Abstract

KRACHT, AARON ARTHUR. A Linear Base Articulated Robot Arm for Surgical Endoscopy. (Under the direction of Dr. Edward Grant).

This project involved developing a surgical robot assistant using an articulated robot running on a linear axis. The research concentrated on studying the localization of an endoscopic tool. The kinematics involved in this type situation requires that a constant point in space (trocar point) is maintained along a rigid tool while repositioning the manipulator. Results show that the localization algorithm and interactive interface developed is capable of using this unique robot configuration to perform the desired task. For this system, error was used as the performance metric. Positioning of the endoscopic manipulator relative to the world coordinate frame was possible to within 0.05 inch. Error in maintaining a constant point in space is evident during repositioning however this was caused by limitations in the robot arm.

A Linear Base Articulated Robot Arm for Surgical Endoscopy

by

Aaron Arthur Kracht

A thesis submitted to the Graduate Faculty of
North Carolina State University
in partial fulfillment of the
requirements for the Degree of
Master of Science

Electrical Engineering

Raleigh, North Carolina

2006

Approved by:

Dr. Edward Grant
Chair of Advisory Committee

Dr. John Muth

Dr. Ola Harrysson

Biography

Aaron Arthur Kracht was born on January 27, 1981 in Aurora, Colorado to Jerry and Cynthia Kracht. At a very young age Aaron had a particular fascination with the solar system. Many nights in Colorado were spent star gazing and imagining life in space. At the age of 9, his family moved to High Point, NC. Due to all the trees around his new home, Aaron's fascination for space dwindled but it resurfaced later in his life. In middle school and high school Aaron had an interest in problem solving, designing, and experimenting. This helped him to work hard and excel in mathematics and science. Late in his high school career he decided to become an engineer and applied to North Carolina State University's engineering program. Accepted as an undeclared engineer, Aaron took his first year in college to decide which type of engineering to concentrate in. At this point he decided to concentrate in Computer and Electrical Engineering because of a new found fascination for electronics. During his undergraduate studies, Aaron's coursework was spread over a lot of different disciplines because he had not decided on what to concentrate in. Then, toward the end of his undergraduate studies, Aaron realized the fascination he has with the application of robotics and intelligent machines in space. It was then decided to pursue a Masters in Electrical Engineering with a concentration in robotics and intelligent machines.

Acknowledgements

I can not put into words the gratitude that I have towards everyone in my life.

Thanks for all that you do.

Table of Contents

List of Figures

Chapter 1 – Introduction

Robots are typically thought to be used for industrial purposes however they are beginning to gain the attention of the medical field. A current application pertains to using the precision and stability of a robot arm to assist in minimally invasive surgery.

1.1 – Robotically Assisted Surgery

Surgery is a medical procedure involving an incision in the human body performed to repair damage or arrest disease [20]. In a fully invasive operation, an open incision is made that is large enough for the surgeon to view the internal organs and perform the operation. For example, in open-heart surgery a long incision is made along the sternum, after which the sternum is split and retracted (median sternotomy) [21]. These open incisions increase trauma to a patient beyond what is experienced from the actual repair. After the operation, the patient must heal from trauma associated with the repaired organs and from the open incision. This results in long recovery times and opportunities for infection [16].

Endoscopic Surgery, also called Minimally Invasive Surgery (MIS), is a type of operation that has been developed to reduce trauma associated with making these large open incisions. This type of operation involves viewing the operational field on a television monitor by inserting a special camera, called an endoscope, through a small incision in the skin (see Figure 1). To perform the operation,"long, thin, manually operated instruments" are inserted through other small incisions called trocars [16]. This type of surgery reduces the size of the open incisions and therefore results in less pain and scarring after surgery, faster recovery times, and less risk of infection [9].

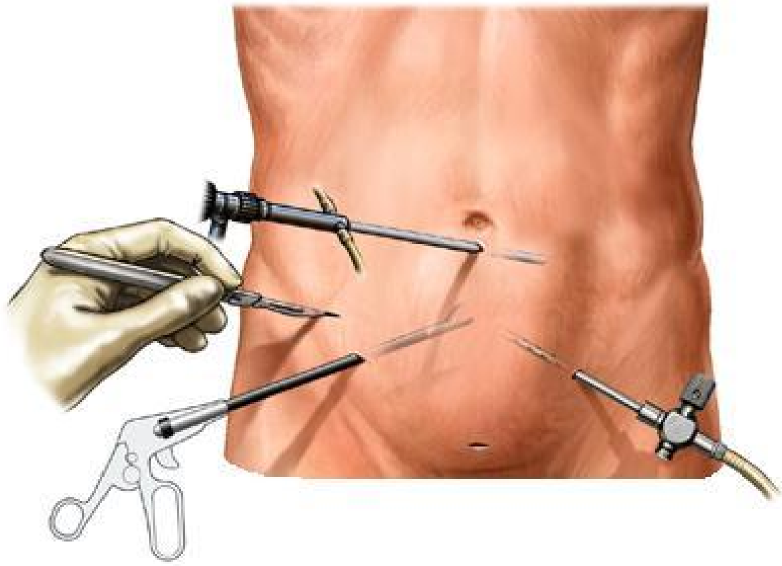


Figure 1 - Manual Endoscopic Surgery

Endoscopic surgery does have benefits over fully invasive surgery but there are also several disadvantages associated with performing this type of operation. The two basic disadvantages involve viewing the internal operational field and the surgical tool interface. Using an endoscopic camera, the operational field is viewed on a 2-D television monitor providing only a limited view with no depth perception. Advances in visual technology have provided surgeons with equipment that allow them to view the field in 3-D however this technology is expensive [23]. Another disadvantage is that the endoscope is operated by a surgical assistant. This requires the surgeon to communicate motion instructions which becomes difficult when giving instructions such as how far to move the endoscope and in which direction. It has also been reported that small tremors from the scope-holding assistant, magnified onto the television monitor, can cause nausea among the surgical team [23].

Disadvantages associated with the surgical tool interface involve the endoscopic tools and the way that the operation is performed. Often the endoscopic tools are heavy, lacking ergonomic design, and do not have the same DOF/dexterity as a human hand (see Figure 2) [10, 12]. As mentioned before, these tools are inserted through small incisions in the patient

Figure 2 - Manual Endoscopic Tools

to perform the operation. This creates the fulcrum/lever effect whereas the surgeon is required to transpose each hand motion to get the desired internal motion [3]. For example, if it is desired to move the tool manipulator to the left within the patient the surgeon must transpose this motion in his mind and move the tool handle to the right (see Figure 3). Surgeons must also deal with the amplification of tremors in their hands due to the fulcrum/lever effect and the length of the tools [16]. Performing an operation through small incisions using endoscopic tools removes the ability for a surgeon to use the sense of touch to gain more information about the internal tissue (haptic feedback). Tools that provide surgeons with this type of information are being developed, however they are not yet in use [8].

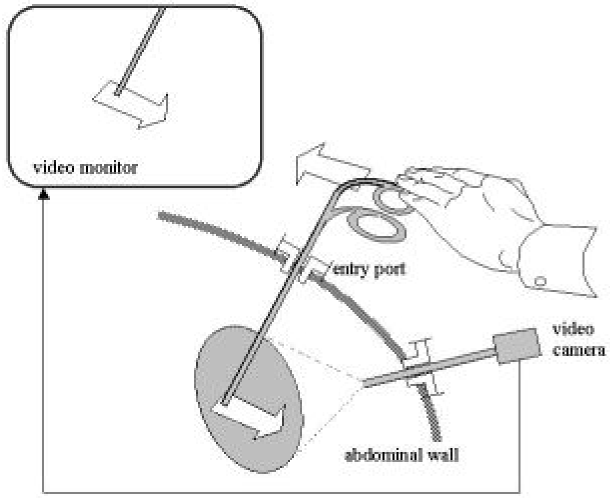


Figure 3 - Fulcrum/Lever Effect [3]

In the late 1980's, researchers motivated by the limitations of endoscopic surgery turned to robotic technology to make improvements [4]. Although the idea of using robotics for telepresence surgery was not a new idea at this time, research and development into this concept rapidly progressed around the success of MIS techniques and became possible with advancements in computing power [4]. By 1998, two major companies had developed surgical systems approved by the FDA for commercial use. Computer Motion in Goletta, California developed the Zeus Surgical System and Intuitive Surgical in Sunnyvale, California developed the daVinci Surgical System. Currently only the daVinci is in production due to the merger of Computer Motion and Intuitive Surgical in 2003.

daVinci Surgical System

The daVinci Surgical System is a surgical robotic system that assists surgeons in performing an endoscopic operation. It consists of three main components (see Figure 4): the surgeon console (left), patient-side cart (right), and laparoscopic tower (center) [10]. The surgeon console provides the operator with a 3-D image of the surgical field and complete control over the surgical robot. Surgeons can control the robot arms and thus the orientation of the endoscopic tools by using the surgical controls placed below the 3-D view port. Controlling the orientation of the endoscope is provided through the hand controls and foot pedals. The patient side cart consists of 2 or 3 robotic arms to manipulate the endoscopic tools and a robotic arm to orient the endoscopic camera. The laparoscopic tower holds additional equipment such as light sources for the cameras, the harmonic scalpel generator, insufflators, and a viewing monitor for surgical assistants [10].

Figure 4 - Intuitive Surgical daVinci Surgical System

The daVinci Surgical System offers several benefits to a surgeon over performing traditional endoscopic surgery. As mentioned before, one of the difficulties of endoscopic surgery is viewing the operational field. To improve this surgeons are provided with direct control of the endoscope eliminating the need to communicate motion instructions to a surgical assistant. Since a robot arm holds the endoscope, any chance of nausea associated with tremor is also eliminated. An additional advantage is that all motion is matched with the current magnification. This means that the distance required to move the robot arm, with the surgical controls, from the top of the 3-D console view to the bottom is the same under all magnifications. Surgeons that perform complex or microsurgical operations such as in pediatric surgery benefit from this ability [2].

The surgeon console on the daVinci Surgical System provides surgeons with a comfortable ergonomic interface to perform an endoscopic operation. The surgical controls are located naturally below the 3-D view port and the endoscope controls are placed at the feet. At anytime during the operation, a clutch mechanism is provided to allow a surgeon to move the surgical controls to a comfortable position without moving the robotic arms. Since the operation is performed through a computer, the surgeon’s hand position is monitored and then filtered to remove involuntary motions caused by tremor [8]. The computer also removes the fulcrum effect to give direct mapping of the surgeon’s hands to the robotic manipulators. Intuitive also developed the Endowrist Instrument to improve the DOF/dexterity of the endoscopic tools during a procedure. This tool provides the same degrees of freedom as the human hand as seen in Figure 5.

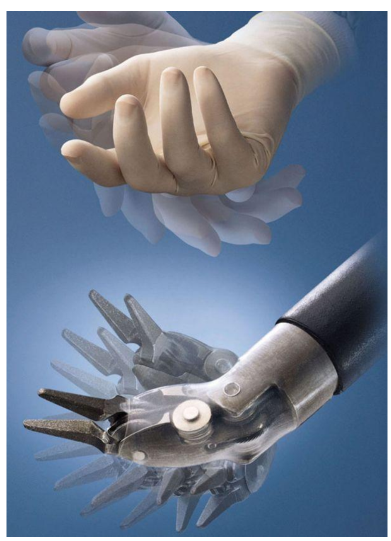


Figure 5 - Intuitive Surgical EndoWrist Manipulator

The daVinci Surgical System provides a lot of advantages over traditional endoscopic surgery, but there still are some disadvantages to using this system. First, similar to traditional endoscopic surgery, the system does not provide haptic feedback to the surgeon. Surgeons must pick up on forces applied to tissue by using visual queues and prior experience with the system [6]. Second, the system is relatively large and not portable. Due to the large size of the system, trocar points must be placed a certain distance apart or else there is a chance the robotic arms will collide during surgery. This is a current problem for pediatric surgery because the patient is so small that the trocar points must be close together [2]. Last, the system comes with a very large price tag of approximately 1 million dollars. This price raises a lot of questions as to the cost effectiveness of the system and completely removes smaller hospitals from considering the system.

Robotically assisted surgery has provided many advantages to surgeons, with some disadvantages, however it is still a relatively new technology. It is better to view current systems as prototypes of future systems [6].

1.2 – Thesis Goals

The main goal of this thesis is to use a linear base articulated robot arm to perform the task of positioning an endoscopic tool. The research concentrated on the localization of the tool. An additional goal was to design a system that would attach to the robot arm and provide functionality to an Intuitive Surgical EndoWrist. To control the surgical robotic system a human-machine interface was created.

1.3 – Thesis Outline

In Chapter 2 a broader view of medical robots is discussed and Chapter 3 gives the details concerning the design of the robot system for this thesis. Chapter 4 provides an analysis of the surgical system along with simulation and experimental data and Chapter 5 will present conclusions and future research.

Chapter 2 – Medical Robots

Medical robots are *tools* that are being used by doctors to increase there capabilities and abilities to diagnose and cure diseases. They are not replacing doctors, but they are providing them with ways to perform better. Medical robots use range from autonomously performing a specific task during an operation to providing a human-machine interface that helps perform the entire procedure. There are several commercial companies that offer these systems however they are not being widely used. Some of the reasons for this revolve around questions regarding there effectiveness, safety, and cost [7].

Before talking more specifically about the different types and tasks of medical robots, it will help to give a definition of a robotic system. A robot is a mechanical system that is capable of performing a physical task [21]. It can be broken down into three main components: a control device, actuators and mechanical parts, and sensors. These three components allow a robot to interact with its environment. The controller is the brains of the system processing information and changing the actuators and mechanical parts based on this information. The actuators and mechanical parts provide the actual motion of the robot system. The sensors of the system allow the robot to get feedback on its position and in smarter systems data from the environment. Benefits of using robotics are that they can perform motions with very high precision repetitively without fatigue. They are capable of performing tasks that are not possible by humans such as lifting heavy objects and can hold a very precise position endlessly. From these benefits it is not hard to see why research is being done to implement robotics into some of the very difficult, fatiguing, and precise procedures that doctors/surgeons are performing.

Medical robots can be divided according to their level of autonomy into three categories: active, semiactive, and passive [15]. Further detail and examples of these medical robots are given in the following sections.

2.1 – Active Medical Robots

Active medical robots autonomously perform a specific task during an operation that was programmed prior to the procedure. These robots take on an active role of performing a task under the supervision of the operator. Using this type of robot typically involves three steps: preoperative planning, program verification, and performing the operation. Preoperative planning involves using medical images and data to decide upon what the robot will do. After deciding on a plan of action, the robot is programmed to perform the specific task allowing it to know what to do. This program is then simulated or actually run on the robot before the operation to verify that it performs the task successfully with no complications. This is an important step for this type of robot because during the actual procedure the operator only supervises the robots action and stops it if any complications occur. The robot is then used to perform the preprogrammed task which may involve bone drilling, radiation application, or holding a sensor or tool.

An example of this type of medical robot is the ROBODOC which is used for orthopedic surgery [14, 18]. This robot is designed to perform the broaching procedure in total hip replacement operations. Before surgery, pins are inserted into the patient's femur, CT scans are taken, and using this data a surgeon selects an implant and chooses its location. This data is stored in the robot controller to be used during the operation. When it is time to use the robot, its position is calibrated using the surgical pins, and then the desired implant shape, location, and orientation is cut. The major benefit in using this robot is the precise cut that it can perform. Manual broaching can leave only 20% of the implant in contact with the bone leading to postoperative problems while with the ROBODOC's precision 96% is in contact [14].

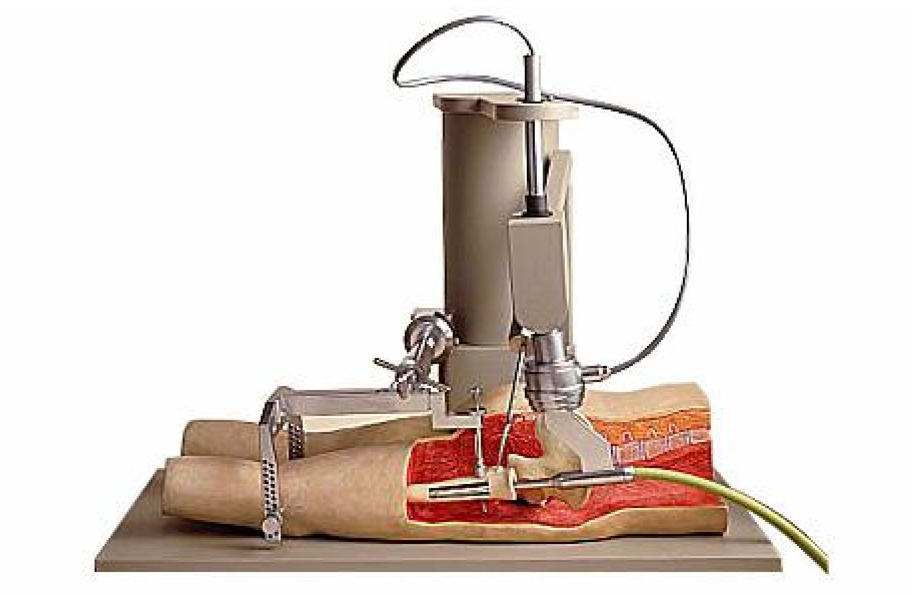


Figure 6 - RoboDoc

Another example of an active robot is an image-guided radiosurgery robot called the CyberKnife [1, 5]. This system consists of an x-ray linear accelerator (LINAC) mounted on a gantry robot arm which positions and orients the LINAC before delivering a dose of radiation. It is typically used to treat intracranial and extracranial tumors and other lesions which require a high degree of radiation delivery accuracy [5]. The order of operation in using this robot is similar to all active robots. What is interesting about this system is that it is not restricted to performing exact preprogrammed positions but uses image data to reposition itself according to the targets location. Initialization of the robot is performed by using previous CT images along with real-time radiographs to determine the coordinates of the tumor or lesion in the patient. The system then begins administering radiation to the prescribed locations in the patient using real-time radiograph images to adjust and reposition the robot if any movements in the patient occur.

Figure 7 - CyberKnife

2.2 – Semiactive Medical Robots

This type of medical robot takes on some of the features of both the passive and active medical robots. Similar to a passive robot, a surgeon will directly control the robotic system however the system will provide some type of constraint. This type of medical robot has been classified as having constrained cooperative control autonomy [19]. An example of this type of robot is the Acrobot which is used for total knee replacement surgery [11]. This system is similar to the RoboDoc in that preoperative data is used to decide upon the geometry and position of the cut in the bone. The difference between the two robots is that the Acrobot manipulator is controlled by the surgeon. During the cutting procedure, the surgeon “guides the robot using a handle with a force sensor attached to the robot tip [11].” The active role that the robot takes is preventing the surgeon from making a cut outside a predefined region. Benefits of using this system are geometrical accuracy of the flat cuts made on the femur and tibia. Other examples of this type of robot are the PADyC [15] and the "Steady-Hand" robot [17].

2.3 – Passive Medical Robots

Passive medical robots are systems that perform motions only under the command of a human operator through some interface (joystick, foot pedal, etc.). These robots are also called surgical assistants, surgical extenders, and telemanipulators [19]. These robots are programmed to listen to external interfaces for motion commands and other tasks. As opposed to active medical robots, there is no preoperative motion commands programmed into the controller. In some cases a surgeons may use medical images to perform preoperative decisions such as robot positioning, but the actual robot is not programmed to perform a motion command on its own. Most of these robots are used for endoscopic procedures were they may perform the task of positioning an endoscope camera or actually manipulate the endoscopic tools under a surgeon's control. An example of this type of robot is the daVinci Surgical System which is discussed in Chapter 1.

Chapter 3 – Robotic Surgical Platform

The robotic surgical system described in this thesis is a tool that would assist a surgeon in performing an endoscopic procedure. It does this by holding and physically manipulating an endoscopic tool under the direct control of the surgeon. It should be noted that the robot does not perform the surgery, but only provides assistance by manipulating the tools position and manipulator at the surgeons command. The system is classified as a passive medical robot, more specifically as a master-slave telemanipulator (see Figure 8). In this type of system, the master has direct control over all movements of the robotic system. This is an essential feature for a surgical robot because only the surgeon should have the ability to move the robot arm. In between the surgeon and the Robotic System is a human-machine interface that provides the surgeon with control of the system.

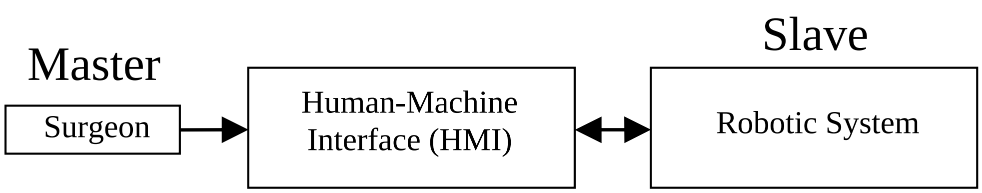


Figure 8 - Master-Slave Telemanipulator

3.1 – Human-Machine Interface

The human-machine interface (HMI) is the instruments provided by which the surgeon can control the robotic system to perform an operation. It is composed of a computer (monitor, cpu, keyboard, mouse), two Saitek evo Force Feedback joysticks, and the robot teach pendant. More details concerning each of these components is discussed in the following sections.

3.1.1 – Joysticks

The two joysticks provide direct control of the robotic system's movements. The first joystick is used to control the robotic arm while the second joystick controls the surgical manipulator. The robotic arm joystick provides XYZ motion control of the endoscopic end effecter in reference to the world frame. Joystick motions along the X and Y axis are converted to X and Y displacements while rotation of the joystick is converted to Z displacements. The manipulator joystick performs five different functions: yaw, pitch, roll, open grip, and close grip. X and Y motions of the joystick are converted to pitch and yaw motions of the manipulator. Rotating the joystick performs the roll option and pressing button 1 or button 2 performs the close or open grip function.

Figure 9 - Saitek evo Force Feedback joystick

Motion	Manipulator Response
X axis	Pitch
Y axis	Yaw
Rotation	Roll
Button 1	Close fingers
Button 2	Open fingers

Figure 10 - EndoWrist Joystick Control

Joystick Motion	Endoscope Response
X axis	X displacement
Y axis	Y displacement
Rotation	Z displacement

Figure 11 - Robot Arm Joystick Control

3.1.2 – Robot Teach Pendant

The robot teach pendant is used to position the robot before inserting the endoscopic tool into the patient. It only allows the user to change the joint angles of the robot arm and the position along the linear axis. The speed dial on the control ranges from 0 to 100, however for safety the robot should never be operated at a speed above 20. See Figure 13 for a detailed explanation of each switches operation.

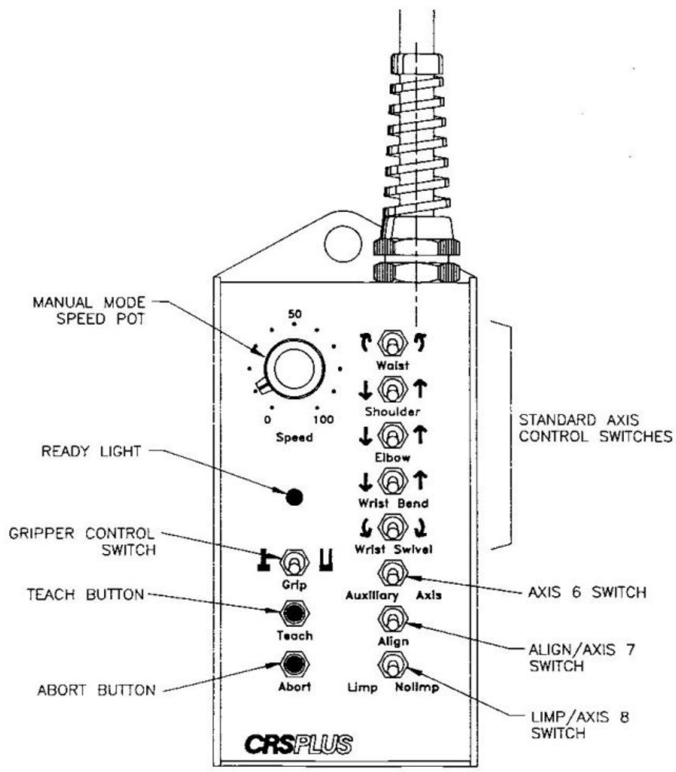
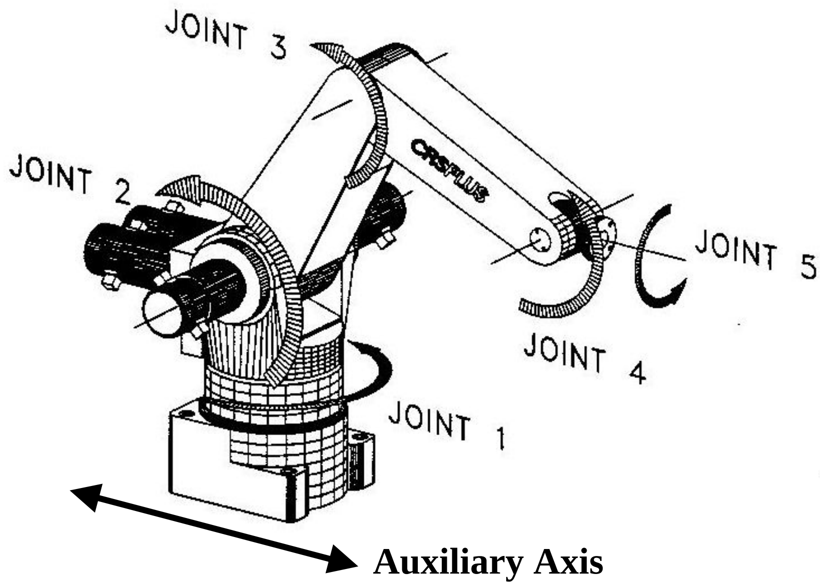


Figure 12 - Robot Teach Pendant



Controller	Action
Waist Switch	Rotates robot around waist (Joint 1). Switch right rotates counterclockwise. Switch left rotates clockwise.
Shoulder Switch	Rotates robots shoulder joint (Joint 2). Switch right raises arm. Switch left lowers arm.
Elbow Switch	Rotates robots elbow joint (Joint 3). Switch right raises robot forearm. Switch left lowers robot forearm.
Wrist Bend switch	Rotates the robots wrist joint (Joint 4). Switch left raises robot manipulator. Switch right lowers robot manipulator. (Changes the *pitch* of the tool position)
Wrist Swivel switch	Swivel the robot wrist joint (Joint 5). Switch right rotates clockwise. Switch left rotates counterclockwise (Changes the *roll* of the tool position)
Auxiliary Axis switch	Translates robot along length of table. Switch left translates toward homing bracket. Switch right translates away from homing bracket.
Align switch	Align robot tool with nearest (horizontal or vertical) plane. If pitch is 45 degrees it will align to the vertical plane.
Limp/No Limp switch	Switch left (Limp) disengages all positional servos. Switch right (No Limp) engages all positional servos.
Speed dial	Sets the speed at which the robot will move.
Grip Switch	Operates additional motor (Not Used)
Teach button	Records robot position into memory
Abort button	Immediately stops any motion in progress

Figure 13 - Teach Pendant Controls

3.1.3 – Computer and Software

The computer runs the systems software program which performs many functions but most importantly provides the user with the ability to control the robotic system. There are four different states of the program: Initialization, Trocar Placement, Insertion/Retraction, and Operation. The program starts up in the initialization state. The first task is to establish and verify communication with the Robotic Arm, Surgical Manipulator, and joysticks. Communication does not need to be established with the robot teach pendant since it is part of the robotic system. If communication is not established with any component, the program informs the user and then exits. This is done as a precaution so that all parts of the system are ready for operation before continuing. The second task in this state is to ask the user if the robot arm needs to be initialized. Performing this aligns the arm with the mathematical model of the arm stored in the controller's memory allowing the robot controller to know the position of the arm relative to the world. Initializing the robot can only be executed if the arm is in the homing bracket. This must be done if not previously executed since the robot controllers last power cycle. For safety the user of the system is notified of this action as opposed to automatic execution at the start-up of the program. After this the main program interface window opens and the system transfers into the Trocar Placement state.

The Trocar Placement state is where the user correctly positions the robot arm for the operation with the robot teach pendant. To ensure safety of the patient, user, and robot, movement should not be conducted at a speed exceeding 20 on the teach pendant. After the robot is positioned, insertion of the tool into the patient is manually performed after pressing the 'insertion/retraction' button on the program interface (see Figure 14). This function locks

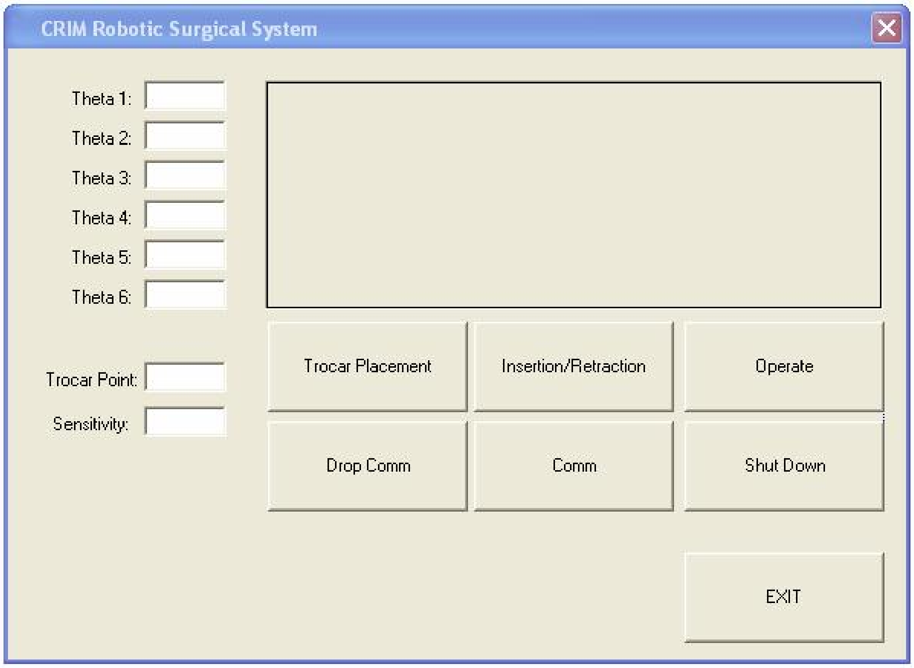


Figure 14 - Program Window

the robot arm from performing any motion and allows the user to insert the tool and attach it to the robot arm safely. To begin performing the operation, the user must type in the trocar length and press the operate button on the program interface. The program enters into the Operation state and begins to poll the two joysticks for motion commands. When the system recognizes a joystick movement, it decodes the values and communicates a motion command to the robotic system. After sending the motion command, the system continues to poll the joysticks for a new motion command. At the completion of the operation or at anytime during the operation, the endoscopic tool can be removed by pressing Insertion/Retraction. This puts the system into the insertion/retraction state which prevents the robotic system from making any motions based on the joysticks or teach pendant. The tool can then be manually detached from the robot arm and removed from the patient. At this state the user will be able to press the Trocar Placement button allowing for control of the robot via the robot teach pendant. It is at this state that the Shut Down button can be pressed which will perform the docking procedure for the robot arm.   An illustration of the program flow is given in Figure 15.

To ensure safety of the device the program interface will deactivate buttons if it is not possible to use them at the current state of the surgical system. For example, the Shut Down button will not be active during the operation state because pressing the button will cause the robot to move in ways that are dangerous to the patient. Additional fields were added to the program interface window to allow for additional functionality. A text box is added so that information can be passed to the user of the system such as current joint angles, state of the system, and errors. Two buttons, Drop Comm and Comm, had to be added to compensate for problems in the robot system communication. After some time passes in the Operation state, communication is dropped between the computer and robot controller. To continue working in the operation state, pressing Drop Comm ensures the communication is broken and pressing Comm will reestablish communication with the robot controller.

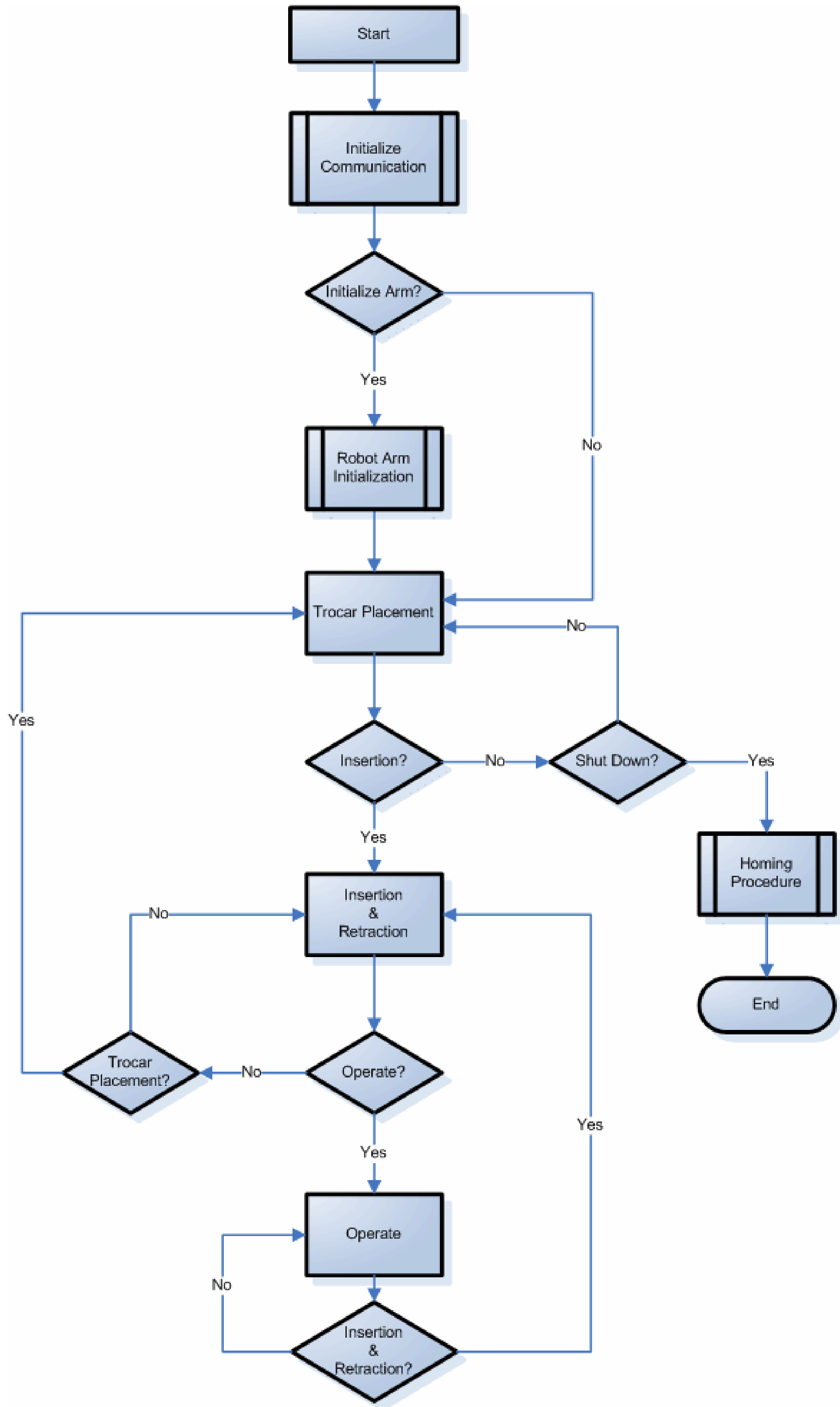


Figure 15 - Program Flow

3.2 – Robotic System

The robotic system is the slave of the master/slave telemanipulator system providing all the physical motion that a surgeon would perform during an endoscopic procedure. The motion that this system provides is separated into two subsystems: the robotic arm and the surgical manipulator.

3.2.1 – Robotic Arm

The robotic system used to position the endoscopic tool is an A250 CRS Plus Small Industrial Robot System. It consists of an articulated 5 degree of freedom robot arm mounted on a linear axis and a RSC Robot System Controller. The robot arm is driven by D.C. servos and position feedback is provided by digital optical encoders with a resolution of 0.005 degrees for joints 1-3, 0.023 degrees for joint 4, and 0.045 degrees for joint 5. Information regarding the linear axis resolution has not been obtainable. The controller contains an Intel 8086/NEC V30 microprocessor and communication is through an RS232 interface. More specifications of the robot arm and the system controller can be found in the appendix. The robots world frame consists of the table that is built around it. It knows its position in space relative to this table.

There are two types of motion that the robot arm must perform: trocar positioning and operative. To perform these motions the forward kinematic equations for the robot arm must be derived along with a solution to each joint angle and the linear position. Then algorithms must be found to move the robot in each of the different types of motion.

Forward Kinematic Equations

To perform the movements required, the kinematic equations for the arm were derived using the Denavit-Hartenberg (D-H) notation. First, a reference frame was placed at joints 1 thru 4 of the arm and at the location of the trocar. A frame was placed at the trocar point so that its position and orientation could be used to perform the trocar kinematics which will be discussed later. A frame was not assigned to joint 5 because the task of the robot arm is to correctly position the endoscopic manipulator in space relative to the world frame. To explain the reason for this the motion at the trocar point will be explained in further detail. If we look at the kinematics performed at the trocar point forward we see that correct positioning of the tool requires rotation at the trocar and translation of the tool. This type of motion is similar to a cylindrical polar robot which has two rotations about its pivot point and has an arm that translates to the desired point in its workspace (see Figure 16). From this we can see that a rotation of the tool using joint 5 is not needed to *position* the manipulator and is excluded to allow for derivation of the trocar kinematics. The correct *orientation* of the manipulator can be carried out by the functions provided by tool attachment.

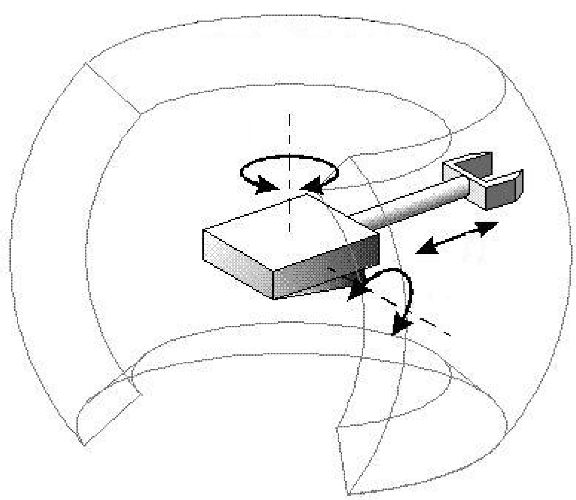


Figure 16 - Spherical Polar Robot

After the reference frames were assigned, a D-H parameter table which relates frame {*i*} to {*i+1*} was created. In the sketch below the variables L2, L3, and L4 are equal to 10 inches. The variables L5 and L6 can change to reposition the endoscopic tool. There is a 2 inch d_i value associated with the last frame due to the tool attachment which is not illustrated below but can be seen in Figure 19. The transformation matrix relating the robot base to the frame at the trocar point was then found by multiplying each individual link's transform.

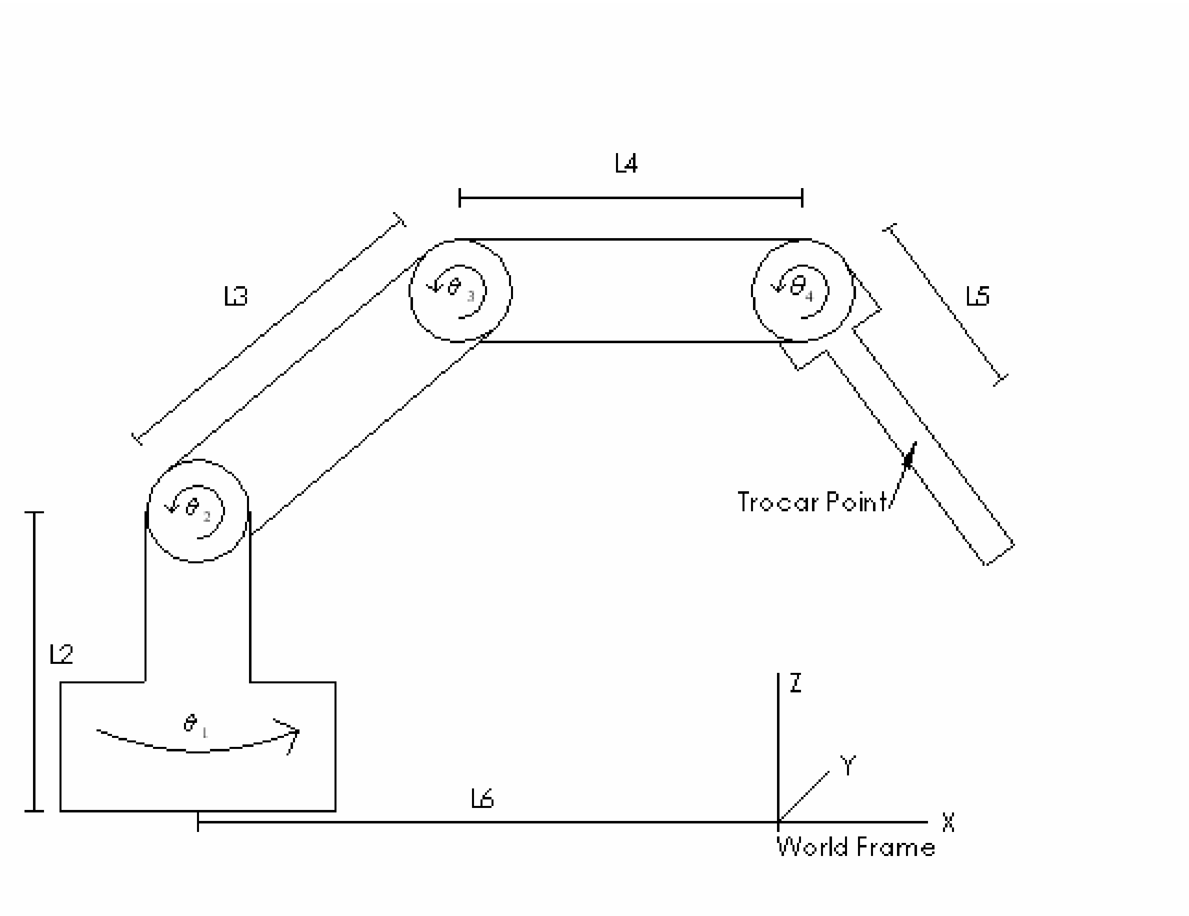


Figure 17 - Sketch of Robot Configuration

i	**α_{i-1}**	**a_{i-1}**	**θ_i**	**d_i**
1	0	L6	2_1	0
2	0	0	0	L2
3	0	+90	0	0
4	0	0	2_2	0
5	0	L3	2_3	0
6	0	L4	2_4	0
7	0	L5	0	0
8	0	-90	0	0
9	0	0	0	2

Figure 18 - Denavit-Hartenberg Parameter Table

$$T = \begin{bmatrix} \text{c1c234} & \text{-s1} & \text{-c1s234} & L5\text{c1c234} + 10\text{c1c23} + 10\text{c1c2} - 2\text{c1s234} - L6 \\ \text{s1c234} & \text{c1} & \text{-s1s234} & L5\text{s1c234} + 10\text{s1c23} + 10\text{s1c2} - 2\text{s1s234} \\ \text{s234} & 0 & \text{c234} & L5\text{s234} + 10\text{s23} + 10\text{s2} + 10 + 2\text{c234} \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

With the knowledge of the transformation matrix, the individual joint angles can be calculated using the following equations:

$$q_1 = atan2(T_{21}, T_{11})$$

$$q_{234} = atan2(T_{31}, T_{33})$$

Rearranging the equations for p_x and p_y gives:

$$c23 = -\frac{\left(L5s1c234 + 10s1c2 - 2s1s234 - T_{24}\right)}{10s1}$$

$$s23 = -\frac{\left(L5s234 + 10s2 + 10 + 2c234 - T_{34}\right)}{10}$$

These equations are then substituted into $(s23)^2 + (c23)^2 = 1$ which gives an equation with q_2 as the only unknown. Rearranging of the terms gives us:

$$a\cos q_2 + b\sin q_2 = c$$

$$a = \frac{20L5T_{21} + 40T_{23} - 20T_{24}}{s1}$$

$$b = 20L5T_{31} + 200 + 40T_{33} - 20T_{34}$$

$$\begin{aligned} c = &\left(-(L5T_{21})^2 - (2T_{23})^2 - T_{24}{}^2 - 4L5T_{23}T_{21} + 2L5T_{21}T_{24} + 4T_{23}T_{24}\right) * s1^{-2} \\ &-(L5T_{31})^2 - (2T_{33})^2 - T_{34}{}^2 - 20L5T_{31} - 4L5T_{31}T_{33} + 2L5T_{31}T_{34} \\ &-40T_{33} + 20T_{34} + 4T_{33}T_{34} - 100 \end{aligned}$$

Using this format q_2 can be solved using the formula:

$$q_2 = atan2\left(b,a\right) \pm atan2\left(\sqrt{a^2 + b^2 - c^2}, c\right)$$

The rest of the joint angles can then be solved as follows:

$$q_3 = atan2(s23, c23) - q_2$$

$$q_4 = atan2(s234, c234) - q_2 - q_3$$

$$L6 = L5c1c234 + 10c1c23 + 10c1c2 - 2c1s234 - T_{14}$$

Trocar Placement

Motion during the Trocar Position state involves using the robot teach pendant to position the endoscopic tool at the location of the trocar (see section 3.1.2). At this time motion commands are communicated directly to the robot systems controller from the teach pendant without the computer's aid. Since the motion commands at this time are handled by the teach pendant there is no need to derive equations to perform this motion.

Operative Control

To perform the operative control (also called trocar kinematics) some facts must be stated. The robot must maintain a constant point at which the tool enters the patient (*trocar* *point*). After the surgeon inserts the endoscopic tool into the patient, the robotic system knows every joint angle and the trocar length (L5) needed to solve the transformation matrix equations. From the transformation matrix the ***trocar point*** and orientation relative to the world frame is known. This information is valuable because the X, Y, and Z displacements of the endoscopic manipulator will be relative to the world frame.

$$ {}^{0}_{10}T = \begin{bmatrix} \text{c1c234} & \text{-s1} & \text{-c1s234} & p_x \\ \text{s1c234} & \text{c1} & \text{-s1s234} & p_y \\ \text{s234} & 0 & \text{c234} & p_z \\ 0 & 0 & 0 & 1 \end{bmatrix} $$

The frame located at the trocar point lies on the endoscopic tool shaft (the point which is desired to remain static). It is oriented with the X axis pointing in the direction of the end manipulator. Therefore, the manipulators point in space, relative to the world frame, can be found by a translation of the total tools length minus the trocar length (L5) along the X axis. It should be noted that the manipulator frame will have the same orientation as the trocar frame since only a translation has occurred and they both lie on a rigid link.

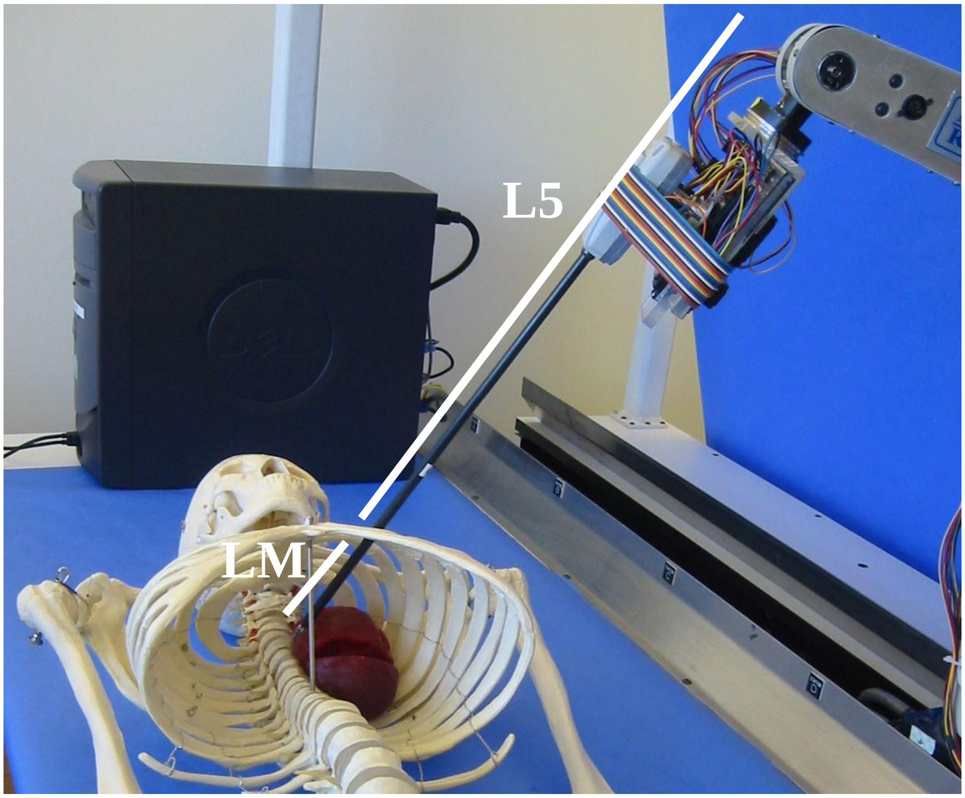


Figure 19 - Distance Labels on Endoscopic Tool

$$LM = ToolLength - L5$$

$$P_m = {}_{10}^{0}T * \begin{bmatrix} LM & 0 & 0 & 1 \end{bmatrix}^T$$
$$= \begin{bmatrix} p_{mx} & p_{my} & p_{mz} & 1 \end{bmatrix}^T$$

Changing the position of the manipulator would give us a *new* position:

$$P_{mn} = \begin{bmatrix} p_{mx} + d_x & p_{my} + d_y & p_{mz} + d_z \end{bmatrix}^T$$

Where d_x, d_y, and d_z are the relative change in the world frame. Using the equation previously used to get the manipulator point, the new orientation of the trocar point frame can be solved for with a new manipulator point.

$$P_{mn} = {}_{10}^{0}T * \begin{bmatrix} LM & 0 & 0 & 1 \end{bmatrix}^T$$

$$\begin{bmatrix} p_{mx}+d_x \\ p_{my}+d_y \\ p_{mz}+d_z \\ 1 \end{bmatrix} = \begin{bmatrix} LMc1c234+p_x \\ LMs1c234+p_y \\ LMs234+p_z \\ 1 \end{bmatrix}$$

Rearranging these equations by putting known values on the left and unknown values on the right gives:

$$p_{mx}+d_x-p_x=LMc1c234$$

$$p_{my}+d_y-p_y=LMs1c234$$

$$p_{mz}+d_z-p_z=LMs234$$

It should be noted that p_x, p_y, and p_z are known because they represent the trocar's location in space which remains constant. From these three equations, the following values can be derived:

$$q_1=atan2((p_{my}+d_y-p_y),(p_{mx}+d_x-p_x))$$

$$q_{234}=atan2((p_{mz}+d_z-p_z),((p_{my}+d_y-p_y)/s1))$$

$$LM=\left(p_{mz}+d_z-p_z\right)/s234$$

$$LM=\left(p_{my}+d_y-p_y\right)/\left(s1c234\right)$$

These equations were analyzed in order to ensure that a divide-by-zero does not result. The equation for q_{234} has an s1 in the denominator which does not constitute a problem. The orientation of the robot would not allow q_1 to become 0 or 180 degrees during an operation. When calculating LM, a check must be done to ensure that a divide-by-zero does not occur. If s234 equals zero, then the next equation with s1c234 should be used. This switch is possible because s1 will not be a problem as previously discussed, and if s234 is zero then c234 equals 1.

Given the new manipulator position and the values derived in the previous equations, the new orientation of the trocar frame can be defined. In most cases a new value for the trocar length (L5) must also be redefined after a movement of the manipulator. This is possible by knowing the relationship between L5 and LM.

$$T_{10}^{0} = \begin{bmatrix} \text{c1c234} & \text{-s1} & \text{-c1s234} & p_x \\ \text{s1c234} & \text{c1} & \text{-s1s234} & p_y \\ \text{s234} & 0 & \text{c234} & p_z \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$L5 = ToolLength - LM$$

After redefining the transformation matrix and L5, equating the new joint angles of robot is possible by using the equations previously defined. These new joint angles would then be sent to the robot controller.

3.2.2 – Surgical Manipulator

The surgical manipulator consists of two parts: the endoscopic tool and the tool attachment. The endoscopic tool is an EndoWrist Needle Driver 8mm developed by Intuitive Surgical for the daVinci Surgical System. This tool is designed to provide a surgeon with more degrees of freedom than classic endoscopic tools and it is designed to mimic the movement of the human hand (see Figure 5). The motion of the tool's manipulator is purely mechanical and is performed by rotating discs located on the handle side of the tool (opposite the manipulator). These discs can perform five different motions: Pitch, Yaw, Roll, Open Grip, and Close Grip.

Manipulator Motion	Disc
Roll	Disc 1
Yaw	Disc 3 and Disc 4 (Both CCW or Both CW)
Pitch	Disc 2
Open Grip	Disc 3 (CW) and Disc 4 (CCW)
Close Grip	Disc 3 (CCW) and Disc 4 (CW)

Figure 20 - Function of EndoWrist Discs

The tool attachment controls the motion of the endoscopic tool manipulator by rotating these discs. Rotation is executed by three Hitec HS-422 hobby servo motors and a Parallax S-148 continuous rotation servo motor. Two different motors were used because of the degree at which each disc can rotate. Discs 2 through 4 rotate less than ± 90 degrees which allows standard hobby servo motors to maintain there position. Disc 1 rotates approximately ± 180 degrees which is not possible with a standard servo motor therefore a continuous rotation servo is used to rotate this disc. To maintain knowledge of this servos position, a Nubotics WW-01 servo encoder was added. Connection from the servo motors to the EndoWrist was performed by an aluminum mounting hub which rotated the shaft of an EndoWrist disc attachment specially manufactured for this prototype.

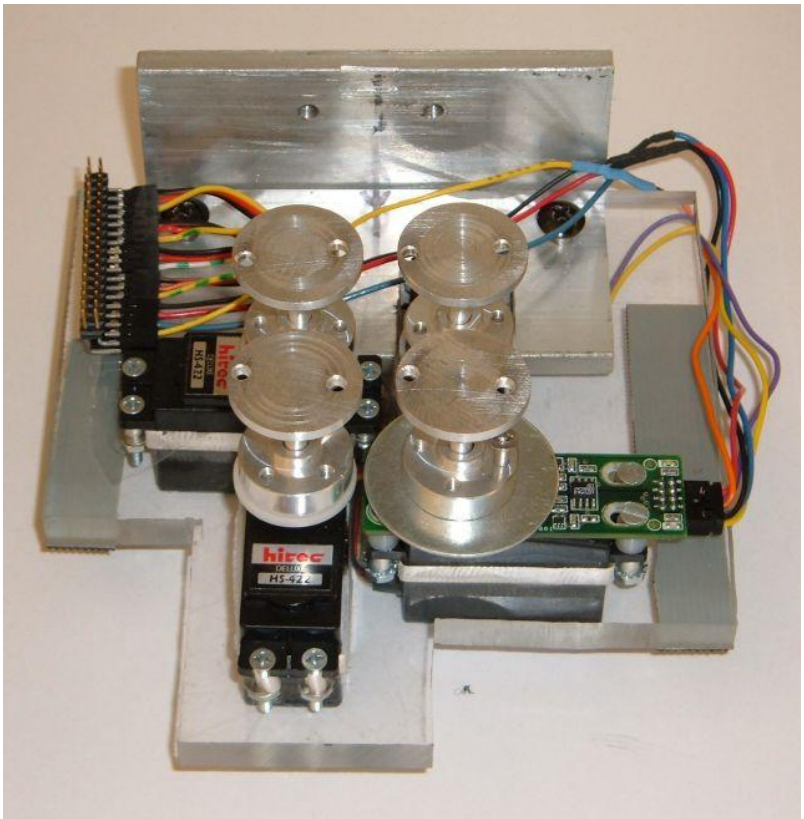


Figure 21 - Tool Attachment

Figure 22 - EndoWrist Attachment and Aluminum Hub Mount

The rotational position of the servo motors is driven by an Acroname BrainStem GP 1.0 module which communicates with the computer through a USB serial interface connector.

The processor is a Microchip PIC18C252 which runs at 40MHz. This module was chosen because it has 4 high-resolution servo outputs, digital I/O pins, separate logic and servo power supply, and serial port communication.

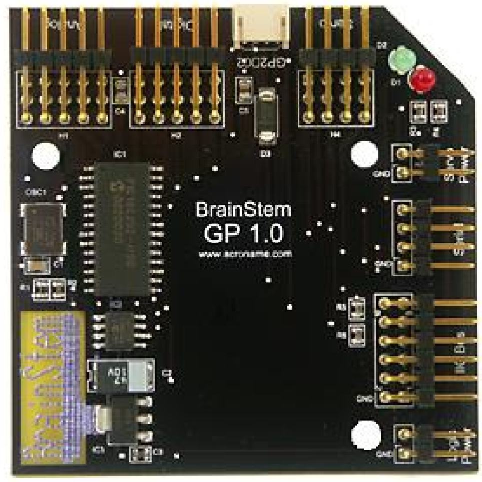


Figure 23 - Acroname BrainStem GP 1.0

Chapter 4 – Analysis, Simulation, and Experimentation

The surgical robotic system that was developed in this project has many of the benefits that other similar surgical systems have such as the elimination of tremor and the fulcrum/lever effect. These benefits are inherited in any system using a robotic arm to hold the endoscopic tool. Therefore an analysis of the system will be centered on its ability to provide precise and reliable motion.

4.1 – Tool Attachment

The tool attachment developed and manufactured for this system provided the ability to control an Intuitive Surgical EndoWrist manipulator improving the manipulative ability of traditional endoscopic surgery. Controlling every function of the tool was possible through the Human-Machine interface with very little delay in its response. Position commands were sent to the BrainStem at a rate of 30 Hz allowing for real time positioning of the servos. The delay was seen in the response of the standard servos which can rotate 60 degrees in 0.21/0.16 seconds at 4.8/6.0 volts. This would involve a little under a second to complete a full rotation however this type of motion most likely would not be seen in a surgical procedure.

As mentioned before, this device performs all the motions that would be needed in a surgical procedure however the capability of the system is not suitable for surgery. The three common hobby servo motors do not have much precision in positioning which is needed in this part of the system. They may create the need to reorient the position of the robot arm if the motors can not move the manipulator to reach a specific location. The continuous rotation servo and optical encoder circuitry would also cause problems. A problem with this motor is that modified servo motors for continuous rotation typically have drifting trouble. Any motor that may drift and change its orientation during a procedure is not acceptable. The optical encoder circuitry used for this motor is not suited for its intended purpose.

Currently, the system performs a function without taking other factors into consideration. As mentioned before, motion of the wrist is performed by rotating Disc 2 but it was observed that rotating Disc 2 while holding the other Discs constant would also perform a rotation of the fingers. This is not taken into consideration and the user would have to compensate for this with the joystick. Also, the current system does not take into consideration the orientation of the manipulator which could damage the tool. If the manipulator is positioned at a certain extreme position then any other motion performed does not take this into account. The problem with this can be seen with the motion of the wrist. If the fingers are at an extreme position, then rotation of the wrist could force the fingers toward a position that is not possible damaging the EndoWrist.

4.2 – Robot Arm/System

It was found that the robot arm used for this system is capable of performing the basic motions required for changing the endoscopic tool position while maintaining a constant trocar point. Since the primary concentration in this thesis was to use a linear base articulated arm to perform trocar kinematics for endoscopic surgery, a detailed analysis into its performance was conducted.

4.2.1 – Simulation

Before any control of the arm was performed, the kinematic equations and the control algorithm were simulated using Matlab (see appendix). This simulation calculates all the joint values and the linear axis position relative to the world coordinate frame. These values were then used with the individual link transforms derived in Chapter 2 to present a visual graphic of the robot and endoscopic tool's position and orientation. An example of this is shown in the image below where three different robot orientations are shown which perform a displacement of the manipulator along the Y axis.

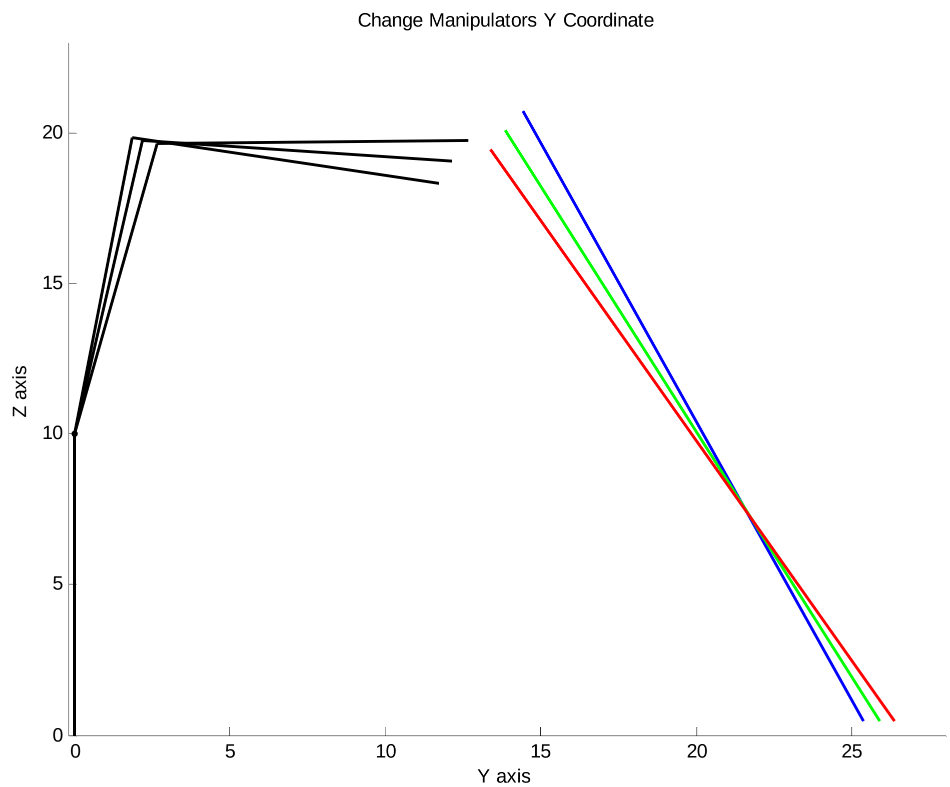


Figure 24 - Matlab Simulation Output

The colored lines represent the endoscopic tool which is not directly attached to the robot flange because the tool attachment creates a 2" Z axis offset relative to the frame at joint 4. It can be seen that the endoscopic tool maintains a constant trocar point while performing the desired motion.

At the initial design stages this program was used to verify that a constant trocar point was maintained after the desired motion was performed. Analyzing the position and orientation of the robot arm was also conducted to verify that the robot was capable of performing the motions. After a complete understanding of the robot kinematics and the trocar kinematics designed to move the endoscope, the computer was programmed to control the robot arm. As a precaution, before any commands were sent from the computer to the robot arm, the simulation environment was used to verify that the program was calculating the correct values. Subsequently, the simulation environment was used to verify that at a new orientation and trocar length (L5) changing the position of the endoscopic manipulator would not cause problems such as illegal joint positions. After implementing control of the robot arm with the computer program, analysis of the robot arms performance was conducted.

4.2.2 – Analysis of Motion

Experiments were conducted to verify that the control algorithm would command the robot to the correct position while maintaining a constant trocar point. Before this was conducted, the accuracy and precision of the robotic system was analyzed. This was done so that it would be known if errors in the surgical systems motion could be attributed to the robot arm. Problems were found with the linear axis and the actual position the robot controller declared after a command. It was found that the linear axis was not moving the commanded distance but moved with an approximate ratio of 18/18.875”. This was compensated for in the computer program but could cause some small errors in positioning.

The second problem that was observed with the robot is that after sending a motion command the joint angles would not move to the exact position but there would be an error associated with the new position. Error in joint angle position will be evident in any system but an analysis was conducted to verify that it would not effect the final position of the endoscopic manipulator to any considerable degree. Using the simulation environment, 40 different motion commands were sent to the robot controller. The average error and maximum error in each joint angle was recorded and can be seen in Figure 25. It was found that with a maximum error in each joint there would be an error in the manipulator position of approximately (0.0055, 0.0035, 0.0032).

	Joint 1	Joint 2	Joint 3	Joint 4	Joint5	Linear axis
Average Error	0.004953	0.00322	0.00165	0.003117	0.00235	0.000596
Max Error	0.0104	0.0050	0.0038	0.0103	0.0451	0.0177

Figure 25 - Average and Max Error of Robot Controller

After it was known that the robot has minimal error associated with its positioning, analysis was conducted on the kinematics developed in this thesis. The first experiment that was conducted involved repositioning the endoscopic manipulator in the X, Y, and Z directions to verify that the system would move a specified distance with minimal error.

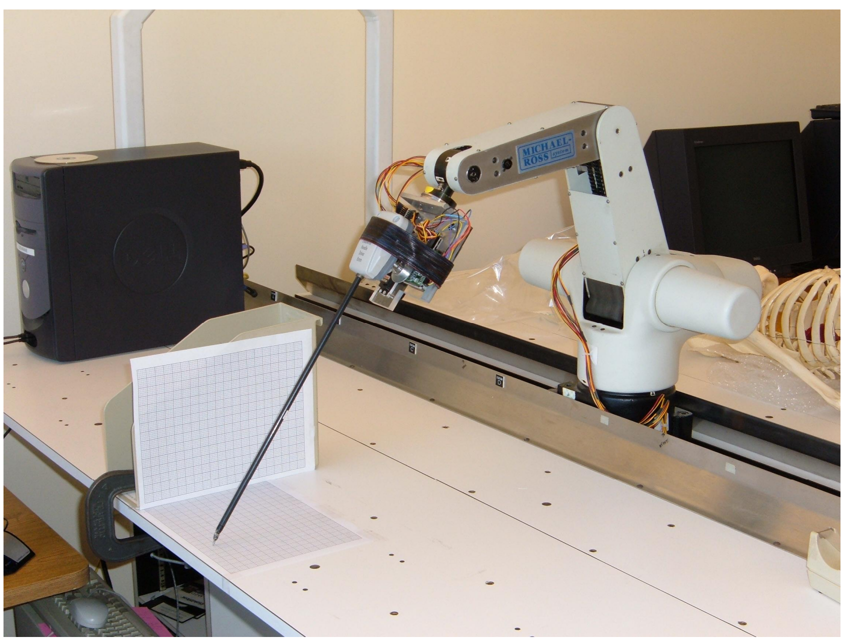


Figure 26 - Experiemental Robot Position

Experiments were conducted on the precision of the robots motion by placing graphic paper with 0.1 inch lines on the XY and YZ axis of the robot with the lines parallel to the robot axis. Verification of the graphic papers alignment was conducted using the robotic arm. The endoscopic manipulator was then placed very near to one of these planes and commanded to move 1 inch in the XY or YZ plane depending on its current position. This distance was chosen because it was decided that any error associated with a movement would become more observable as the distance increased. Experiments were conducted on smaller motions (approximately 0.1 inch) but there was no observable error. Further analysis on smaller motions would give the resolution of the system but they were not conducted due to the lack of precise measuring equipment.

Each movement was performed and recorded using three different trocar lengths: 9, 11.5, and 14 inches. This was done because the amount of motion that the robot must perform to conduct a movement will depend on this distance (see Figure 27). The three trocar lengths chosen represent three important locations on the length of the endoscopic tool. The 9 inch length is a distance that is closer to the robot flange than the center of the tool length. This would mean that the robot would only have to perform small motions to move the manipulator a large distance. The 11.5 inch distance is in the middle of the tool length which requires the robot end to move the same distance as the manipulator end. The 14 inch length is past the middle of the tool from the robot flange and represents the robot flange moving a greater distance than the manipulator. Examples of the images used to record this data are presented below in Figure 29 – Figure 32.

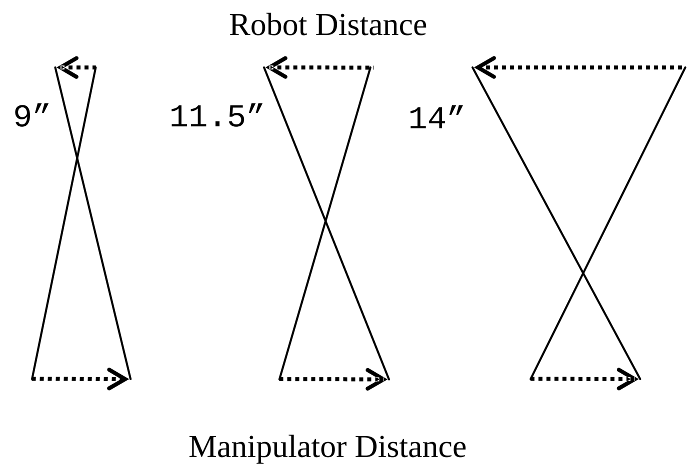


Figure 27 - Robot vs. Manipulator Distance

To find the error in each movement the pixels in each image were measured from a common point on the graph paper to the center of the manipulator. To convert the pixel value to inches, the number of pixels per 0.1 inch on the graph paper was measured. This was done on each image to account for any change that could occur from the different camera angles. This displacement from the common point was then used to determine the error associated with each movement in the XY or YZ plane.

	Average Abs. Error X	**Average Error Abs. Y**	**Average Error Abs. Z**
TL = 9”	(0.004, 0.023, 0)	(0.0 26, 0.027, 0)	(0, 0.006, 0.013)
TL = 11.5”	(0.013, 0.023, 0)	(0.029, 0.027, 0)	(0, 0.006, 0.014)
TL = 14”	(0.012, 0.033, 0)	(0.037, 0.027, 0)	(0, 0.014, 0.008)

Figure 28 - Average Absolute Error

From the results it can be seen that the system is controlled with reasonable accuracy and that there is no direct correlation between the manipulator error and trocar length. Limitations of the system such as manufacturing imprecision, joint position errors from the controller, robot actuation, and human errors account for the error seen in positioning.

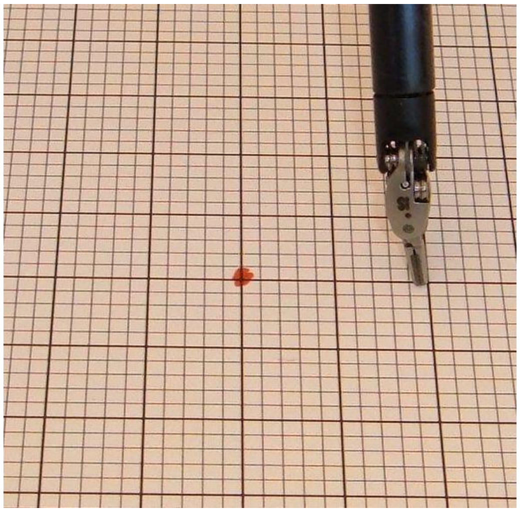


Figure 29 - X Displacement in XY Plane

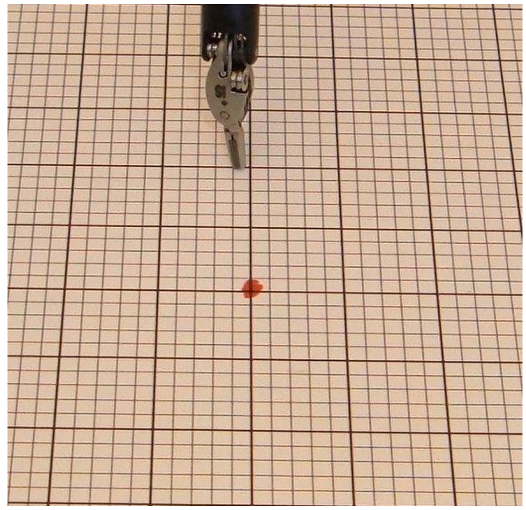


Figure 30 - Y Displacement in XY Plane

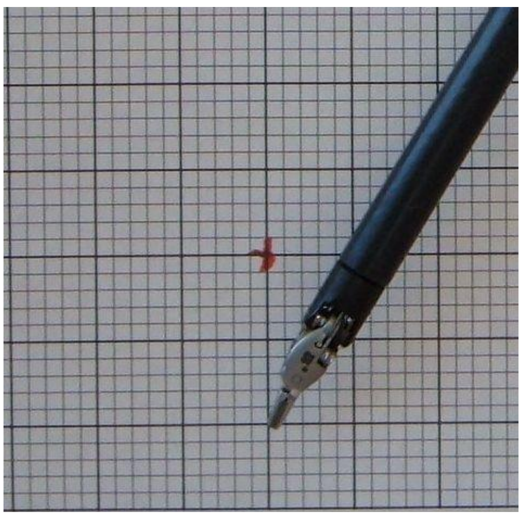


Figure 31 - Z Displacement in YZ Plane

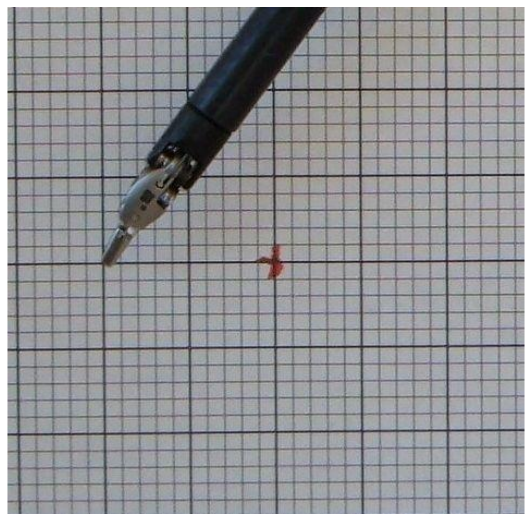


Figure 32 - Y Displacement in YZ Plane

The second experiment was verifying that the robot maintained a constant trocar point as it moved the manipulators position. This was conducted similar to the previous experiment. The endoscopic tool was aligned with the graphic paper on the YZ plane and motions were performed while observing that the trocar point is maintained (see Figure 33 and Figure 34). From video and the images captured after each movement it can be seen that the robot does keep the trocar point constant.

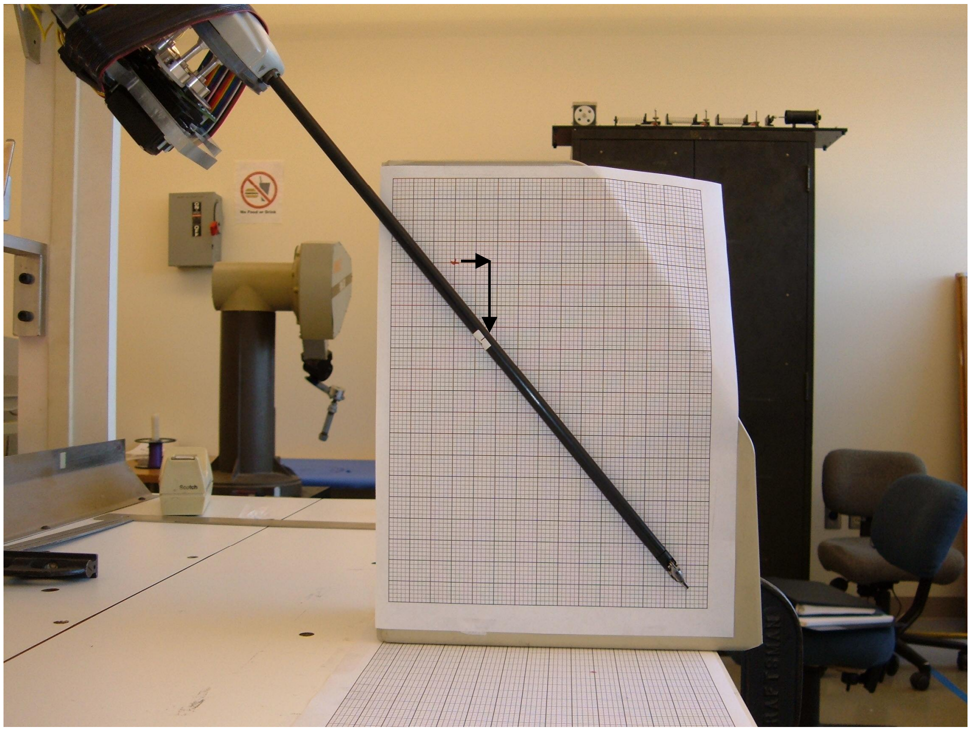


Figure 33 - Trocar Point

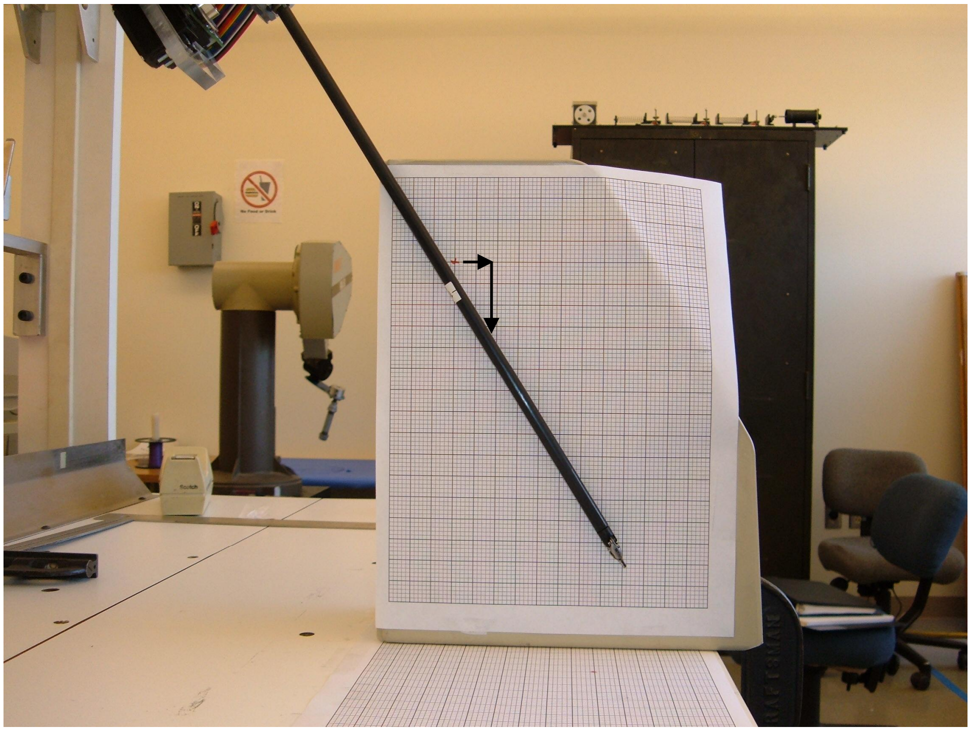


Figure 34 - Trocar Point after Repositioning

For a complete analysis, the actual motion that the robot arm performs during repositioning should also be taken into consideration. Motion commands that are sent specify an absolute angle of each joint and a position on the linear axis. After the controller receives the command, it calculates a velocity for each joint angle that would cause them to all reach the new angle/position at the same time. This results in smooth motion with the robot manipulator traveling in an arc. The system does offer a straight line command however using this command results in very fast jerky motion. It was decided that straight line travel was not needed at this time and that the arcs in travel do not cause errors in positioning. Another observation to make in the robot arm is the ‘sticky’ type motion of joint 4. This creates error in maintaining the trocar point during the actual motion however from the results it can be seen that it does not affect the final position. The sticky motion in joint 4 is a mechanical problem and does not reveal a problem in the system control.

The results of these experiments reveal that accurate control of the CRS Robotic System is possible but it should be noted that the original intention of the system is not as a master/slave telemanipulator. This system was intended, under normal operation, to perform a preprogrammed set of tasks stored in the controller’s memory and not processing real time motion commands. It was found that sending motion commands over the RS232 interface to quickly will overwrite the robot controller memory buffer corrupting the data. Due to this the current system does not control the robot arm in real-time but sends delayed commands so that a motion can be executed and cleared from memory before a new one is sent.

Chapter 5 – Conclusions and Future Research

The main goal of this thesis was to design a robotic surgical system using a linear base articulated arm for positioning an endoscopic tool. Additional goals were to create a device that would provide functionality to an Intuitive Surgical EndoWrist and a human-machine interface to control the surgical system. The system described is a demonstration of using a unique robot arm configuration to perform this motion however it is not intended for an actual surgical operation. Robotic arms with similar configurations have also been used as surgical robots [13, 22] but I have not seen any that have a configuration such as the one presented here.

Analysis of the system showed that repositioning the endoscopic manipulator is possible to within 0.05 inches. This algorithm was able to maintain the trocar point however error was observed during repositioning. This error is due to the robot arm actuation and presents itself as a limitation of the system. This research presents a unique robot arm configuration for endoscopic surgery with precise localization and opens the door for several new and interesting future research topics.

5.1 – Future Research

Using this system, it may be possible to create an intelligent surgical operating system. As previously mentioned, this robotic system knows its relative position in the world frame. If the system is then provided with real time medical imaging to gain information about the relative position of objects in its workspace, then it is possible that the system could be commanded to perform small tasks autonomously.

Another research topic would involve further analysis on the different positions and orientations of this robot configuration. Provided with the kinematics for endoscopic surgery in this thesis research can be conducted into relative positioning of the patient and the trocar to the robot arm. Such research would bring into view the robots workspace and positioning limitations. It may also reveal new orientations of an endoscopic tool that are capable with this design which previously have not been possible.

Future research can also be conducted into a precise control system for manipulating and Intuitive Surgical EndoWrist. If the internal mechanics of this tool were completely understood then precise positioning of the manipulator would be possible. This would involve the combination of the internal mechanics model and a kinematic model of the manipulator relative to the wrist joint.

Research could also be conducted on the control required to insert the endoscopic tool with the robot arm.

References

[1] Adler, John R., Jr., et al. "Image-guided Robotic Radiosurgery." Neurosurgery 44.6 (1999): 1299-1306.

[2] Allmen, Daniel Von. "Physical constraints in the application of robotics to pediatric surgery." Vice Chancellor's Workshops in Interdisciplinary Research: Robotics: The Good, the Bad, and the Ugly. Chapel Hill, 1 Feb. 2006.

[3] Ben-Porat, Ori, Moshe Shoham, and Joachim Meyer. "Control Design and Task Performance in Endoscopic Teleoperation." Presence 9.3 (2000): 256-267.

[4] Camarillo, David B., Thomas M. Krummel, and J. Kenneth Salisbury, Jr. "Robotic Technology in Surgery: Past, Present, and Future." The American Journal of Surgery 188 (Suppl to Oct. 2004): 2S-15S.

[5] Chang, Steven D., et al. "An Analysis of the Accuracy of the CyberKnife: A Robotic Frameless Stereotactic Radiosurgical System." Neurosurgery 52 (2003): 140-147.

[6] Chitwood, W. Randolph, Jr. "Robotic Cardiac Surgery: From Technology to Patient Care." The Honor Society of Phi Kappa Phi North Carolina State University Chapter Spring 2006 Lecture. Raleigh, 9 Feb. 2006.

[7] Cleary, Kevin, and Charles Nguyen. "State of the Art in Surgical Robotics: Clinical Applications and Technology Challenges." Computer Aided Surgery 6 (2001): 312-328.

[8] Dario, Paolo, Blake Hannaford, and Arianna Menciassi. "Smart Surgical Tools and Augmenting Devices." IEEE Transactions on Robotics and Automation 19.5 (2003): 782-792.

[9] Griffin, R. Morgan. "Surgery Lite: Understanding Endoscopic Surgery." WebMD Feature 26 Oct. 2005 http://www.webmd.com/content/Article/114/111263.htm.

[10] Hanly, Eric J. "Multiservice Laparoscopic Surgical Training using the daVinci Surgical System." The American Journal of Surgery 187 (2004): 309-315.

[11] Jakopec, M., et al. "The First Clinical Application of a "Hands-On" Robotic Knee Surgery System." Computer Aided Surgery 6 (2001): 329-339.

[12] Marohn, Michael R., and Eric J. Hanly. "Twenty-First Century Surgery Using Twenty-First Century Technology: Surgical Robotics." Current Surgery 61.5 (2004): 466-473.

[13] Mayer, Hermann, et al. "An Experimental System for Robotic Heart Surgery." Proceedings. 18th IEEE Symposium on Computer-Based Medical Systems (2005): 55-60.

[14] Pransky, Joanne. "Surgeons' Realizations of RoboDoc." Industrial Robot 25.2 (1998): 105-108.

[15] Schneider, Olivier, and Jocelyne Troccaz. "A Six-Degree-of-Freedom Passive Arm with Dynamic Constraints (PADyC) for Cardiac Surgery Application: Preliminary Experiments." Computer Aided Surgery 6 (2001): 340-351.

[16] Stevens, Jason M., and Gregory D. Buckner. "Actuation and Control Strategies for Miniature Robotic Surgical Systems." Journal of Dynamic Systems, Measurement, and Control 127 (2005): 537-549.

[17] Taylor, Russel, et al. "A Steady-Hand Robotic System for Microsurgical Augmentation." The International Journal of Robotics Research 18.12 (1999): 1201-1210.

[18] Taylor, Russel H., et al. "An Image-Directed Robotic System for Precise Orthopaedic Surgery." IEEE Transactions on Robotics and Automation 10.3 (1994): 261-274.

[19] Taylor, Russel H., and Dan Stoianovici. "Medical Robotics in Computer-Integrated Surgery." IEEE Transactions on Robotics and Automation 19.5 (2003): 765-781.

[20] "Surgery." WordNet Search – 2.1. Online.

[21] Wikipedia. Online.

[22] Venere, Emil. "Lower cost, portable surgical robots could be smooth operators." Purdue News. 2 March 2006. http://www.purdue.edu/UNS/html4ever/2006/060302.Peine.robots.html.

[23] Zenati, Marco. "Robotic Heart Surgery." Cardiology in Review 9.5 (2001): 287-294.

Appendix

Appendix A – CRS Robot Specifications

Structure	Articulated 5 DOF
Drive motor	Permanent magnet DC Servo
Bearings	ABEC Class 1 - 0.375" ID
Max voltage	+/-25 Vdc
Max current	10.8 amps
Mech. time const.	11.62 msec
Max speed @ 25V	3600 rpm
Peak torque	100 oz-in
Brush life	8000 hours @ 1200 rpm
Transmission	
Waist rotate	Harmonic drive
Shoulder	Harmonic drive
Elbow	Harmonic Drive/chain
Wrist bend (pitch)	bevel-/spur-gear/chain
Tool roll	bevel-/spur-gear/chain/gear
Payload	kg
Maximum design	2.0
Full speed/acc	1.0
Reach - Waist to tool flange	22 inches
Reach (by link)	Inches
Base to shoulder	10
Shoulder to elbow	10
Elbow to wrist pivot	10
Wrist pivot to tool flange	2
Joint travel ranges	degrees
Waist rotate	+/-175
Shoulder	+110, -0
Elbow	+0, -130
Wrist bend (pitch)	+/-110

Figure 35 - Robot Arm Specifications (Part 1)

Joint speeds at 100% program speed	rad/sec
A150 Series:	
Waist rotate	1.74
Shoulder3	1.08
Elbow	1.74
Wrist bend (pitch)	3.14
Tool roll	6.28
A250 Series:	
Waist rotate	3.05
Shoulder3	2.18
Elbow	3.05
Wrist bend (pitch)	3.14
Tool roll	6.28

Joint default acceleration rates	rad/sec^2
A150 Series:	
Waist rotate	5.45
Shoulder	5.45
Elbow	5.45
Wrist bend (pitch)	24.54
Tool roll	49.09
A250 Series:	
Waist rotate	12.93
Shoulder	12.93
Elbow	12.93
Wrist bend (pitch)	58.18
Tool roll	116.36

Position Feedback	Optical incremental encoders
Resolution	1000 pulse/rev.
Index	Marker pulse 1 per rev.
Output	Chnls A, B, Z sq.wave TTL

Figure 36 - Robot Arm Specifications (Part 2)

Joint Resolution	deg
Waist rotate	0.005
Shoulder	0.005
Elbow	0.005
Wrist bend (pitch)	0.023
Tool roll	0.045

Joint Resolution	inches @ tool flange
Waist rotate	0.0019
Shoulder	0.0009
Elbow	0.0009
Wrist bend (pitch)	0.0008
Tool roll	0.0016

Figure 37 - Robot Arm Specifications (Part 3)

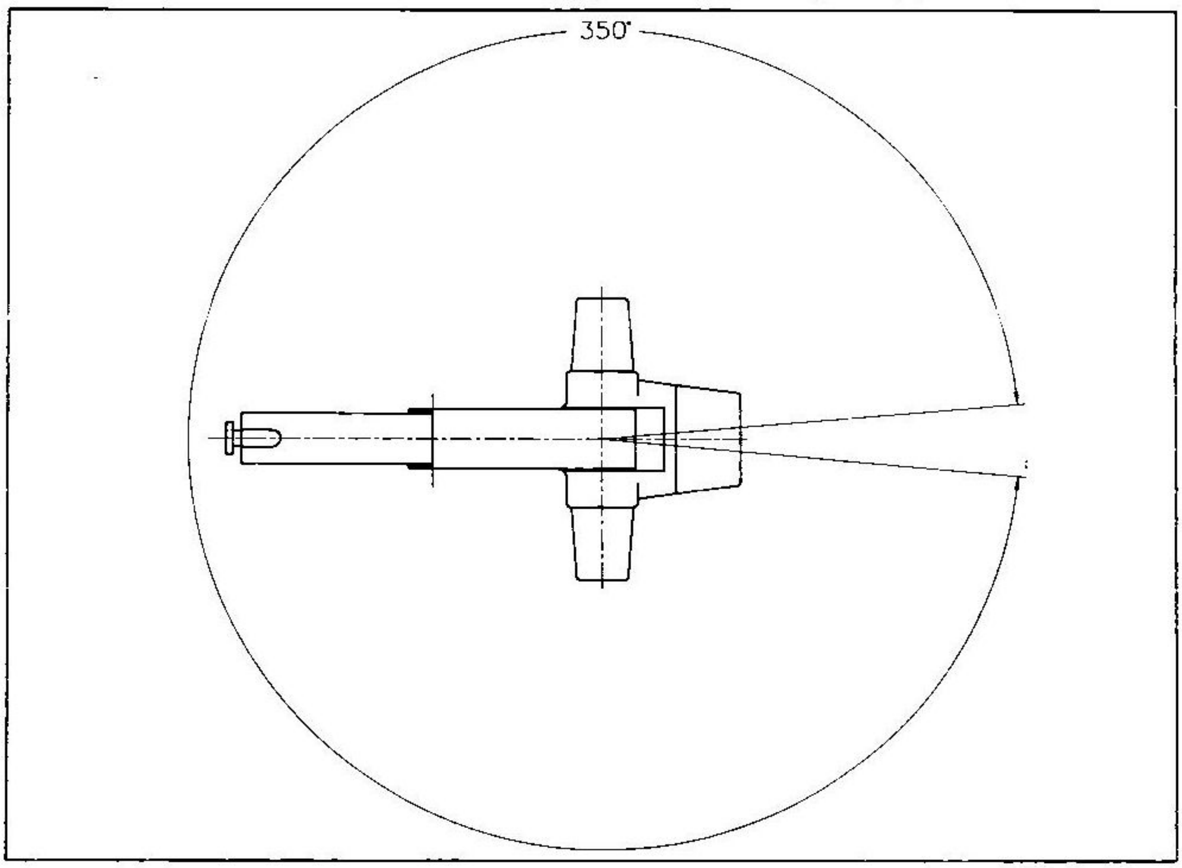


Figure 38 - Robot Workspace Plan

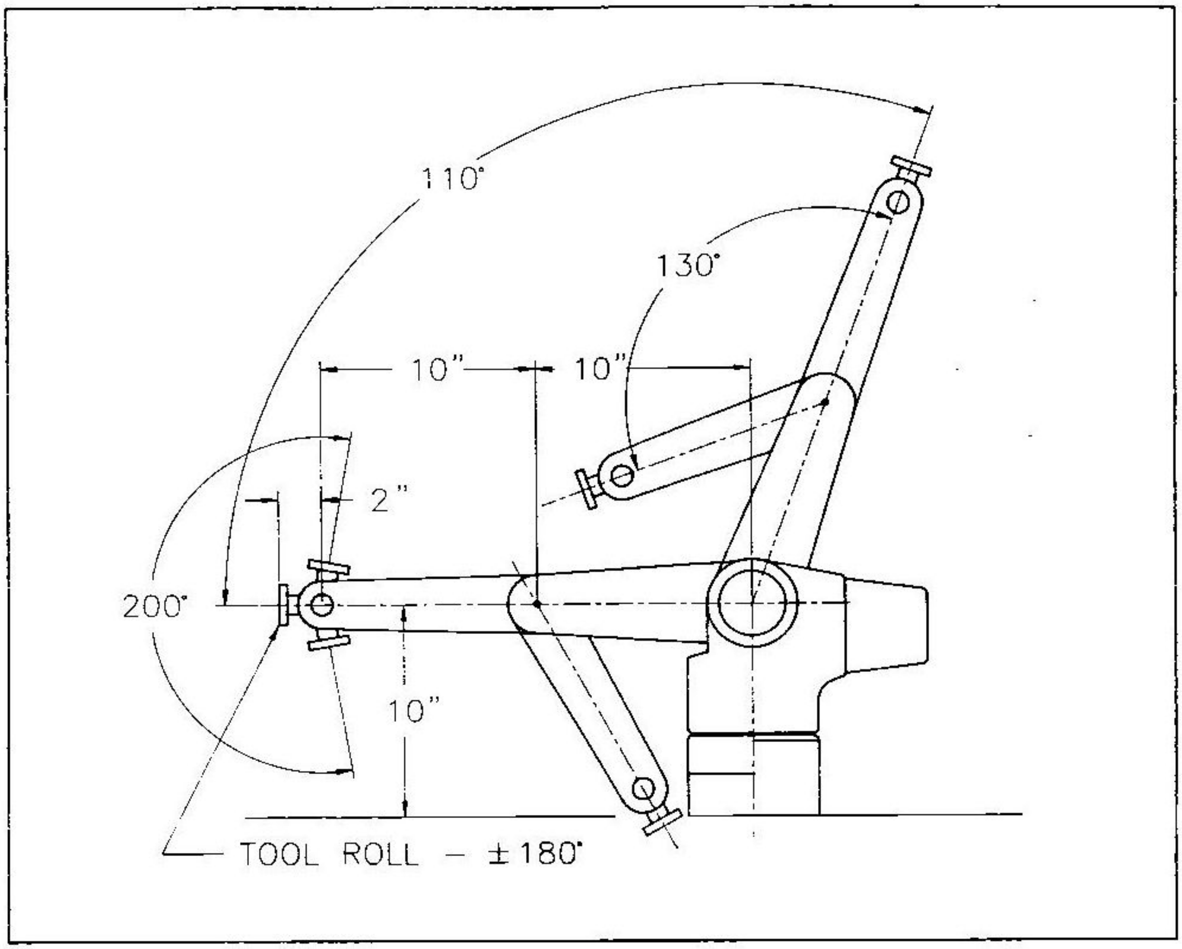


Figure 39 - Robot Workspace Elevation

HOME

TOKEN: /020

FORMAT:

[Line#] HOME

DESCRIPTION:

The HOME command aligns the arm with the mathematical model of the arm stored in the controller memory. When the robot is first turned on, the controller does not know the arm position relative to the world (as defined by the centre of the manipulator base). The operator must Home the robot to provide the synchronization between controller and arm.

This HOME command uses the encoder zero cross and the available home switch to initialize each axis.

NOTES:

1. **Before the HOME command is used, the calibration data saved by the @@CAL command must be in controller memory. This is the factory default condition.**

- Double Space or leave them like the manual?

TOKEN: /006

FORMAT:

[Line#] JOINT <JOINT #>,<DEGREES>

DESCRIPTION:

<JOINT #>
Can use constants, variable names, arrays or expressions.
<DEGREES>
Can use constants, variable names, arrays or expressions.

The joints of the robot will move the specified distance in degrees. Optional extra axes will move the specified number of units from their current location. The desired units are selected by the @XPULSES and @XRATIO commands were executed.

Robot joints will move in a joint de-coupled fashion. The joint number describes which joint is moved by the following legend:

1. **Waist**
2. **Shoulder**
3. **Elbow**
4. **Wrist Bend (Pitch)**
5. **Wrist Swivel (Roll)**

The JOINT command will execute at a speed as specified in the last SPEED command. The sense of each joint can be seen in FIGURE 2-2.

WARNINGS:

1. **Each robot joint has a limiting travel which should not be exceeded. This limit is checked prior to the execution of the JOINT command when the robot has been Homed.**
2. **Joint Interpolated moves can be done before the robot has been homed, but limits will not be checked. Care must be taken with this command until the robot is Homed.**

EXAMPLE:

1. **Move Joint #1 (Base) by +45 degrees.**

```
200 JOINT 1,45
```

TOKEN: **/132**

FORMAT:

[Line#] MA <j1>,<j2>,<j3>,<j4>,<j5>,<j6>

DESCRIPTION:

<j1>…<j5>

Can use constants, variable names, arrays and expressions.

This command moves the robot to the pose determined by the absolute joint angles specified, in Joint Interpolated motion. The joint angles are entered i9n radians.

WARNINGS:

All five angles must be entered, each separated by a comma (preferred) or a space.

TOKEN: /045

FORMAT:

[Line#] MANUAL [JOI | CYL | WORLD | TOOL]

DESCRIPTION:

The Manual mode allows the operator to move the robot using the 8 joint selector switches on the Teach pendant, along with the gripper toggle switch and the speed selector knob.

During Manual mode, all non-motion commands remain active. Note that the system prompt changes from the usual >> after the MANUAL command is entered. The system console may operate slightly slower in the Manual mode due to the extra scanning which is required for the Teach pendant.

There are two Manual modes of the robot control. In the Joint Manual mode, individual joints are controlled with each toggle switch on the Teach pendant. In Cylindrical Manual mode, the motion of joints 2 and 3 are coordinated to provide a motion of the wrist that is radial, with constant elevation, or vertical (in the Z axis), with constant radial distance from the robot base. The base rotate and the two wrist axers are controlled the same as in Joint Manual mode. Only JOINT MANUAL mode can be used before the robot ahs been Homed.

NOTES:

1. **The manual mode will lock out any motion request that may come from the keyboard. The RAPL-II error message 'ISO ERROR' will appear. This is a safety requirement that is enforced with a parameter setting.**
2. **When moving the robot in joint mode, it is possible to move the robot beyond its current POSE definition. Any subsequent motion will then move the location with the new pose value. This may produce unwanted results.**

TOKEN: **/131**

FORMAT:

[Line#] MI <j1>,<j2>,<j3>,<j4>,<j5>,<j6>

DESCRIPTION:

<j1>…<j6>

Can use constants, variable names, arrays or expressions.

This command moves the robot joints by the incremental joint angles specified, in Joint Interpolated motion. The joint angles are entered in radians.

WARNINGS:

All six angles must be entered, each separated by a comma (preferred) or a space.

TOKEN: /096

FORMAT:

NOHELP

DESCRIPTION:

This command turns the syntax building feature (Help mode) OFF. When controlling the robot through a host computer interface, the syntax builder should be turned off.

TOKEN: /046

FORMAT:

[Line#] NOMANUAL

DESCRIPTION:

This command turns off the function of the Teach Pendant. This will return the system prompt to the >> symbol. (Refer to description of MANUAL command.) Issuing any RAPL-II motion command automatically executes a NOMANUAL command.

TOKEN: /141

FORMAT:

PASSWORD <PASSWORD>

DESCRIPTION:

This command with the correct PASSWORD value permits access to the monitor, set-up commands and the diagnostic command level (refer to APPENDIX A of the Technical Manual for a complete list of these commands). After issuing the correct PASSWORD, these commands will also be displayed when the ? command is issued.

This mode may be disabled by entering an incorrect PASSWORD.

WARNING:

After entry of the command and before the <cr> to terminate the command line, the terminal echo is disabled so that a private password entry can be made. If the operator aborts the PASSWORD command using an ABORT sequence, the terminal echo will remain disabled until re-enabled by the <Ctrl-E> or by the use of the CONFIG command.

<u>TOKEN:</u> /016

<u>FORMAT:</u>

[Line#] SPEED <VALUE>

<u>DESCRIPTION:</u>

<VALUE>

Can use constants, variable names, arrays or expressions.

This command changes the speed for all subsequent motions (except the GOPATH command).

The value of the speed corresponds to a percentage of full speed. The range permitted is from 1% to 150% of full speed. For Joint Interpolated motion, 100% is the maximum speed at which the robot will move in a coordinated fashion from any one programmable point to any other programmable point. For straight line motion, 100% is the linear speed entered by the @MAXSPD command an displayed in the STATUS display.

In immediate mode, with HELP on, the current speed is displayed when the SPEED command is entered. This is the easy way of determining the current speed setting.

When the controller is powered-up, the SPEED setting is limited 10% for all A460 robots. Any increase in speed beyond this limit before the robot is homed is ignored.

TOKEN: /031

FORMAT:

[Line#] W2

DESCRIPTION:

This command displays the current actual robot position in motor, joint, and Cartesian coordinates. If more than the standard 5 axes are installed, the extra axes will be displayed under the first, second and third values in the MOTOR and JOINT displays. In the WORLD display, they will be displayed only if a @TRACK or @GANTRY command has been issued.

TOKEN: /159

FORMAT:

[Line#] XZERO <Axis#>

DESCRIPTION:

<Axis#>

Can use constants, variable names, arrays or expressions.

This command zeros the position registers for the specified axis (axis 1-8). This is useful when doing a calibration sequence for an axis, as it allows a known position to be loaded into the position registers.

EXAMPLE:

1. **To enter a value of zero (0) into the current position register for axis six (6):**

   >>XZERO AXIS #: 6<cr>

TOKEN: /157

FORMAT:

@@CALZC [<axis#>, <axis#>...]

DESCRIPTION:

This command executes a calibration sequence on the specified axes. The home switches are not used in this procedure. The calibration is made at the next encoder zero-cross event.

Appendix C – Matlab Simulation Files

RKEWXYZ.m

```
function [manip_x, manip_y, manip_z] = RKEWXYZ( x, y, z, trocar_pt)

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%% Program: RKEWXYZ.m
%% Date:    March 3, 2006
%%
%% Purpose: To create the forward kinematic tranformation matrix for
%%          the A251 CRS Plus robot.  The transformation matrix is used
%%          to simulate motions of a surgical robot.  The desired change
%%          in manipulator position and current trocar point (L5) is taken
%%          as inputs.  The program performs three changes in manipulator
%%          position and then the robot is plotted to visualize and verify
%%          the motions performed by the robot.
%%
%% Inputs: x -> desired change in x position of robot manipulator
%%         y -> desired change in y position of robot manipulator
%%         z -> desired change in z position of robot manipulator
%%         trocar_pt -> initial length from robot flange to incision
%%
%% Outputs: manip_x -> x position of manipulator relative to world frame
%%          manip_y -> y position of manipulator relative to world frame
%%          manip_z -> z position of manipulator relative to world frame
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

syms c1 s1 c2 s2 c3 s3 c4 s4 c5 s5 L5 L6 cy sy cz sz LM;

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%      Robot Link Lengths       %%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
L2 = 10;
L3 = 10;
L4 = 10;

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%      Define reference points along robot tool       %%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
robot_end = 0;
manip_end = 23;

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%              Symbolic Transformation Matrices                     %%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
T0_1 = [c1  -s1  0  -L6; s1  c1  0  0; 0   0  1   0; 0  0  0 1];
T1_2 = [1    0   0   0;  0   1   0  0; 0   0  1  L2; 0  0  0 1];
T2_3 = [1    0   0   0;  0   0  -1  0; 0   1  0   0; 0  0  0 1];
T3_4 = [c2  -s2  0   0;  s2  c2  0  0; 0   0  1   0; 0  0  0 1];
T4_5 = [c3  -s3  0   L3; s3  c3  0  0; 0   0  1   0; 0  0  0 1];
T5_6 = [c4  -s4  0   L4; s4  c4  0  0; 0   0  1   0; 0  0  0 1];
T6_7 = [1    0   0   L5; 0   1   0  0; 0   0  1   0; 0  0  0 1];
```

```
T7_8 = [1    0   0   0;  0   c5  -s5 0;0   s5 c5 0 ; 0  0  0 1];
T8_9 = [1    0   0   0;  0   0   1  0; 0  -1  0   0; 0  0  0 1];
T9_10 = [1    0   0   0;  0   1   0  0; 0   0  1  2; 0  0  0 1];

T0_2  = T0_1*T1_2;
T0_3  = T0_2*T2_3;
T0_4  = T0_3*T3_4;
T0_5  = T0_4*T4_5;
T0_6  = T0_5*T5_6;
T0_7  = T0_6*T6_7;
T0_8  = T0_7*T7_8;
T0_9  = T0_8*T8_9;
T0_10 = T0_9*T9_10;

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%  Defines rotation matrix for solution of Trocar Kinematics  %%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
T0_1a = [cz -sz  0   0; sz cz 0 0; 0  0 1 0; 0 0 0 1];
T1_2a = [1   0   0   0; 0  0 -1 0; 0  1 0 0; 0 0 0 1];
T2_3a = [cy -sy  0   0; sy cy 0 0; 0  0 1 0; 0 0 0 1];
T3_4a = [1   0   0  LM; 0  1  0 0; 0  0 1 0; 0 0 0 1];
T4_5a = [1   0   0   0; 0  0  1 0; 0 -1 0 0; 0 0 0 1];

T0_2a = T0_1a*T1_2a;
T0_3a = T0_2a*T2_3a;
T0_4a = T0_3a*T3_4a;
T0_5a = T0_4a*T4_5a;

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%  Initial Robotic Position taken from controller (degrees) %%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
t1(1) = input('Joint 1 =');
t2(1) = input('Joint 2 =');
t3(1) = input('Joint 3 =');
t4(1) = input('Joint 4 =');
t5(1) = input('Joint 5 =');
t6(1) = input('Joint 6 =');

%% Robot gives joint angles in relation to world coordinate frame
%% therefore joint angles need to be converted to relate to previous
%% joint frame

t1(1) = (t1(1)*pi)/180;
t2(1) = (t2(1)*pi)/180;
t3(1) = (t3(1)*pi)/180 - t2(1);
t4(1) = (t4(1)*pi)/180 - (t2(1)+t3(1));
t5(1) = (t5(1)*pi)/180 - (pi/2);

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%% Initial Transformation Matrix  %%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
T = double(subs(T0_10,[c1 s1 c2 s2 c3 s3 c4 s4 c5 s5 L5 L6],
[cos(t1(1)) sin(t1(1)) cos(t2(1)) sin(t2(1)) cos(t3(1)) sin(t3(1))
 cos(t4(1)) sin(t4(1)) cos(t5(1)) sin(t5(1)) trocar_pt t6(1)]));

Tm = double(subs(T0_10,[c1 s1 c2 s2 c3 s3 c4 s4 c5 s5 L5 L6],
[cos(t1(1)) sin(t1(1)) cos(t2(1)) sin(t2(1)) cos(t3(1)) sin(t3(1))
```

```
 cos(t4(1)) sin(t4(1)) cos(t5(1)) sin(t5(1)) manip_end t6(1)]));

for n = 1:3

   t = calcTheta(T, trocar_pt);
   t1(n) = t(1);
   t2(n) = t(2);
   t3(n) = t(3);
   t4(n) = t(4);
   t5(n) = t(5);
   t6(n) = t(6);

    % XYZ location of the robot arm end (tool attachment point)
    rob = double(subs(T0_10,[c1 s1 c2 s2 c3 s3 c4 s4 c5 s5 L5 L6],
          [cos(t1(n)) sin(t1(n)) cos(t2(n)) sin(t2(n)) cos(t3(n))
           sin(t3(n)) cos(t4(n)) sin(t4(n)) cos(t5(n)) sin(t5(n))
           robot_end t6(n)]));
    rob_x(n) = rob(1,4);
    rob_y(n) = rob(2,4);
    rob_z(n) = rob(3,4);

    % XYZ location of trocar point
    troc = double(subs(T0_10,[c1 s1 c2 s2 c3 s3 c4 s4 c5 s5 L5 L6],
           [cos(t1(n)) sin(t1(n)) cos(t2(n)) sin(t2(n)) cos(t3(n))
            sin(t3(n)) cos(t4(n)) sin(t4(n)) cos(t5(n)) sin(t5(n))
            trocar_pt t6(n)]))
    troc_x(n) = troc(1,4);
    troc_y(n) = troc(2,4);
    troc_z(n) = troc(3,4);

    % XYZ location of the EndoWrist manipulator
    manip = double(subs(T0_10,[c1 s1 c2 s2 c3 s3 c4 s4 c5 s5 L5 L6],
                   [cos(t1(n)) sin(t1(n)) cos(t2(n)) sin(t2(n)) cos(t3(n))
                    sin(t3(n)) cos(t4(n)) sin(t4(n)) cos(t5(n)) sin(t5(n))
                    manip_end t6(n)]));
    manip_x(n) = manip(1,4);
    manip_y(n) = manip(2,4);
    manip_z(n) = manip(3,4);

    %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
    %%     Find XYZ location of each robot joint     %%
    %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
    T01 = double(subs(T0_1,[c1 s1 c2 s2 c3 s3 c4 s4 L6],
          [cos(t1(n)) sin(t1(n)) cos(t2(n)) sin(t2(n)) cos(t3(n))
           sin(t3(n)) cos(t4(n)) sin(t4(n)) t6(n)]));
    T01x(n) = T01(1,4);
    T01y(n) = T01(2,4);
    T01z(n) = T01(3,4);

    T02 = double(subs(T0_2,[c1 s1 c2 s2 c3 s3 c4 s4 L6],
          [cos(t1(n)) sin(t1(n)) cos(t2(n)) sin(t2(n)) cos(t3(n))
           sin(t3(n)) cos(t4(n)) sin(t4(n)) t6(n)]));
    T02x(n) = T02(1,4);
    T02y(n) = T02(2,4);
    T02z(n) = T02(3,4);

    T03 = double(subs(T0_3,[c1 s1 c2 s2 c3 s3 c4 s4 L6],
```

```
          [cos(t1(n)) sin(t1(n)) cos(t2(n)) sin(t2(n)) cos(t3(n))
           sin(t3(n)) cos(t4(n)) sin(t4(n)) t6(n)]));
    T03x(n) = T03(1,4);
    T03y(n) = T03(2,4);
    T03z(n) = T03(3,4);

    T04 = double(subs(T0_4,[c1 s1 c2 s2 c3 s3 c4 s4 L6],
          [cos(t1(n)) sin(t1(n)) cos(t2(n)) sin(t2(n)) cos(t3(n))
           sin(t3(n)) cos(t4(n)) sin(t4(n)) t6(n)]));
    T04x(n) = T04(1,4);
    T04y(n) = T04(2,4);
    T04z(n) = T04(3,4);

    T05 = double(subs(T0_5,[c1 s1 c2 s2 c3 s3 c4 s4 L6],
          [cos(t1(n)) sin(t1(n)) cos(t2(n)) sin(t2(n)) cos(t3(n))
           sin(t3(n)) cos(t4(n)) sin(t4(n)) t6(n)]));
    T05x(n) = T05(1,4);
    T05y(n) = T05(2,4);
    T05z(n) = T05(3,4);

    T06 = double(subs(T0_6,[c1 s1 c2 s2 c3 s3 c4 s4 L6],
          [cos(t1(n)) sin(t1(n)) cos(t2(n)) sin(t2(n)) cos(t3(n))
           sin(t3(n)) cos(t4(n)) sin(t4(n)) t6(n)]));
    T06x(n) = T06(1,4);
    T06y(n) = T06(2,4);
    T06z(n) = T06(3,4);

    %% Calculating New Manipulator XYZ Position %%
    Xn = manip(1,4); + x;
    Yn = manip(2,4); + y;
    Zn = manip(3,4); + z;

    dtzn = atan2((Yn-T(2,4)),(Xn-T(1,4)));
    dtyn = atan2((Zn-T(3,4)),((Yn-T(2,4))/sin(dtzn)));
    dtxn = ((Zn-T(3,4))/sin(dtyn));

    for n = 1:3
        for m = 1:3
            T(n,m) = double(subs(T0_5a(n,m),[cy sy cz sz LM],
                         [cos(dtyn) sin(dtyn) cos(dtzn) sin(dtzn) dtxn]));
        end
    end

    trocar_pt = manip_end - dtxn;

end

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%     Plot YZ robot and tool location for each rotation     %%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
figure(1)
hold on;

line([rob_y(1),manip_y(1)],[rob_z(1),manip_z(1)],'Color','b');
line([rob_y(2),manip_y(2)],[rob_z(2),manip_z(2)],'Color','g');
line([rob_y(3),manip_y(3)],[rob_z(3),manip_z(3)],'Color','r');
```

```
for n = 1:3
line([T01y(n),T02y(n)],[T01z(n),T02z(n)],'Color','k','LineWidth',2);
line([T02y(n),T03y(n)],[T02z(n),T03z(n)],'Color','k','LineWidth',2);
line([T03y(n),T04y(n)],[T03z(n),T04z(n)],'Color','k','LineWidth',2);
line([T04y(n),T05y(n)],[T04z(n),T05z(n)],'Color','k','LineWidth',2);
line([T05y(n),T06y(n)],[T05z(n),T06z(n)],'Color','k','LineWidth',2);
end

ylabel('Z axis');
xlabel('Y axis');
hold off;

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%     Plot XZ robot and tool location for each rotation     %%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
figure(2)
hold on;

line([rob_x(1),manip_x(1)],[rob_z(1),manip_z(1)],'Color','b');
line([rob_x(2),manip_x(2)],[rob_z(2),manip_z(2)],'Color','g');
line([rob_x(3),manip_x(3)],[rob_z(3),manip_z(3)],'Color','r');

for n = 1:3
line([T01x(n),T02x(n)],[T01z(n),T02z(n)],'Color','k','LineWidth',2);
line([T02x(n),T03x(n)],[T02z(n),T03z(n)],'Color','k','LineWidth',2);
line([T03x(n),T04x(n)],[T03z(n),T04z(n)],'Color','k','LineWidth',2);
line([T04x(n),T05x(n)],[T04z(n),T05z(n)],'Color','k','LineWidth',2);
line([T05x(n),T06x(n)],[T05z(n),T06z(n)],'Color','k','LineWidth',2);
end

ylabel('Z axis');
xlabel('X axis');
hold off;

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%     Plot XY robot and tool location for each rotation     %%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
figure(3)
hold on;

line([rob_x(1),manip_x(1)],[rob_y(1),manip_y(1)],'Color','b');
line([rob_x(2),manip_x(2)],[rob_y(2),manip_y(2)],'Color','g');
line([rob_x(3),manip_x(3)],[rob_y(3),manip_y(3)],'Color','r');

for n = 1:3
line([T01x(n),T02x(n)],[T01y(n),T02y(n)],'Color','k','LineWidth',2);
line([T02x(n),T03x(n)],[T02y(n),T03y(n)],'Color','k','LineWidth',2);
line([T03x(n),T04x(n)],[T03y(n),T04y(n)],'Color','k','LineWidth',2);
line([T04x(n),T05x(n)],[T04y(n),T05y(n)],'Color','k','LineWidth',2);
line([T05x(n),T06x(n)],[T05y(n),T06y(n)],'Color','k','LineWidth',2);
end

ylabel('Y axis');
xlabel('X axis');
hold off;
```

calcTheta.m

```
function [t] = calcTheta(T, trocar_pt)

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%% Program: calcTheta.m
%% Date:    March 3, 2006
%%
%% Purpose: Joint angles of the robot are calculated given the
%%          transformation matrix and trocar point
%%
%% Inputs: T -> the current transformation matrix of the robot
%%         trocar_pt -> length from robot flange to incision
%%
%% Outputs: t -> calculated joint angles
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

   %% Calculate Theta 1 %%
   t(1) = atan2(T(2,1),T(1,1));
   if(t(1) > 3.1416)
       t(1) = t(1) - 6.2832;
   elseif(t(1) < -3.1416)
       t(1) = t(1) + 6.2832;
   end

   %% Temporary values to solve for Theta 2 %%
   a = (20*trocar_pt*T(2,1) + 20*2*T(2,3) - 20*T(2,4))/(sin(t(1)));
   b = 20*trocar_pt*T(3,1) + 200 + 20*2*T(3,3) - 20*T(3,4);
   c = (( -(trocar_pt*T(2,1))^2 - (2*T(2,3))^2 - (T(2,4))^2 -
          2*2*trocar_pt*T(2,3)*T(2,1)+2*trocar_pt*T(2,1)*T(2,4)+
          2*2*T(2,3)*T(2,4))/((sin(t(1)))^2))+trocar_pt*T(3,1)*(2*T(3,4)-
          20-2*2*T(3,3) - trocar_pt*T(3,1)) + 2*T(3,3)*(-2*T(3,3)-20) +
          T(3,4)*(-T(3,4)+20) + 2*2*T(3,3)*T(3,4) - 100;

   %% Solve Theta 2 %%
   t(2) = atan2(b,a) - atan2(sqrt(a^2 + b^2 - c^2),c);
   if(t(2) > 3.1416)
       t(2) = t(2) - 6.2832;
   elseif(t(2) < -3.1416)
       t(2) = t(2) + 6.2832;
   end

   c23 = (trocar_pt*T(2,1) + 10*sin(t(1))*cos(t(2)) + 2*T(2,3) - T(2,4)) /
         (-10*sin(t(1)));
   s23 = (trocar_pt*T(3,1) + 10*sin(t(2)) + 10 + 2*T(3,3) - T(3,4))/(-10);

   %% Solve Theta 3 %%
   t(3) = atan2(s23,c23) - t(2);
   if(t(3) > 3.1416)
       t(3) = t(3) - 6.2832;
   elseif(t(3) < -3.1416)
       t(3) = t(3) + 6.2832;
   end
```

```
s234 = T(3,1);
c234 = T(2,1)/sin(t(1));

%% Solve Theta 4 %%
t(4) = atan2(s234,c234) - t(2) - t(3);
if(t(4) > 3.1416)
    t(4) = t(4) - 6.2832;
elseif(t(4) < -3.1416)
    t(4) = t(4) + 6.2832;
end

%% Solve Length 6 (linear axis position) %%
t(6) = trocar_pt*T(1,1) + 10*cos(t(1))*cos(t(2) + t(3)) +
       10*cos(t(1))*cos(t(2)) + 2*T(1,3) - T(1,4);
```

RK_data.m

```
function  RK_data(trocar_pt)

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%% Program: RK2.m
%% Date:    March 3, 2006
%%
%% Purpose: Used to perform experimention.  Input the intial joint
%%          values given from controller.  Data of that current
%%          orientation is given and then program asks for the desired
%%          change in position.  After change in position given, the
%%          program outputs data to change the robot to new position.
%%          Command to send to robot is outputed to a Data.txt file
%%          allowing for ease of sending commands with HyperTerminal
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

syms c1 s1 c2 s2 c3 s3 c4 s4 L5 L6 cy sy cz sz LM;

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%      Robot Link Lengths       %%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
L2 = 10;
L3 = 10;
L4 = 10;

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%      Define reference points along robot tool       %%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
robot_end = 0;
manip_end = 23;

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%             Symbolic Transformation Matrices                      %%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
T0_1 =  [c1  -s1  0  -L6; s1  c1  0  0; 0   0  1   0; 0  0  0 1];
T1_2 =  [1    0   0   0;  0   1   0  0; 0   0  1  L2; 0  0  0 1];
T2_3 =  [1    0   0   0;  0   0  -1  0; 0   1  0   0; 0  0  0 1];
T3_4 =  [c2  -s2  0   0;  s2  c2  0  0; 0   0  1   0; 0  0  0 1];
T4_5 =  [c3  -s3  0   L3; s3  c3  0  0; 0   0  1   0; 0  0  0 1];
T5_6 =  [c4  -s4  0   L4; s4  c4  0  0; 0   0  1   0; 0  0  0 1];
T6_7 =  [1    0   0   L5; 0   1   0  0; 0   0  1   0; 0  0  0 1];
T7_8 =  [1    0   0   0;  0   1   0  0; 0   0  1   0; 0  0  0 1];
T8_9 =  [1    0   0   0;  0   0   1  0; 0  -1  0   0; 0  0  0 1];
T9_10 = [1    0   0   0;  0   1   0  0; 0   0  1   2; 0  0  0 1];

T0_2  = T0_1*T1_2;
T0_3  = T0_2*T2_3;
T0_4  = T0_3*T3_4;
T0_5  = T0_4*T4_5;
T0_6  = T0_5*T5_6;
T0_7  = T0_6*T6_7;
T0_8  = T0_7*T7_8;
T0_9  = T0_8*T8_9;
```

```
T0_10 = T0_9*T9_10;

T0_5a = [cz*cy, -sz, -cz*sy, cz*cy*LM;
         sz*cy,  cz, -sz*sy, sz*cy*LM;
         sy,      0,     cy,    sy*LM;
          0,      0,      0,       1];

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%  Initial Robotic Position taken from controller (degrees) %%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
t1(1) = input('Joint 1 =');
t2(1) = input('Joint 2 =');
t3(1) = input('Joint 3 =');
t4(1) = input('Joint 4 =');
t5(1) = input('Joint 5 =');
t6(1) = input('Joint 6 =');

%% Robot gives joint angles in relation to world coordinate frame
%% therefore joint angles need to be converted to relate to previous
%% joint frame

t1(1) = (t1(1)*pi)/180;
t2(1) = (t2(1)*pi)/180;
t3(1) = (t3(1)*pi)/180 - t2(1);
t4(1) = (t4(1)*pi)/180 - (t2(1)+t3(1));
t5(1) = (t5(1)*pi)/180 - (pi/2);

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%% Initial Transformation Matrix  %%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
T = double(subs(T0_10,[c1 s1 c2 s2 c3 s3 c4 s4 L5 L6],
    [cos(t1) sin(t1) cos(t2) sin(t2) cos(t3) sin(t3) cos(t4) sin(t4)
     trocar_pt t6]));

%% Define no initial change so current data will be shown %%
x = 0;
y = 0;
z = 0;


while (x < 100)|(y < 100)|(z < 100)

    fid = fopen('Data.txt','w');
    Tm = double(subs(T0_10,[c1 s1 c2 s2 c3 s3 c4 s4 L5 L6],
           [cos(t1) sin(t1) cos(t2) sin(t2) cos(t3) sin(t3) cos(t4)
            sin(t4) manip_end t6]));

    Xn = Tm(1,4) + x;
    Yn = Tm(2,4) + y;
    Zn = Tm(3,4) + z;

    dtzn = atan2((Yn-T(2,4)),(Xn-T(1,4)));
    dtyn = atan2((Zn-T(3,4)),((Yn-T(2,4))/sin(dtzn)));
    dtxn = ((Zn-T(3,4))/sin(dtyn));
```

```
    for n = 1:3
        for m = 1:3
            T(n,m) = double(subs(T0_5a(n,m),[cy sy cz sz LM],
                     [cos(dtyn) sin(dtyn) cos(dtzn) sin(dtzn) dtxn]));
        end
    end
    trocar_pt = manip_end - dtxn;


   t = calcTheta(T, trocar_pt);
   t1 = t(1);
   t2 = t(2);
   t3 = t(3);
   t4 = t(4);
   t5 = t(5);
   t6 = t(6);

   % XYZ location of the EndoWrist manipulator
   manip = double(subs(T0_10,[c1 s1 c2 s2 c3 s3 c4 s4 L5 L6],
           [cos(t1) sin(t1) cos(t2) sin(t2) cos(t3) sin(t3) cos(t4)
            sin(t4) manip_end t6]));
   Tt = double(subs(T0_10,[c1 s1 c2 s2 c3 s3 c4 s4 L5 L6],
        [cos(t1) sin(t1) cos(t2) sin(t2) cos(t3) sin(t3) cos(t4) sin(t4)
         trocar_pt t6]));

   %% Verify that new move maintains the same trocar point
   if (abs(T(1,4)-Tt(1,4)) < 0.001) &&
      (abs(T(2,4)-Tt(2,4)) < 0.001) &&
      (abs(T(3,4)-Tt(3,4)) < 0.001)

       MA = [t1 t2 (t2+t3) (t2+t3+t4) (t5+pi/2) t6]
       joints = [MA(1)*(180/3.1416) MA(2)*(180/3.1416) MA(3)*(180/3.1416)
                 MA(4)*(180/3.1416) MA(5)*(180/3.1416) MA(6)]
       World = [(manip(1,4)+t6) manip(2,4) manip(3,4)]
       fprintf(fid,'MA %.4f,%.4f,%.4f,%.4f,%.4f,%.4f\n',MA(1), MA(2),
               MA(3), MA(4), MA(5), (MA(6)*18/18.875));
       status = fclose(fid);
   end

    x = input('x =');
    y = input('y =');
    z = input('z =');

end
```