was generated from the Imatest (see below) application, and printed using a laser printer.

The experimental setup can be seen in Figure 3.2. It consists of an adjustable metal arm to hold the laparoscope in a horizontal position, a device to hold the robotic camera in position, and the vision test target.

The experiments were conducted in a darkened room with no outside light. The light sources used were the xenon light source for the laparoscope and white LEDs from the robotic camera. The control unit and light source for the laparoscope can

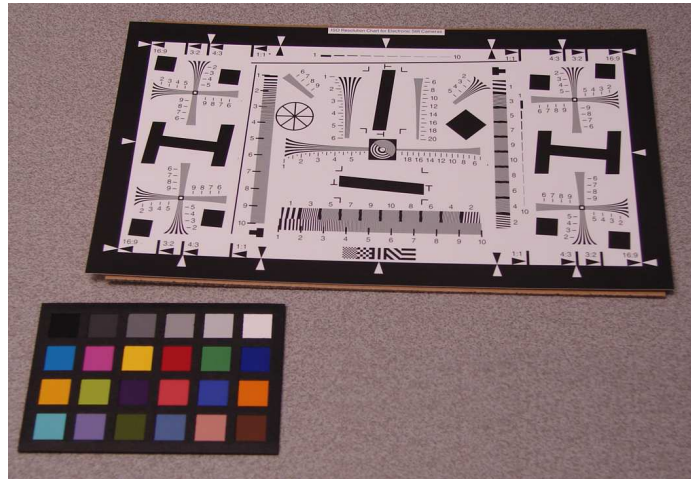


Figure 3.1: Vision Targets

be seen in Figure 3.3. The video output from both systems is NTSC composite. A Sensoray Model 2250 [71] USB 2.0 frame grabber, connected to a laptop PC, was used to capture frames of video for later analysis.

After all test images were captured, analyses was done using the Imatest [72] application. This program includes modules for MTF, color error, and distortion analysis.

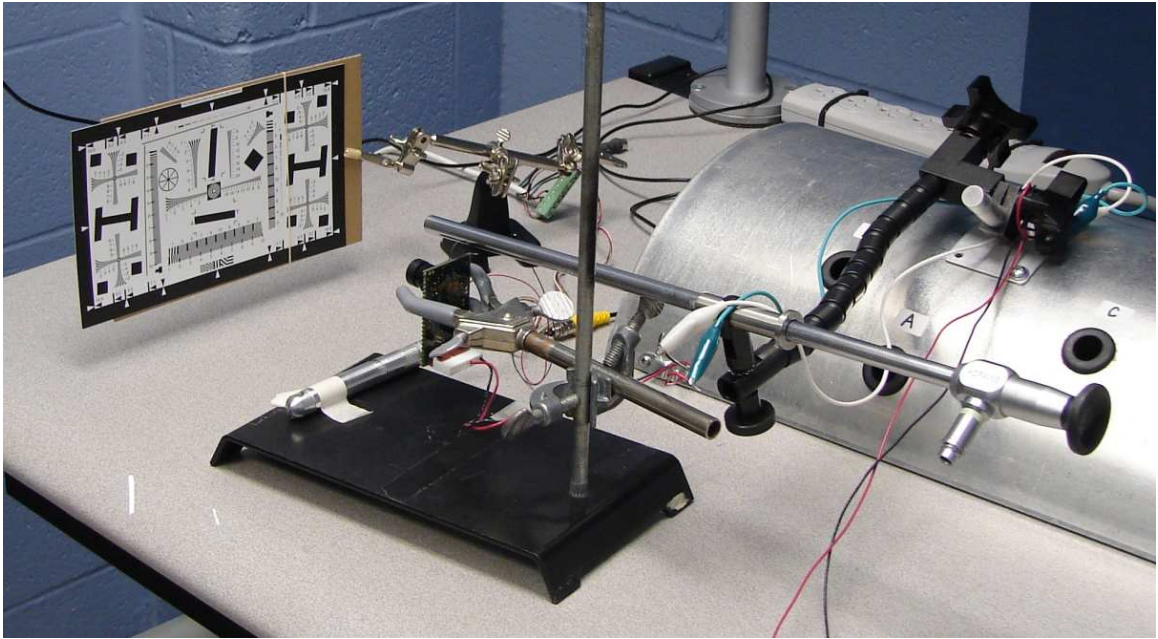


Figure 3.2: Experimental Setup



Figure 3.3: Laparoscope Control and Video Capture

3.3 Experimental Results

3.3.1 MTF Testing

A commonly used performance metric of optical systems is the modulation transfer function, or MTF. Several series of black and white bars of equal width are imaged by the system under test. Each set of bars has smaller widths than the previous. The finest set in which the line structure can be seen shows the resolution limit of the system, expressed in line pairs per millimeter.

The Modulation Transfer Function is defined as [73]:

$$MTF(\nu) = \frac{M_i}{M_o}, \quad (3.1)$$

where M_i and M_o are the modulation of the image and the modulation of the object, respectively. The modulation is defined as:

$$M = \frac{Y_{max} - Y_{min}}{Y_{max} + Y_{min}}, \quad (3.2)$$

where Y is the luminance maximum and minimum values. Figure 3.4 shows an example set of bars, and how MTF is calculated. The square wave in the image represents the brightness of the bars at a single spatial frequency, and the sine wave represents the output of the imaging system. The imaging system blurs the square wave, which results in a sine wave output. Y_{max} and Y_{min} are the maximum and minimum luminance values of the imager output. A plot of the MTF over all spatial frequencies defines the MTF of the system.

The ISO 12233 resolution chart was used to acquire test images for measuring MTF for both systems. Several frame grabs from each imaging system were saved to disk and analyzed with the Imatest application. Imatest uses Fourier analysis

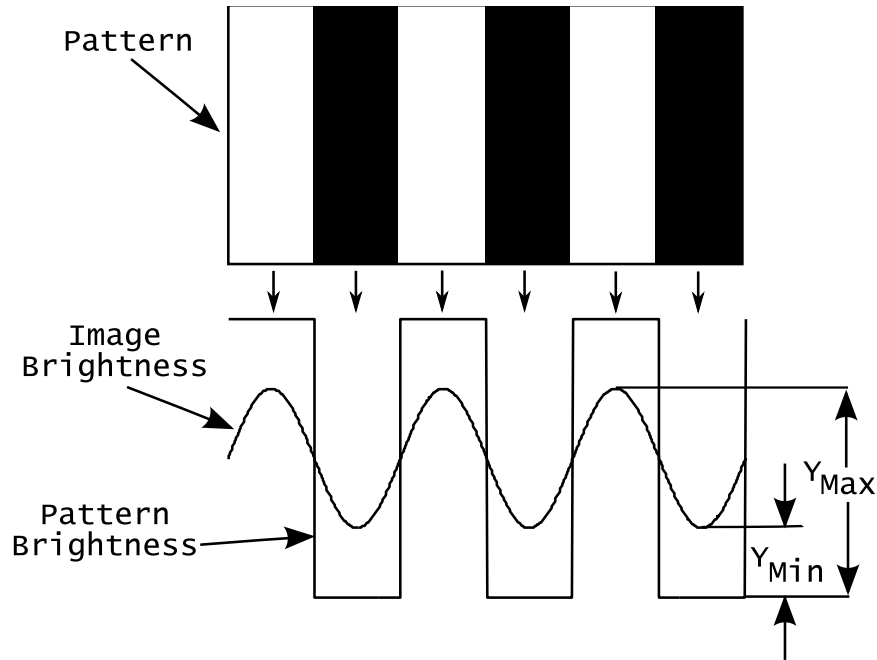


Figure 3.4: MTF Target

to compute MTF from a knife edge section of the resolution chart. These MTF curves are plotted in Figure 3.5. Each curve is a different estimate of the MTF from different regions in the captured images. Figure 3.6 shows the average curves with both systems plotted on the same axis. The curves show MTF plotted against spatial frequency in cycles per pixel. Values above the Nyquist frequency of $0.5cy/px$ have no meaning and are not displayed. Both systems have high MTF values below $0.05cy/px$, but the Micron camera drops off much faster than the laparoscope above this frequency. Qualitatively, this indicates that at spatial frequencies above approximately $0.05cy/px$, the laparoscope reproduces the scene with a better contrast ratio or sharpness.

Again, the laparoscope has significantly better results than the *in vivo* system. This is most likely caused by the lower quality optics in the *in vivo* system, because the MTF of the system is defined to be the product of the MTFs for each component of the system (lens, imager, etc) [73]. In the design of these devices, optics quality

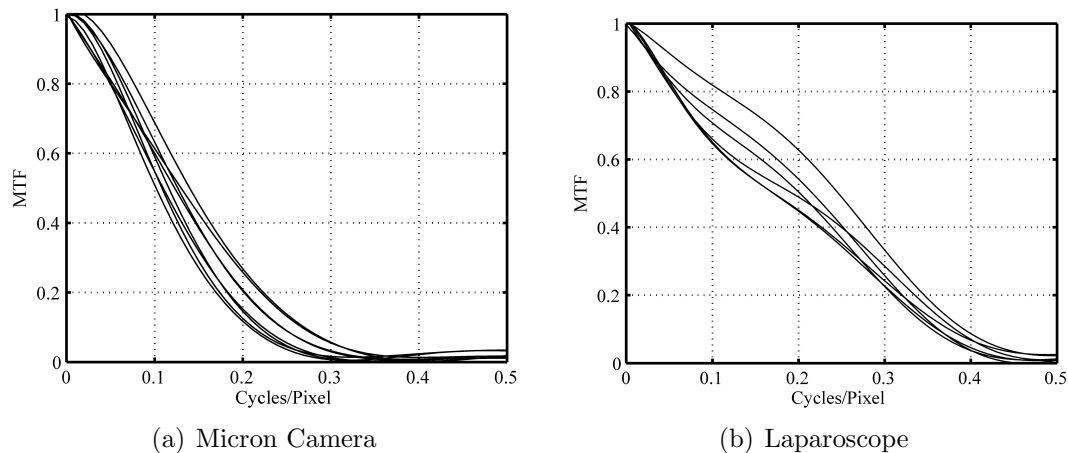


Figure 3.5: MTF Curves for Both Systems

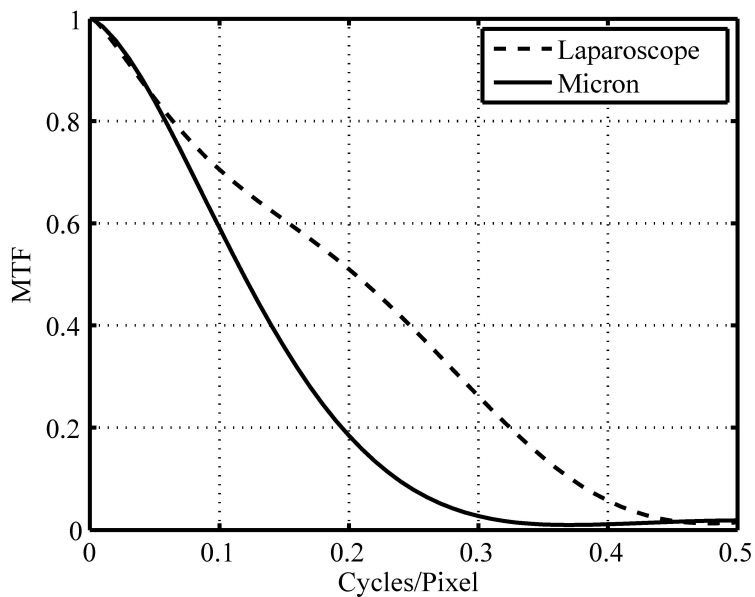


Figure 3.6: MTF Comparison

must be sacrificed for space, given the small physical size of the *in vivo* system. The laparoscope system is able to have higher quality optics, because the optics are all external.

3.3.2 Color Accuracy

Color accuracy of the two systems was measured using a Macbeth ColorChecker. The ColorChecker was placed in uniform illumination, and several frames were captured for analysis. The Imatest program converts the test images to CIELAB color space, and compares the test image color value to the actual color value for each color patch in the ColorChecker chart. Plots of these color differences can be seen in Figure 3.7 (*in vivo* camera) and Figure 3.8 (Laparoscope). These plots show the ideal color value and the actual color value, plotted in CIELAB color space. Mean and RMS color errors are also given. These are summarized in Table 3.1.

| System | μ_{error} | RMS_{error} |
|-----------------------|---------------|---------------|
| <i>In vivo</i> Camera | 8.46 | 10.90 |
| Laparoscope | 14.70 | 16.60 |

Table 3.1: Color Error

Figure 3.9 shows color error for each system, plotted against color zone number. The data in Table 3.1 and Figure 3.9 shows the *in vivo* camera had significantly less color error than the laparoscope.

3.3.3 Distortion

Optical lens systems suffer from distortion artifacts. This distortion is a non-linear effect that causes straight lines in an image to appear curved. For modeling this non-linear effect, an infinite series is required. However, usually only radial distortion needs to be considered, which can be modeled with one parameter (κ_1) [74]. This is modeled as:

$$r_u = r_d(1 + \kappa_1 r_d^3). \quad (3.3)$$

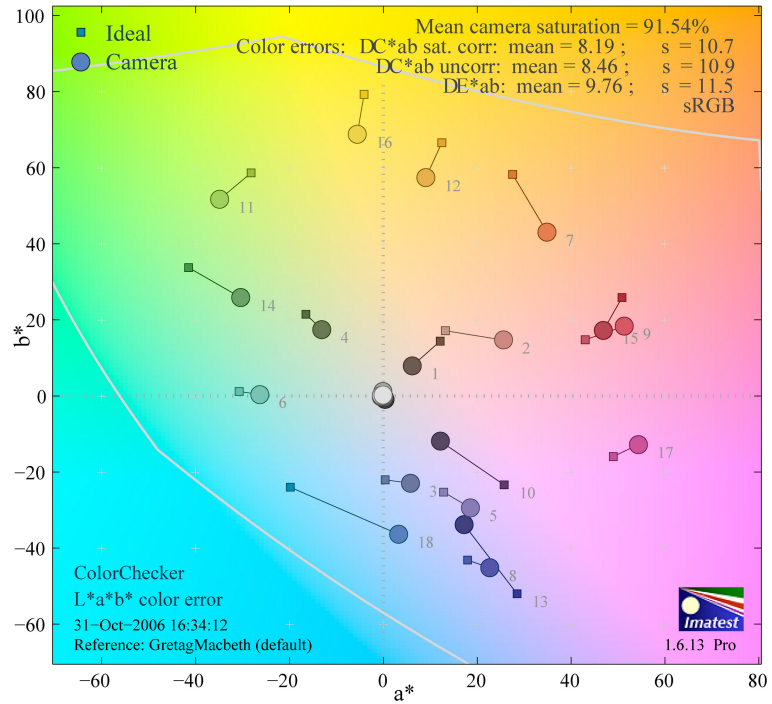


Figure 3.7: Micron Camera Color Error

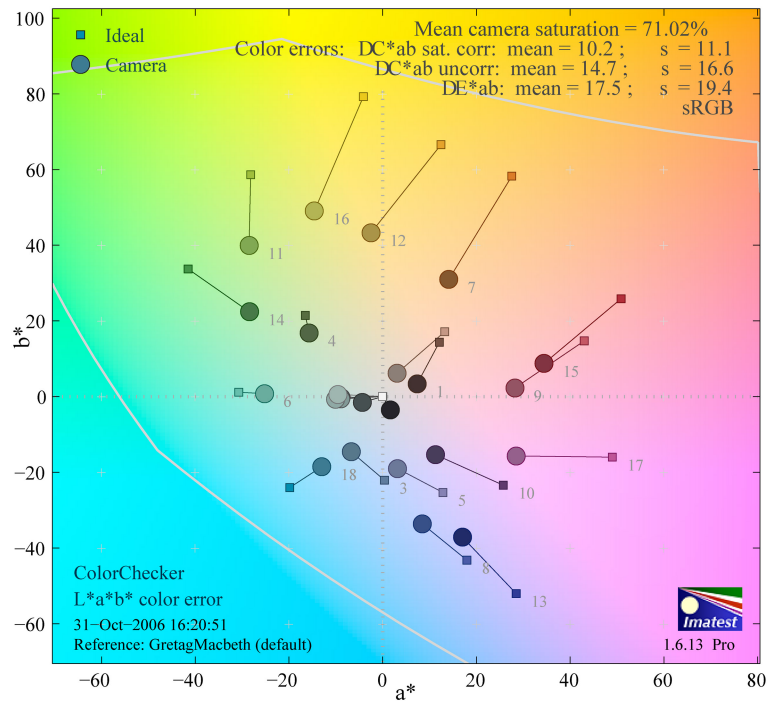


Figure 3.8: Laparoscope Color Error

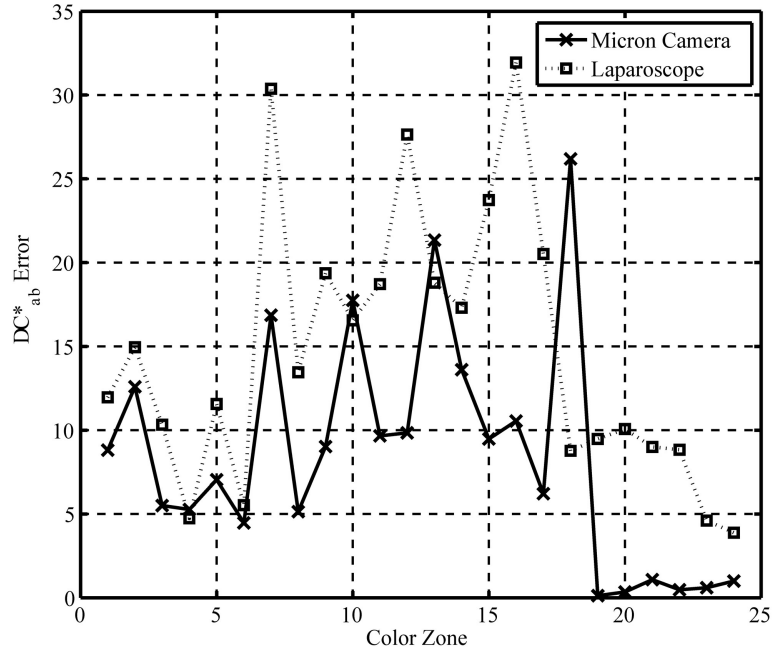
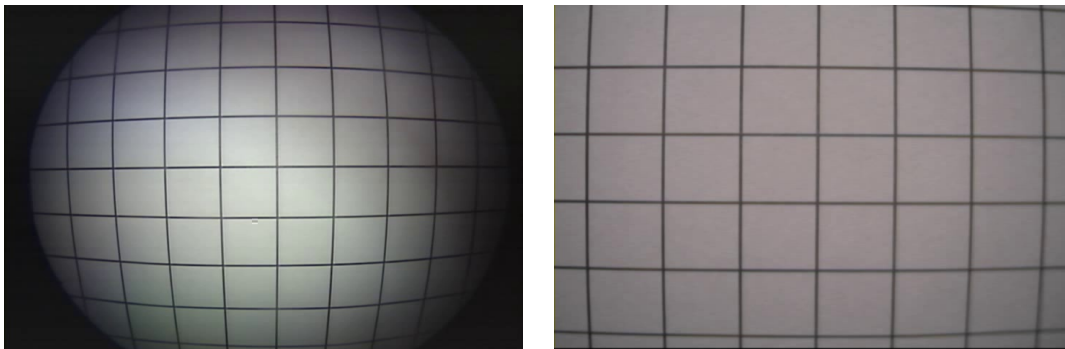


Figure 3.9: Color Error for Both Systems

Equation 3.3 relates the undistorted radius r_u and the distorted radius r_d . This one term model of distortion is also referred to as barrel ($\kappa_1 > 0$) or pincushion ($\kappa_1 < 0$) distortion.



(a) Laparoscope Distortion

(b) *In vivo* Camera Distortion

Figure 3.10: Lens Distortion

An example of lens distortion can be seen in Figures 3.10(a) (Laparoscope) and 3.10(b) (*In vivo* camera). The test target for the preceding Figures is a square grid pattern. As is evident from the images, the laparoscope has significant radial distortion.

tion. The *in vivo* camera has very little distortion. The results from Imatest confirm this quantitatively, and can be seen in Table 3.2.

| System | κ_1 |
|-----------------------|------------|
| <i>In vivo</i> Camera | 0.06 |
| Laparoscope | 0.35 |

Table 3.2: Radial Distortion

3.4 Summary

The experiments discussed in this chapter involved comparing two video systems used in laparoscopic surgery, a standard laparoscope and an *in vivo* device developed in part by the author. Benchtop tests were done with both systems, which tested resolution, color accuracy, and distortion. Analysis was completed on sample images captured from both systems. The laparoscope scored better on MTF testing, but the *in vivo* system scored better on color accuracy and distortion. It was hypothesized that the lower MTF value for the *in vivo* system was caused by lower quality (smaller) optics used for this system. These results show that *in vivo* robotic camera devices can produce video that is comparable to standard laparoscopes and can therefore show great potential for use in MIS procedures. Because this system will be used in MIS procedures, other metrics can be used to measure the efficacy of these devices. The analysis shown here could be expanded upon, by including metrics such as noise in the analysis. Other metrics include data from observing surgeons performing surgical tasks while using different systems for visual feedback. Preliminary studies to this effect have already been completed and will be discussed in Chapter 4. Other metrics could include image or video comparisons, using actual *in vivo* images captured during animal surgeries. These images could then be rated by surgeons as to their quality

and usefulness, although this metric is more subjective than the others presented.

Chapter 4

NEEMO Experiments

4.1 NEEMO Mission Background

NASA Extreme Environment Mission Operations, or NEEMO, is a NASA project for simulating long term space exploration. The NEEMO missions are conducted in Aquarius, an underwater habitat located 3.5 miles offshore of Key Largo, Florida. Starting in 2001, there have been 13 NEEMO missions, each with 4 Aquanauts and 2 National Undersea Research Center support technicians. The missions last up to 3 weeks, and consist of underwater EVAs and various experiments. These missions are meant to simulate conditions of life during long term space flight, such as living aboard the International Space Station.

Aquarius is the only undersea laboratory dedicated to marine science in the world [75]. It is owned by the National Oceanic and Atmospheric Administration (NOAA) and managed by the University of North Carolina at Wilmington (UNCW). Operating 60 feet below the surface, it is deployed next to a coral reef. It provides life support systems, living and sleeping quarters, and sophisticated research facilities for up to 6 inhabitants. Aquarius is comprised of several components. The habitat

module is a 82 ton pressure vessel which is 14 meters long by 3 meters in diameter, composed of 3 compartments. A 20 m³ wet porch contains an open moon pool, dive equipment storage, and a shower. The 14 m³ entry lock contains space for computers and experiments, power equipment, life support equipment, and bathroom facilities. The 40 m³ main lock is the primary living space. It contains bunks, computer work stations, kitchen facilities, dining and areas, and life support controls. The Life Support Buoy is a 10 meter diameter buoy. It contains generators, air compressors, radio communication equipment and oxygen storage flasks. It supplies Aquarius with air, power, and communication capability through a three inch diameter umbilical. The third component of Aquarius is the baseplate. It is a 116 ton structure which provides a stable and level base for the habitat.

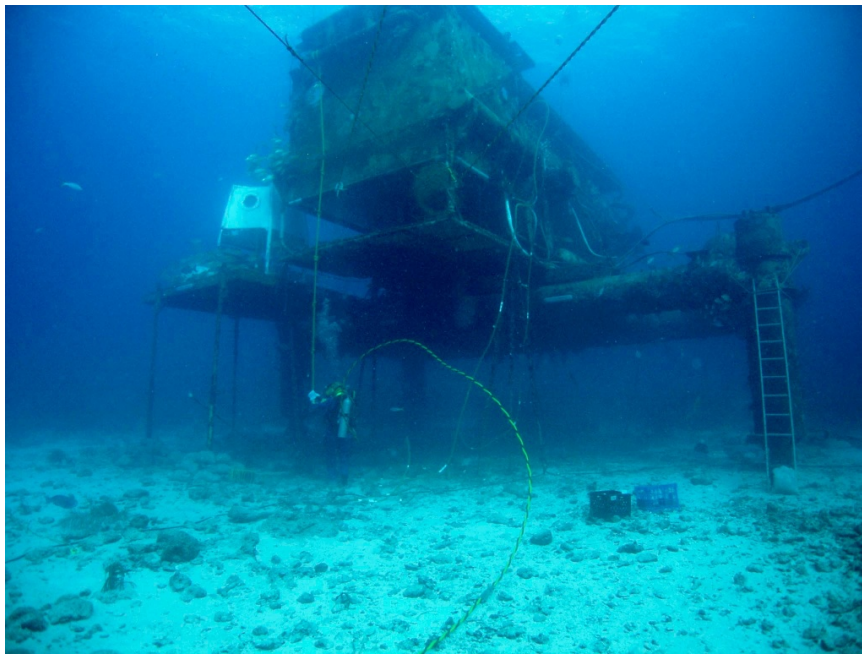


Figure 4.1: External View of Aquarius Habitat

The biggest advantage of Aquarius is longer diving times. Aquarius aquanauts can increase their diving time to almost ten times that of a surface diver through saturation diving. This enables them to conduct research much faster than if they

had to use conventional diving techniques. A typical 10 day saturation mission on Aquarius would take 60 to 70 days using surface based techniques. This has enabled scientists working on Aquarius to gain much more information on coral reefs as well as new technologies for future undersea habitats.

The aquanauts on NEEMO missions are NASA astronauts and other scientists. Aquarius provides an environment similar to living on the International Space Station (ISS) in many ways. First, the habitat is similar in size to the modules of the ISS. Also, the crew must coordinate their work with the remote mission control center on land in Key Largo. The crew must be isolated from the outside world for up to three weeks. This isolation is used to develop crew-mission control communications techniques and to study the impact on behavior and physiology.

NEEMO 9 was an 18 day mission conducted April 3 to 20, 2006. The mission was a joint project between the Canadian Center for Minimal Access Surgery (CMAS) at McMaster University, the U.S. Army Telemedicine and Advanced Technology Research Center (TATRC), the National Space Biomedical Research Institute (NSBRI), the University of Nebraska, and NASA. The goals of the mission were to evaluate new medical diagnostic and therapeutic technologies to improve medical care in remote and harsh environments, as well as to develop procedures and techniques for lunar exploration using remotely operated vehicles, tracking systems, and navigation devices. The four member crew for NEEMO 9 included three NASA astronauts and one laparoscopic surgeon.

4.2 Experiments

The experiments developed by the UNL researchers for the NEEMO mission were of two types: surgical task comparison and telementoring. The task comparison ex-

periments were designed to test the usefulness of the *in vivo* surgical camera robots compared to using a standard laparoscope. The telementoring experiment was designed to show that *in vivo* robots could be used in surgical situations in which an experienced surgeon in a remote location could telementor crew members through an unfamiliar surgical procedure. All experiments involved the use of one or both of the previously described miniature *in vivo* robots developed for the mission: the MR-20c (NEEMO variant, see Section 2.10) and the T-15b (NEEMO variant, see Section 2.11). The experiments were performed using the abdominal cavity simulator shown in Figure 4.2. The crew used standard laparoscopic instruments to manipulate simulated tissue while receiving visual feedback from one of the *in vivo* camera robots or a standard laparoscope. The video from the simulator was viewed on a LCD monitor.



Figure 4.2: Abdominal Cavity Simulator

Only one video source was viewed at a time, and the same monitor was used for all devices.

The first task was a rope passing task, which simulated the surgical task of in-

specting a long section of bowel. A 30 cm length of rope was marked with colored sections every 5 cm (See Figure 4.3). Using laparoscopic graspers and only touching the colored portions, the rope was passed back and forth five times. This was completed by each crew member once with the tilt robot used for visual feedback, and once using the laparoscope for visual feedback. The tilt camera used a LED light source for illumination, while the laparoscope used an integrated xenon light source.

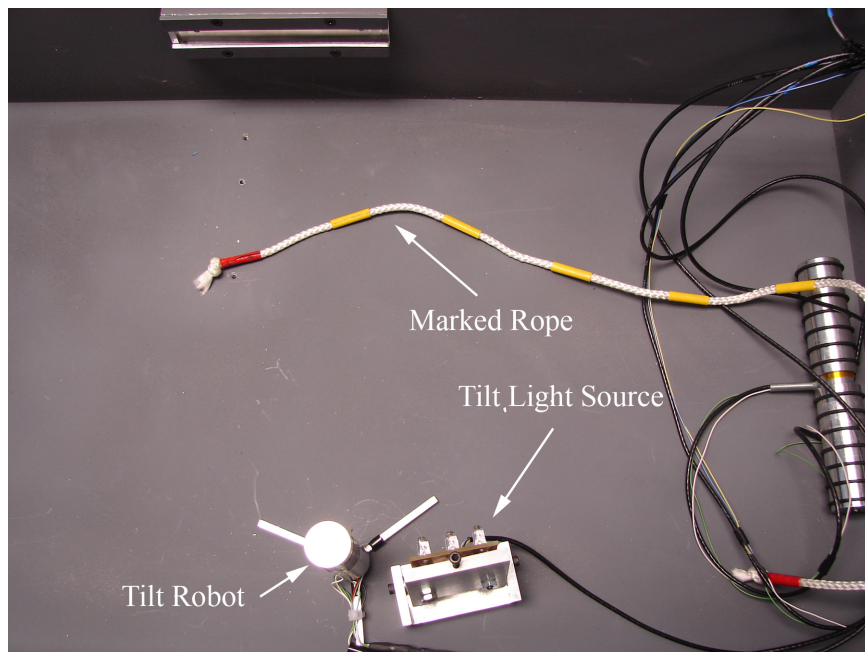


Figure 4.3: Rope Passing Setup

The second task was a stretch and dissect task. This task consisted of using a laparoscopic grasper to grasp and stretch a piece of simulated tissue, and using another tool to cut the tissue at a marked location. Two sets of five samples were used: one placed on an incline, requiring the crew member to pull the tissue to the right; and the other placed on the rear wall of the simulator, which required the crew member to pull the sample forward and cut it (See Figure 4.4).

The third task was a telementoring task using the mobile camera robot, along with laparoscopic tools to perform a simulated appendectomy. The task was setup with a

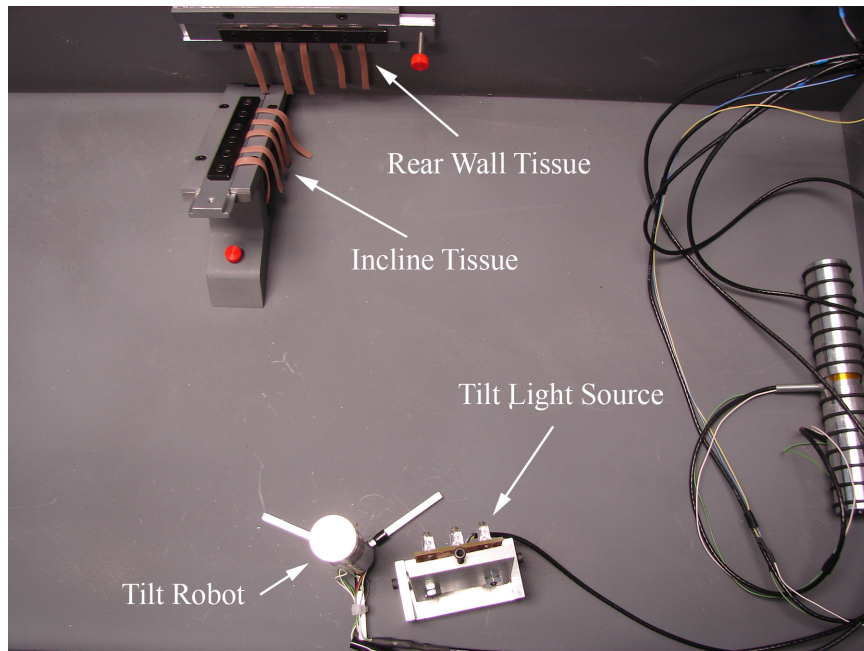


Figure 4.4: Stretch and Dissect Task Setup

small piece of tube, acting as the appendix, attached to the bottom of the surgical trainer box (Figure 4.5). The task was designed to evaluate if a skill set could be built upon and if the crew members could be telementored through an unfamiliar surgical situation. Prior to the mission, each crew member was trained on the use of the mobile camera robot and surgical tools, but not on the specifics of this task. Immediately before starting this task, the crew member watched a 30 sec video which described the appendectomy procedure. To start the task, the mobile camera robot was moved into a position to view the simulated appendix. The crew member was then telementored through the procedure over a video conferencing link from the University of Nebraska Medical Center in Omaha, Nebraska. The procedure included stapling, dissecting, and removing the simulated appendix.

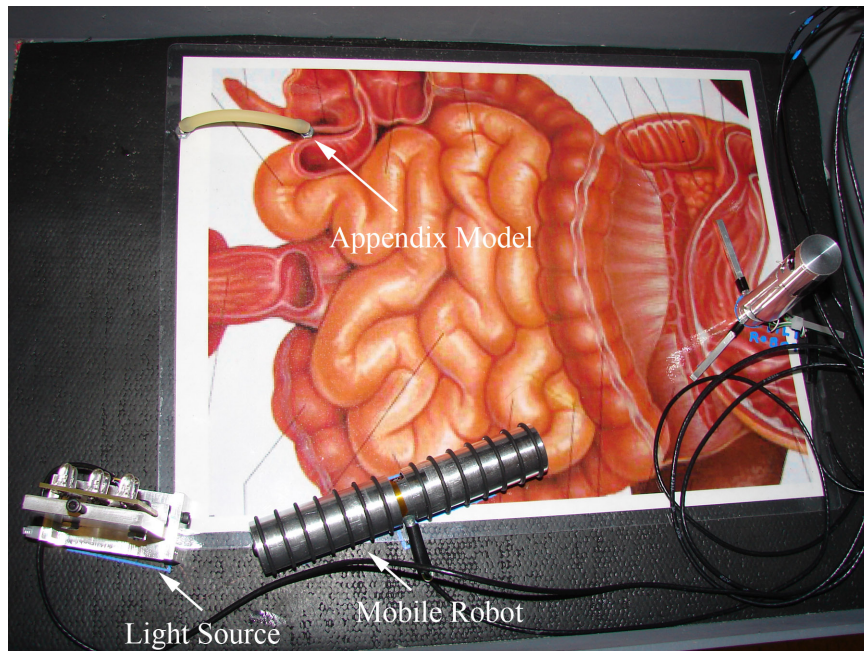


Figure 4.5: Appendectomy Task Setup

4.3 Results and Analysis

Each crew member completed the rope passing test twice, once using visual feedback from the tilt robot (Figure 4.6(a)) and once using the laparoscope (Figure 4.6(b)). Each task was timed and video was recorded for later analysis. The performance metrics chosen for comparison were: hits, misses, grasps, and time. A hit was defined as a grasp of a colored rope segment, a miss was defined as grasping a segment between the colored segments, and grasps were the sum of hits and misses. A perfect score of 100 hits could be achieved by passing the rope forwards and backwards five times while hitting each segment and never missing. Repeated measures ANOVA (Analysis of Variance) tests were performed on each metric, with the results summarized in Table 4.1. The hypothesis was that the results using the tilt camera would be better or just as good as the laparoscope results. As can be seen in Table 4.1, all four metrics showed no significant difference between the tilt camera and the laparoscope

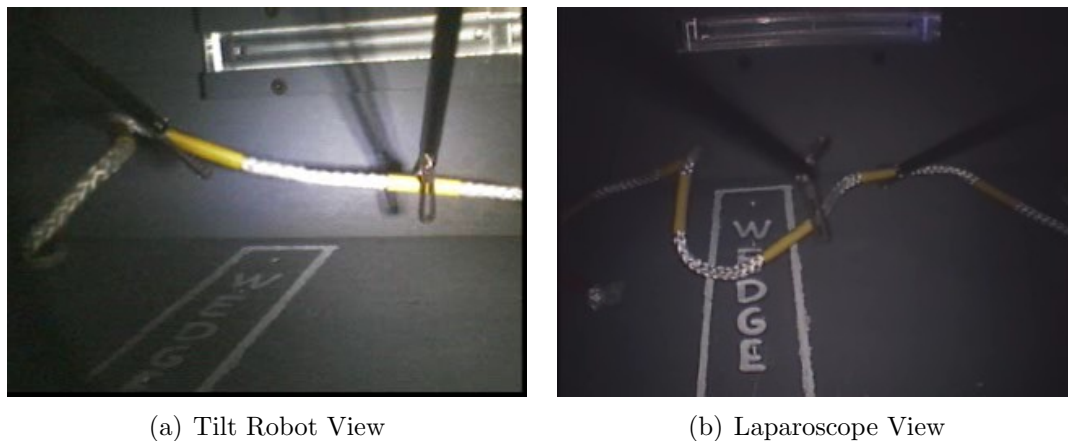


Figure 4.6: Rope Passing Task

($p > 0.05$ in all cases). If the time difference for each participant is graphed, as

| Metric | $F(1, 3)$ | p | μ_{robot} | σ_{robot} | $\mu_{laparoscope}$ | $\sigma_{laparoscope}$ |
|---------|-----------|-------|---------------|------------------|---------------------|------------------------|
| Hits | 0.570 | 0.486 | 97.250 | 6.450 | 100.250 | 5.740 |
| Misses | 4.849 | 0.115 | 11.000 | 4.080 | 17.750 | 3.860 |
| Grasps | 4.699 | 0.119 | 108.250 | 7.590 | 118.000 | 3.460 |
| Time(s) | 0.570 | 0.505 | 260.750 | 85.640 | 285.250 | 53.990 |

Table 4.1: Rope Passing Analysis

in Figure 4.7, it can be seen that participant 4's time difference is in the opposite direction (this participant was faster with the laparoscope) than the other three. If this case is not considered, the results of the ANOVA for the time metric are very different. In this case, there was a significant difference between the tilt device and laparoscope ($p = 0.024$), with the tests using the tilt device showing a lower mean completion time (see Table 4.2).

| Metric | F | p | μ_{robot} | σ_{robot} | μ_{scope} | σ_{scope} |
|---------|--------|-------|---------------|------------------|---------------|------------------|
| $N = 4$ | 0.570 | 0.505 | 260.750 | 85.640 | 285.250 | 53.990 |
| $N = 3$ | 39.613 | 0.024 | 225.670 | 60.140 | 282.000 | 65.640 |

Table 4.2: Rope Passing Time Analysis

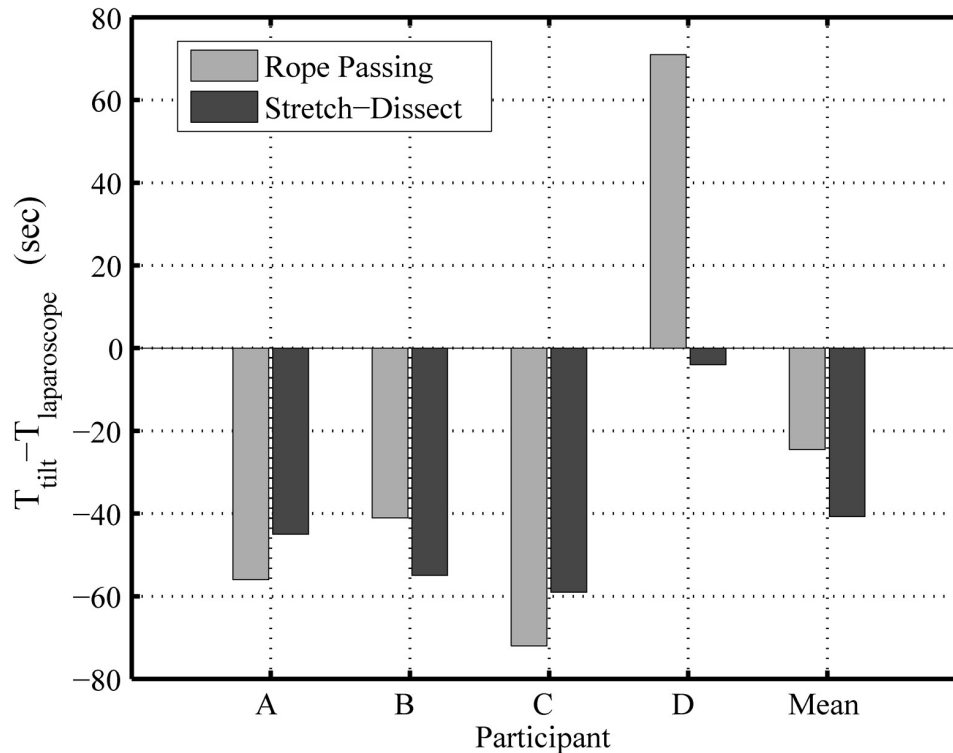


Figure 4.7: Time Difference Graphs

The stretch and dissect task was similar to the rope passing task. Each crew member completed the task twice, once with visual feedback from the tilt robot (Figure 4.8(a)) and once using the laparoscope (Figure 4.8(b)). A full task consisted of grabbing, stretching, and cutting five tissue samples on the side incline mount, followed by five samples on the rear mount. Each task was timed and the video recorded for later analysis. The cut tissue samples were also saved for later analysis. The performance metrics used here were total time and accuracy. The accuracy was measured by the distance from the tissue mark to the actual cut in millimeters and the angle of the cut (an angle of 0° corresponds to a cut perpendicular to the length of the tissue). After measuring the cut tissue samples and analyzing the video, the data were recorded for analysis. For each participant, all distance and angle measurements were averaged over each test. Repeated measures ANOVA analyses

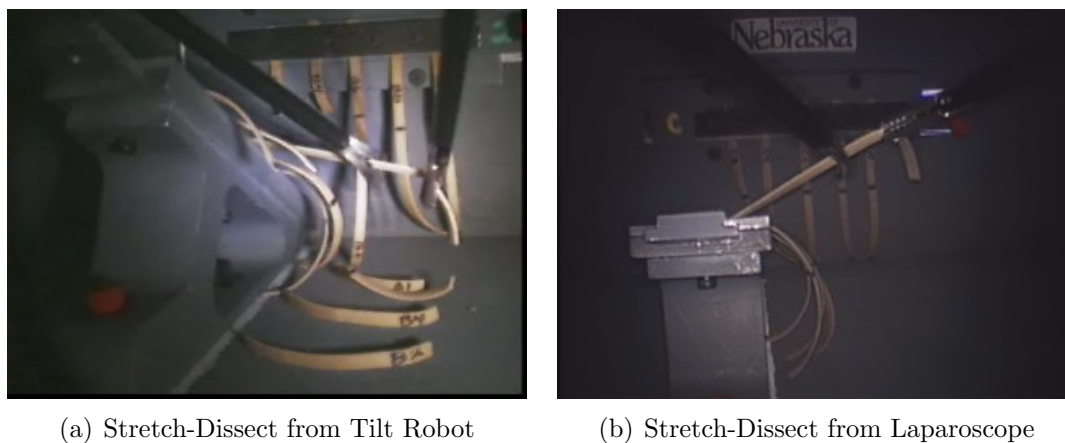


Figure 4.8: Stretch-Dissect Task

were performed on this data, with the results summarized in Table 4.3. Again, it was hypothesized that the results for the tilt camera would be equal to or better than those of the laparoscope. There was no significant difference in results for angular error ($F(1, 3) = 0.727$, $p = 0.456$) or in distance error ($F(1, 3) = 0.238$, $p = 0.659$). There was a significant difference found in total time ($F(1, 3) = 10.462$, $p = 0.048$), with the tilt camera total time ($\mu_{robot} = 106.0$) less than the laparoscope total time ($\mu_{scope} = 146.75$).

| Metric | $F(1, 3)$ | p | μ_{robot} | σ_{robot} | μ_{scope} | σ_{scope} |
|--------------------|-----------|-------|---------------|------------------|---------------|------------------|
| Angular Error(deg) | 0.727 | 0.456 | 11.25 | 2.87 | 14.00 | 4.83 |
| Distance Error(mm) | 0.238 | 0.659 | 1.08 | 0.40 | 0.95 | 0.02 |
| Total Time(s) | 10.462 | 0.048 | 106.00 | 31.62 | 146.75 | 24.53 |

Table 4.3: Stretch and Dissect Analysis

Each crew member successfully completed the appendectomy task, using the mobile robot for visual feedback. No quantitative measures were made during these tasks, but near and far views of the simulated appendix can be seen in Figures 4.9(a) and 4.9(b). The successful completion of this task by all crew members demonstrates that training participants to complete an unfamiliar task using miniature surgical robots.

After each crew member completed all three tasks, a series of survey questions were

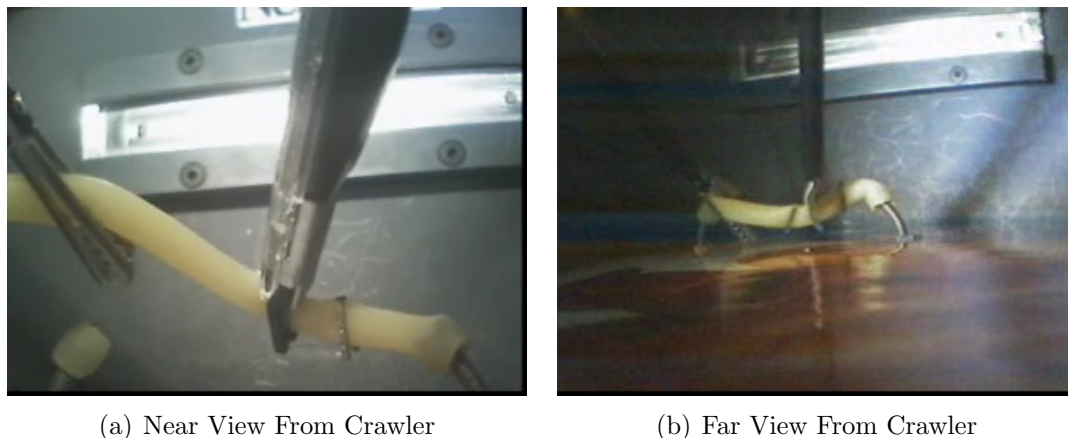


Figure 4.9: Appendectomy Task

completed. The questions asked the crew members to rate they were satisfied with the performance of the tilt and the laparoscope on depth perception, lighting, video feedback, camera control, and overall functionality. These questions were answered on a scale from 1 to 5, with 1 meaning strongly disagree and 5 meaning strongly agree. Repeated measures ANOVA analysis were performed on the survey data, with the results summarized in Table 4.4. There were no significant differences between the tilt

| Metric | df | F | p | μ_{robot} | σ_{robot} | μ_{scope} | σ_{scope} |
|-----------------------|------|-------|-------|---------------|------------------|---------------|------------------|
| Depth Perception | 2 | 0.00 | 1.000 | 4.0 | 1.0 | 4.0 | 0.0 |
| Lighting | 3 | 27.00 | 0.014 | 2.8 | 1.0 | 4.3 | 0.5 |
| Video Feedback | 3 | 2.46 | 0.215 | 3.5 | 1.0 | 4.3 | 0.5 |
| Camera Control | 3 | 0.06 | 0.824 | 3.3 | 1.5 | 3.5 | 1.0 |
| Overall Functionality | 3 | 0.00 | 1.000 | 3.5 | 1.0 | 3.5 | 0.6 |

Table 4.4: Survey Results

robot and the laparoscope in depth perception ($F(1, 2) = 0$, $p = 1$), video feedback ($F(1, 3) = 2.46$, $p = 0.215$), camera control ($F(1, 3) = 0.59$, $p = 0.824$), and overall functionality ($F(1, 3) = 0$, $p = 1$). For the lighting question, the laparoscope had a significantly better score ($F(1, 3) = 27.0$, $p = 0.014$, $\mu_{laparoscope} = 4.3$, $\mu_{robot} = 2.8$).

The results presented in Tables 4.1, 4.3, and 4.4 show, for the most part, no significant difference ($p > 0.05$) between the results using the *in vivo* robots and results using a laparoscope. Examination of the actual results seems to support this result, as the mean times, errors, etc are similar between the two conditions. Before drawing any conclusions, a *post hoc* power analysis must be completed to check the validity of the results. For all of the previously discussed analyses where no significant difference ($p > 0.05$) was found, the *post hoc* analysis will show the chance that a Type II error was made. A Type II error indicates that there is an effect in the population, but the effect was missed by the analysis. For each test, the sample size, S and the effect size r must be calculated using Equations (4.1) and (4.2).

$$S = df_{error} + 1 = N \quad (4.1)$$

$$r = \sqrt{\frac{F}{F + df_{error}}} \quad (4.2)$$

With r and S , the effect power can be found using the Power, Effect Size & Sample Size table [77]. Effect power is related to the probability of committing a Type II error by the following equation: $P_{error} = 1 - Power$. Based on this, all of the results in Table 4.5 have at least a 70% chance of being a Type II error, which would indicate an effect was missed, but is impossible to ascertain the direction of this effect (*i.e.* is the laparoscope or miniature robotic device better). There are several ways of increasing the statistical power of an experiment, the easiest being to increase the sample size. The S_{needed} column in Table 4.5 shows the minimum sample size needed to reduce the chance of a Type II error to 20% (Power = 80%). The data indicate that a sample size of at least 50 should be used to increase the statistical power of the results.

| Metric | p | F | S_{actual} | r | Power | S_{needed} |
|-------------|------|------|--------------|-----|-------|--------------|
| RP Hits | 0.49 | 0.57 | 4.0 | 0.4 | < 0.2 | 44 |
| RP Misses | 0.12 | 4.85 | 4.0 | 0.8 | 0.2 | 10 |
| RP Grasps | 0.12 | 4.70 | 4.0 | 0.8 | 0.2 | 10 |
| RP Time | 0.51 | 0.57 | 4.0 | 0.4 | < 0.2 | 44 |
| SD Ang Err | 0.46 | 0.73 | 4.0 | 0.4 | < 0.2 | 44 |
| SD Dis Err | 0.66 | 0.24 | 4.0 | 0.3 | < 0.2 | 82 |
| Sur Depth | 1.00 | 0.00 | 3.0 | 0.0 | 0.0 | N/A |
| Sur Vid Fdb | 0.22 | 2.46 | 4.0 | 0.7 | 0.3 | 11 |
| Sur Cam Ctl | 0.82 | 0.06 | 4.0 | 0.1 | < 0.2 | 343 |
| Sur Overall | 1.00 | 0.00 | 4.0 | 0.0 | 0.0 | N/A |

Table 4.5: *Post-hoc* Power Analysis

An overall interpretation of the experimental results presented here is difficult to give. A cursory analysis of the results would seem to indicate that there is little difference when using miniature *in vivo* robots as compared to using a laparoscope in these laparoscopic training tests. Several of the results (*e.g.* Total time in rope passing and stretch-dissect) even indicate that the *in vivo* robots could be superior. A more thorough analysis indicates these results have serious “power problems”, due to the very small sample size. This power problem represents a threat to the validity of these results. It must be noted that the small sample size was impossible to change for these experiments, because the NEEMO crew was limited to 4 members. Future studies of this kind could easily be designed to avoid some of the inherent problems encountered in this study. Suggestions for future studies will be discussed further in Chapter 7.

4.4 Summary

NASA Extreme Environment Mission Operations, or NEEMO, is a NASA project for simulating long term space exploration. The missions are conducted in an underwa-

ter habitat called Aquarius near Key Largo, Florida. NEEMO 9 was conducted from April 3 to 20, 2006. University of Nebraska researchers, including the author, developed some of the experiments performed by the NEEMO crew. The UNL experiments were surgical task comparison and telementoring using *in vivo* surgical robots. The task comparison experiments involved using an *in vivo* robot and a laparoscope for visual feedback while performing surgical training tasks. It was hypothesized that the *in vivo* robots would be as good, or better, than the laparoscope for visual feedback. The telementoring experiments involved telementoring the crew members through a simulated surgical procedure. The statistical results from these experiments were inconclusive as a result of power problems related to the small sample size. Further experiments need to be completed to obtain more reliable results.

The author had several roles in the UNL NEEMO experiments:

- Design and testing of printed circuit boards (PCBs) used in the *in vivo* robots for these experiments.
- Design and testing of PCB used in the control box used in the experiments.
- Software development and testing of the control box software, firmware on the robot PCBs, and Windows PC based software used to control the *in vivo* robots
- Traveled to Johnson Space Center in Houston, Texas for preliminary training of the NEEMO crew on the robots that were used in the NEEMO experiments.
- Traveled to Key Largo, Florida to deliver the experimental equipment and train the crew immediately prior to the NEEMO mission.
- Performed statistical analyses on the experimental data obtained from the NEEMO experiments.

Chapter 5

Survivable Pig Studies

In order for the *in-vivo* robots described in this dissertation to be used in procedures involving humans, they must meet the following requirements:

1. The devices must be usable without causing tissue damage or transmitting infection.
2. The devices must be useful in a laparoscopic procedure.

To this end, an experiment was designed using the ceiling pan-tilt (CPT) devices described in Section 2.16 in three survivable cholecystectomy animal surgeries. A porcine animal model was used in all three procedures, which were performed on different dates. In the first procedure, the CPT was used to augment the view of the surgical field from the laparoscope. In the final two procedures, the CPT was the sole means of visual feedback for the surgeon performing the procedure. A detailed description of the procedures and results follow.

5.1 Ceiling Pan-tilt (CPT) Camera

A small robotic camera, the CPT-100, was designed and built for the survivable pig experiments. The device can be seen in Figure 5.1. This device attaches to the ab-

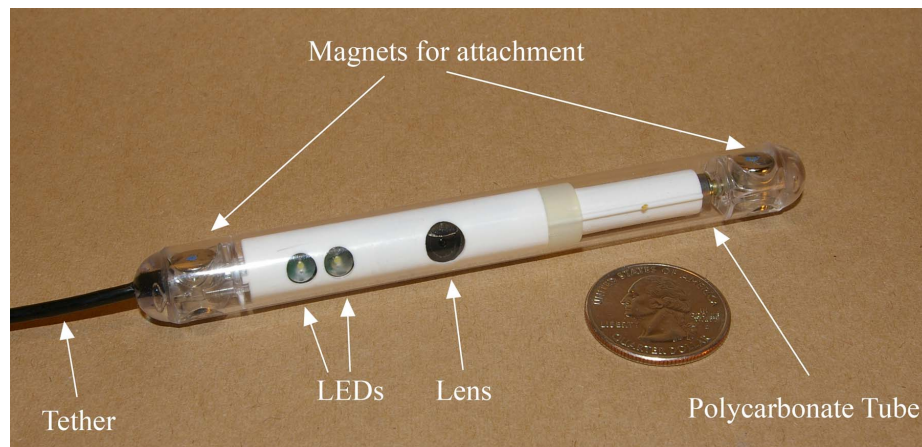


Figure 5.1: CPT-100 Robotic Camera

dominal wall by means of small embedded magnets and an external magnetic handle. The surgeon moves the handle, which changes the view of the surgical field. The CPT-100 also has a motor allowing the camera to tilt inside the clear polycarbonate body tube. These two methods of movement allow for a full view of the surgical field. The device also features two light emitting diodes (LEDs) for illumination. The device connects to an external breakout box which contains a power supply, USB connector for command data, and a connector for video output. The CPT-100 is controlled from a PC using software developed by the author. This software enables LED brightness control, tilt motor control, and camera settings adjustment.

5.2 Pre-Operative Procedure

In order to satisfy item 1 of the requirements, the device must be sterilizable. Two sterilization methods currently in use are ethylene oxide gas (EtO) and STERRAD,

a gas plasma process from Ethicon. Both processes were separately used to ensure compatibility and effectiveness with the CPT. Ethylene oxide gas was used to sterilize the CPT used in Animal B. The CPTs used in Animals A and C were sterilized with STERRAD. A brief description of each process follows.

EtO sterilization uses toxic ethylene oxide gas, a potent anti-microbial agent which kills viruses, bacteria, and fungi [78]. EtO can be used on instruments where the high temperature and pressure of steam sterilization can be damaging. STERRAD was developed in the early 1980s as a alternative to ethylene oxide gas and steam sterilization [79]. The technology uses a low-temperature hydrogen peroxide gas plasma. It is suited to heat and moisture sensitive instruments.

The animals used in this study were anesthetized pigs. The protocol for this study was approved by the University of Nebraska-Lincoln and University of Nebraska Medical Center Institutional Animal Care and Use Committee (IACUC). Each animal's weight and preoperative medications and anesthetics are shown in Table 5.1.

In addition to these medications, an IV line was established to administer any additional medications. During each procedure, each animal was maintained with isoflurane and supplemented with oxygen. The animals also underwent a surgical skin scrub and hair removal.

A partial layout of the operating room is shown in Figure 5.2. Only items relevant to the operation of the CPT devices are shown. The anesthetized animal was located on the OR table. The robot control console contained a PC, several LCD moni-

| Animal | Weight | Medications |
|---------------|---------------|--------------------------------------|
| A | 30.6kg | |
| B | 28.7kg | Telazol Xylazine Ketamine Isoflurane |
| C | 29.5kg | |

Table 5.1: Preoperative medications and Anesthetics

tors, and video distribution components. The OR towers each had a CRT monitor, one used for displaying the CPT image, and one used for the laparoscope image (if needed). These towers also contained the laparoscope camera controller, light source, and insufflation equipment. The surgeon stood behind the OR table in a position to enable viewing of the CRTs on the OR towers. The robot controller (the author) sat behind the robot control console. The robot controller adjusted LED brightness, tilt motor position, and camera settings, all under the direction of the surgeon performing the operation. The tether from the CPT device was connected to a breakout box,

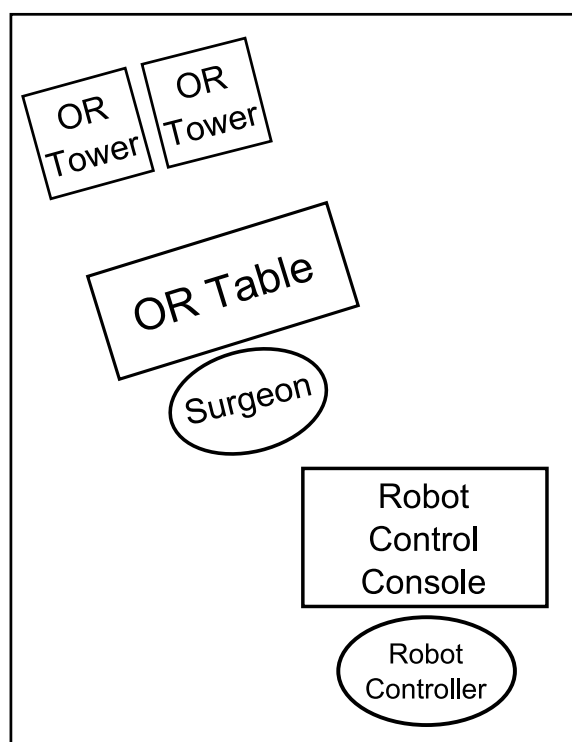


Figure 5.2: OR Layout for Survivable Surgeries

located directly underneath the OR table. The breakout box contained connections for video output, power, and USB. A USB cable ran from the PC on the robot control console to the breakout box. Motor, LED, and camera control signals were carried by this cable. An AC to DC converter supplied power to the breakout box, which

contained a DC regulator for supplying power to the CPT. The analog video output signal was routed to a video distribution amplifier on the robot control console. The distribution amplifier supplied a video signal to the CRT on the OR tower, an LCD on the robot control console, and a digital video recorder on the robot control console.

5.3 Operative Procedure

After each animal was prepped and anesthetized, a nasogastric tube was inserted and the stomach was aspirated. Locations for ports and the CPT robot were identified and marked. The peritoneal cavity was insufflated with CO₂ using a Veress needle.

Next, a 12 mm trocar was placed near the abdominal midline for insertion of the CPT and laparoscope (laparoscope used for animal A only). The sterilized CPT was removed from a sealed bag and connected to the breakout box. Power, USB, and video connections were made. The control software on the PC was used to verify functionality of the focusing mechanism, tilt capability, LED brightness, and camera settings of the CPT. Camera settings were changed to provide the best image, camera focus was set to a position deemed optimal by the surgeon. The CPT was inserted into the insufflated peritoneal cavity. During insertion, the CPT was drawn to the abdominal wall by using an external magnetic handle. This was done to minimize the chance that the clear outer tube of the CPT would make contact with any organs or tissue, which could obscure the operating field from the imager. Once attached to the abdominal wall, the magnetic handle was used to position the CPT for the procedure. After positioning, the LED brightness, focus, and tilt were again adjusted from the PC to provide the best view of the surgical environment. Additional trocars were introduced for standard laparoscopic tools.

The surgeon used the video from the CPT as visual feedback to perform the

cholecystectomy. First, the triangle of Calot was exposed by lifting the gallbladder (Figure 5.3(a)). A duckbill grasper was used to apply downward pressure of the anterior edge of Hartmann's pouch. The peritoneum of the superior leaf of the cystic pedicle was divided from the liver and the inferior peritoneal leaf was exposed. The incision was completed around the anterior edge of the cystic pedicle and the liver. The front and back structures of the cystic pedicle were dissected to expose the cystic duct and cystic artery. The cystic duct was anteriorly separated from the cystic artery using an electrosurgical hook knife (Figure 5.3(b),5.3(c)). The cystic artery was clipped and divided with the hook knife. The cystic duct was clipped at its junction with the gall bladder (Figure 5.3(d),5.3(e),5.3(f)), completing dissection of the cystic pedicle. Finally, the gallbladder was separated from the liver (Figure 5.3(g),5.3(h)), inserted into a bag (Figure 5.3(i)), and removed from the abdomen.

After the removal of the gallbladder, the gallbladder bed, duodenal bulb, and hepatic fixture of the transverse colon were inspected. The subhepatic pouch and right parahepatic space were irrigated until the returning fluid was clean. The CPT device was retracted by pulling on the electrical tether. To complete the procedure, the trocars were removed, the port incisions sutured, and local anesthetic was applied to the suture sites.

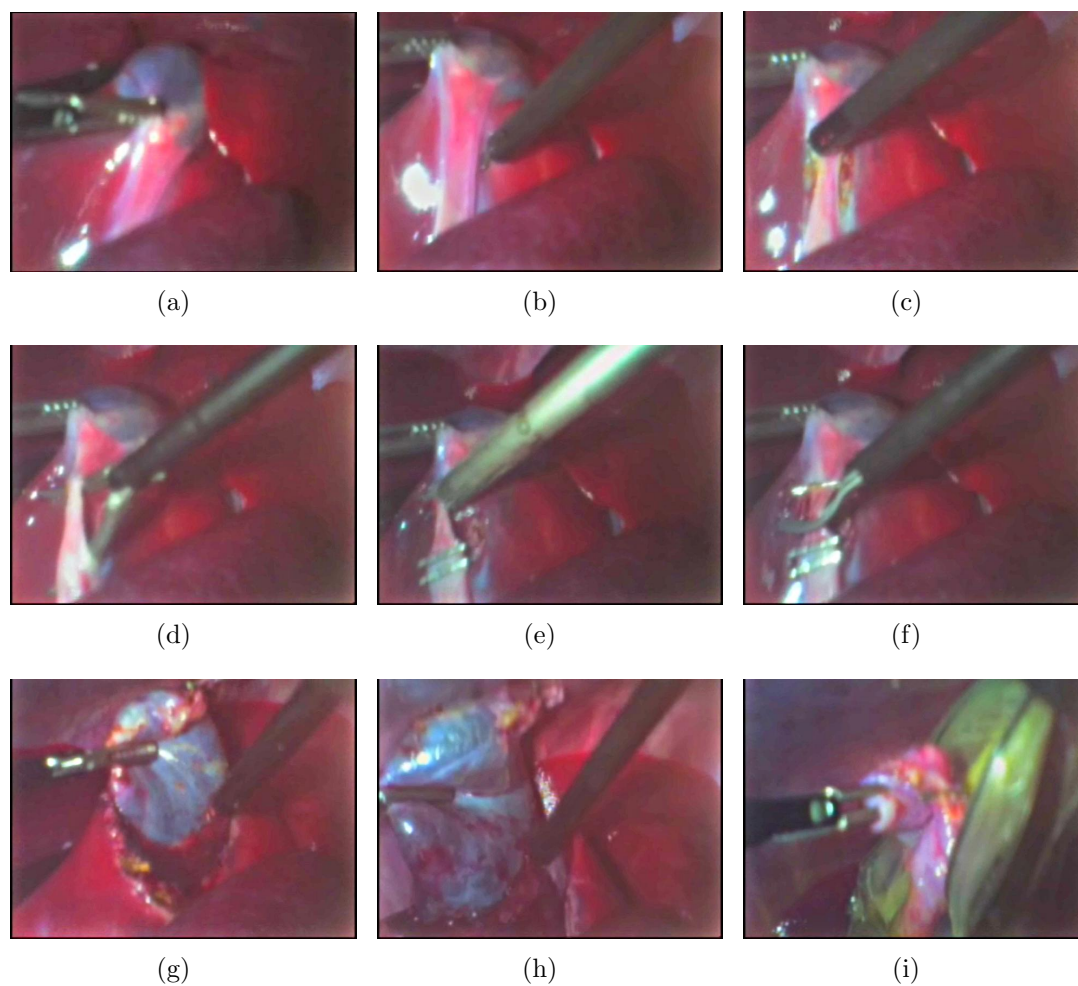


Figure 5.3: *In vivo* Images from CPT

5.4 Post-Operative Observations

After the procedure, each animal was transferred to a care facility where they received care (food, water, washing, etc.) for 14 days. Each animal was monitored for signs of infection, intestinal obstruction, urinary infection, confusion, etc. At the end of this period, the animal was placed under general anesthesia. A visual exam of the suture sites and the site of the magnetic handle was performed to look for signs of infection, trauma, adhesions, or other abnormalities. Blood samples were taken before the animal was euthanized. The abdominal cavity was accessed and a visual exam

of the internal organs was performed, with several biopsies taken for pathological examination.

5.4.1 Animal A

The surgery for animal A took place on April 10, 2007 [80]. Immediately following the surgery, the animal was observed to be doing well, with no obvious signs of pain. For the next 14 days, the animal continued to do well, it was active, had a normal appetite, and showed no signs of pain or distress. Some minor swelling was observed around the trocar incisions. On April 24, 2007, the animal was sedated, blood was collected, and the animal was euthanized. The necropsy showed a mild adhesion of the abdominal wall to the surface of the liver. The stomach, liver bed, and small bowel all appeared normal. No infections or abnormalities were noted. Biopsies were taken of the liver, small bowel, and interior abdominal wall. The pathological report showed mild congestion of the liver and no histopathological abnormalities on the small bowel sample. The abdominal wall sample showed fibrosis and mild chronic inflammation.

5.4.2 Animal B

The surgery for animal B took place on May 29, 2007 [81]. Immediately following the surgery, the animal was observed to be doing well, with no bleeding present. For the next 14 days, the animal continued to do well, was active, no bleeding was present, and it seemed alert. On June 12, 2007, the animal was sedated, blood was collected, and the animal was euthanized. The necropsy showed no abnormalities, adhesions, or other problems. Biopsy records for Animal B could not be located.

5.4.3 Animal C

The surgery for animal C took place on July 17, 2007 [82]. Immediately following the surgery, the animal was observed to be doing well, with no evidence of pain, had a good appetite, and its incisions were clean and dry. For the next 14 days, the animal continued to do well, it was active, had a normal appetite, showed no signs of pain or distress. On July 31, 2007, the animal was sedated, blood was collected, and the animal was euthanized. The necropsy revealed a very small inclusion cyst with fluid in the subxiphoid port incision that was consistent with a cutaneous abscess. The peritoneum was within normal limits. There were small adhesions to the gallbladder fascia. The small bowel, colon, stomach were all within normal limits. There was no evidence of adhesions or other abnormalities in the peritoneal cavity. Biopsies of stomach, small intestine, and liver were taken. All biopsies showed no significant histopathological abnormalities.

5.5 Summary

A miniature *in vivo* camera device (CPT-100) was designed, built, and tested to show that using these *in vivo* robots in survivable surgeries is feasible. Each device was sterilized and used in a survivable animal surgery. In the first surgery, the CPT augmented the view of the surgical field from the laparoscope. In the final two procedures, the CPT was used as sole visual feedback during the procedure. Each animal was observed for 14 days after the surgery, with all animals showing no negative effects from the surgery or the use of the CPT device. After 14 days, each animal was euthanized and necropsies were performed. Animals A and C each showed a mild adhesion, and Animal C had a small inclusion cyst present. All other tissues and organs appeared to be normal. The biopsies taken from animal A showed

small problems, and biopsies from animal C showed no abnormalities. These results show:

1. Small *in vivo* surgical robots can be successfully sterilized using different sterilization procedures with no damaging effects to the robot.
2. These *in vivo* robots can be used successfully as the sole means of visual feedback in laparoscopic procedures.
3. These *in vivo* robots have been shown to cause no significant damage or infection in survivable animal surgeries.

The author's contributions in the survivable pig studies included the following:

1. Design and testing of the printed circuit boards (PCBs) in the CPT-100.
2. Developed and tested the firmware on the CPT-100 used for USB communications, motor control, LED control, and camera control.
3. Developed and tested Windows PC software including a graphical user interface (GUI) used to control the CPT-100.
4. In charge of electronics setup for the CPT-100 devices prior to the animal surgeries.
5. Under direction of the surgeon, controlled the CPT-100 devices during the three animal surgeries.

Chapter 6

Surgical Tasks

The majority of the *in vivo* robotic devices previously described in Chapter 2 were designed to augment or replace a laparoscope or endoscope in minimally invasive surgery, although some have incorporated simple tools such as biopsy graspers (BMR-22a, Sec 2.12). These devices have been used successfully in several animal surgeries as visual feedback devices. One of the latest trends in minimally invasive surgery is natural orifice transluminal endoscopic surgery, or NOTES, which was discussed in Section 1.1.4. Consequently, some of the latest work in robotic surgery has been in developing a stable platform for NOTES. In addition to NOTES procedures, this *in vivo* surgical platform could be used in a system that is capable of performing supervised autonomous surgical tasks. Rather than controlling the joints of the robot directly, the surgeon instead issues commands to direct the robot where to move and what task should be done, e.g. grasping, cutting, etc. This would enable the surgeon to perform simple procedures over a high latency communication channel.

6.1 A Dexterous *In vivo* Manipulator Robot

In order to move the *in vivo* robots from surgical assistants to a platform that enables NOTES, a new design approach is needed that provides a stable platform for manipulation and visualization within the surgical environment, while also allowing flexibility for insertion. The AB3 robot for NOTES, shown in Figure 6.1, consists of

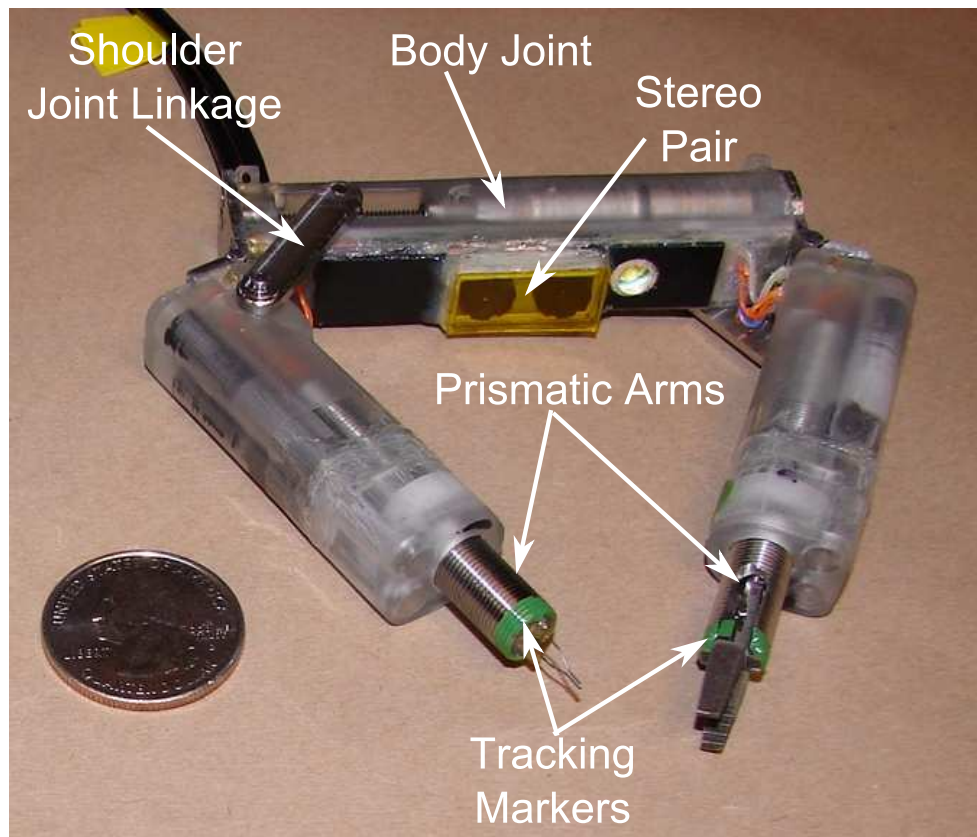


Figure 6.1: AB3 Robot

two prismatic arms connected to a central body by rotational shoulder joints. The linkage used for articulation of the shoulder joint can be disconnected, allowing flexibility at the shoulder joint for insertion. The left arm has a gripper end effector and the right has a cautery end effector. The body contains a stereo camera pair for visual feedback and magnets for attachment to the interior of the abdominal wall.

Colored markers on the robot are used for real-time position tracking. Iterations of this NOTES robot design have been demonstrated in multiple surgeries in a porcine model including peritoneal exploration and partial cholecystectomy [83].

6.2 Supervised Autonomous Tasks

In addition to using these robots for NOTES, these devices can also be used to perform supervised autonomous surgical tasks. Such tasks could be useful in situations where the patient is in a location far from a trained surgeon. A surgeon at a remote location could control the robot even if the communication channel between surgeon and patient is of low bandwidth or has very high latency. This could have applications in several areas, including:

- Long term spaceflight
- First responders on a battle field
- First responders in locations far from a hospital

In order to accomplish this, the robot would need to be able to perform simple tasks autonomously. To investigate this, a system was designed using an existing NOTES robot prototype, the AB3. A block diagram of the hardware can be seen in Figure 6.2. A PC containing a frame grabber is the center of the system. The user interface, tracker and PID controller are implemented here in C++. The framegrabber receives video input from an overhead camera and the stereo imaging pair on the AB3 robot. The actuators on the AB3 robot are driven from an external control box, which is connected to the PC via a USB connection.

This system has three main components: visual tracker, controller, and stereo vision. The user is presented with a video capture from the robot. The user then

selects a point on the image for the robot to move to (e.g. a piece of tissue to grasp or cut). The system uses a stereo correspondence algorithm to compute the location of this point in 3D space. Once this point is verified by the user as correct, the controller moves the appropriate end effector to the desired position receiving visual feedback from the tracker. This process is repeated until the desired task is completed.

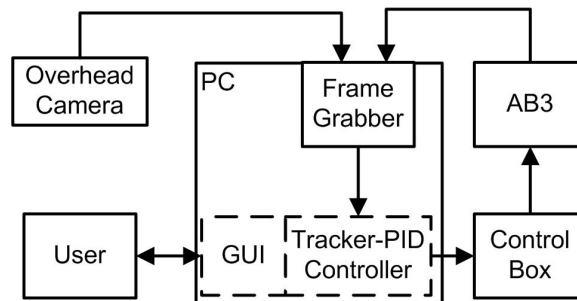


Figure 6.2: System Block Diagram

6.2.1 Tracker

To enable positional feedback for each joint controller, a color tracking system was implemented. In many robotic systems, optical encoders mounted on each motor could be used for this purpose, but the addition of encoders for this application was not feasible. Encoders would add significant size to each motor, and thus to the entire robot. In addition to the increase in size, encoders would add 4-5 wires per motor to the tether of the robot. For these reasons, an optical tracking system was implemented. It was decided to affix small color dots to specific points on the robots joints and track their positions using a color segmentation method. The first step in this method was to determine the color for the dots to optimize the performance of the tracker. A method similar to the one developed by Uemura, et al. [52, 51] was used. This method uses histograms in HSV (Hue-Saturation-Value) color space. A graphical representation of HSV color space can be seen in Figure 6.3 [84].

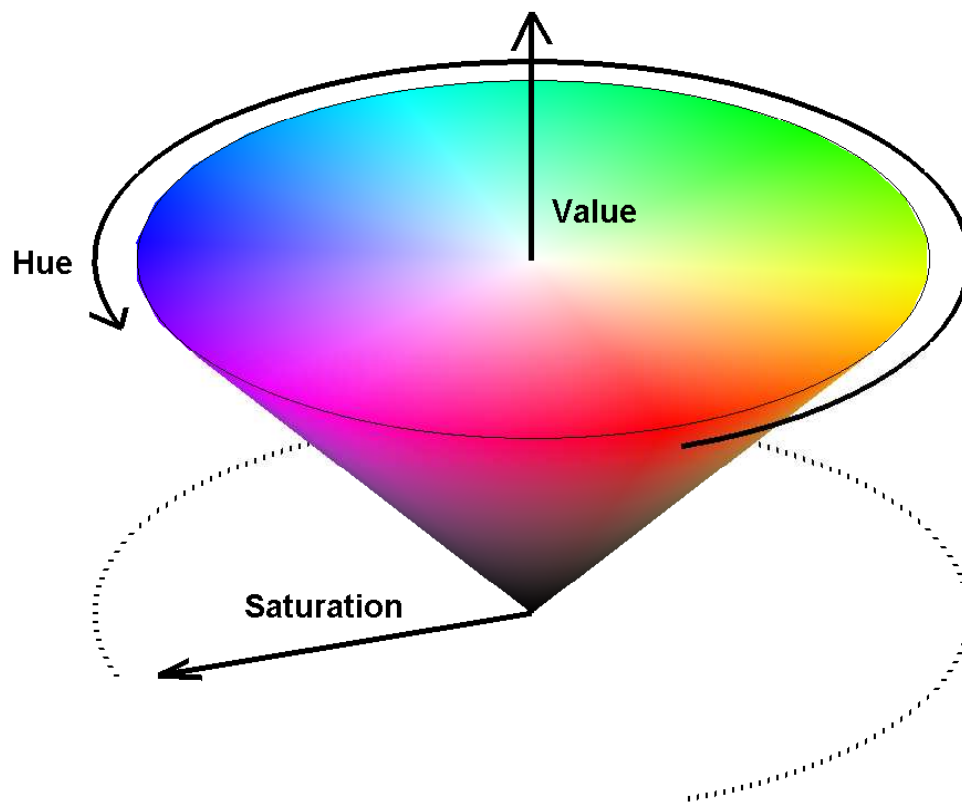


Figure 6.3: HSV Color Space

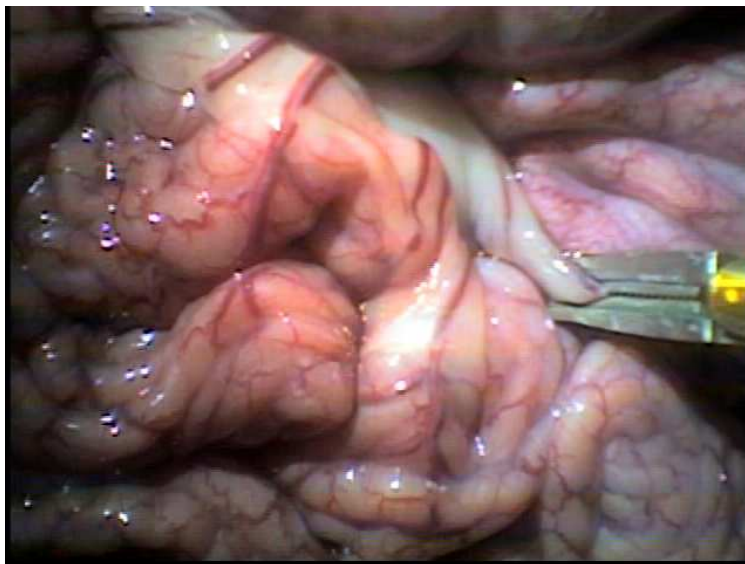
In this color space, hue is defined by the CIE as *the attribute of a visual sensation according to which an area appears to be similar to one of the perceived colors, red, yellow, green, and blue, or a combination of two of them* [85]. In the HSV cone, this value is represented as the angular coordinate of the color in question, measured in degrees, with a value of “pure” red ((1,0,0) in RGB value) corresponding to 0° and 360°. Saturation is defined as *the colorfulness of an area judged in proportion to its brightness*; or, the more a color is concentrated at a specific wavelength, the higher its saturation. Pure white would therefore have a saturation of 0. In the HSV cone, saturation is represented as the radial distance from the center of the cone, with 0 at the center and 1 at the edge. Value, or brightness, can roughly be described as the total amount of radiant energy. It is represented as height in the HSV cone, with 0

at the point, and 1 at the base.

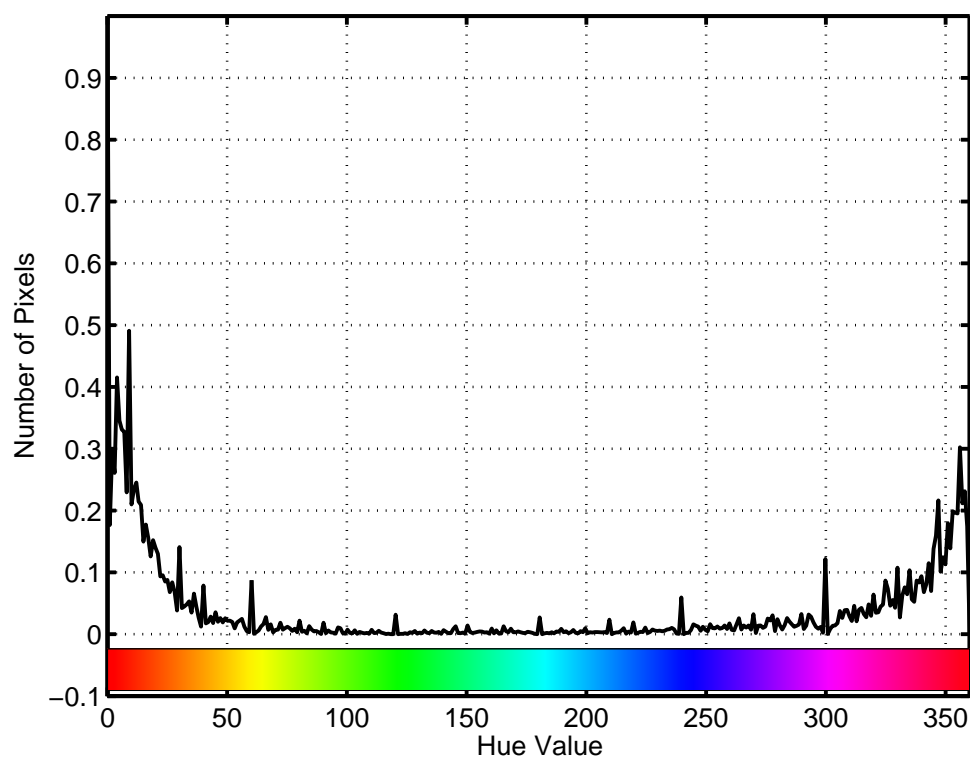
To determine the color for the tracker markers, several *in vivo* images were analyzed. Two of these *in vivo* test images can be seen in Figures 6.4(a) and 6.5(a). These images are of large bowel and liver. Figures 6.4(b) and 6.5(b) show the hue value histograms obtained using Matlab after converting the images from RGB to HSV color space.

In both the histogram images, most of the pixels are concentrated in the orange-red to violet-red regions, which is what one would intuitively expect for *in vivo* images. After examining several of these images, green ($H \approx 120^\circ$) was chosen as the marker color. These markers were then made and attached to the robot. Due to the geometry of the robot, positions of the imagers, and lens selection, the color markers were not visible over the entire workspace of the robot when using the robot's cameras. This effect is illustrated in Figure 6.6, which shows the overhead and robot camera views for a small joint angle. In this image, the marker on the upper arm is not visible. Therefore, to speed up development of the entire controller, it was decided to first implement the tracker from an overhead view. In a final stand-alone system, overhead tracking would not be feasible, because the *in vivo* robot would be the only source of visual feedback from the operating environment. However, the methods described below would have similar implementations tracking from an robot camera view versus an overhead view.

A flowchart of the tracker implementation can be seen in Figure 6.7. The tracker was written in C++ using OpenCV, an open source computer vision library [86], which was designed for real-time computer vision applications. First, the tracker is initialized with the starting marker positions. This is done either from a initialization file, or manually by the user clicking on the approximate locations in a video frame displayed in the user interface. The video image is digitized by the frame grabber and



(a) Original Image

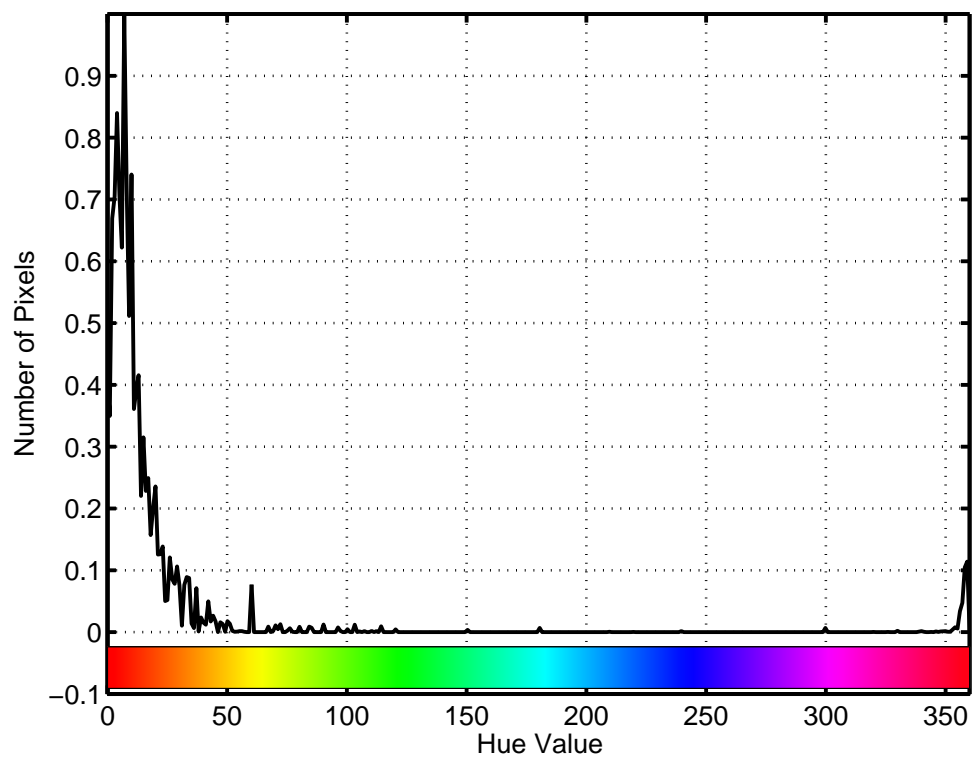


(b) Hue Histogram

Figure 6.4: Large Bowel



(a) Original Image



(b) Hue Histogram

Figure 6.5: Liver and Abdominal Wall

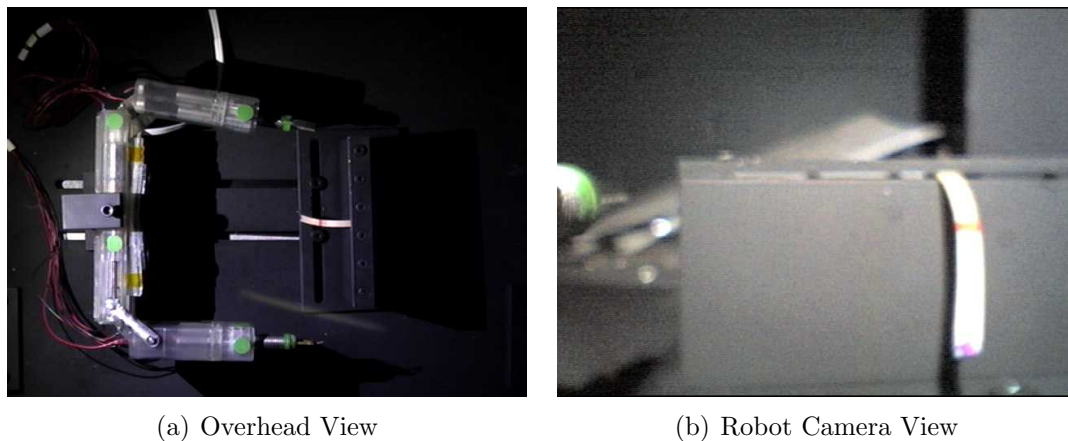


Figure 6.6: Tracker Marker Obstruction

converted from RGB to HSV using an OpenCV library function. A binary image is calculated using the the hue channel according to Equation 6.1.

$$M_H(x, y) = \begin{cases} 1 & H_{Low} < I(x, y) < H_{High} \\ 0 & \text{else.} \end{cases} \quad (6.1)$$

Here, H_{Low} and H_{High} are the range of hue values of the colored markers, $M_H(x, y)$ is the pixel at (x, y) in the hue mask image, and $I(x, y)$ is the pixel at (x, y) in the original image. An example of this can be seen in Figures 6.8(a) and 6.8(b). The white pixels in Figure 6.8(b) have a value of 1. This mask has significant amounts of noise in it. To reduce this noise, another mask image is calculated using the same method as in Equation 6.1, but on the saturation channel. This mask can be seen in Figure 6.8(c). These two binary masks are combined using a pixel by pixel logical AND operation, resulting in the final mask image, Figure 6.8(d).

After isolating the areas where the possible markers are in the image, these areas need to be located in the image. Using the OpenCV function *cvFindContours*, the connected components are located in the resultant binary image. The central moments are then calculated for each connected component. An area thresholding

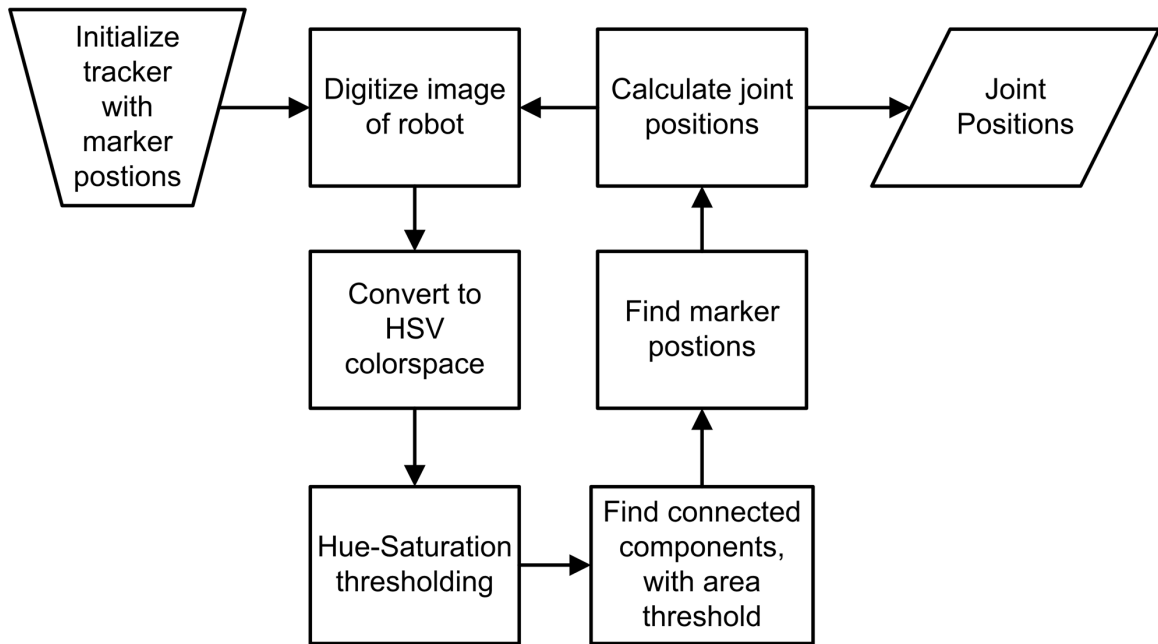


Figure 6.7: Tracker Operation

operation removes all components below a certain area in pixels. The centroid for each of the remaining moments is stored in a list. This list is compared to the last known positions of each marker and all distances are calculated. An example list is shown in Table 6.1. In this situation, three markers are being tracked and four possible markers have been identified. The data in the table d_{Xy} represents the distance in pixels from marker X to possible marker y . Each column is then sorted and ranked, with the shortest distance being ranked one. The possible marker location with a rank of one is then used for the next location for each marker. This simple ranking system has been shown to work in normal benchtop operating conditions. A more sophisticated system would need to be implemented to track a faster moving robot, to deal with temporary occlusions, or large changes in lighting. One possibility could be the feature correspondence algorithm proposed by Zhongxiang and Ronghua [49], on which the method described here is loosely based. This method uses multiple features to compute the best match for each item.

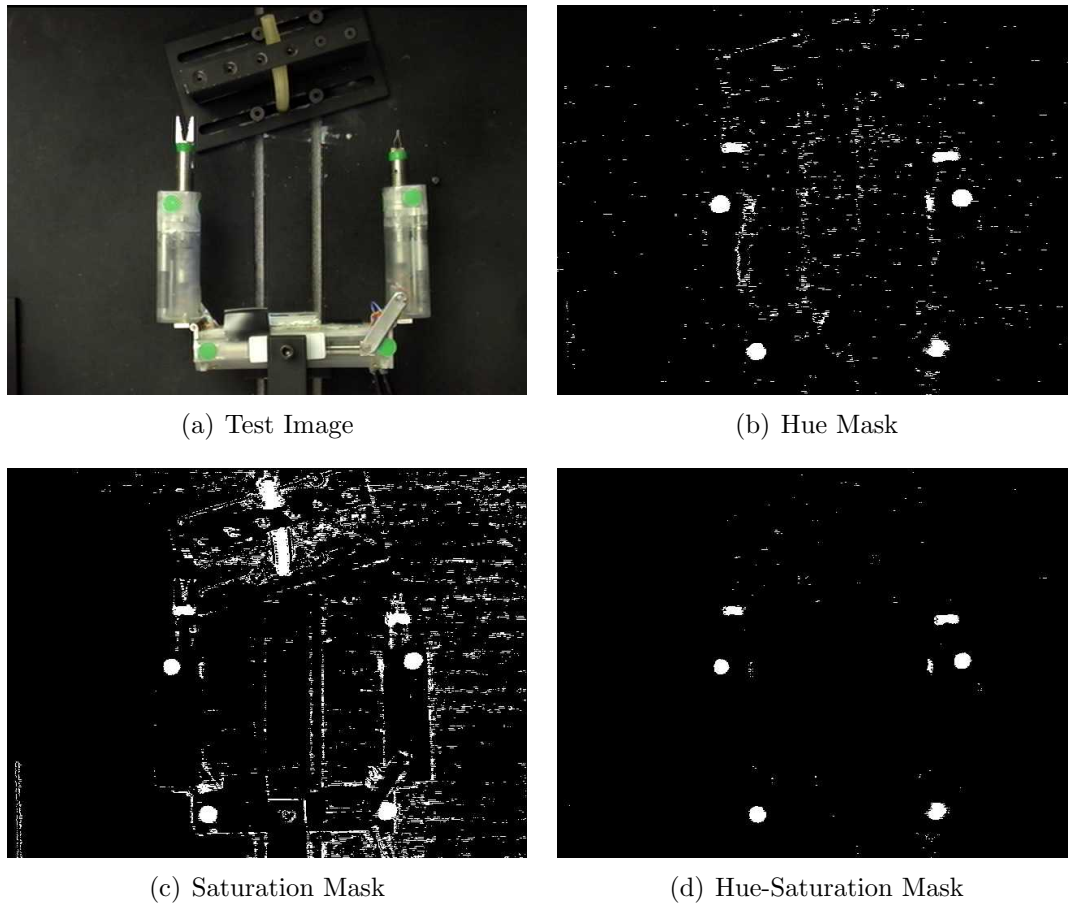


Figure 6.8: Hue-Saturation Masks

The next step is calculating the end effector positions using the joint marker positions. This process is summarized below:

1. Convert marker coordinates to right handed coordinates
2. Calculate rotation angle (rotation of image and/or robot)
3. Calculate robot origin in image coordinates
4. Convert markers to robot coordinates
5. Calculate wrist length and shoulder angle using inverse kinematic equations
6. Calculate end effector position

| Possible Markers | Current Marker | A | B | C |
|------------------|----------------|----------|----------|----------|
| 1 | | d_{A1} | d_{B1} | d_{C1} |
| 2 | | d_{A2} | d_{B2} | d_{C2} |
| 3 | | d_{A3} | d_{B3} | d_{C3} |
| 4 | | d_{A4} | d_{B4} | d_{C4} |

Table 6.1: Distance Table

First, the marker coordinates must be converted from the left handed image coordinates to a right handed coordinate system, shown in Figure 6.9. The equations for

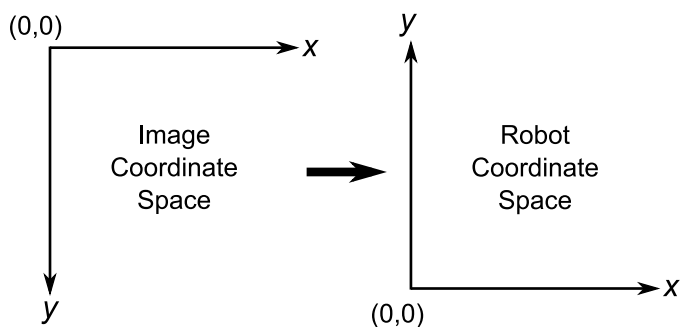


Figure 6.9: Image Coordinate Conversion

conversion are:

$$x_{right} = x_{left} \tag{6.2}$$

$$y_{right} = y_{max} - y_{left},$$

where y_{max} is the height of the image capture in pixels. A pixels to millimeters scaling factor, S_{mm} , is then calculated using the positions of markers $M1$ and $M2$ (see Figure 6.10). Since these are fixed markers, the distance between them can be measured and stored as a constant in the program. Next, an angular correction is calculated. This is to correct for any rotation in the robot or the camera. The correction is calculated

using markers $M1$ and $M2$:

$$\theta_{rot} = \tan^{-1} \left(\frac{y_{M2} - y_{M1}}{x_{M2} - x_{M1}} \right). \quad (6.3)$$

θ_{rot} is then used to calculate the center point of the robot body (body marker center

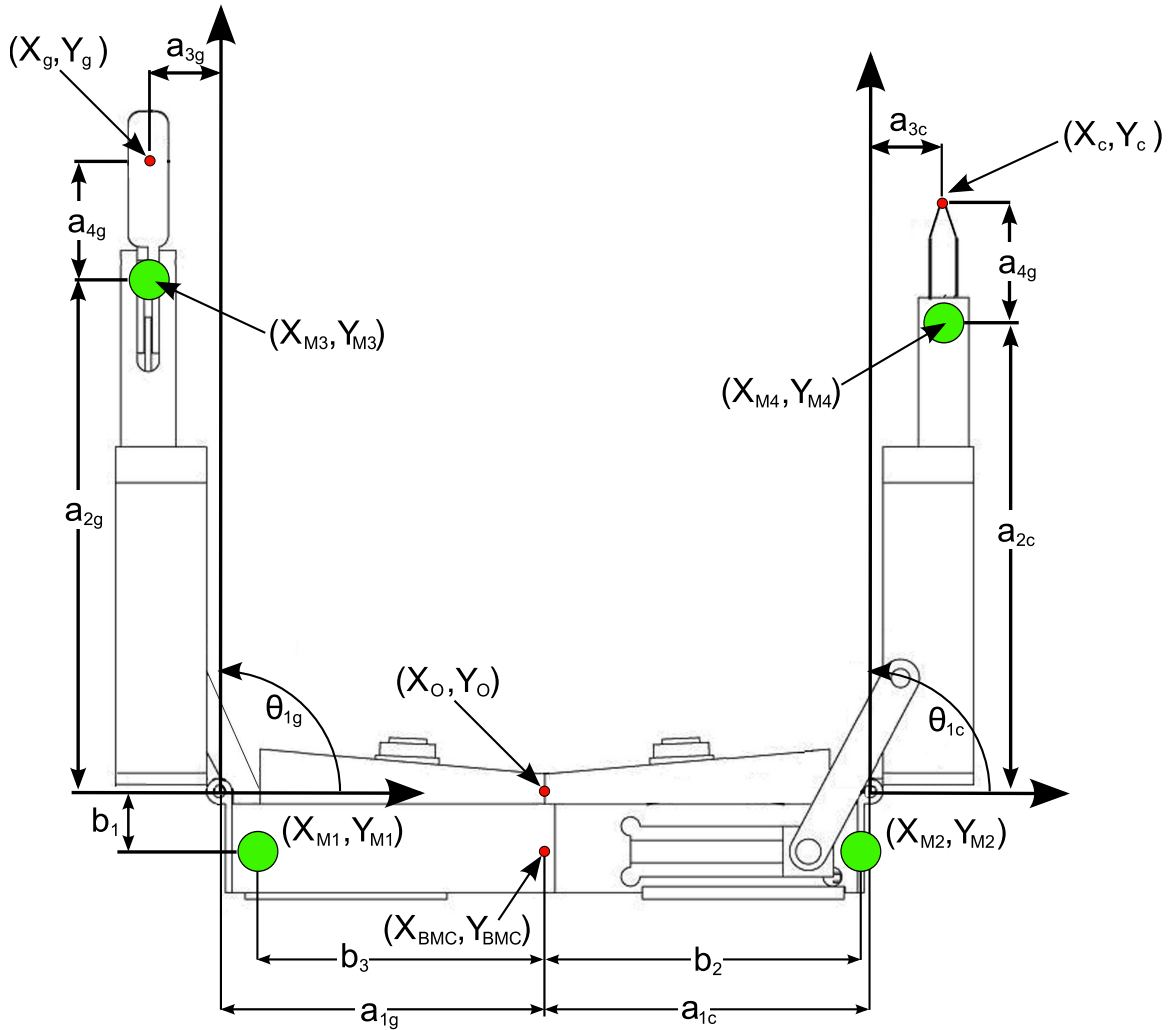


Figure 6.10: AB Robot Measurements

or BMC):

$$\begin{aligned} x_{BMC} &= \frac{x_{M1} + b_2 * \cos(\theta_{rot}) + x_{M2} - b_3 * \cos(\theta_{rot})}{2} \\ y_{BMC} &= \frac{y_{M1} + b_2 * \sin(\theta_{rot}) + y_{M2} - b_3 * \sin(\theta_{rot})}{2}. \end{aligned} \quad (6.4)$$

The BMC, pixels to mm scale factor (S_{mm}), and b_1 from Figure 6.10 are used to transform the marker points from image space to robot space, (See Equations 6.5 and 6.6). Matrix \mathbf{S} is used to scale the positions from pixels to millimeters, using the previously calculated S_{mm} value. \mathbf{T}_1 is a matrix used to calculate the translation of the body marker center to the robot origin, a shift of b_1 mm along the y axis. \mathbf{R} is the rotational matrix used to calculate the transformation due to the image rotation, θ_{rot} . The second translation matrix, \mathbf{T}_2 , is used to calculate the shift of the body marker center, (x_{BMC}, y_{BMC}) .

$$\mathbf{S} = \begin{bmatrix} S_{mm} & 0 & 0 & 0 \\ 0 & S_{mm} & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad \mathbf{T}_1 = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & b_1 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}, \quad (6.5)$$

$$\mathbf{R} = \begin{bmatrix} \cos(\theta_{rot}) & -\sin(\theta_{rot}) & 0 & 0 \\ \sin(\theta_{rot}) & \cos(\theta_{rot}) & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad \mathbf{T}_2 = \begin{bmatrix} 1 & 0 & 0 & -x_{BMC} \\ 0 & 1 & 0 & -y_{BMC} \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}. \quad (6.6)$$

Combining these matrices, we get the overall translation-rotation-scaling matrix:

$$\mathbf{T}_{ov} = S_{mm} \begin{bmatrix} \cos(\theta_{rot}) & -\sin(\theta_{rot}) & 0 & -x_{BMC} \cos(\theta_{rot}) + y_{BMC} \sin(\theta_{rot}) \\ \sin(\theta_{rot}) & \cos(\theta_{rot}) & 0 & -x_{BMC} \sin(\theta_{rot}) - y_{BMC} \cos(\theta_{rot}) + b_1 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}. \quad (6.7)$$

Equation 6.7 is then used to translate the marker locations from image space to robot space. Using the inverse kinematic equations 6.8–6.11 derived by Wood [56], the joint variables θ_{1g} , θ_{1c} , a_{2g} , and a_{2c} are calculated. For the cautery arm, $x = x_{M4}$, $y = y_{M4}$; and for the gripper arm, $x = -x_{M3}$, $y = y_{M3}$.

$$a_2 = \sqrt{x^2 + y^2 - 2xa_1 + a_1^2 - a_3^2} \quad (6.8)$$

$$\theta_1 = \tan^{-1} \left(\frac{A}{B} \right) \quad (6.9)$$

$$A = \frac{xa_3 + ya_2 - a_1a_3}{x^2 + y^2 - 2xa_2 + a_2^2} \quad (6.10)$$

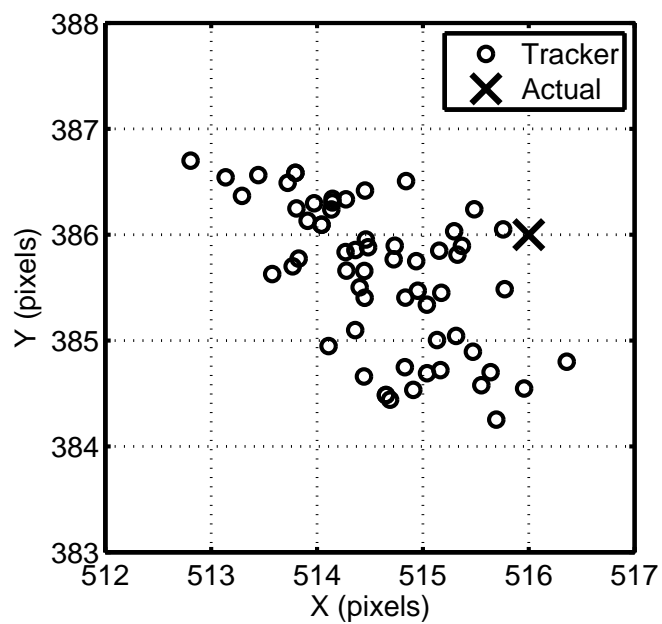
$$B = \frac{a_2(x - a_1) - ya_3}{x^2 + y^2 - 2xa_1 + a_1^2} \quad (6.11)$$

Using the kinematic model of the device provided by Wood [56], the positions of the end effectors can now be calculated in the robot frame using Equations 6.12 and 6.13. The a_4 term is added to account for the distance between marker $M3$ or $M4$ and the tip of the end effector. The previous calculations did not take this factor into consideration.

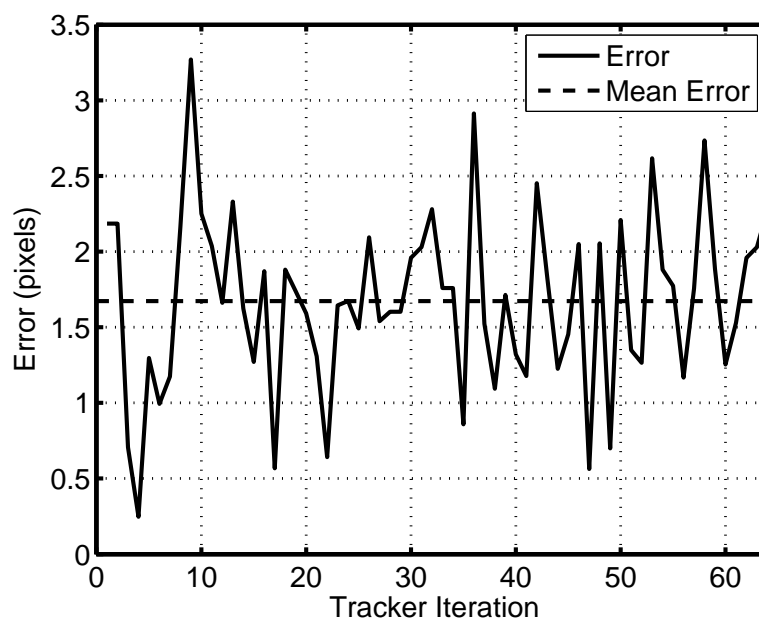
$$\begin{aligned} x_g &= -(a_{2g} + a_{4g}) \cos(\theta_{1g}) - a_{3g} \sin(\theta_{1g}) - a_{1g} \\ y_g &= (a_{2g} + a_{4g}) \sin(\theta_{1g}) - a_{3g} \cos(\theta_{1g}) \end{aligned} \quad (6.12)$$

$$\begin{aligned} x_c &= (a_{2c} + a_{4c}) \cos(\theta_{1c}) + a_{3c} \sin(\theta_{1c}) + a_{1c} \\ y_c &= (a_{2c} + a_{4c}) \sin(\theta_{1c}) + a_{3c} \cos(\theta_{1c}) \end{aligned} \quad (6.13)$$

Assessing tracker performance proved to be somewhat difficult. Tracking the actual position of the end effectors with high accuracy would require additional hardware and modifications to the robot, such as using an external optical tracking system. It was decided to find actual position by manually selecting the end effector position from the image captures, and recording these values. When these values were added to a scatter plot with the calculated tracker position, the data showed similar amounts of variability, even when the data set consisted of images where the end effector did not change position. Rather than introduce this measurement error into the evaluation of the tracker, only the stationary tracker data will be analyzed. Figure 6.11(a) shows a scatter plot of 60 consecutive calculated locations of the stationary cautery end effector. The actual value of the end effector was found by analyzing several of the images at higher magnification. An error plot for this data is plotted in Figure 6.11(b). The mean error for this data is 1.625 pixels, which is an error of approximately 0.4 mm. Compared to the size of the simulated tissue (3.6 mm or 11%) and the open gripper size (6.5mm or 6.1%), this amount of error was determined to be acceptable, and no further analysis was conducted on the tracker error.



(a) Tracker Scatter



(b) Tracker Error

Figure 6.11: Tracker Performance

6.2.2 Controller

With the tracker functioning, the position controller could be designed. The first step in designing the controller is to model the system, with a separate model for each joint. Using the tracker previously developed, open loop step tests were performed and the joint position data were recorded for later analysis. Data were recorded for step input magnitudes of 25%, 50%, 75%, and 100% of maximum in both forward and reverse directions. These data were imported into Matlab and fit to a linear model. Inspection of the step output data indicated that a simple model would most likely be sufficient for modelling this system. The step data was fit to the form $y(t) = mt + b$ for each step magnitude and direction, for a total of eight fits. The system transfer functions were then calculated according to the following:

$$Y(s) = X(s)H(s) \Rightarrow H(s) = \frac{Y(s)}{X(s)}, \quad (6.14)$$

where $Y(s)$ is the Laplace transform of the output, $X(s)$ is the Laplace transform of the input (unit step in this case), and $H(s)$ is the transfer function of the system. $Y(s)$ and $X(s)$ are found using 6.15 and 6.16.

$$y(t) = mt + b \xleftrightarrow{\mathcal{L}} Y(s) = \frac{sb + m}{s^2} \quad (6.15)$$

$$x(t) = u(t) \xleftrightarrow{\mathcal{L}} X(s) = \frac{1}{s} \quad (6.16)$$

The transfer function of each joint is then found to be:

$$H(s) = \frac{sb + m}{s}. \quad (6.17)$$

where m and b are the values found in the time domain fit. The Matlab *step* function is used to calculate a simulated step response for each transfer function. This simulated response is plotted with the recorded response. These plots can be seen in Figures 6.15-6.18 on pages 142-145 at the end of this section.

These eight transfer functions are then analyzed to find the best transfer function for each joint. Eight step simulations are calculated for each transfer function, one for each of the experimental step inputs tested. The differences between each of these simulations and the actual experimental data are calculated, and RMS error values are calculated. These RMS values are stored in an array. Next, the mean value over each row (or each fit) is calculated. The corresponding transfer function of the minimum of these means is deemed to be the best fit. This process is better illustrated in Table 6.2

The above process is completed for each joint. The resulting transfer functions are shown in Equation 6.18.

$$\begin{aligned}
 H_{gw}(s) &= \frac{-0.4316s+1.585}{s} & H_{cw}(s) &= \frac{-0.2325s+1.407}{s} \\
 H_{gs}(s) &= \frac{0.01367s-0.07039}{s} & H_{cs}(s) &= \frac{0.0146s+0.01185}{s}
 \end{aligned}
 \tag{6.18}$$

A PID control scheme was decided upon, due to the ease of implementation of

| Fits ↓ | Experimental Output ⇒ | y_1 | y_2 | ⋯ | y_8 | Mean Error |
|-----------|--------------------------|-------------|-------------|---|-------------|--|
| $H(s)_1$ | | $e_{h1,y1}$ | $e_{h1,y2}$ | ⋯ | $e_{h1,y8}$ | ⇒ $\mu_1(e_{h1,y1}, e_{h1,y2}, \dots)$ |
| $H(s)_2$ | | $e_{h2,y1}$ | $e_{h2,y2}$ | ⋯ | $e_{h2,y8}$ | ⇒ $\mu_2(e_{h2,y1}, e_{h2,y2}, \dots)$ |
| ⋮ | | ⋮ | ⋮ | ⋮ | ⋮ | ⋮ |
| $H(s)_8$ | | $e_{h8,y1}$ | $e_{h8,y2}$ | ⋯ | $e_{h8,y8}$ | ⇒ $\mu_8(e_{h8,y1}, e_{h8,y2}, \dots)$ |
| | | | | | Best Fit | ⇐ $min(\mu_1, \mu_2, \dots)$ |

Table 6.2: Transfer Function Decision Table

this type of controller. The control system block diagram can be seen in Figure 6.12. In this diagram, R is the command position, G is the transfer function of the PID controller, H is the system transfer function, T is the tracker, and Y is the PWM output signal.

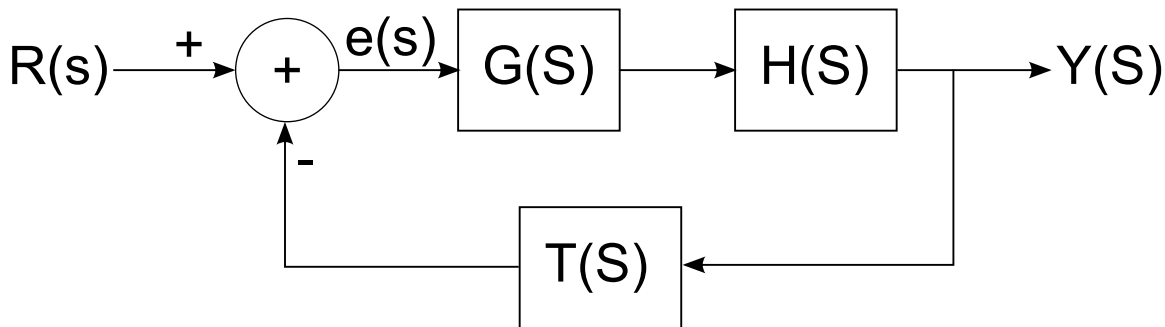


Figure 6.12: Joint Control System

The transfer function for a continuous time PID controller is:

$$G(s) = K_p + \frac{K_i}{s} + K_d s. \quad (6.19)$$

where K_p is the proportional gain, K_i is the integral gain, and K_d is the derivative gain. Looking at each term in the time domain we have:

$$g(t) = K_p e(t) + K_i \int_0^t e(\tau) d\tau + K_d \frac{d}{dt} e(t), \quad (6.20)$$

where $e(t)$ represents the error at time t . To implement these terms in a digital computer, the following approximations are used [87]:

$$\begin{aligned} p(k) &= K_p e(k), \\ i(k) &= i(k-1) + K_i e(k), \\ d(k) &= K_d (e(k) - e(k-1)), \end{aligned} \quad (6.21)$$

where $e(k)$ is the error term at time k . The equation for the output of the controller at time k , which is implemented in software is given in Equation 6.22.

$$u[k] = K_p e[k] + K_i i_{int} + K_d (e[k] - e[k - 1])$$

$$i_{int} = \left\{ \begin{array}{ll} \sum_{t=0}^k e[t] & i_{min} < i_{int} < i_{max} \\ i_{max} & i_{int} > i_{max} \\ i_{min} & i_{int} < i_{min} \end{array} \right\} \quad (6.22)$$

A separate controller was designed for each joint and implemented in C++. A flowchart of this controller can be seen in Figure 6.13.

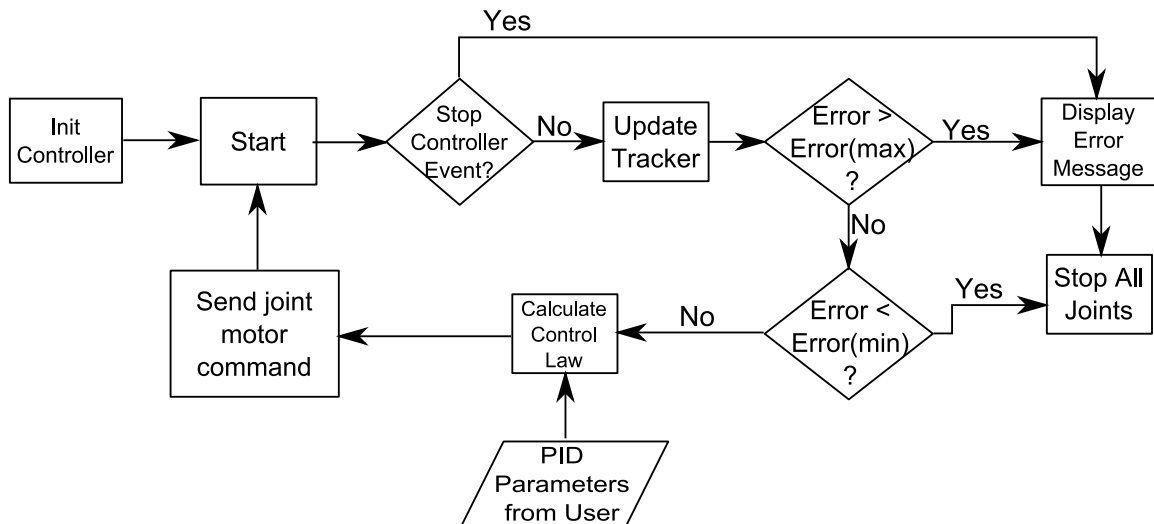


Figure 6.13: PID Controller Implementation

After modeling the system and discovering it can be described as a quasi-zero order system, the traditional process of analytically determining PID constants was not investigated. Instead, the PID constants were manually tuned on the benchtop. This process resulted in a controller in which no derivative control was necessary. Therefore, the final system is described as a P-I controller. A plot of the gripper end effector position while executing a movement from 88° to 62° with PI constants

$p = 500$, $i = 15$, and $i_{max} = 20$ can be seen in Figure 6.14(a). At the start of motion, the error is very large which results in a P value of nearly 100. When the output PID value exceeds 100 for a short time after the integral term starts to grow, the output is limited to 100, because a PWM duty cycle of over 100% cannot be realized. As the error drops, the P value drops off, and with it, the output. At approximately 2.8 seconds, the integral component has reached its limit of 20. This maximum was introduced to stop the integral term from becoming dominant. At 3.4 seconds, all values drop to zero when the joint has reached the “in position” window, or minimum error, (See Figure 6.13). This window, set at 0.5° in this example, is necessary to prevent the controlling from oscillating over the set point indefinitely, since it is unlikely it will achieve the precise command position. When controlling the shoulder joints, the angle is not variable being controlled, rather it is the slider position. Due to the geometry of the slider-crank mechanism, the slider-joint angle relation is not linear [88]:

$$s = L_2 \sin(\Phi) + L_1 \sqrt{1 - \left(\frac{e + L_2 \cos(\Phi)}{L_1}\right)^2} \quad (6.23)$$

$$\Phi = \theta_{joint} - 105^\circ,$$

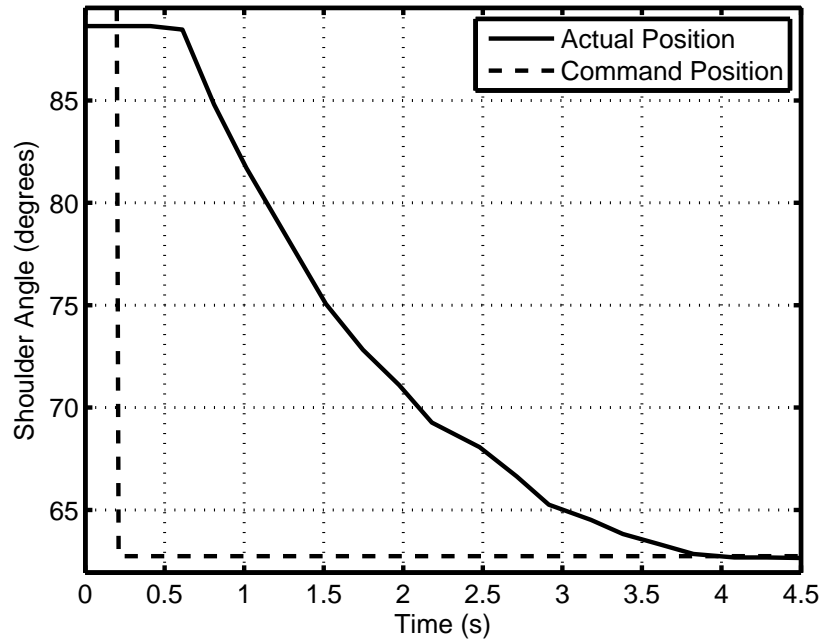
where s is the slider position, and θ_{joint} is the measured joint angle.

A separate controller was implemented for each joint. The PID constants used can be seen in Table 6.3. The “Position Limits” parameters were introduced to stop

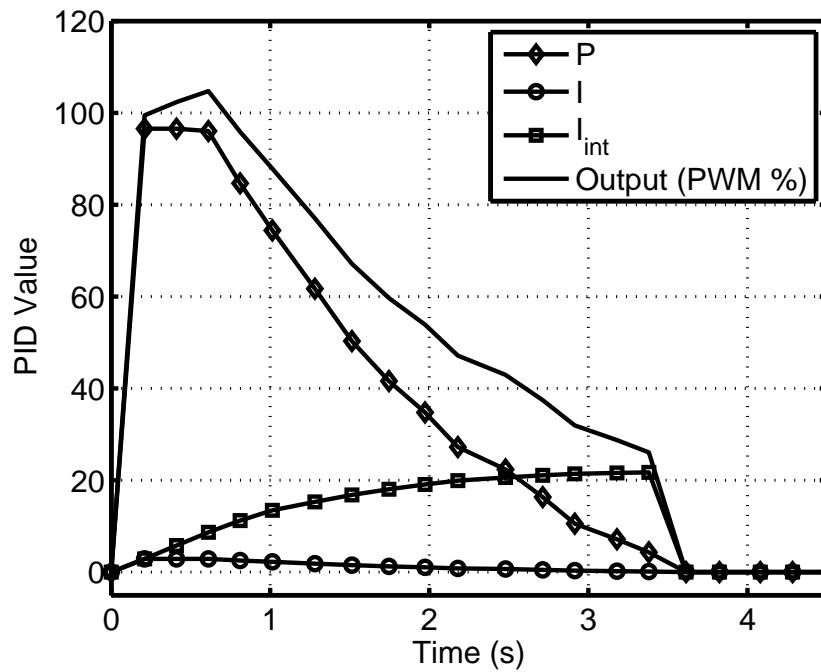
| Joint | p | i | i_{max} | In Position | Position Limits |
|------------------|-------|------|-----------|-------------|--------------------------|
| Cautery Wrist | 12.0 | 2.0 | 50.0 | 0.5 mm | 70 mm-100 mm |
| Gripper Wrist | 12.0 | 2.0 | 50.0 | 0.5 mm | 78 mm-105 mm |
| Cautery Shoulder | 500.0 | 10.0 | 20.0 | 0.5° | 88° - 140° |
| Gripper Shoulder | 500.0 | 15.0 | 20.0 | 0.5° | 85° - 130° |

Table 6.3: PID constants

all joint motion and prevent damage to the robot if a command position is given that is out of the movement range.



(a) Gripper Angle



(b) Controller PID Values

Figure 6.14: Position and PID Internal Values

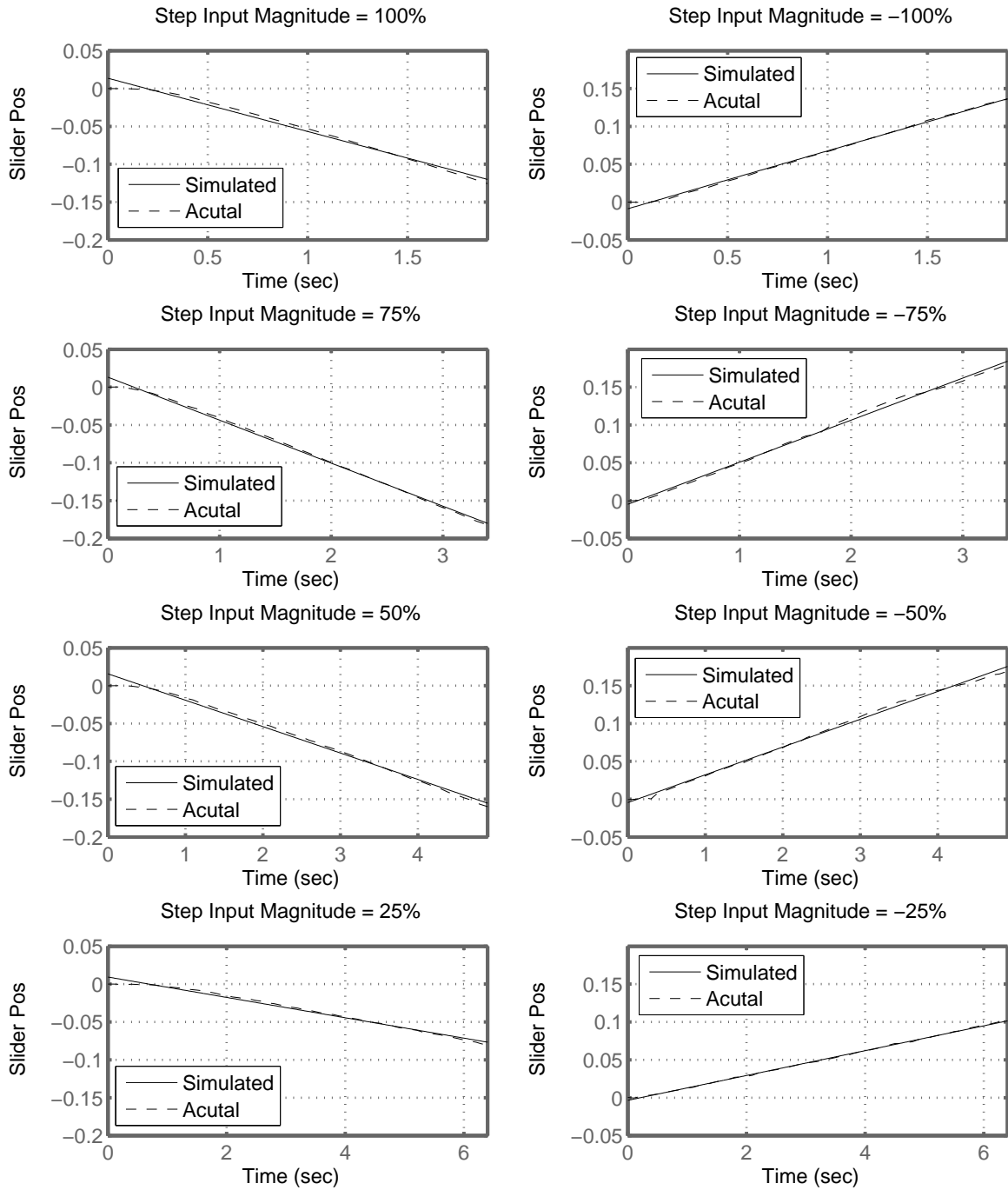


Figure 6.15: Gripper Shoulder Models

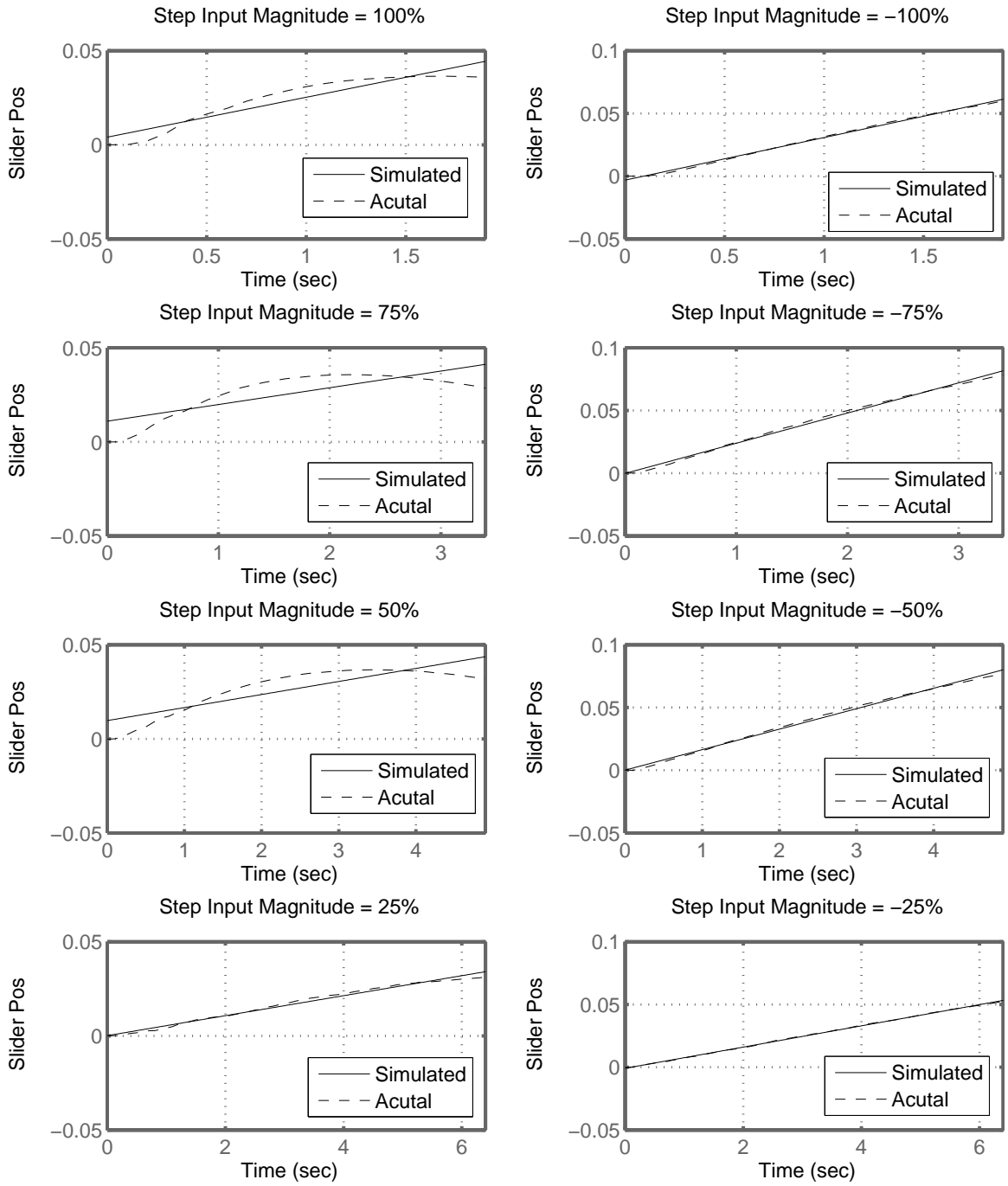


Figure 6.16: Cautery Shoulder Models

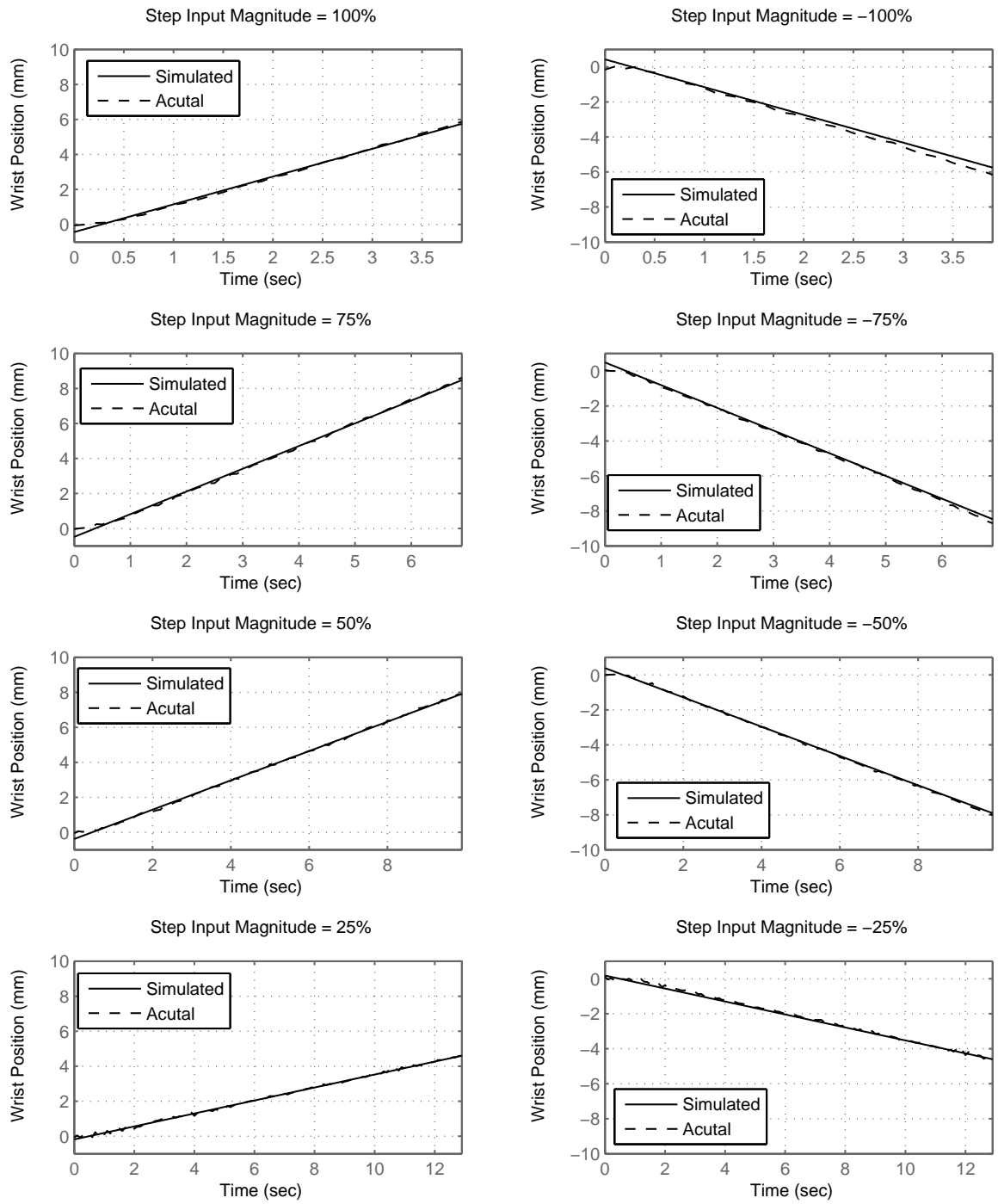


Figure 6.17: Gripper Wrist Models

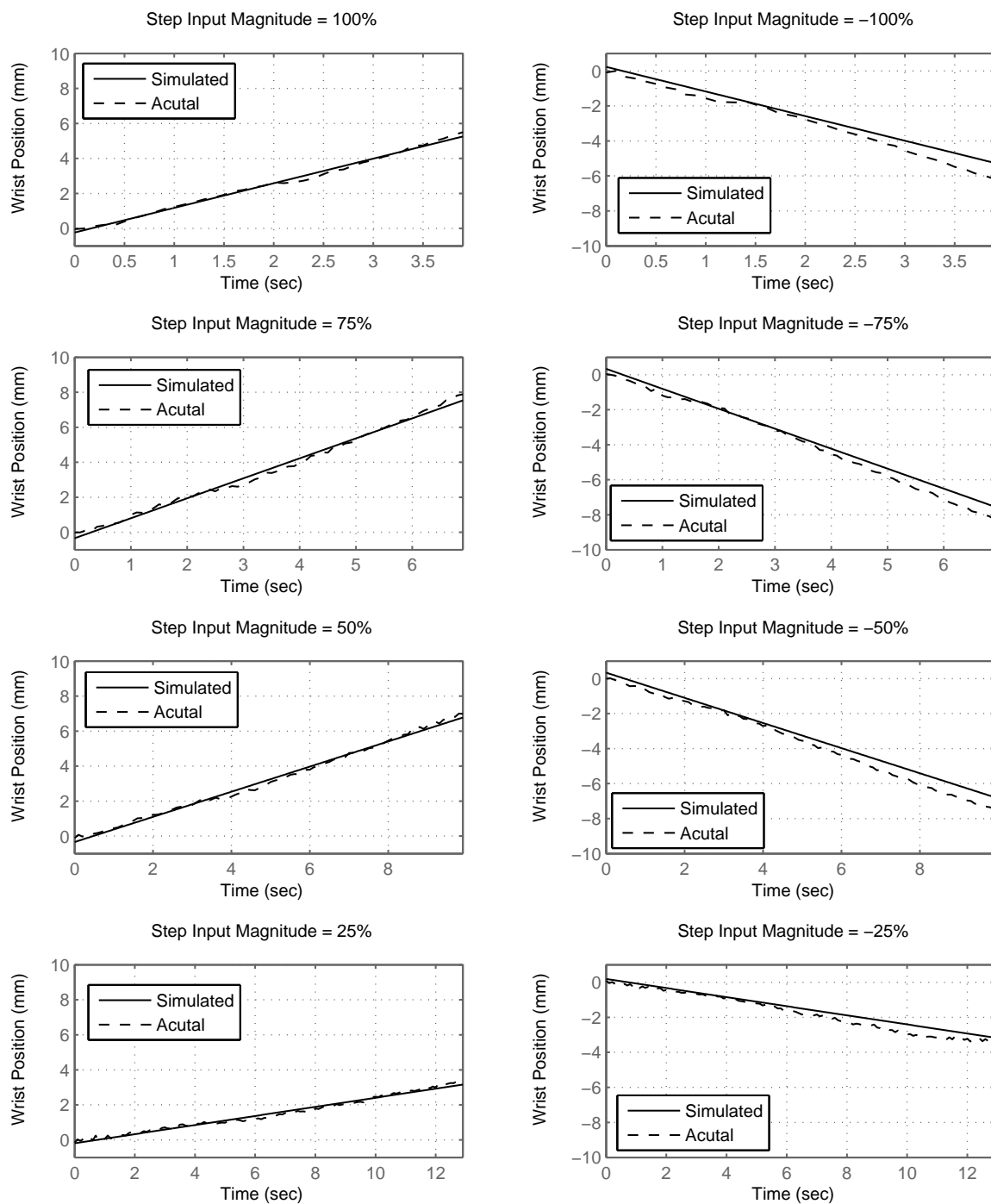


Figure 6.18: Cautery Wrist Models

6.2.3 Stereo Vision

The stereo vision component of the system is used to compute the desired end effector position based on user input. The user is presented with one of two images from the stereo imager pair on the robot, where he or she selects a piece of tissue to grasp or dissect. The software then uses a stereo correspondence algorithm to find the corresponding point in the image from the other stereo imager. The software then computes the three-dimensional coordinates from the disparity and known camera geometry. This system illustrates the two basic steps in computer based stereo imaging: correspondence and calculating three-dimensional coordinates from a pair of two-dimensional coordinates. This stereo vision algorithm was not developed by the author; rather an existing stereo vision algorithm was integrated into the system by the author.

Correspondence refers to calculating disparity values for features in an image. A point in the field of view of a pair of imagers will project onto different points of the image plane in each imager. The difference in the positions of these corresponding projected points is referred to as disparity. Manually finding these corresponding points is usually trivial, but developing an algorithm to accomplish this is a highly complex topic, and is a heavily investigated topic in computer vision. Numerous algorithms have been developed and several of these were evaluated and ranked by Scharstein and Szeliski [89].

When the correspondence problem is solved, three-dimensional coordinates can be calculated from the pair of corresponding image points. A process for accomplishing this in the system described here is shown below. Figure 6.19 shows a top-down projection of a single camera imaging an object at point P . Point $P(x, y)$ is the point of interest. It lies a distance r_{obj} from the camera, O , and has coordinates (x_{obj}, y_{obj}) . The horizontal field of view is defined as θ_{fov} , with $-x_{max}$ and x_{max} defining the field

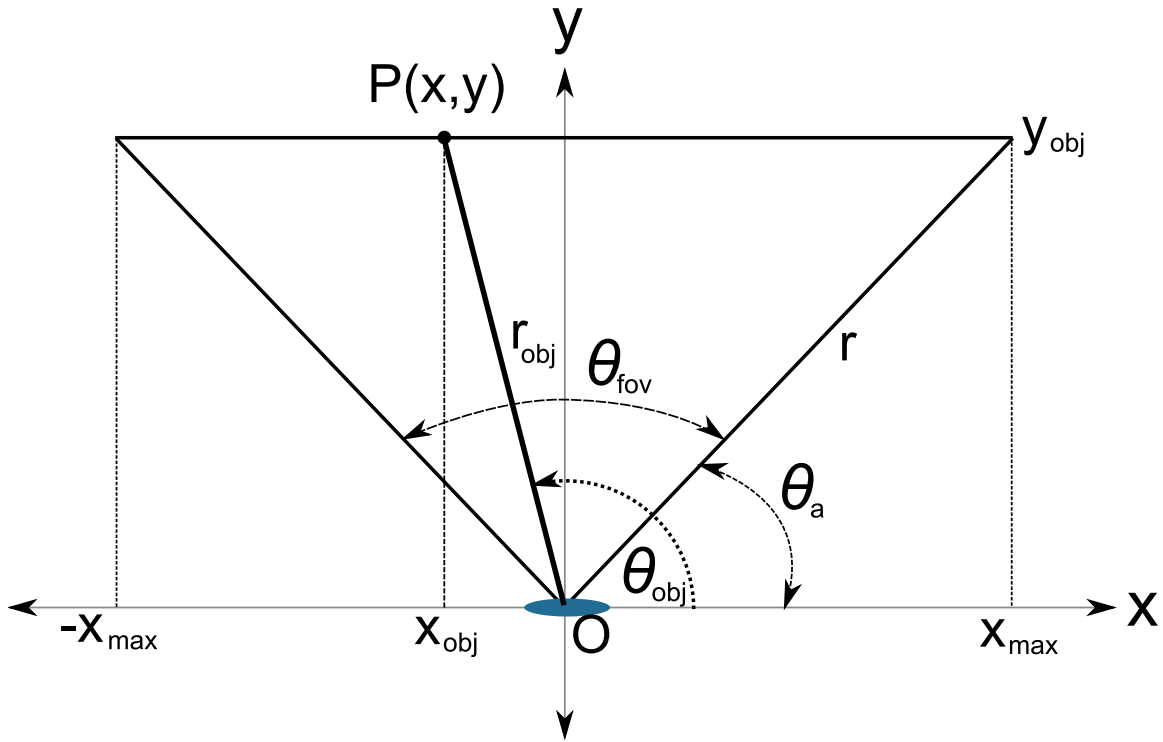


Figure 6.19: Single Camera

that is viewable at distance y_{obj} from the camera. The quantity of interest is θ_{obj} , the angle of the point with respect to the x -axis. Another angle necessary for the calculations, θ_a , is defined as:

$$\theta_a = \frac{\pi - \theta_{fov}}{2}. \quad (6.24)$$

The field of view of the camera is projected onto the imager with a width of N pixels. We will define $-\frac{N}{2}$ as the leftmost pixel, corresponding to $-x_{max}$ and $\frac{N}{2}$ as the rightmost pixel, corresponding to x_{max} . The equations of x_{max} and y_{max} can then be written as:

$$\begin{aligned} x_{max} &= r \cos(\theta_a) \\ y_{max} &= r \sin(\theta_a). \end{aligned} \quad (6.25)$$

Using the relationship between x_{max} and the total number of pixels given above, the

coordinates of a point on the line $y = y_{obj}$ is:

$$\begin{aligned} x(n) &= r \cos(\theta_a) \left(\frac{2n}{N}\right) \\ y(n) &= r \sin(\theta_a), \end{aligned} \tag{6.26}$$

where n is the horizontal pixel position of the point. Using this, the angle θ_{obj} can be found:

$$\begin{aligned} \tan(\theta_{obj}) &= m_{obj} = \frac{y_{obj}}{x_{obj}} = \frac{\sin(\theta_a)}{\cos(\theta_a)\left(\frac{2n}{N}\right)} = \frac{N \tan(\theta_a)}{2n} \\ \theta_{obj} &= \tan^{-1} \left(\frac{N \tan(\theta_a)}{2n} \right). \end{aligned} \tag{6.27}$$

Figure 6.20 shows the setup extended to two cameras. This time, the origin is centered between the two cameras separated by a distance d . This distance d is also known as the *baseline*. The slopes of the two lines, $m_{obj,1}$ and $m_{obj,2}$ are found using Equation 6.27.

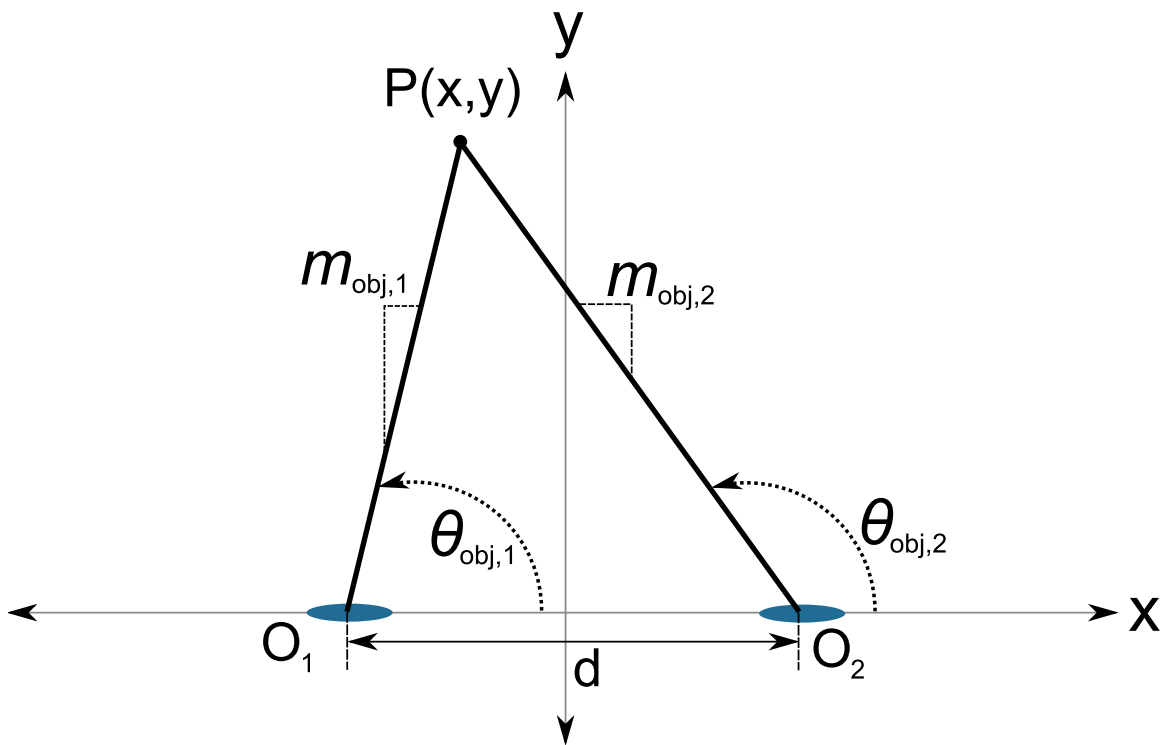


Figure 6.20: Dual Cameras

The two line segments are defined as:

$$y_i = m_{obj,i}x + b_i \quad i = [1, 2]. \quad (6.28)$$

The intercepts are calculated to be:

$$b_i = \frac{m_{obj,i}d}{2}. \quad (6.29)$$

Because these two lines intersect, the coordinates (x_{obj}, y_{obj}) of point P can then be found:

$$\begin{aligned} x_{obj} &= \frac{b_2 - b_1}{m_1 - m_2} \\ y_{obj} &= m_{obj,i}x_{obj} + b_i. \end{aligned} \quad (6.30)$$

These give the x and y coordinates of a point in one horizontal plane. Extending these results to three dimensions is analogous to the previous calculations, but with the vertical field of view $(\theta_{fov,v})$, the vertical number of pixels V , and the vertical pixel position, v (see Figure 6.21). The vertical slope, $m_{obj,v}$ is given by;

$$m_{obj,v} = \frac{V \tan(\theta_{a,v})}{2v}. \quad (6.31)$$

The z coordinate is then given by:

$$z_{actual} = m_{obj,v} \left(\sqrt{x_{obj}^2 + y_{obj}^2} \right). \quad (6.32)$$

The x coordinate remains unchanged, but the y coordinate must be adjusted due to the angle $\theta_{obj,z}$:

$$\begin{aligned} x_{actual} &= x_{obj} \\ y_{actual} &= y_{obj} \cos(m_{obj,v}). \end{aligned} \quad (6.33)$$

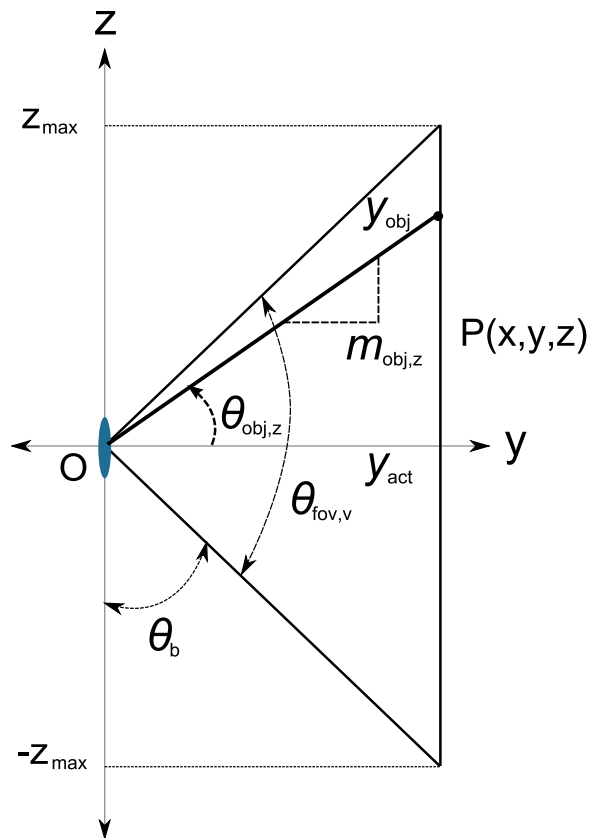


Figure 6.21: Stereo Vertical Adjustment

These results are only valid for a system in which the optical axes of the cameras are parallel, and the cameras are placed at $y = 0, z = 0$. Due to time constraints and the complexity of stereo correspondence algorithms, it was decided to use an existing algorithm designed by Psota [90], rather than designing an algorithm specific to this application. In future development of this system, this algorithm would be replaced with an approach customized to this problem to allow for real-time performance.

6.2.4 Software

To integrate the tracker, stereo vision algorithm, and PID controller, a Windows based graphical user interface (GUI) was developed by the author. This interface built on the existing AB3 control software and is based in one main dialog, shown in Figure

6.22. The “Tracker Control” area of the main dialog is the user interface to the tracker. The program allows for live tracking, using images captured from a framegrabber or off-line tracking, using previously captured images. The image source is set by a radio button selection. When operating in off-line mode, the user must choose a sequence of files to use for tracking, by clicking the *Choose File...* button. The filename must be part of a sequence of images, with a suffix of *_xx*, where *xx* is a number. When a file is selected, the file name and index are displayed on screen. To update the tracker, buttons marked *+* and *-* are used. These load the next image and previous image files, respectfully, and update the tracker state. There are several controls for adjusting the color tracker. The *Hue Low*, *Hue High*, *Sat Low*, *Sat High* fields define the hue and saturation ranges used to calculate the mask images shown in Figure 6.8. The default values are loaded from a configuration file at program startup. The *Update HS* button enables the user to manually adjust the hue and saturation ranges. The *Color Calibrate* button enables a guided color calibration method to find the hue and saturation ranges. An overhead frame capture is presented to the user, who manually selects areas (using the mouse) in the image corresponding to the tracking markers. When area selection is complete, all selected pixels are combined, and the program calculates the range of hue and saturation represented by the user selections and updates the values in the program. The *Position Calibrate* button allows the user to interactively enter the position of the tracker markers. The current overhead capture is presented to the user, who clicks on each point in specific order. This updates the trackers current “guess” of the marker positions. The current marker positions are also displayed in the field below the *Update HS* button. The *Show Capture* button displays the current overhead capture image. The *Start Tracking* starts live tracking of end effector positions updated at 10 Hz when using the framegrabber. The *AB3 Meas* button allows the user to fine tune the device dimensional constants used in

calculating end effector position.

The “Controller Parameters” area allows user interaction with the PID controller. The constants P, I, D, I_{max} , and E_{max} can be changed by the user at any time. These default values are loaded at startup from a configuration file. A static field displaying the current position of each joint is also displayed. These fields are updated while the controller is operating. Below the current position fields are user editable fields to enter the desired joint positions. *Home* position buttons move all joints back to their default positions. Step cursor buttons enable the user to move the joints by 10 mm (wrist joints) or 5° (shoulder joints) under PID control. The *Controller State* indicator displays when the controller thread is active and the *Controller Fault* indicates when a controller error condition has occurred. The controller can be started and stopped with the *Start Move* and *Stop Controller* buttons, respectfully. The gripper actuator is opened and closed via two buttons.

The stereo vision component is also controlled from this dialog. The *Set End Eff Target* button presents an overhead capture to the user, who clicks on the image to store the desired end effector positions. The *Find Target Psota Method* button presents the user with a left camera image. The user clicks on the desired target point, and the algorithm calculates the three dimensional position. After calculation is complete, the program prompts the user for the desired end effector. The *Find Target Manual Method* button presents the left and right images to the user. The user must manually find and click on the corresponding points in the images. The program uses the pixel positions to calculate the point in three dimensional space.

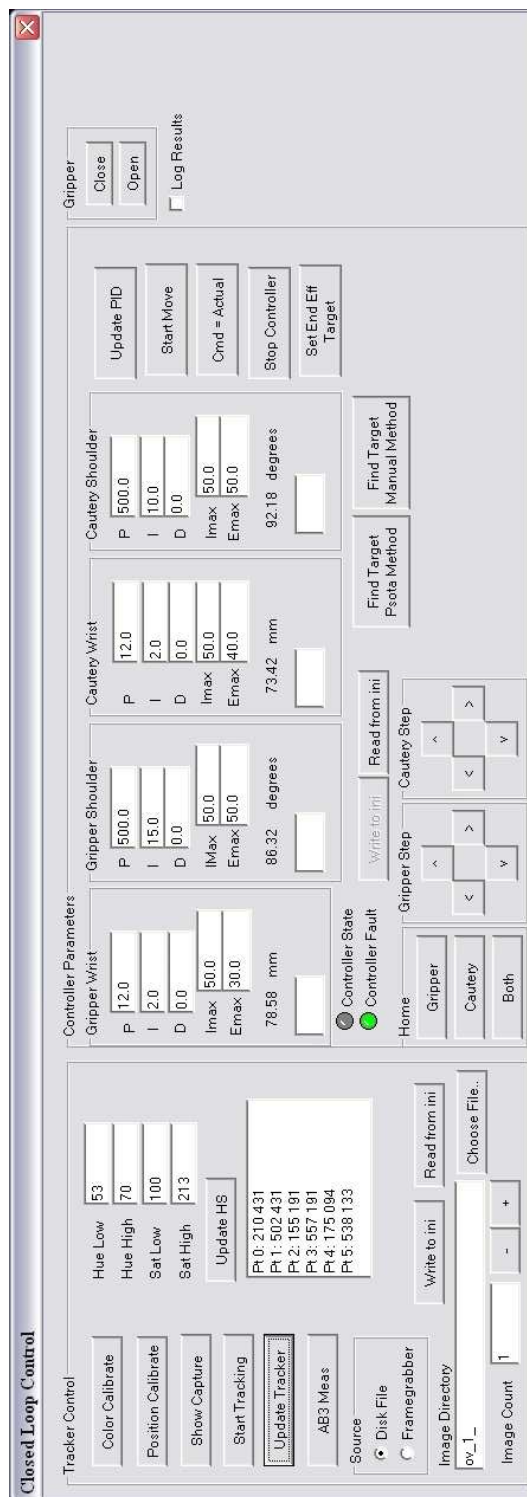


Figure 6.22: Supervised Autonomous Task User Interface

The PID controller, stereo algorithm interface, and tracker are all implemented in the *digitalController* class. A collaboration diagram of this class is shown in Figure 6.23. The main dialog class of the GUI instantiates a *digitalController* object.

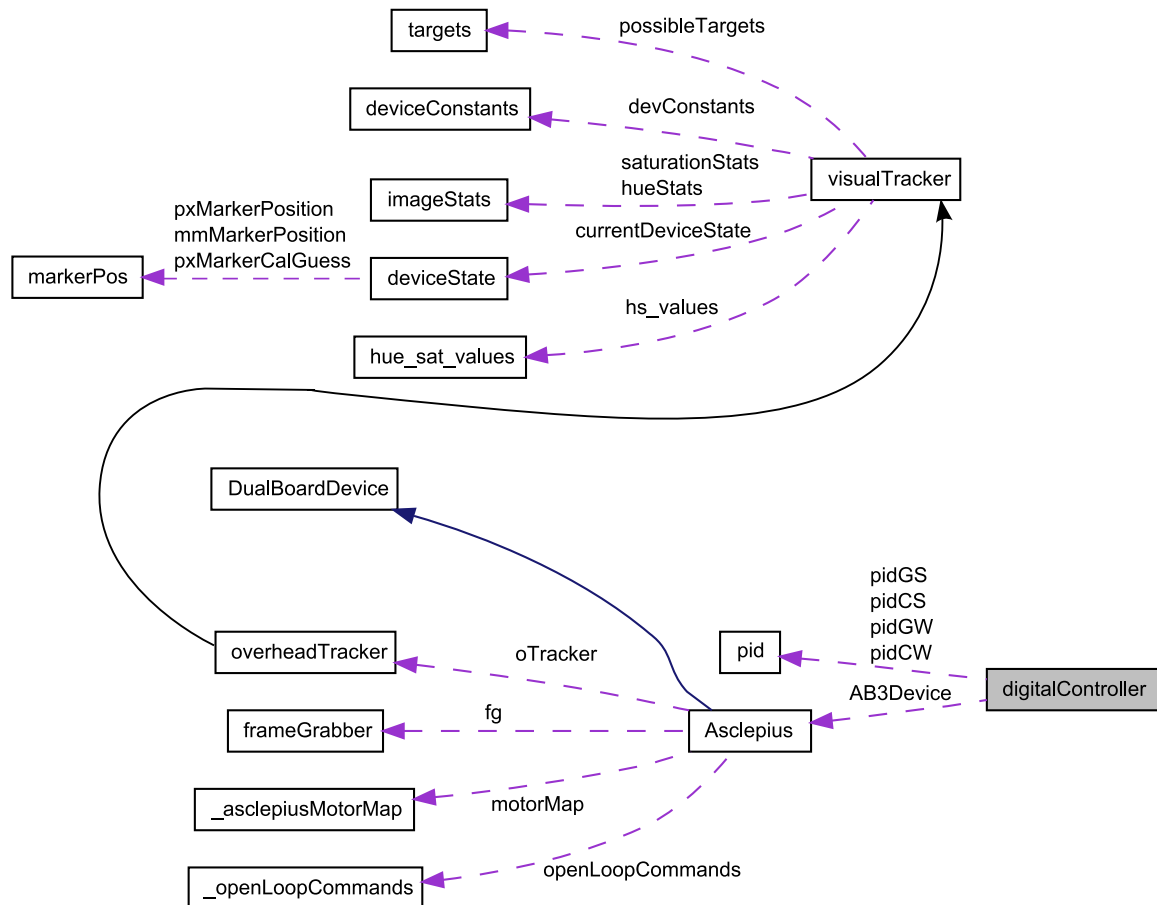


Figure 6.23: *digitalController* Collaboration Diagram

This class implements functions for control law calculations for all joints, as well as worker thread to implement the control loop. The *digitalController* object contains a member, *AB3Device*, a pointer to an *Asclepius* object. The *Asclepius* object is the interface to the AB3 device. The *digitalController* object also contains four *pid* member objects, one for each joint. The *pid* object stores all values needed by the PID controller, including constants, current position, error, command position, and gain terms. The *Asclepius* class has member attributes of *overheadTracker*, *frameGrabber*,

_asclepiusMotorMap, and *_openLoopCommands* objects. The *_openLoopCommands* class stores data for open loop control, and is not used with closed loop PID control. The *_asclepiusMotorMap* contains joystick mapping information used when controlling the AB3 device with the Virtual Incision controller. The *frameGrabber* class wraps the Sensoray framegrabber library in a C++ class. It supports up to three simultaneous framegrabber boards. The *overheadTracker* class, which is derived from the *visualTracker* class, implement all tracking functions.

6.3 Results

Several benchtop tests of the system were conducted. The robot was placed in a mount to keep it stationary, while a piece of rubber band was placed in a separate mount in the middle of the robot's workspace to simulate a piece of tissue. A digital camcorder was mounted above the robot and tissue model. The overhead view of the setup can be seen in Figure 6.24.

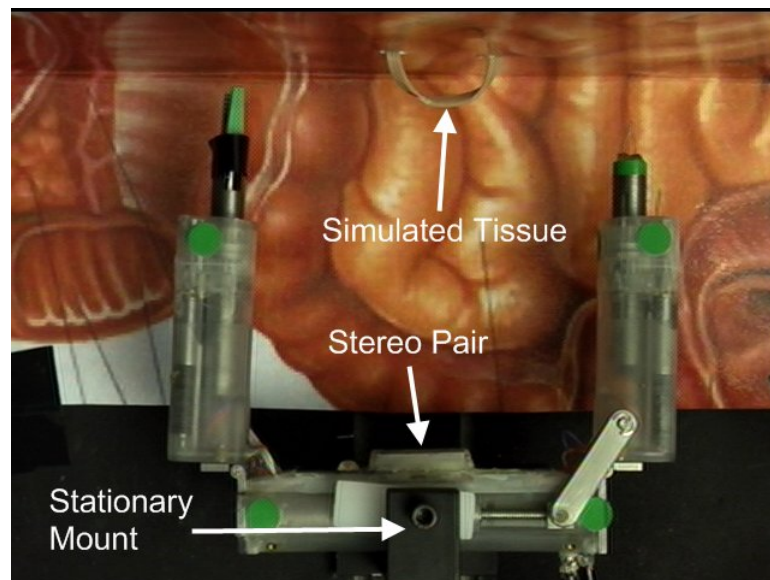


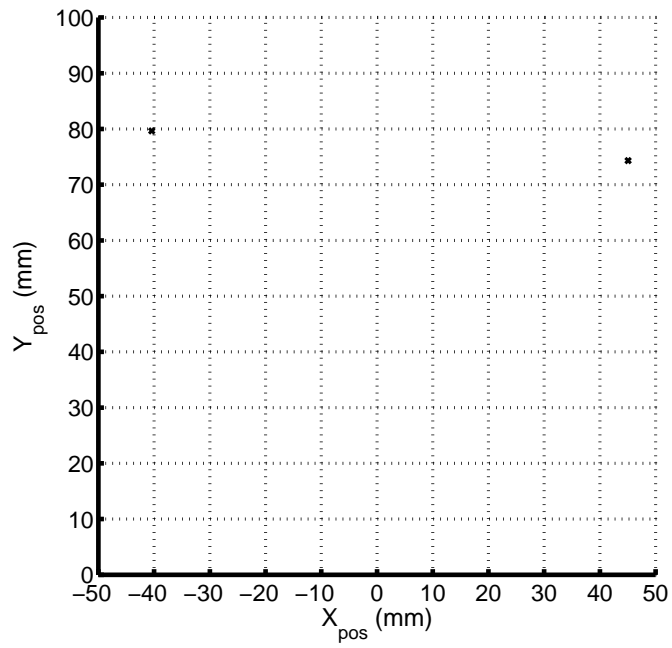
Figure 6.24: Supervised Autonomous Task Setup

The benchtop tests consisted of a simulated stretch and dissect task. The steps in this task are listed below, with end effector position plots and camera views following in Figures 6.25-6.31. The graphs marked “End Effector Paths” are plots of the calculated end effector positions at each time step. Those graphs with only a single dot at the end effector position represent steps where that particular joint did not move.

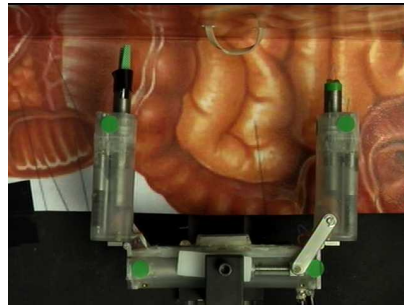
1. User selected location on tissue model to grasp by clicking on red mark in Figure 6.25(c). Stereo matching algorithm calculates final position and displays to user for confirmation prior to movement.
2. Robot moved gripper arm to selected location, position plot in Figure 6.26(a), final position in Figures 6.26(b)-6.26(d).
3. User manually closed gripper from a on-screen button in the GUI. This joint is controlled in an open-loop velocity mode, and care must be taken not to over extend the joint in either direction.
4. User retracts gripper arm using a control in the GUI which translates the joint by a set distance. This stretches tissue to prepare for dissection. See Figure 6.27(a).
5. User selected point to cut with cautery arm, and stereo matching algorithm calculates final position with user confirmation. Cautery arm then moves into position (Figure 6.28(a)).
6. With the cautery arm in the correct position, the dissection would occur here, but the cautery device was not functional at the device at the time of testing.
7. Cautery arm retracted after simulated dissection, as in Step 4, see Figure 6.29.

8. User extended gripper arm back to original position, as in Step 4, see Figure 6.30.
9. User manually opens gripper and releases tissue, as in Step 3.
10. User moves arms back to starting position, by clicking “Home joints” control in the GUI, Figure 6.31.

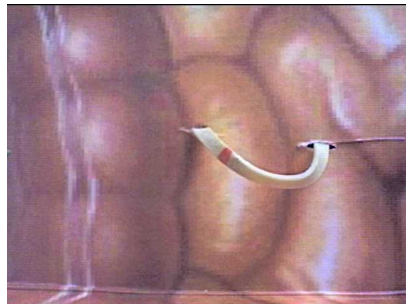
All images and data were saved for later analysis. Figure 6.32 shows positional data from the cautery end effector during the movement in step 6 of the task, calculated after the test from saved data. A graph of the tracker positional error over this movement is shown in Figure 6.33. The maximum measured error during tracking was 0.7 mm, with a mean error of 0.3mm. These figures are typical of all movements over both arms.



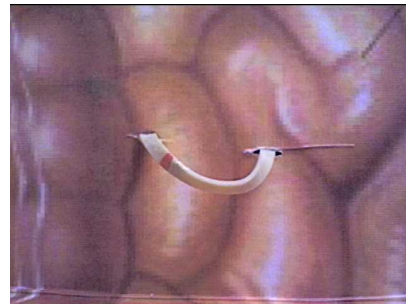
(a) End Effector Paths



(b) Overhead View

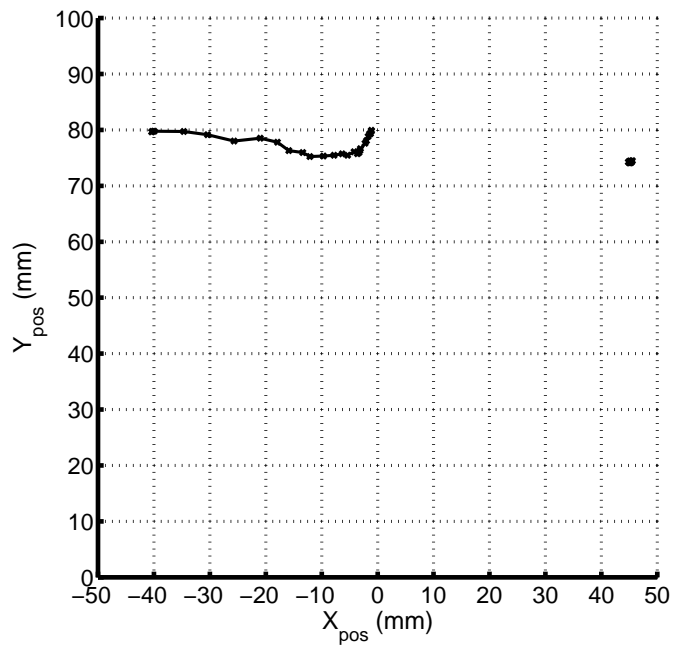


(c) Left View

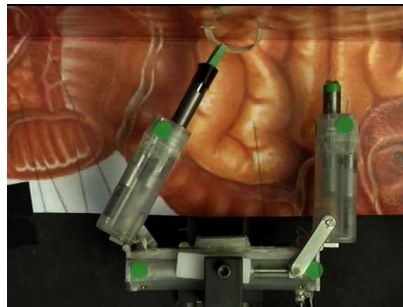


(d) Right View

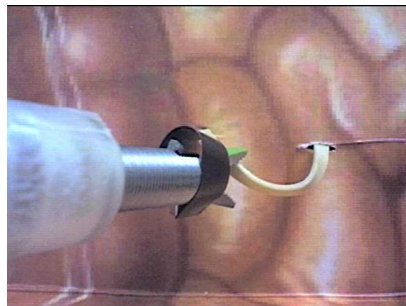
Figure 6.25: Task Start



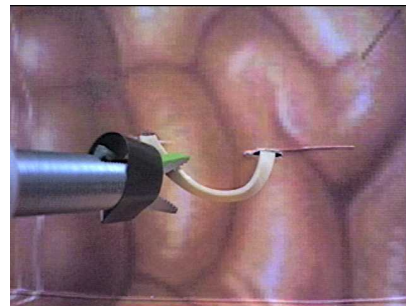
(a) End Effector Paths



(b) Overhead View

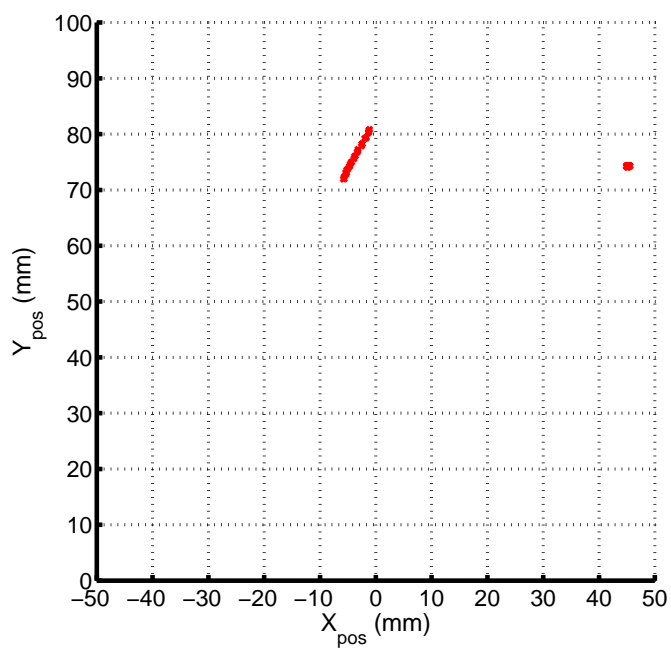


(c) Left View

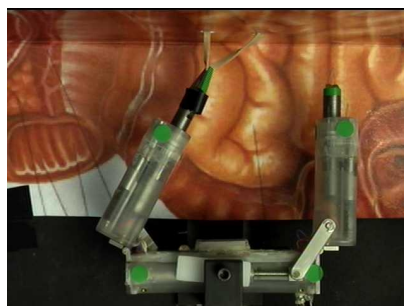


(d) Right View

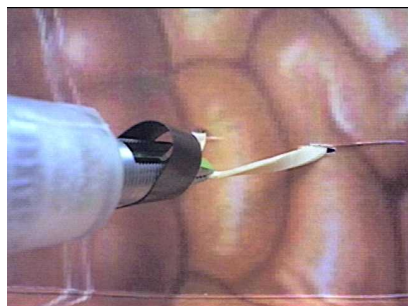
Figure 6.26: Task Step 1



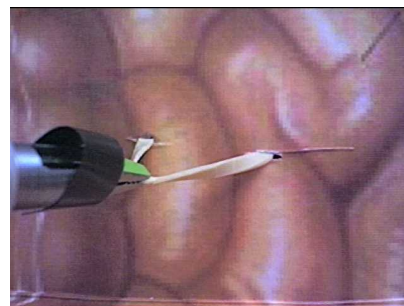
(a) End Effector Paths



(b) Overhead View

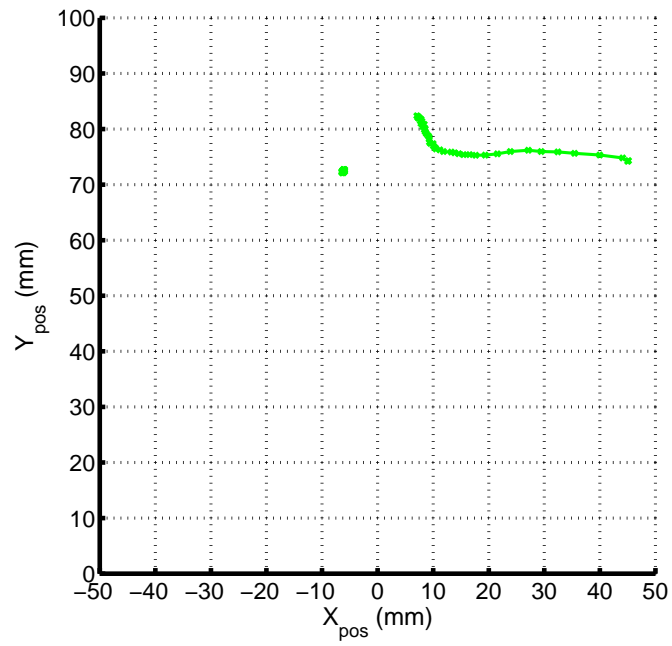


(c) Left View

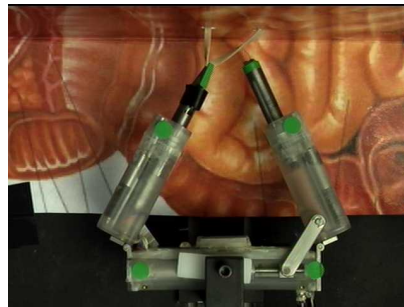


(d) Right View

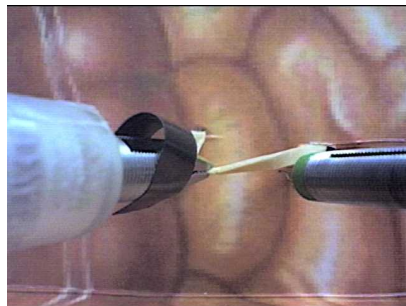
Figure 6.27: Task Step 2



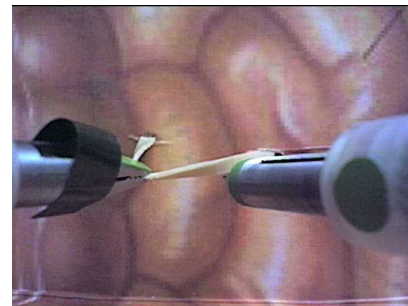
(a) End Effector Paths



(b) Overhead View

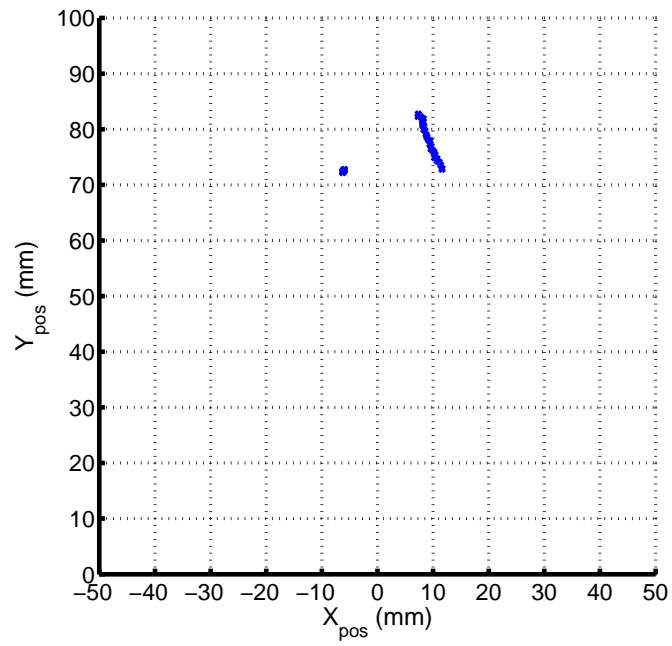


(c) Left View

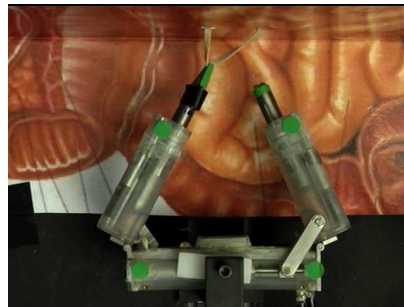


(d) Right View

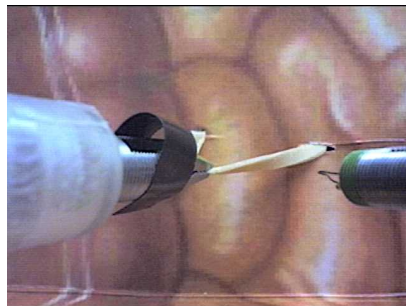
Figure 6.28: Task Step 3



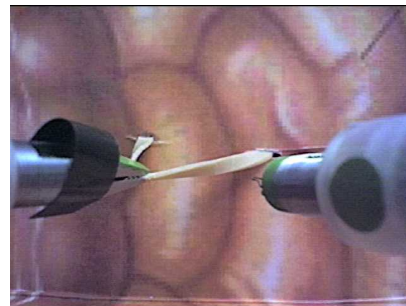
(a) End Effector Paths



(b) Overhead View

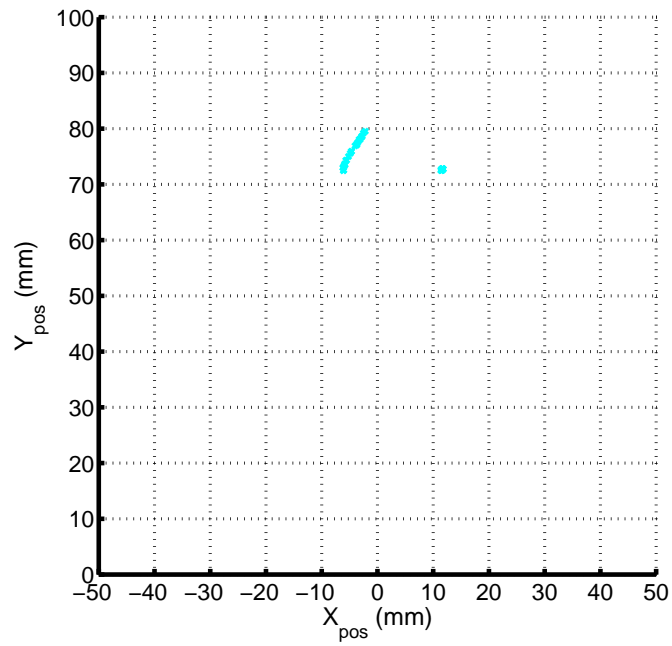


(c) Left View

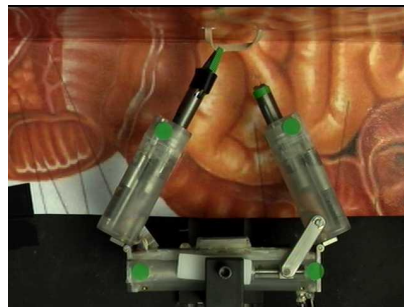


(d) Right View

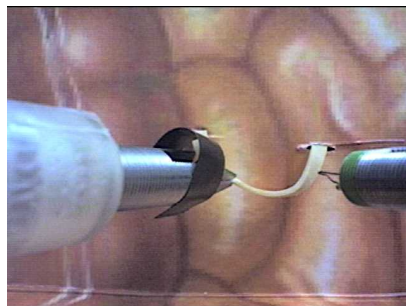
Figure 6.29: Task Step 4



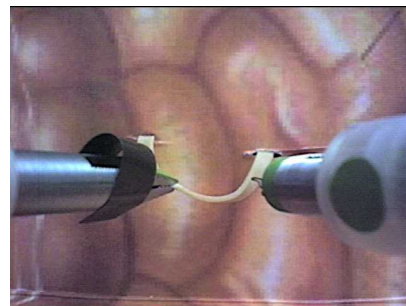
(a) End Effector Paths



(b) Overhead View

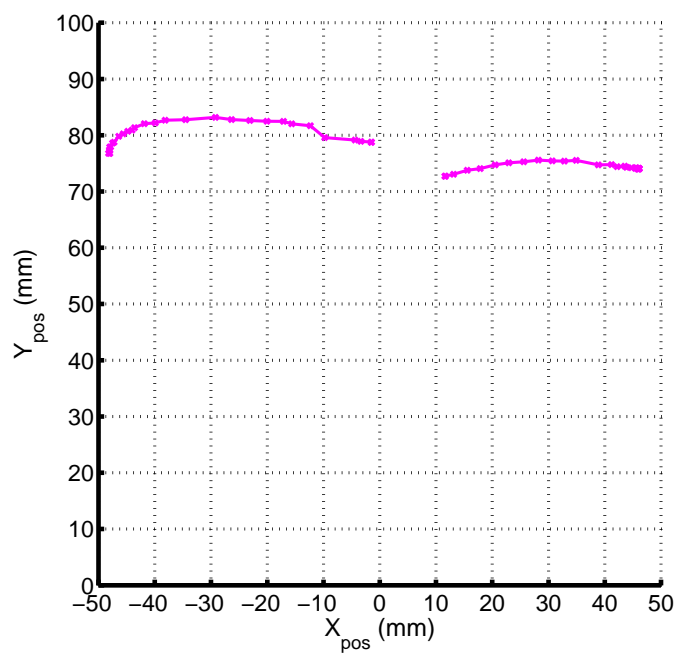


(c) Left View

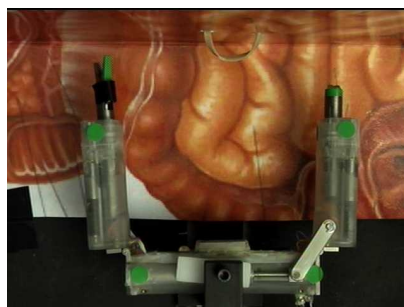


(d) Right View

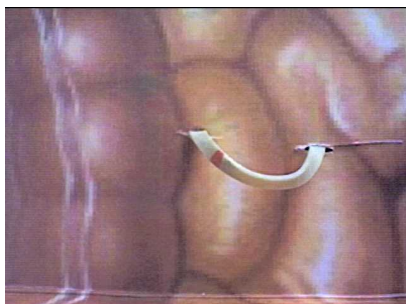
Figure 6.30: Task Step 5



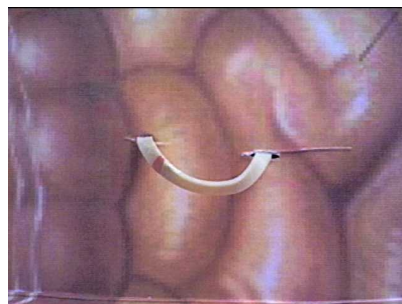
(a) End Effector Paths



(b) Overhead View



(c) Left View



(d) Right View

Figure 6.31: Task Step 6

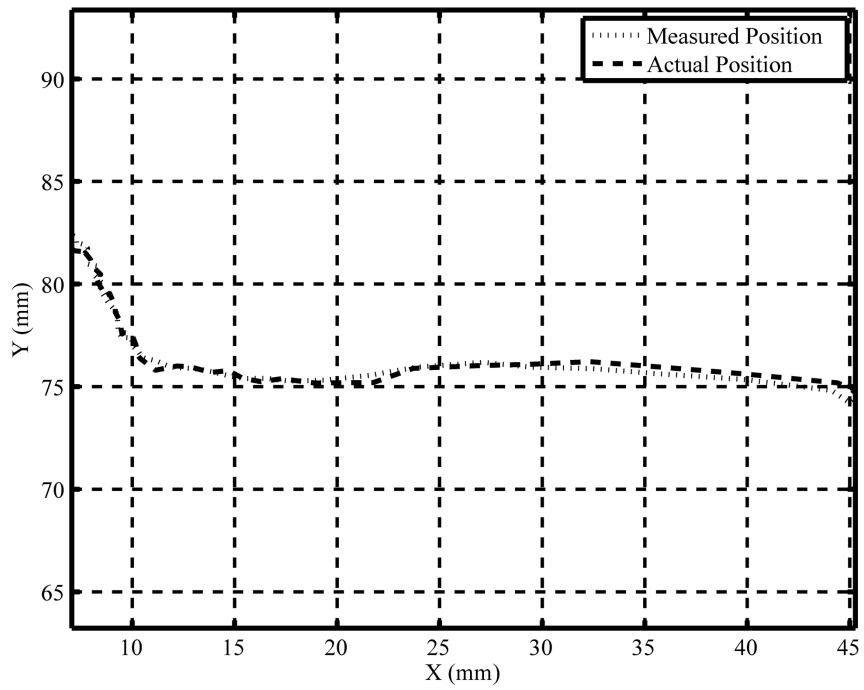


Figure 6.32: End Effector Position

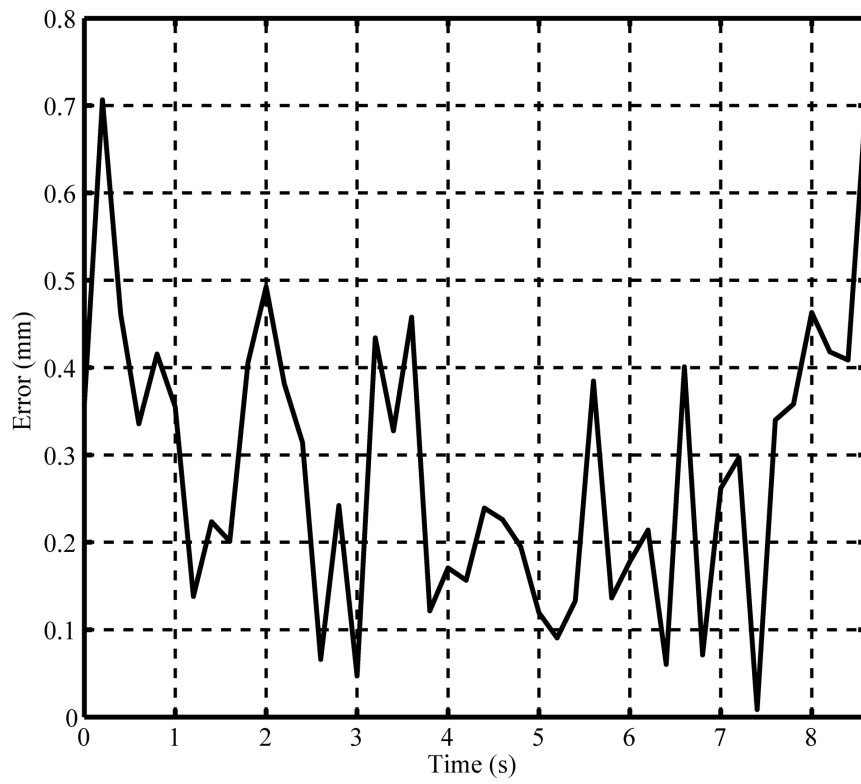


Figure 6.33: Positional Error

6.4 Summary

A completely *in vivo* surgical platform capable of completing supervised autonomous surgical tasks could be very useful in several situations where a patient is a long distance from a trained surgeon. A device such as this could be controlled remotely by a surgeon without the need for a low latency communications channel for real-time video. A prototype system was designed and tested around an existing robotics platform, the AB3 (See section 2.20). This system consisted of a visual tracker, a controller, and stereo vision. User interaction was through a custom designed GUI running on a standard PC. The user selects a point from one of the robots two cameras and the system automatically moves the specified end effector to that point. The tracker locates and tracks colored dots mounted on the robot, and calculates end effector positions in real time. A PID position controller was then designed for each joint. Each joint was characterized by performing open-loop step input tests and analyzing the position data recorded by the tracker. The PID parameters were manually tuned to optimize the controller performance. To locate points of interest that the robot will interact with, a stereo vision system is needed. This system can calculate three-dimensional coordinates from a pair of two-dimensional images using a stereo matching algorithm. To save time, an existing algorithm was used in the prototype system, but future development of the system will include designing a stereo vision specifically for this application. Benchtop tests were conducted using the prototype system. These tests were simulated stretch and dissect procedures, in which the robot (with minimal user interaction) grasps a piece of simulated tissue and dissects it with a cautery tool. The system performed well, but several areas will need to be addressed before a system such as this is used in an *in vivo* environment. These will be discussed in Chapter 7.

The work presented in this chapter, to the best knowledge of the author, represents

some of the first contributions to the field of surgical robotics in using *in vivo* robots to automate surgical tasks. There has been some research done in automating surgical tasks, but this research has used visually servoed external robots. This research includes automating tasks in keratoplasty (corneal shaping) by Zong et al. [44] and automating suture tying by Hynes et al. [43]. Similar research is studying the use of visual feedback to automate steering of medical needles in soft tissue [91].

Chapter 7

Conclusions and Future Work

The work outlined in this dissertation has shown that miniature *in vivo* robots can be used in minimally invasive surgical procedures for providing visual feedback, and in the near future, to be a completely intracorporeal surgical platform capable of performing supervised autonomous surgical tasks. The author was the sole electronics and software designer for 19 different *in vivo* robots described here. These robots varied in both form and function, and included mobile robots, fixed base cameras, wireless cameras, a mobile robot with biopsy capability, a prototype wireless crawler, ceiling pan-tilt cameras, and three types of dexterous manipulators. The design work for these included printed circuit board layout, firmware, interface, PC based control software, motor control, a digital controller, and a vision based real-time tracker.

To show that these *in vivo* cameras have visual quality comparable to standard laparoscopes, comparisons using several visual quality metrics were completed. These results show that a laparoscope has somewhat higher visual quality than the *in vivo* cameras, but it is believed that with improved optics and lighting, the *in vivo* cameras could have comparable visual quality. During experiments conducted in cooperation with NASA during the NEEMO 9 mission, an *in vivo* camera robot was compared

with a laparoscope as a visual feedback device for completing simulated laparoscopic surgical tasks. The results indicate that the *in vivo* camera is no worse, and possibly as good as a standard laparoscope, but further experiments of this kind will be needed to confirm this. A sterilizable *in vivo* camera robot was designed and built to show that these types of devices can be used safely in minimally invasive surgery. Three survivable animal surgeries were completed using a sterilized CPT camera device as sole visual feedback for a cholecystectomy. All three procedures were completed successfully, and the miniature robot caused no tissue damage or infection. Finally, a dexterous manipulator robot was used to demonstrate a proof-of-concept that *in vivo* surgical platforms could be used to perform supervised autonomous surgical tasks. Devices such as these could be used in emergency procedures, long term space flight, or anywhere the patient is not near a trained surgeon. A surgeon in a remote location could still use these devices to perform simple procedures, even if there is no low latency communication channel available. Finally, some suggestions for improvements and future work related to the topics previously discussed will now be given.

7.1 Statistical Comparisons

Chapter 4 described experiments done in cooperation with NASA on the NEEMO 9 mission. The goal of these experiments was to compare the usefulness of the *in vivo* robotic devices described here with a standard laparoscope. The NEEMO crew members all completed multiple tasks using each camera system for visual feedback. The analyses of these results indicate that there is no significant difference in performance metrics between the *in vivo* cameras and laparoscopes. However, the experiments suffered from low statistical power, due to the small number of participants. Several

suggestions for the design of future studies are given below:

1. Participants should be surgeons, because these are the people who will be using the devices in practice.
2. A larger sample size will be necessary to ensure adequate statistical power. A sample size of 50 participants would reduce the chance of a Type-II error to 20%.
3. The participants should be randomly assigned to perform tasks with the *in vivo* robot or the laparoscope; or if each participant uses both systems the order in which they are used should be randomly assigned.
4. Use a more sophisticated statistical model to analyze the data. Rather than the ANOVA models used here, an ANCOVA (analysis of covariance) model could be used. With this model, covariates such as amount of laparoscopic experience, could be factored into the analysis which would improve the statistical power of the results.

7.2 Supervised Autonomous Task Completion

The prototype system for supervised autonomous task completion developed in Chapter 6 worked well under benchtop conditions, but several aspects of it need to be improved before it is a viable *in vivo* solution. The overhead tracker that was implemented worked to develop the position controller, but is clearly not an acceptable long term solution, as there are several problems with the current implementation:

1. No “overhead” camera in an *in vivo* environment.
2. Need to initialize tracker with approximate positions of each marker.

3. Error due to inaccurate device measurements.

In a standalone *in vivo* system, the system would need to track joint positions using only views from the robot cameras, as no overhead camera would be present. Tracking from the robot view introduces new problems, including how to deal with occlusions when the joint is not in the field of view. Fortunately, this is a well studied area of computer vision, and many algorithms have been developed that are able to cope with occlusions [92] [46] [47]. In addition to a modified tracking algorithm, the locations of the device cameras could be designed in such a way that minimizes areas where the tracker markers are not visible. This would need to take into consideration other factors, including physical size and configuration of the device, and the camera's view of the operating environment.

The tracker also required initialization of the approximate starting positions of each marker. One possible way to eliminate this requirement would be to use different color markers for each joint. Looking back at Figures 6.4(b) and 6.5(b) pages 124-125, it can be seen that there is a large range of hue values that do not appear in the *in vivo* test images. In theory, several different colors of marker could be tracked in the same way the markers in this system were tracked.

The joint controllers described in Section 6.2.2 were shown to function adequately for the simple benchtop tests presented here. Before *in vivo* tests were attempted, they should be improved in several aspects. A global path planner should be added that could avoid collisions with other tools and unwanted tissue interaction. A controller should also be added for the gripper end effector to allow the software to control the opening-closing actuation of the end effector.

As mentioned previously, an existing stereo correspondence algorithm was used in the prototype system. In future systems, this should be replaced with a solution designed specifically for this application. The algorithm used here was accurate, but

too slow for use in a surgical environment. The new algorithm should be implemented in C++, possibly using the OpenCV library, which has several stereo vision library functions. The stereo vision system could also be used to improve the tracker. If a unique marker would be attached to each end effector, the color tracker previously described could locate the markers position in both the left and right images, the three-dimensional coordinates of the end effector could then be determined, and finally joint angles from that position. This is a simplification of the problem, and does not address such issues as end effector orientation or any of the inverse kinematic calculations that would be necessary to implement such a system. However, this implementation has one significant advantage not previously mentioned: tracking joint position visually and in real-time would eliminate the need for optical or magnetic encoders on the motors of a device. This has the potential to significantly reduce the size of such devices.

Appendix A

Electronics

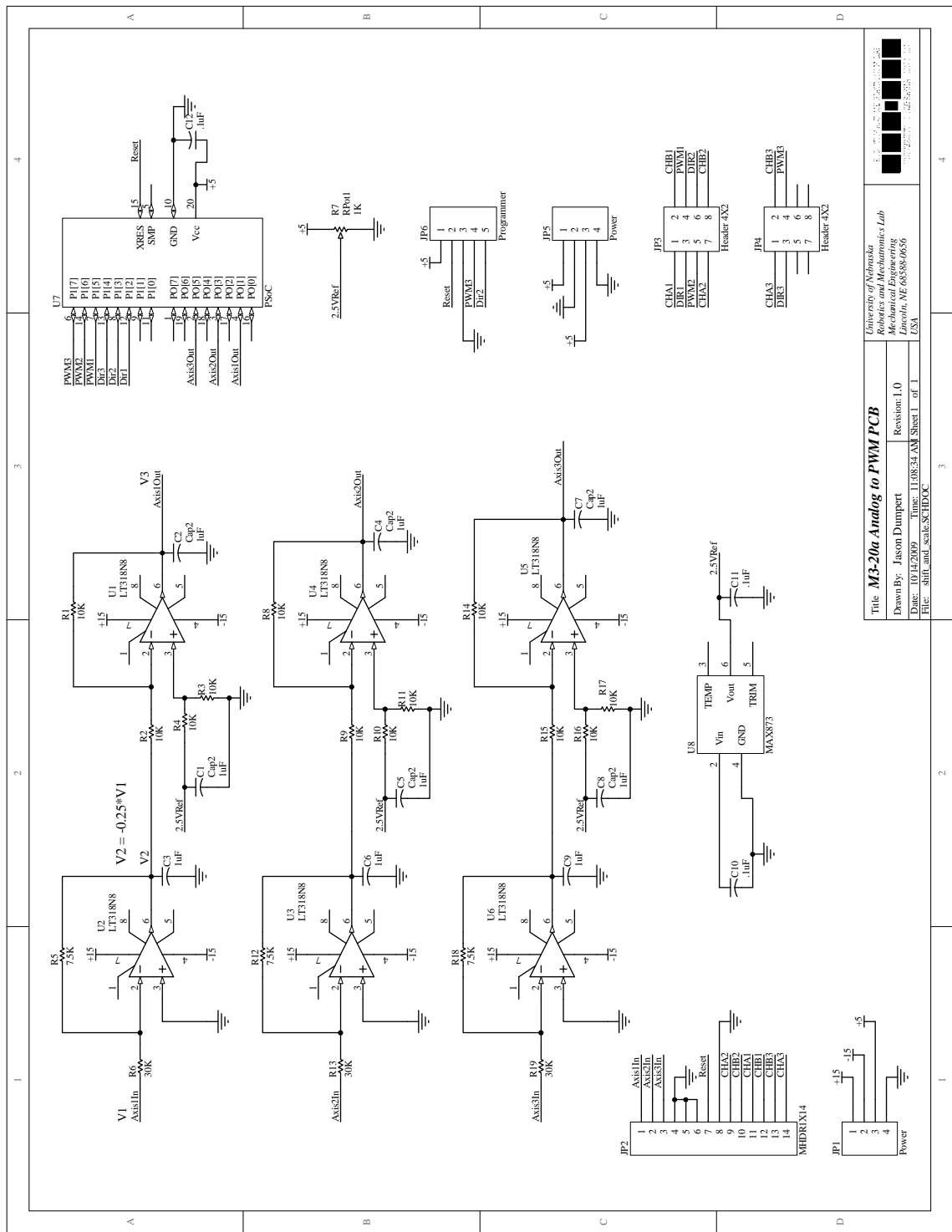


Figure A.1: M3-20a Analog to PWM PCB

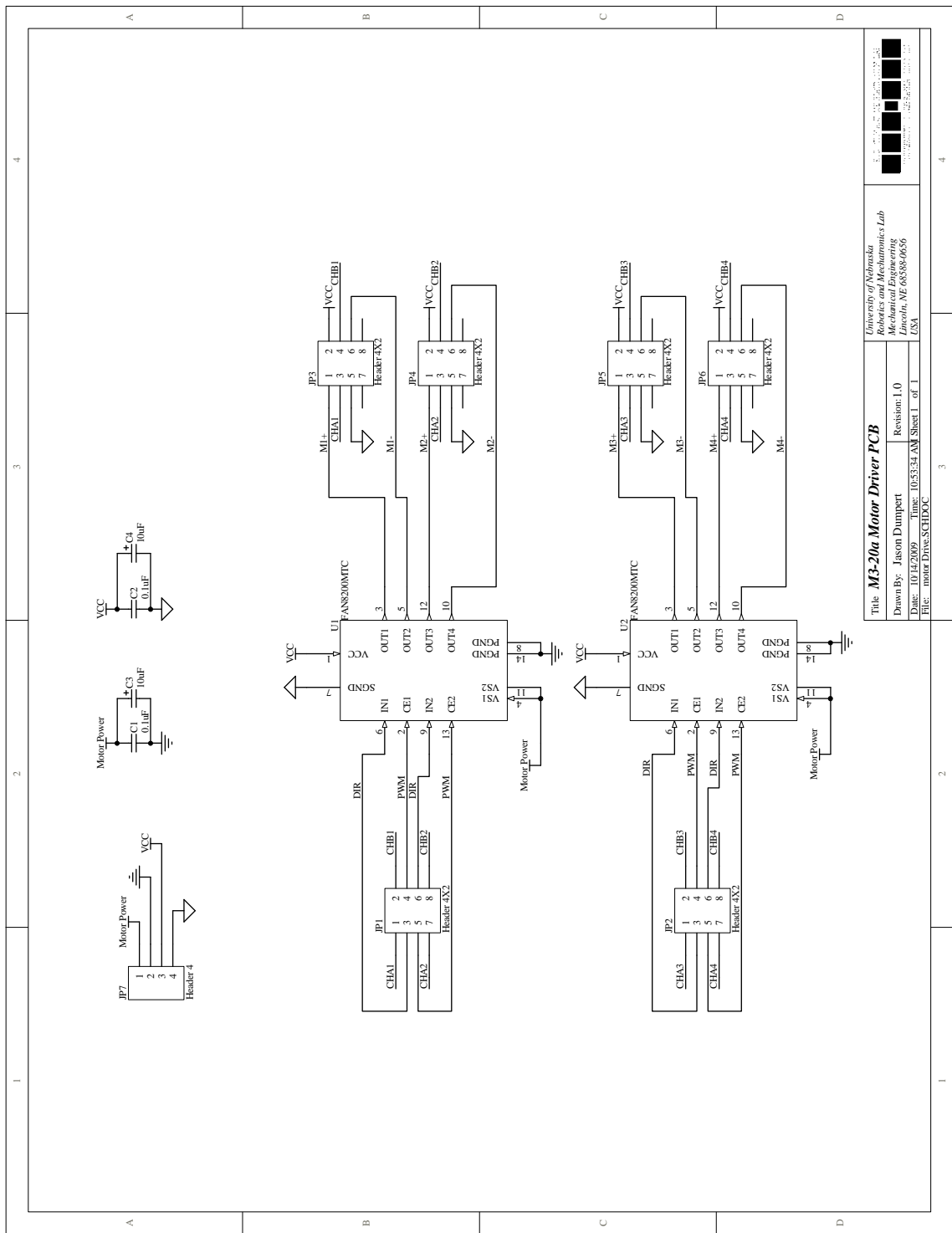
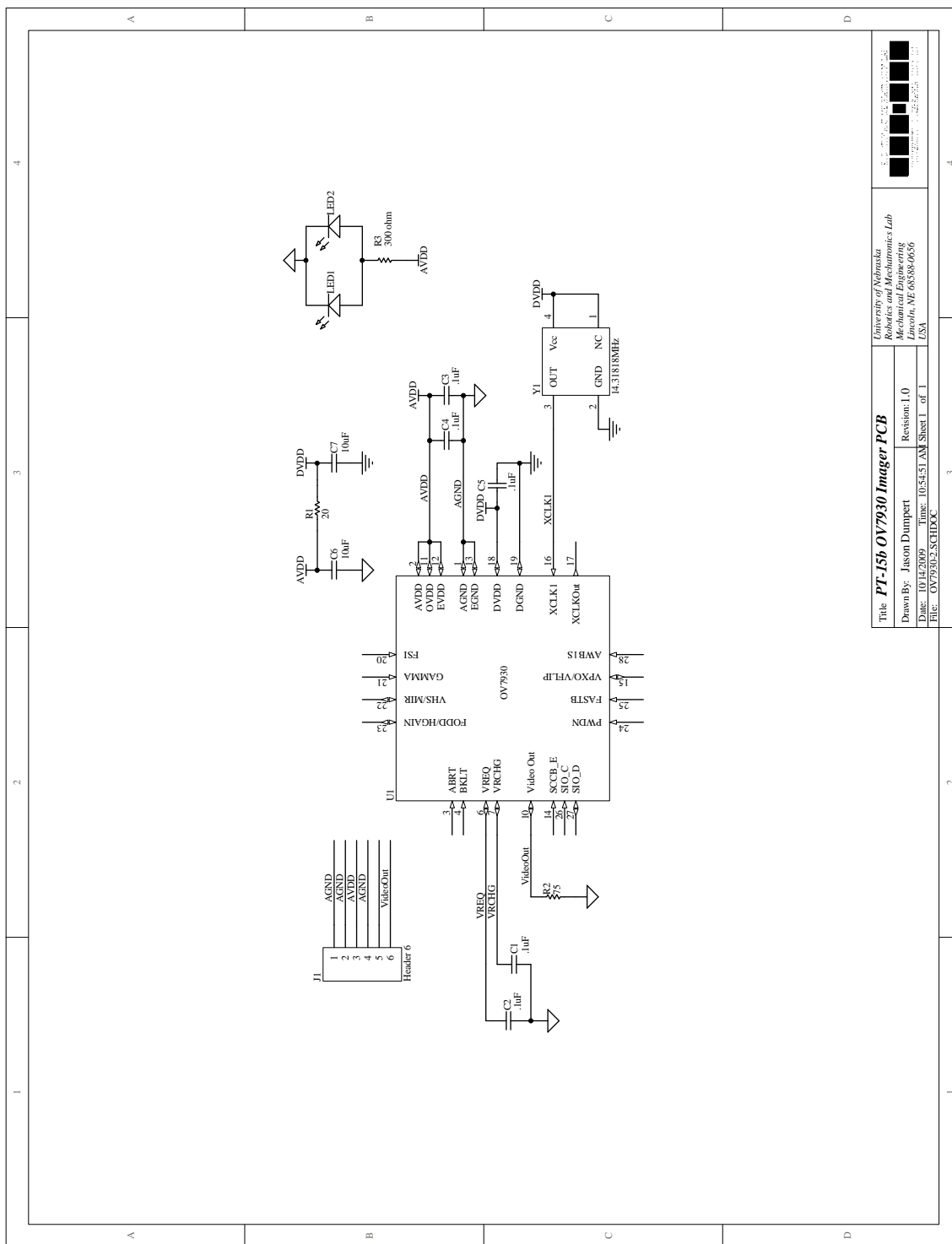


Figure A.2: M3-20a Motor Driver PCB



Title: PT-15b OV7930 Imager PCB

Drawn By: Jason Dummerp

Date: 10/12/2009

Time: 10:54:51 AM

Revision: 1.0

Sheet 1 of 1

File: OV79302.SCHDOC

University of Nebraska
Lincoln
Mechanical Engineering
Lincoln, NE 68588-0656
USA

Figure A.3: PT-15b Imager PCB

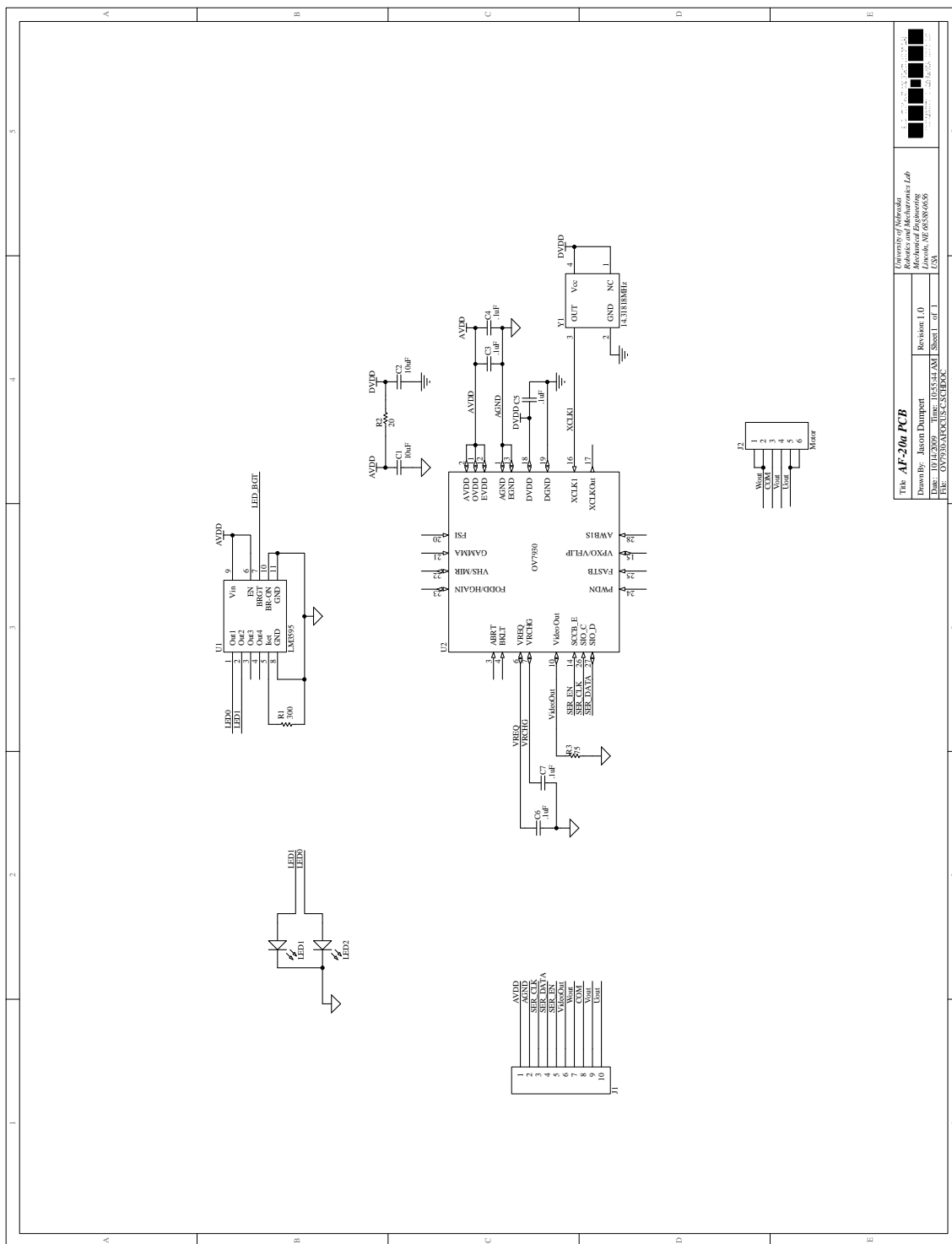
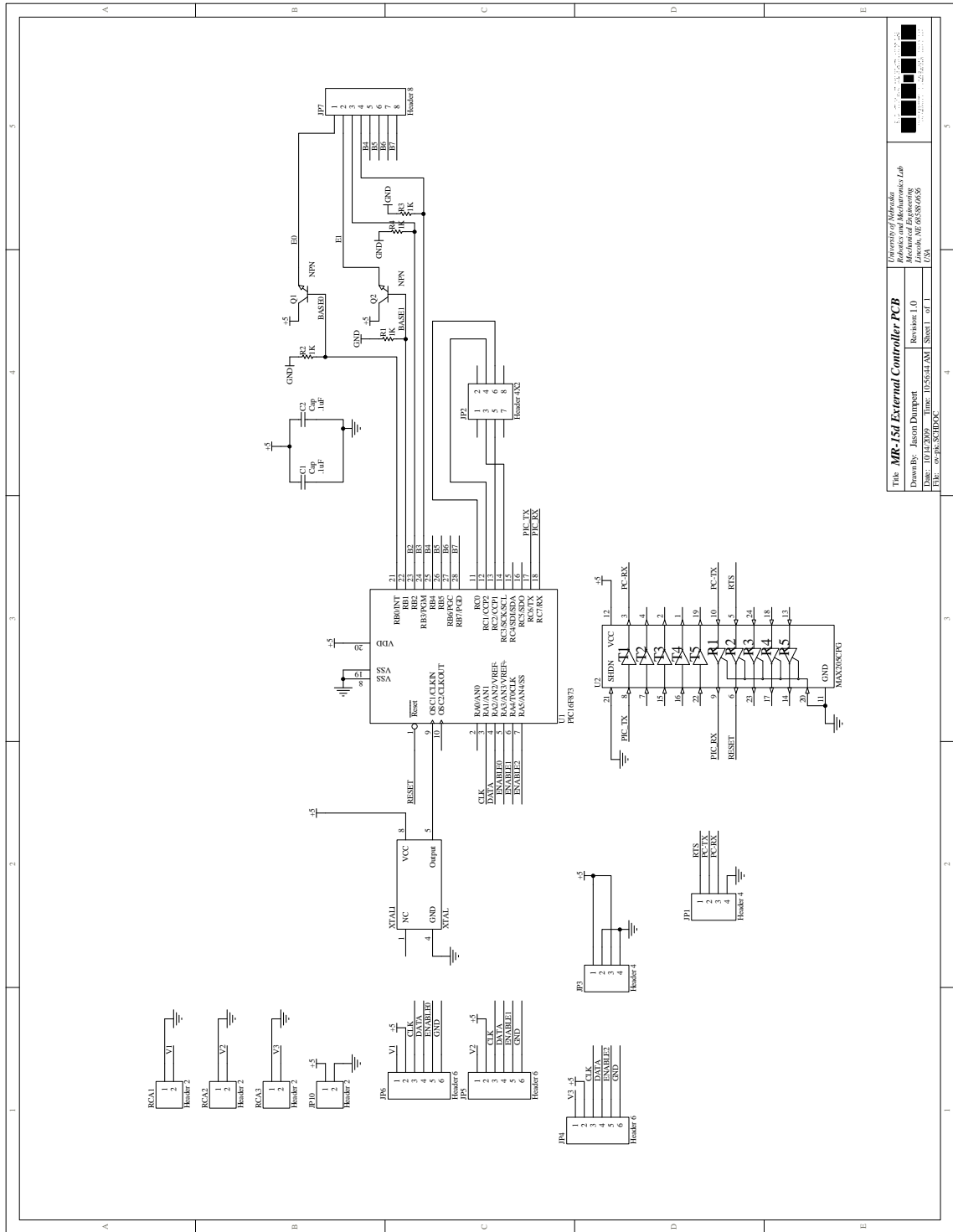


Figure A.4: AF-20a PCB



University of Nebraska
 Robotics and Mechatronics Lab
 Drawn By: Jason Dampert
 Date: 10/12/2009
 File: 1D-5634-AM
 Title: MR-15d External Controller PCB
 Revision: 1.0
 Sheet: 1 of 1
 USA

Figure A.5: MR-15d Controller PCB

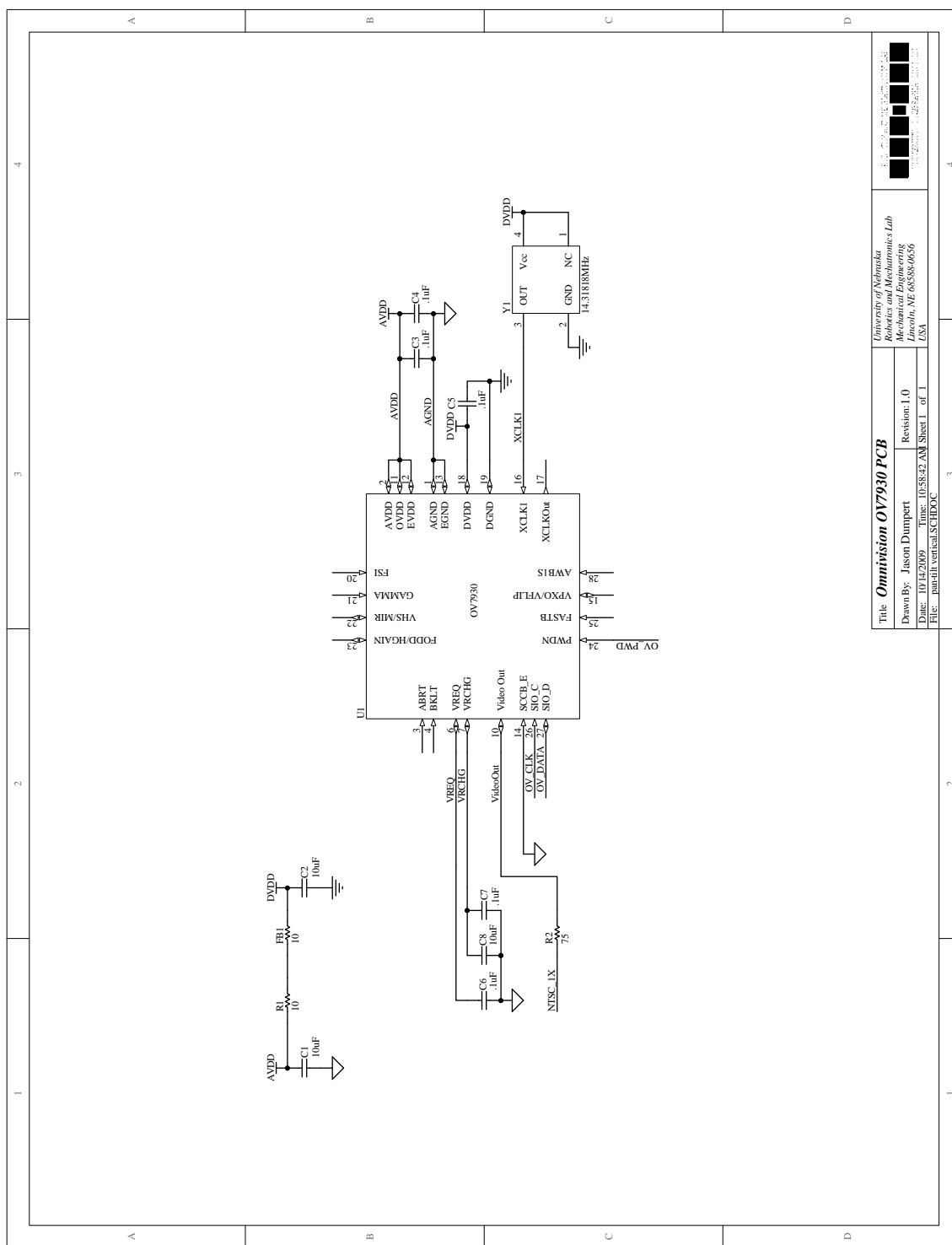
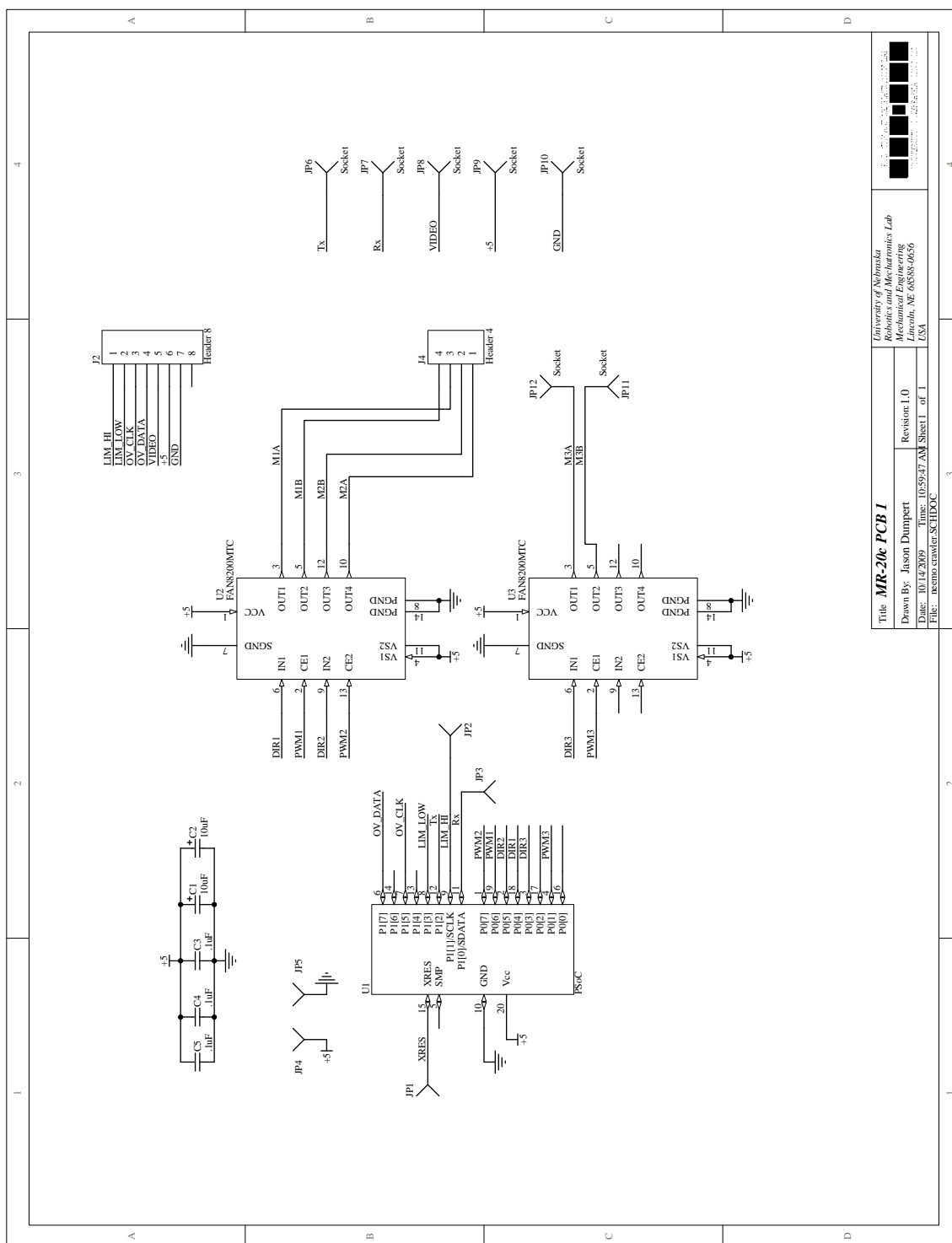


Figure A.6: OV7930 PCB

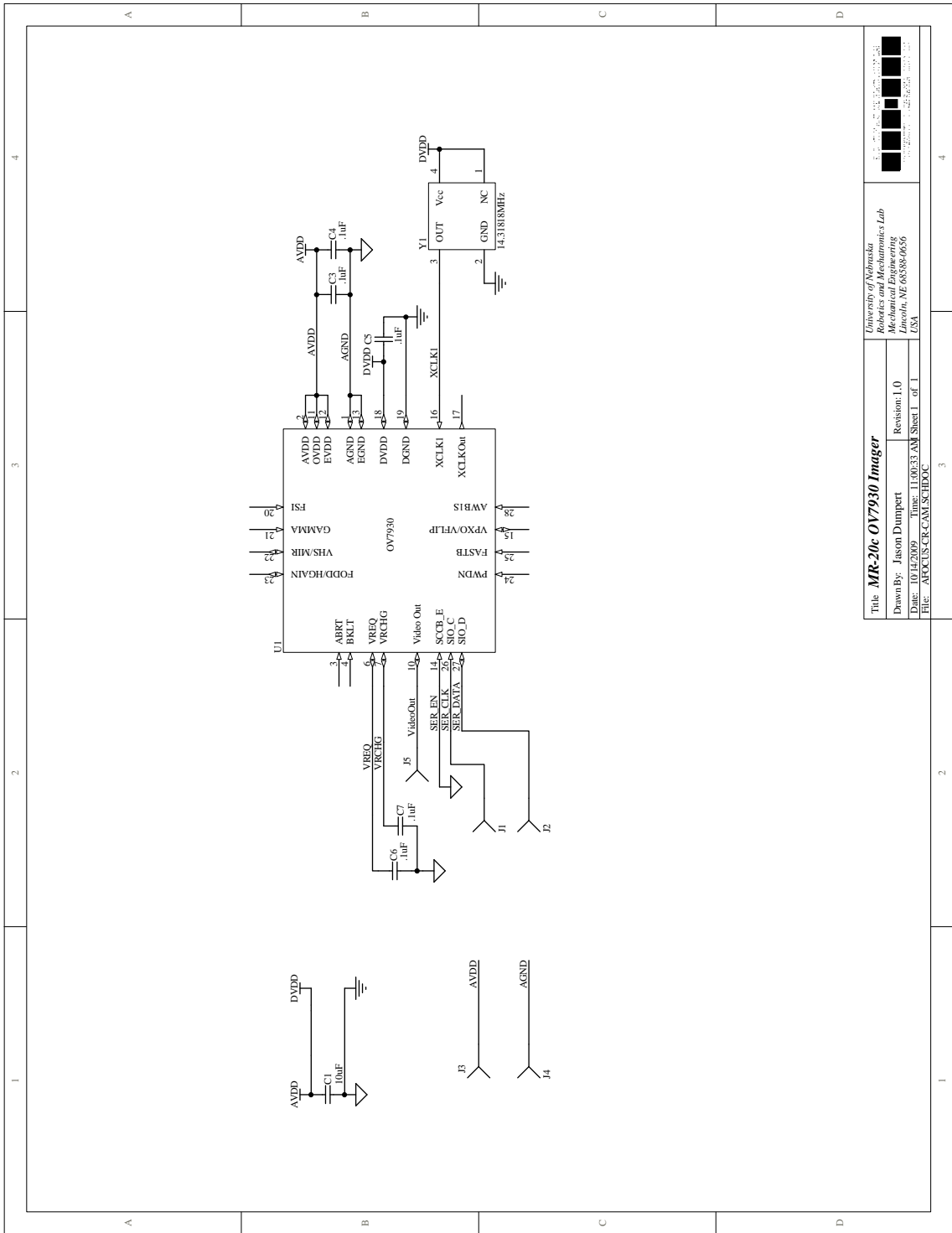


Title MR-20c PCB 1

Drawn By: Jason Dumpert
 Date: 10/14/2009
 Time: 10:59:47 AM
 Sheet 1 of 1
 File: memo crawler SCHDOC

University of Nebraska
 Mechanical Engineering Lab
 Lincoln, NE 68588-0656
 USA

Figure A.7: MR-20c Control PCB



Title: **MR-20c OV7930 Imager**
 Drawn By: Jason Dummerp
 Date: 10/12/2009
 File: AF0CUS_CIR_CAM_SCHDOC

University of Nebraska
 Mechanical Engineering Lab
 Lincoln, NE 68588-0656
 USA

Revision: 1.0
 Time: 11:00:33 AM
 Sheet 1 of 1

Figure A.8: MR-20c and T-15b Imager PCB

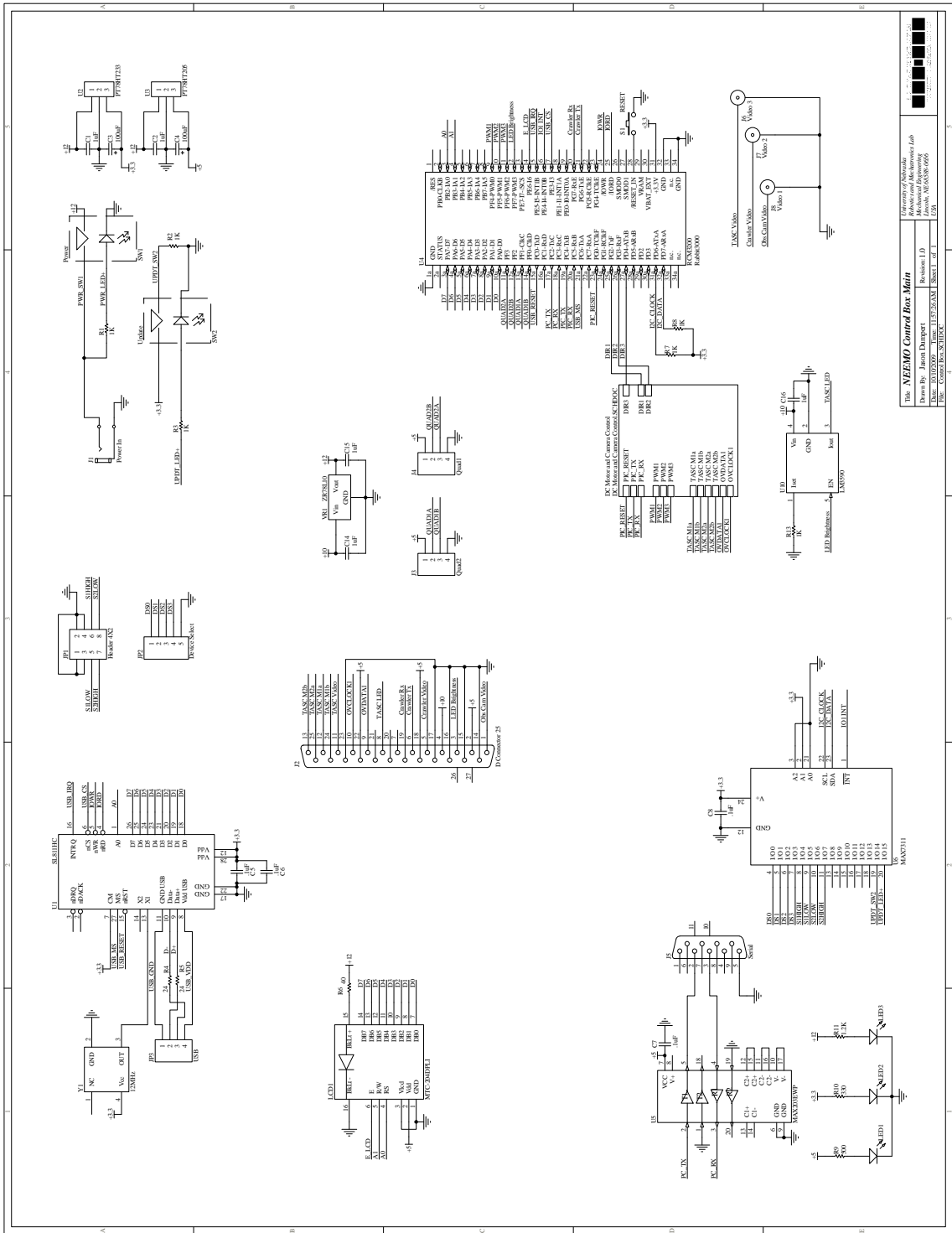
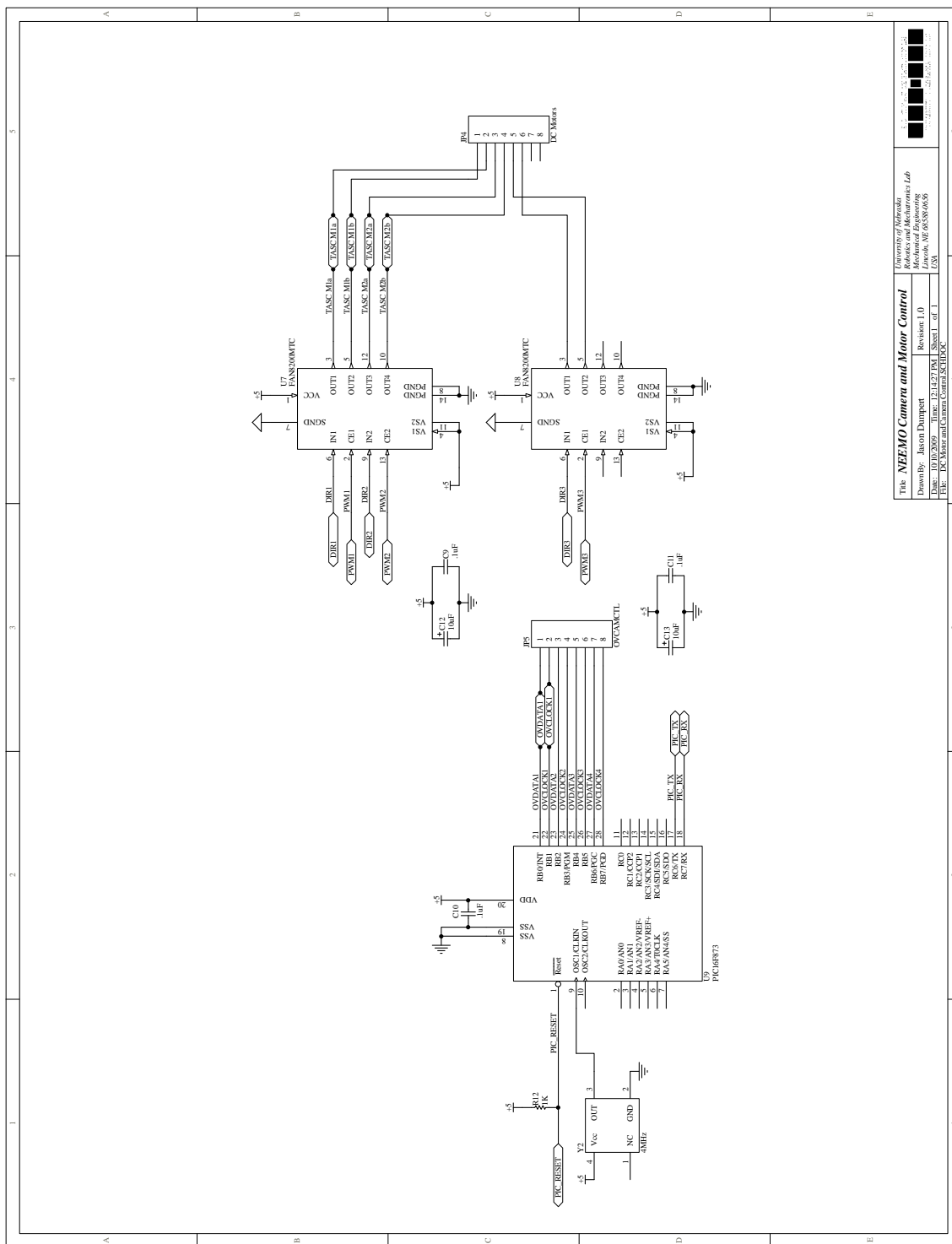


Figure A.9: NEEMO Controller Box PCB-Main

Fig. NEEMO Control Box Main
Revision: 1.0
Date: 01/2008
Author: J. B. J. van den Berg
Lab: University of Nebraska
Department of Electrical and Electronic Engineering
Lincoln, NE 68583-0805



| | |
|--|-------------------|
| University of Nebraska Robotics and Mechatronics Lab Lincoln, NE 68583-0655 USA | |
| Title: NEEMO Camera and Motor Control | Revision: 1.0 |
| Drawn by: Jacom Dampert | Sheet 1 of 1 |
| Date: 10/10/2009 | Time: 12:14:27 PM |
| File: DC_Motor and Camera Control.SCHDOC | |

Figure A.10: NEEMO Controller Box PCB-Imager and Motor Drive

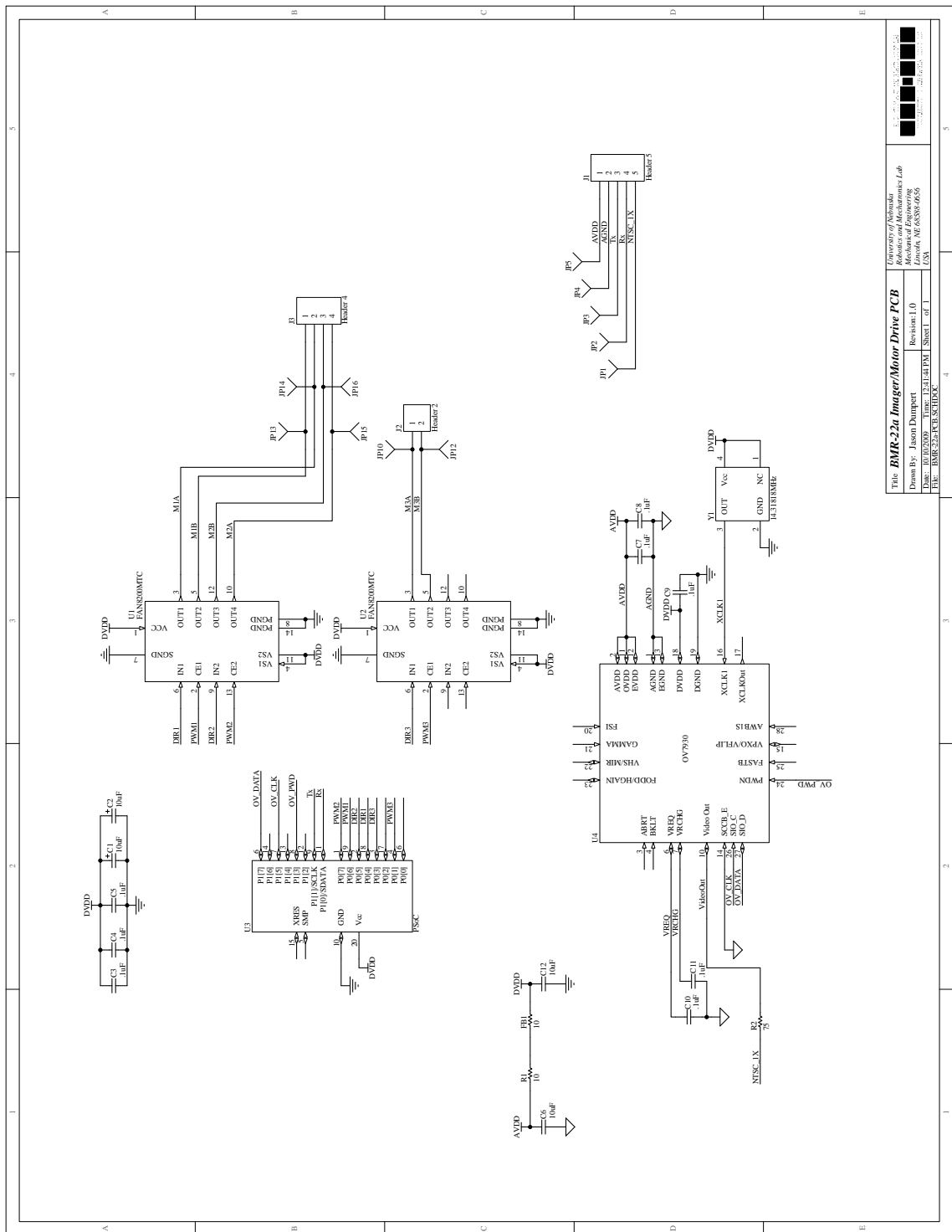
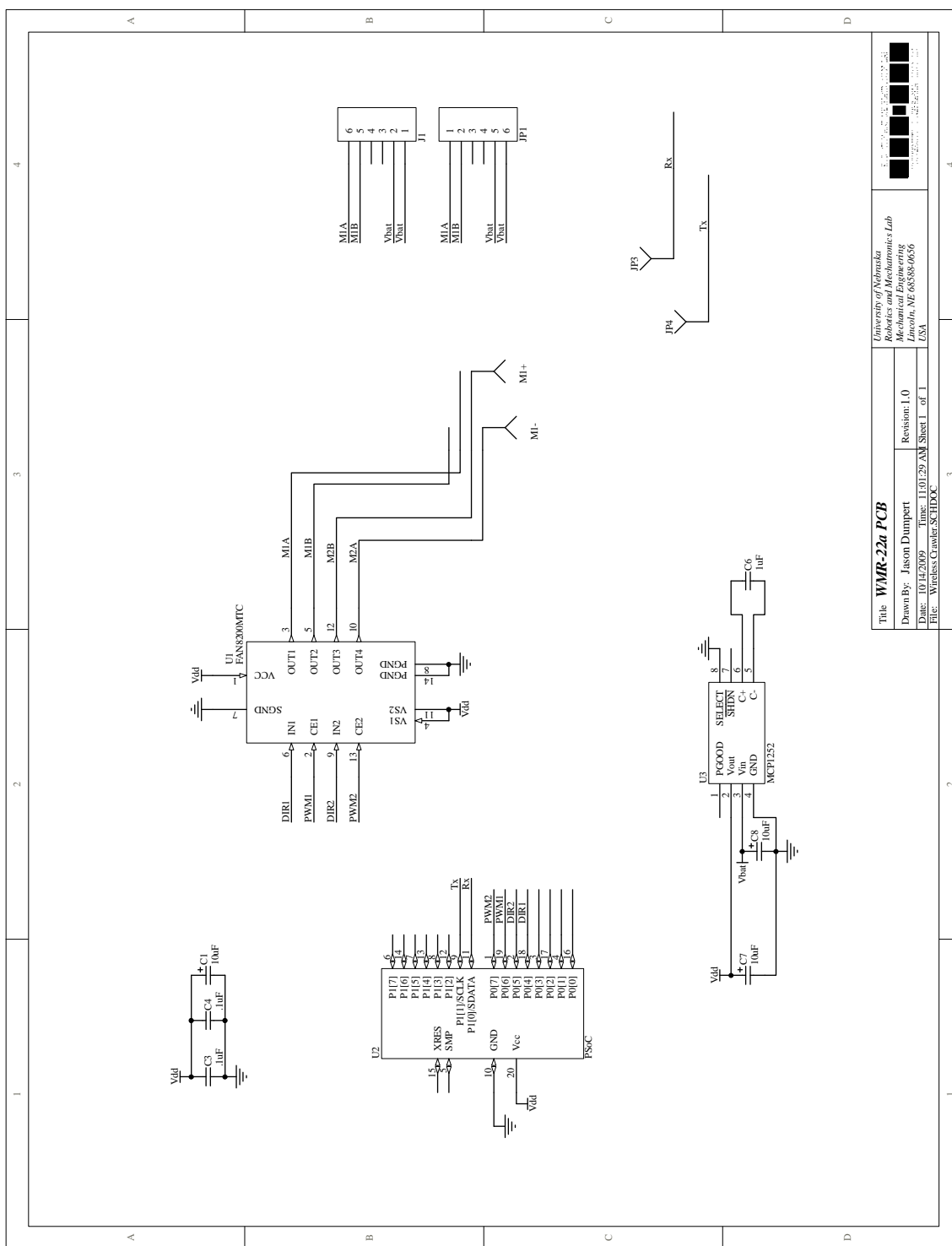


Figure A.11: BMR-22a PCB



Title: **WMR-22a PCB**
 Drawn By: Jason Dumpert
 Date: 10/12/2009
 File: WirelessCrawler.SCHDOC
 University of Nebraska
 Mechanical Engineering Lab
 Lincoln, NE 68588-0636
 Revision: 1.0
 Time: 11:01:29 AM
 Sheet 1 of 1
 USA

Figure A.12: WMR-22a PCB

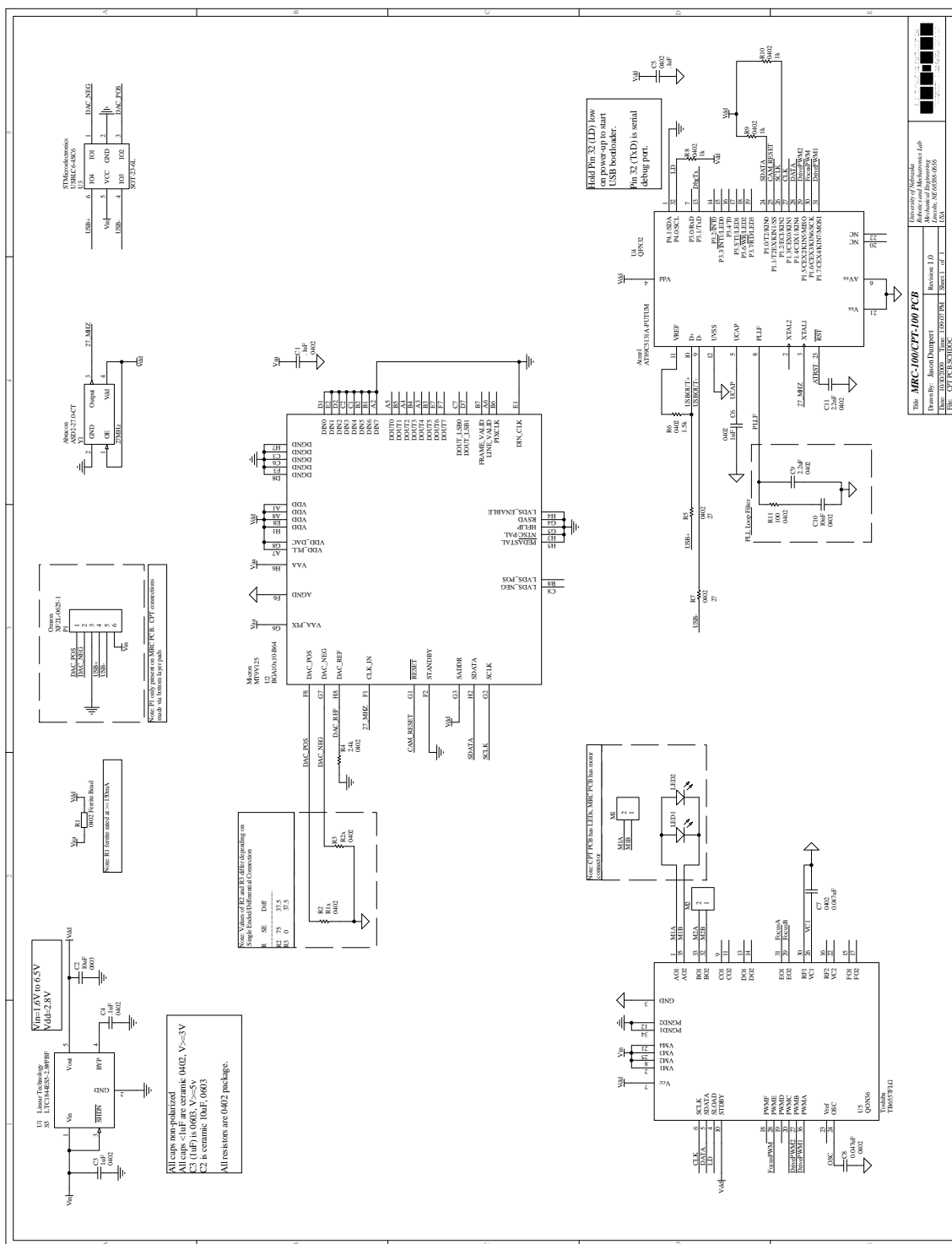


Figure A.13: MRC-100/CPT-100 PCB

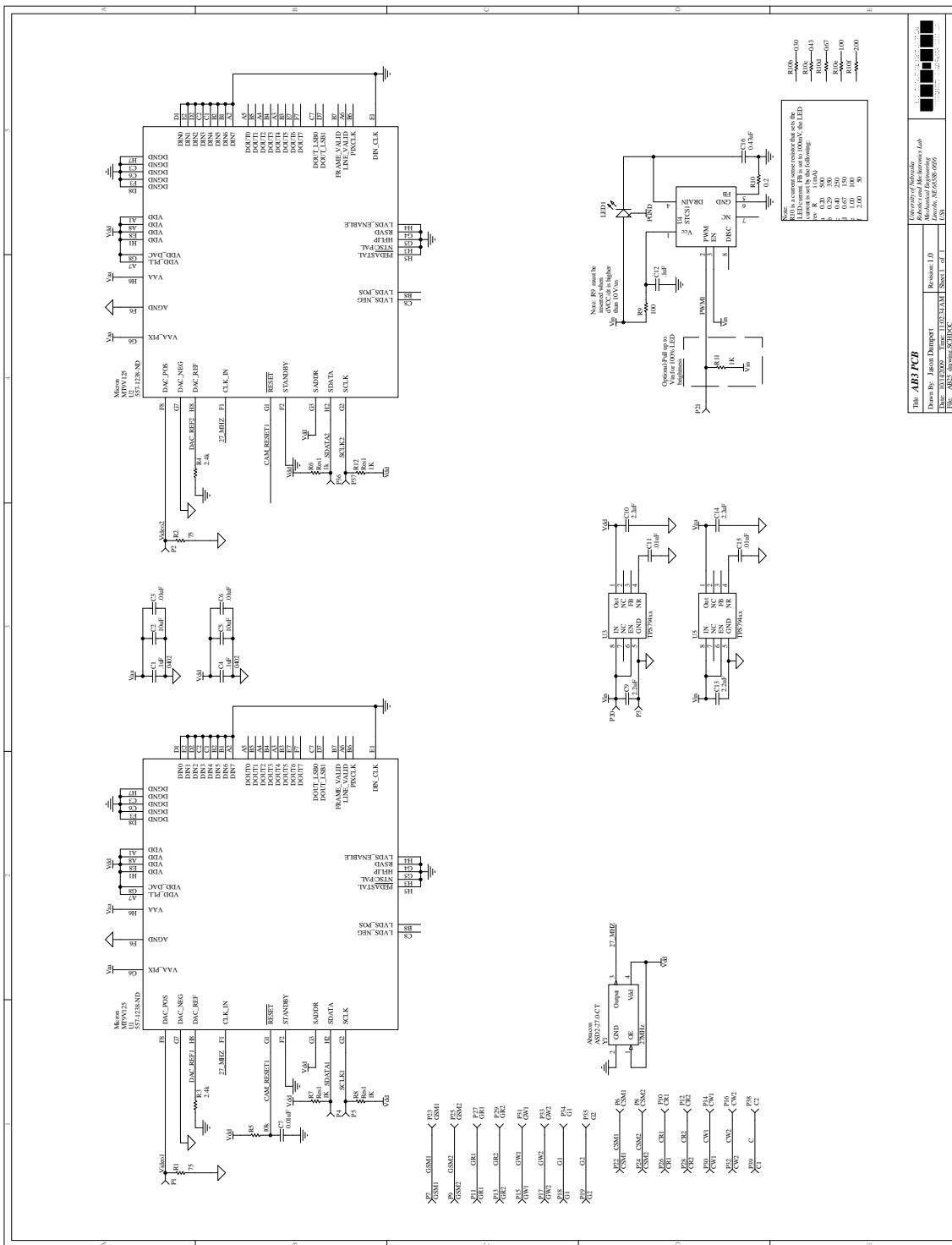


Figure A.14: AB3 PCB

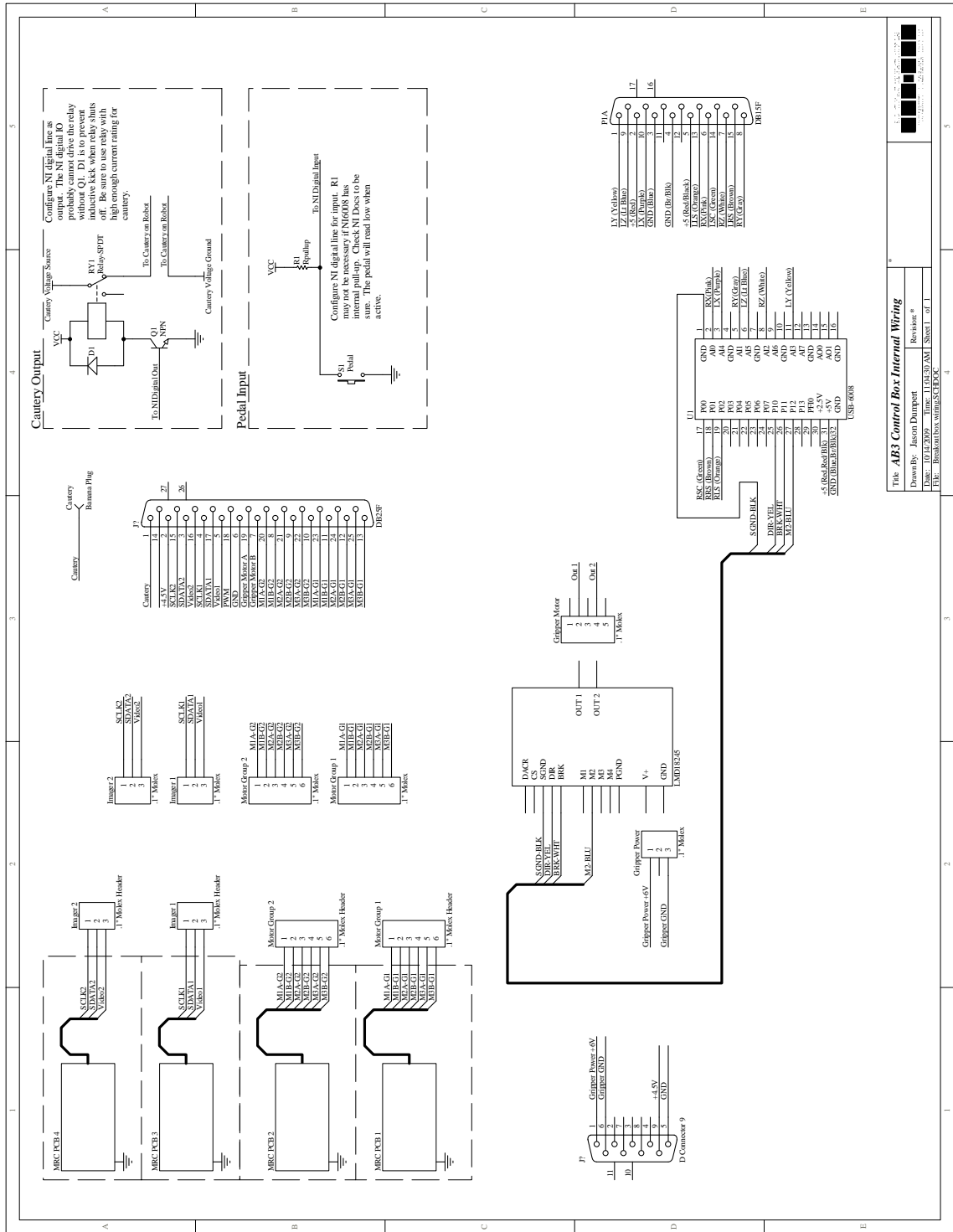


Figure A.15: AB3 Interface Box

Bibliography

- [1] C. M. Tarnay, K. B. Glass, and M. G. Munro, “Incision characteristics associated with six laparoscopic trocar-cannula systems: A randomized, observer-blinded comparison,” *Obstetrics & Gynecology*, vol. 94, pp. 89–93, 1999.
- [2] F. Tendick, S. S. Sastry, R. S. Fearing, and M. Cohn, “Applications of micromechatronics in minimally invasive surgery,” *IEEE/ASME Transactions on Mechatronics*, vol. 3, no. 1, pp. 34–42, 1998.
- [3] A. G. Harrell and B. T. Heniford, “Minimally invasive abdominal surgery: lux et veritas past, present, and future,” *The American Journal of Surgery*, vol. 190, pp. 239–243, 2005.
- [4] M. Uhrich, R. Underwood, J. Standeven, N. Soper, and J. Engsborg, “Assessment of fatigue, monitor placement, and surgical experience during simulated laparoscopic surgery,” *Surgical Endoscopy*, vol. 16, pp. 635–639, 2002.
- [5] G. Ballantyne, “Robotic surgery, telerobotic surgery, telepresence, and telementoring: Review of early clinical results,” *Surgical Endoscopy*, vol. 16, pp. 1389–1402, 2002.
- [6] L. Kavoussi, R. Moore, J. Adams, and A. Partin, “Comparison of robotic versus human laparoscopic camera control,” *The Journal of Urology*, vol. 154, no. 6, pp. 2134–6, 1995.

- [7] A. Partin, J. Adams, R. Moore, and L. Kavoussi, "Complete robot-assisted laparoscopic urologic surgery: a preliminary report." *Journal of the American College of Surgeons*, vol. 181, no. 6, pp. 552–7, Dec 1995.
- [8] W. Geis, H. Kim, E. J. Brennan, P. McAfee, and Y. Wang, "Robotic arm enhancement to accommodate improved efficiency and decreased resource utilization in complex minimally invasive surgical procedures," *Studies in health technology and informatics*, vol. 29, pp. 471–81, 1996.
- [9] M. Ghodoussi, S. Butner, and Y. Wang, "Robotic surgery - the transatlantic case," in *Proceedings of the IEEE International Conference on Robotics and Automation, 2002.*, vol. 2, 2002, pp. 1882–1888.
- [10] H. Reichenspurner, R. J. Damiano, M. Mack, D. H. Boehm, H. Gulbins, C. Detter, B. Meiser, R. Ellgass, and B. Reichart, "Use of the voice-controlled and computer assisted surgical system zeus for endoscopic coronary artery bypass grafting," *J Thorac Cardiovasc Surg*, vol. 118, pp. 11–16, 1999.
- [11] F. Corcione, C. Esposito, D. Cuccurullo, A. Settembre, N. Miranda, F. Amato, F. Pirozzi, and P. Caiazzo, "Advantages and limits of robot-assisted laparoscopic surgery," *Surgical Endoscopy*, vol. 19, pp. 117–119, 2005.
- [12] K. Moorthy, Y. Munz, J. Dosis, J. Hernandex, S. Martin, F. Bello, T. Rockall, and A. Darzi, "Dexterity enhancement with robotic surgery," *Surgical Endoscopy*, vol. 18, pp. 790–795, 2004.
- [13] Intuitive Surgical Inc, "Intuitive Surgical Inc Form 10-K Annual Report (2008)," Feb 2008.
- [14] V. B. Kim, W. H. H. Chapman, R. J. Albrecht, B. M. Bailey, J. Young, L. Nifong, and W. Chitwood, "Early experience with telemanipulative robot-assisted

- laparoscopic cholecystectomy using da vinci,” *Surgical Laparoscopy, Endoscopy & Percutaneous Techniques*, vol. 12, no. 1, pp. 33–44, 2002.
- [15] K. Hyosig and J. Wen, “Robotic assistants aid surgeons during minimally invasive procedures,” *Engineering in Medicine and Biology Magazine, IEEE*, vol. 20, no. 1, pp. 94–104, Jan/Feb 2001.
- [16] D. B. Camarillo, T. M. Krummel, and J. K. J. Salisbury, “Robotic technology in surgery: past, present, and future,” *The American Journal of Surgery*, vol. 188, no. Suppl to October 2004, pp. 2S–15S, 2004.
- [17] Given Imaging Ltd., “PillCam SB,” Online, September 2009. [Online]. Available: <http://www.givenimaging.com/en-us/Patients/Pages/pageSmallBowel.aspx>
- [18] G. Mone, “How it works: The endoscope camera in a pill,” *Popular Science*, March 2008. [Online]. Available: <http://www.popsci.com/node/19963>
- [19] E. Seibel, R. Carroll, J. Dominitz, R. Johnston, C. Melville, C. Lee, S. Seitz, and M. Kimmey, “Tethered capsule endoscopy, a low-cost and high-performance alternative technology for the screening of esophageal cancer and barrett’s esophagus,” *Biomedical Engineering, IEEE Transactions on*, vol. 55, no. 3, pp. 1032–1042, March 2008.
- [20] E. J. Seibel, R. S. Johnston, and C. D. Melville, “A full-color scanning fiber endoscope,” *Proc. SPIE, Opt. Fibers Sens. for Med. Diagnost. Treatment Appl. VI*, vol. 6083, pp. 608 303–18, March 2006.
- [21] University of Washington, “Camera in a pill offers cheaper, easier window on your insides,” *ScienceDaily*, January 2008. [Online]. Available: <http://www.sciencedaily.com/releases/2008/01/080124161613.htm>

- [22] T. Hu, P. Allen, and D. Fowler, “In-vivo pan/tilt endoscope with integrated light source,” in *IEEE/RSJ International Conference on Intelligent Robots and Systems, 2007. IROS 2007.*, 29 2007–Nov. 2 2007, pp. 1284–1289.
- [23] P. Dario, M. Carrozza, L. Lencioni, B. Magnani, and S. D’Attanasio, “A microrobotic system for colonoscopy,” in *Proceedings of the IEEE International Conference on Robotics and Automation, 1997*, vol. 2, Apr 1997, pp. 1567–1572 vol.2.
- [24] P. Dario, M. C. Carrozza, and A. Pietrabissa, “Development and in vitro testing of a miniature robotic system for computer-assisted colonoscopy,” *Computer Aided Surgery*, vol. 4, pp. 1–14, 1999.
- [25] P. Breedveld, “Development of a rolling stent endoscope,” in *The First IEEE/RAS-EMBS International Conference on Biomedical Robotics and Biomechatronics*, Feb. 2006, pp. 921–926.
- [26] B. Kim, Y. Jeong, H.-Y. Lim, T. S. Kim, J.-O. Park, P. Dario, A. Menciassi, and H. Choi, “Smart colonoscope system,” in *2002 IEEE/RSJ International Conference on Intelligent Robots and Systems EPFL*, Lausanne, Switzerland, 2002.
- [27] A. Menciassi, J. H. Park, S. Lee, S. Gorini, P. Dario, and J.-O. Park, “Robotic solutions and mechanisms for a semi-autonomous endoscope,” in *2002 IEEE/RSJ International Conference on Intelligent Robots and Systems EPFL*, Lausanne, Switzerland, 2002.
- [28] A. Slatkin, J. Burdick, and W. Grundfest, “The development of a robotic endoscope,” in *IEEE/RSJ International Conference on Intelligent Robots and Sys-*

tems 95: 'Human Robot Interaction and Cooperative Robots', vol. 2, Aug 1995, pp. 162–171.

- [29] N. Patronik, M. Zenati, and C. N. Riviere, “Preliminary evaluation of a mobile robotic device for navigation and intervention on the beating heart,” *Computer Aided Surgery*, vol. 10, no. 4, pp. 225–232, 2005.
- [30] A. N. Kalloo, V. K. Singh, S. B. Jagannath, H. Niiyama, s. L. Hill, C. A. Vaughn, C. A. Magee, and S. V. Kantsevov, “Flexible transgastric peritoneoscopy: a novel approach to diagnostic and therapeutic interventions in the peritoneal cavity,” *Gastrointestinal Endoscopy*, vol. 60, no. 1, pp. 114–117, 2004.
- [31] M. S. Wagh, B. F. Merrifield, and C. C. Thompson, “Survival studies after endoscopic transgastric oophorectomy and tubectomy in a porcine model,” *Gastrointestinal Endoscopy*, vol. 3, pp. 473–478, 2006.
- [32] E. Lima, T. Henriques-Coelho, C. Rolanda, J. M. Pego, D. Silva, and J. L. Carvalho, “Transvesical thoracoscopy: a natural orifice transluminal endoscopic approach for thoracic surgery,” *Surgical Endoscopy*, vol. 21, no. 6, pp. 854–858, 2007.
- [33] M. Bessler, P. Stevens, L. Milone, M. Parikh, and D. Fowler, “Transvaginal laparoscopically assisted endoscopic cholecystectomy: a hybrid approach to natural orifice surgery,” *Gastrointest Endoscopy*, vol. 6, pp. 1243–1245, 2007.
- [34] R. Zorron, M. Filgueiras, L. Maggioni, L. Pombo, G. Carvalho, and A. Oliveira, “NOTES transvaginal cholecystectomy: report of the first case,” *Surg Innov*, vol. 14, pp. 279–283, 2007.

- [35] USGI Medical, “USGI announces first NOTES transgastric cholecystectomy procedures,” June 2007. [Online]. Available: <http://www.usgimedical.com/news/releases/062507.htm>
- [36] A. N. Kalloo, D. Rattner, W. Brugge, C. J. Gostout, R. H. Hawes, S. V. Kantsevov, M. Marohn, J. Pasricha, J. Ponsky, W. Richards, R. Rothstein, N. Soper, L. Swanstrom, and C. Thompson, “ASGE/SAGES working group on natural orifice transluminal endoscopic surgery white paper,” *Gastrointestinal Endoscopy*, vol. 62, no. 2, pp. 199–203, 2005.
- [37] USGI Medical, “Transport multi-lumen operating platform,” June 2009. [Online]. Available: <http://www.usgimedical.com/eos/components-transport.htm>
- [38] D. J. Scott, S.-J. Tang, R. Fernandex, R. Bergs, M. T. Goova, I. Zeltser, F. J. Kehdy, and J. A. Cadeddu, “Completely transvaginal NOTES cholecystectomy using magnetically anchored instruments,” *Surgical Endoscopy*, vol. 21, no. 12, pp. 2308–2316, 2007.
- [39] J. Hill and W. Park, “Real time control of a robot with a mobile camera,” in *Proceedings of the 9th ISIR*, Washington, D.C., 1979, pp. 233–246.
- [40] A. Casals, J. Amat, and E. Laporte, “Automatic guidance of an assistant robot in laparoscopic surgery,” *Proceedings of IEEE International Conference on Robotics and Automation, 1996*, vol. 1, pp. 895–900 vol.1, Apr 1996.
- [41] G.-Q. Wei, K. Arbter, and G. Hirzinger, “Real-time visual servoing for laparoscopic surgery. controlling robot motion with color image segmentation,” *Engineering in Medicine and Biology Magazine, IEEE*, vol. 16, no. 1, pp. 40–45, Jan.-Feb. 1997.

- [42] A. Krupa, J. Gangloff, C. Doignon, M. 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, Oct. 2003.
- [43] P. Hynes, G. Dodds, and A. Wilkinson, “Uncalibrated visual-servoing of a dual-arm robot for surgical tasks,” in *Proceedings 2005 IEEE International Symposium on Computational Intelligence in Robotics and Automation*, June 2005, pp. 151–156.
- [44] G. Zong, Y. Hu, and D. Liang, “Visually servoed suturing for robotic microsurgical keratoplasty,” in *Proceedings of the 2006 IEEE/RSJ International Conference on Intelligent Robots and Systems*, October 2006, pp. 2358–2363.
- [45] L. G. Shapiro and G. C. Stockman, *Computer Vision*, M. J. Horton, Ed. Prentice-Hall, Inc., 2001.
- [46] A. L. Gilbert, M. K. Giles, G. M. Flachs, R. B. Rogers, and U. Y. Hsun, “A real-time video tracking system,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. PAMI-2, no. 1, pp. 47–56, Jan. 1980.
- [47] H. Tao, H. Sawhney, and R. Kumar, “A sampling algorithm for tracking multiple objects,” in *In Workshop on Vision Algorithms*, 1999, pp. 53–68.
- [48] J. Maccormick and A. Blake, “A probabilistic exclusion principle for tracking multiple objects,” *International Journal of Computer Vision*, vol. 39, no. 1, pp. 57–71, 2000.
- [49] L. Zhongxiang and L. Ronghua, “Complex multiple features tracking algorithm in motion capture,” in *TENCON '02. Proceedings. 2002 IEEE Region 10 Con-*

- ference on Computers, Communications, Control and Power Engineering*, vol. 1, Oct. 2002, pp. 277–280 vol.1.
- [50] M. Han, A. Sethi, W. Hua, and Y. Gong, “A detection-based multiple object tracking method,” in *International Conference on Image Processing*, vol. 5, Oct. 2004, pp. 3065–3068 Vol. 5.
- [51] H. Uemura, J. Tan, and S. Ishikawa, “A robust color tracker employing a two-dimensional color histogram with an anisotropic asymmetric gaussian distribution model,” in *TENCON 2006. 2006 IEEE Region 10 Conference*, Nov. 2006, pp. 1–4.
- [52] H. Uemura, J. K. Tan, and S. Ishikawa, “A color tracker employing a two-dimensional color histogram under changeable illumination,” in *IEEE Industrial Electronics, IECON 2006 - 32nd Annual Conference on*, Nov. 2006, pp. 3273–3278.
- [53] S. H and B. Efron, “Adaptive color space switching for face tracking in multi-colored lighting environments,” in *Automatic Face and Gesture Recognition, 2002. Proceedings. Fifth IEEE International Conference on*, May 2002, pp. 249–254.
- [54] M. Rentschler, “In vivo abdominal surgical robotics: Tissue mechanics modeling, robotic design, and analysis,” Ph.D. dissertation, University of Nebraska, May 2006.
- [55] A. Lehman, “Miniature in vivo robot for natural orifice transluminal endoscopic surgery,” Master’s thesis, University of Nebraska, 2007.
- [56] N. Wood, “Design and analysis of dexterous in vivo robots for notes,” Master’s thesis, University of Nebraska, December 2008.

- [57] *DSP Series Motion Controller Installation Guide*, Motion Engineering, Inc, Santa Barbara, CA, December 1999.
- [58] *CY8C27143, CY8C27243, CY8C27443, CY8C27543, CY8C27643 PSoC Programmable System-on-Chip*, Cypress Semiconductor Corporation, San Jose, CA, April 2009.
- [59] OmniVison Technologies, “OV7930 Color CMOS Analog CAMERACHIP,” Boise, ID, January 2003.
- [60] Microchip Technologies Inc, “PIC16F87x Data Sheet,” 2001.
- [61] Northern Digital Inc, “Comm32 class code,” 2002.
- [62] J. V. Joseph, D. Oleynikov, M. Rentschler, J. Dumpert, and H. R. H. Patel, “Microrobot assisted laparoscopic urological surgery in a canine model,” *The Journal of Urology*, vol. 180, pp. 2202–2205, November 2008.
- [63] Rabbit Semiconductor, “RCM3200 Rabbitcore Series Data Sheet,” 2008.
- [64] Atmel Corporation, “AT89C5131 - 8-bit Flash Microcontroller with Full Speed USB Device,” 2005.
- [65] Toshiba, “TB6557FLG DC Motor Driver,” 2005.
- [66] Micron Technology, Inc, “MT9V125 - SOC VGA Digital Image Sensor,” Boise, ID, July 2005.
- [67] Atmel, “C5131-usb-cdc code,” 2004.
- [68] Cree, Inc, “Cree XLamp XR-E LED datasheet,” 2007.
- [69] Sunex, Inc., “DSL758 Datasheet,” July 2005.

- [70] Karl Storz, “Karl Storz Camera Units,” July 2009. [Online]. Available: http://web-surg.net/karlstorz/general/units/camera_units.htm
- [71] Sensoray, “Model 2250 USB MPEG Framegrabber,” October 2006. [Online]. Available: <http://www.sensoray.com/html/2250data.htm>
- [72] N. Koren, “Imatest documentation,” 2006. [Online]. Available: <http://www.imatest.com/docs/>
- [73] W. J. Smith, *Modern Optical Engineering*. McGraw-Hill Book Company, 1966.
- [74] R. Tsai, “A versatile camera calibration technique for high-accuracy 3d machine vision metrology using off-the-shelf tv cameras and lenses,” *IEEE Journal of Robotics and Automation*, vol. 3, no. 4, pp. 323–344, August 1987.
- [75] National Oceanic and Atmospheric Administration, “NOAA Ocean Explorer: Aquarius Underwater Laboratory,” retrieved June 10, 2009. [Online]. Available: <http://www.oceanexplorer.noaa.gov/technology/diving/aquarius/aquarius.html>
- [76] University of North Carolina Wilmington, “Aquarius Undersea Laboratory Hosts NASA Astronauts for Space Simulation and Training Project,” retrieved June 11, 2006. [Online]. Available: <http://www.uncw.edu/aquarius/archive/2001/10.2001/expd.htm>
- [77] C. P. Garbin, “Power, effect size, and sample size,” 2009. [Online]. Available: http://psych.unl.edu/psycrs/350/unit2/exam_table.pdf
- [78] Andersen Products, Inc., “Ethylene oxide (EtO or EO) sterilization in healthcare facilities,” retrieved August 26, 2008. [Online]. Available: <http://www.anpro.com/support/indexeto.htm>

- [79] Ethicon Inc, “The breakthrough technology behind STERRAD sterilization systems,” retrieved August 26, 2008. [Online]. Available: http://www.aspjj.com/Products_&_Services/STERRAD/Technology/
- [80] University of Nebraska Medical Center, “Postoperative lab reports on survivable animal study-animal A,” April 2007.
- [81] —, “Postoperative lab reports on survivable animal study-animal B,” April 2007.
- [82] —, “Postoperative lab reports on survivable animal study-animal C,” July 2007.
- [83] A. Lehman, J. Dumpert, N. Wood, L. Redden, A. Visty, S. Farritor, B. Varnell, and D. Oleynikov, “Natural orifice cholecystectomy using a miniature robot,” *Surg Endosc*, vol. 23, pp. 260–266, 2009.
- [84] P. Getreuer, “Colorspace matlab package.” [Online]. Available: <http://www.math.ucla.edu/~getreuer/colorspace.html>
- [85] C. Poynton, “Frequently asked questions about color,” 1997. [Online]. Available: <http://www.poynton.com/>
- [86] “OpenCV Wiki.” [Online]. Available: <http://opencv.willowgarage.com/>
- [87] G. F. Franklin, J. D. Powell, and M. L. Workman, *Digital Control of Dynamic Systems*, 3rd ed. Addison Wesley Longman, 1998.
- [88] N. Wood, A. Lehman, J. Dumpert, D. Oleynikov, and S. Farritor, “A prototype in vivo robotic system for notes,” *International Journal of Robotics Research Special Issue on Medical Robotics*, 2008, submitted.

- [89] D. Scharstein and R. Szeliski, “A taxonomy and evaluation of dense two-frame stereo correspondence algorithms,” *Int. J. Comput. Vision*, vol. 47, no. 1-3, pp. 7–42, 2002.
- [90] E. Psota, “Stereoscopic wound measurement device and algorithm,” Master’s thesis, Dept of Electrical Engineering, University of Nebraska-Lincoln, 2006.
- [91] R. Alterovitz, M. Branicky, and K. Goldberg, “Motion planning under uncertainty for image-guided medical needle steering,” *International Journal of Robotics Research*, vol. 27, pp. 1361–1374, Nov 2008.
- [92] D. Reid, “An algorithm for tracking multiple targets,” *Automatic Control, IEEE Transactions on*, vol. 24, no. 6, pp. 843–854, Dec 1979.