DESIGN, CONTROL, AND IMPLEMENTATION OF A THREE LINK ARTICULATED ROBOT ARM

A Thesis

Presented to

The Graduate Faculty of The University of Akron

In Partial Fulfillment

of the Requirements for the Degree

Master of Science

Donald R. Dentler II

August, 2008

DESIGN, CONTROL, AND IMPLEMENTATION OF A THREE LINK

ARTICULATED ROBOT ARM

Donald R. Dentler II

Thesis

Approved:	Accepted:
_____	_____
Advisor Dr. Yueh-Jaw Lin	Dean of the College Dr. George K. Haritos
_____	_____
Committee Member Dr. John S. Gerhardt	Dean of the Graduate School Dr. George R. Newkome
_____	_____
Committee Member Dr. Zhenhai Xia	Date
_____	
Department Chair Dr. Celal Batur	

ABSTRACT

Robots have become commonplace in the manufacturing environment, allowing tasks ranging from the most repetitive to the most complex to be automated. As technology advances, robotics evolves to be both more precise and practical.

The purpose of this research is to study the behavior of a robotic system through the use of a three link articulated robotic arm. An in depth description of the various actuators, controllers, and drivers is included. The arm will be designed following the physical principals governing static and dynamic requirements of motion. The design process includes both examining structural requirements and control implementation. Component selection must be optimized for the design in terms of performance and physical properties. Using the robot arm and simulated motion programs, both forward and inverse coordinate transformation solutions are presented.

TABLE OF CONTENTS

Page

LIST OF TABLES

LIST OF FIGURES

CHAPTER I

INTRODUCTION

Robots have become commonplace in the manufacturing environment, allowing tasks ranging from the most repetitive to the most complex to be automated. As technology advances, robotics evolves to be both more precise and practical.

The purpose of this research is twofold. First, this text should serve as an overview of robotics and the mechanics that govern a robotic system. Second this paper presents the application of robotics principles and general engineering knowledge involved in the design and operation of a robot arm.

Chapter II outlines the elements of a robotic system. Actuators, controller and driver circuits, and computer interfaces are covered in detail. Also detailed are the physical principles that govern these components.

Chapter III describes the engineering principles involved in the structural design of a robotic system. Coordinate transformations, dynamic behavior, and curve tracking conditions are derived and modeled mathematically.

The detailed design of the robot constructed for this research is included in Chapter IV. Motor selection, static and dynamic analysis are covered, as well as calculations justifying these component selections. An assembly procedure is also included.

Chapter V summarizes the works of this thesis with observations and recommendations for extended study on this topic.

CHAPTER II

REVIEW OF ROBOTIC SYSTEMS

Typically, robots are used to perform jobs that are difficult, hazardous or monotonous for humans. They lift heavy objects, paint, weld, handle chemicals, and perform assembly work for days at a time without suffering from fatigue. Robots are defined by the nature of their movement. This section describes the following classifications of robots [1,4]:

- Cartesian
- Cylindrical
- Polar
- Articulated
- SCARA

Cartesian Robot [6,1,4]

Cartesian, or gantry, robots are defined by movement limited by three prismatic joints (figure 1). The workspace is defined by a rectangle resulting from the coincident axes.

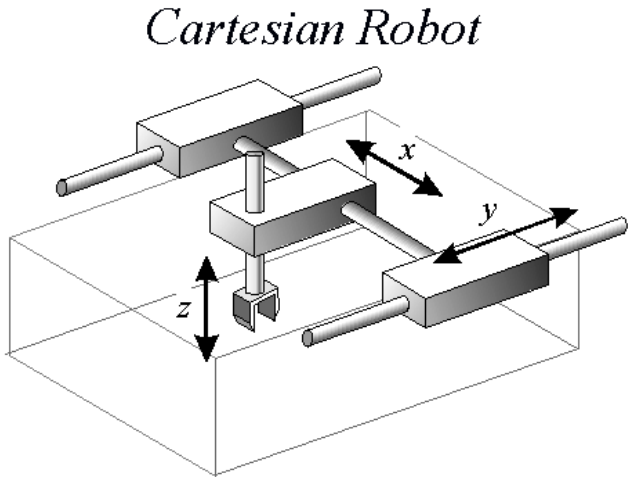


Figure 1: Cartesian Robot Movement

Cylindrical Robot

If one of the Cartesian robot's prismatic joints is exchanged for a revolute joint, a cylindrical robot is formed. A cylindrical robot's movement is defined by a cylindrical coordinate system. Figure 2 demonstrates this unit's thick shelled cylindrical workspace.

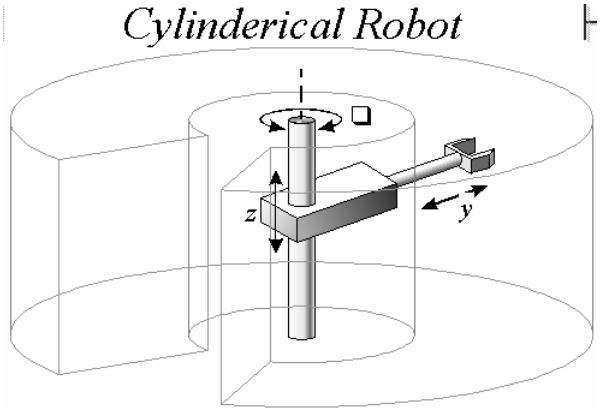


Figure 2: Cylindrical Robot Movement

Spherical Robot

Trading two prismatic joints for revolute joints forms a spherical robot. Spherical, or polar, robots are devices whose axes form a polar coordinate system. This robot arm works within a thick shelled spherical workspace, shown in figure 3.

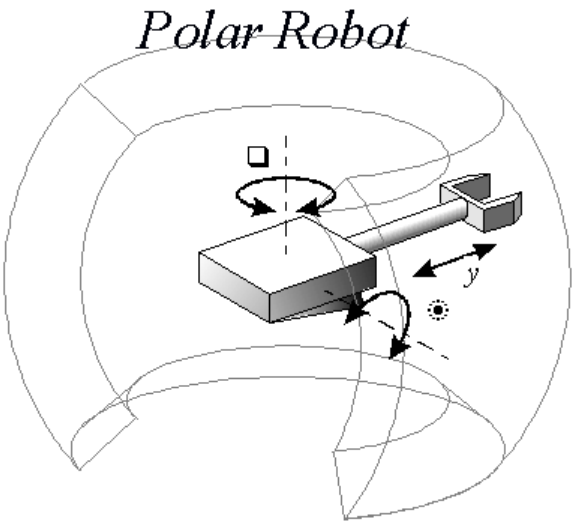


Figure 3: Spherical Robot Movement

Articulated Robot

Substituting a revolute joint for the final prismatic joint turns the arm into an articulated arm. Any robot whose arm has at least three rotary joints is considered to be an articulated robot (figure 4). The workspace is a complex set of intersecting spheres.

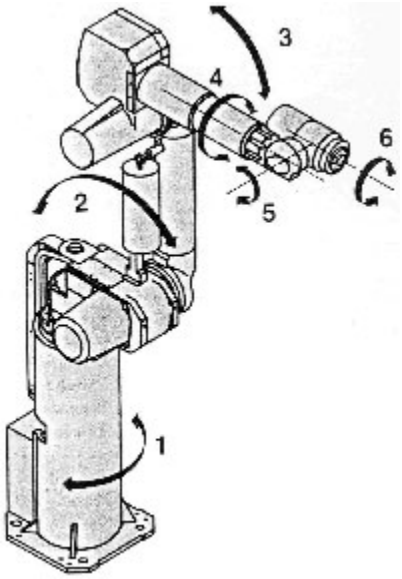


Figure 4: Articulated Robot Motion [2]

It can be seen that the required workspace weighs heavily in the selection of a robotic system. Figure 5 makes a comparison of the previously described robotic systems.

Comparison of robot configuration (from [*McKerrow*]).

Robot	*Joints*	*Coordinates*
		Advantages
Cartesian	prismatic waist prismatic shoulder prismatic elbow	. linear motion in three dimension . simple kinematic model . rigid structure . easy to visualize . can use inexpensive pneumatic drives for pick and place operation. *Disadvantages* . requires a large volume to operate in . work space is smaller than robot volume . unable to reach areas under objects . guiding surfaces of prismatic joints . must be covered to prevent ingress of dust
		Advantages
Cylindrical	revolute waist prismatic shoulder primatic elbow	. simple kinematic model . easy to visualize . good access into cavities and machine openings . very powerful when hydraulic drives used *Disadvantages* . restricted work space . prismatic guides difficult to seal from dust and liquids . back of robot can overlap work volume
		Advantages
Spherical	revolute waist revolute shoulder prismatic elbow	. covers a large volume from a central support . can bend down to pick objects up off the floor *Disadvantages* .complex kinematic model . difficult to visualize
		Advantages
Articulated	revolute waist revolute shoulder revolute elbow	. maximum flexibility . covers a large work space relative to volume of robots . revolute joints are easy to seal . suits electric motors . can reach over and under objects *Disadvantages* . complex kinematics . difficult to visualize . control of linear motion is difficult . structure not very rigid at full reach

Figure 5: Robot Comparison Chart [4]

SCARA Robot

SCARA (which stands for Selectively Compliant Articulated Robot Arm) is a specialty robot which has two parallel rotary joints to provide compliance in a plane. A third prismatic joint allows the arm to translate vertically. SCARA robots differ from articulated robots in that its workspace consists of two concentric cylinders, demonstrated in figure 6. This robot arm is specialized for assembly operations that involve placing parts on top of one another. The gripper can raise, lower, and rotate to orient the component to be assembled.

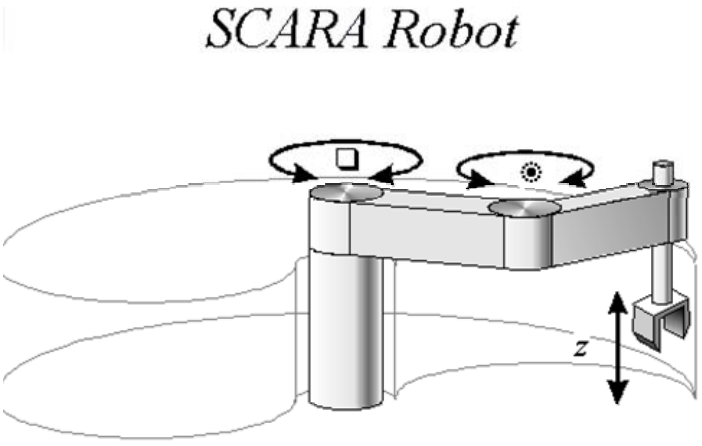


Figure 6: SCARA Robot Motion

Robotic System Components

Robotic system components can be grouped into one of three categories:

- Mechanical structure - This comprises all of the linkages and joints capable of movement.

- Actuator - The mechanism that provides the necessary forces to move the mechanical structure.
- Controller circuit - Supplies the actuators with the input required to achieve the desired position, force, speed, etc.

The design of the mechanical structure will be discussed in Chapter III. Actuators and controller circuits will be reviewed here.

Actuator types

The proper selection of actuator will dictate how effective a robot is in performing a specific task. Actuators can be either mechanical or electrical and have varying strengths and weaknesses as demonstrated in table 1. The basic actuators used for controlling motion include:

- Air Motors
- Hydraulic Motors
- Clutch/Brake
- Stepper Motors
- Servomotors

Table 1: Actuator Comparison

Actuator Type	**Strengths**	**Weaknesses**
Air Motor	-Low Cost -Easily Maintained -Simple to Operate	-Audible Compressor Noise -Inefficient System -Difficult to Regulate Speed
Hydraulic Motor	-High Loads Possible -Simple to Operate	-Slow System -Inefficient System -High Maintenance Requirements
Clutch/Brake	-Low Cost -Effective for Light Loads -Easy to Perform Speed Matching	-Uncontrolled Acceleration -Components Prone to Wear -Non-repeatable System
Stepper Motor	-Simple Control -Constant Load -Accurate Position	-Cannot Vary Load -Can Lose Steps -Resonance Problems
Servomotor	-High Performance -Small Motor Size -Can Operate At High Speeds	-Higher Cost System -Performance Limited by Controls -Speed Limited by Electronics

The most commonly used actuators in robotics are electric motors, be it either a stepper or servo type. Stepper motors perform best in open loop systems and servomotors are best suited for closed loop applications. These two specific actuators will be discussed in some detail along with open and closed loop systems.

Stepper Motors

Stepper motors, or steppers, are mechanically simple when compared to other motors in that there are no internal brushes or contacts. Armature rotation is achieved by switching the magnetic field sequentially.

Types of Steppers

Steppers can be grouped into three categories that differ in terms of internal construction based on the use of permanent magnets and/or iron rotors with laminated steel stators:

- Permanent magnet
- Variable reluctance
- Hybrid.

Permanent Magnet Steppers

The permanent magnet (PM) motor (shown in figure 7) has a permanent magnet attached to the rotor. It is a relatively low speed, low torque device with large step angles of either 45° or 90°. Its simple construction and low cost make it an ideal choice for non-industrial applications, such as a line printer wheel positioner.

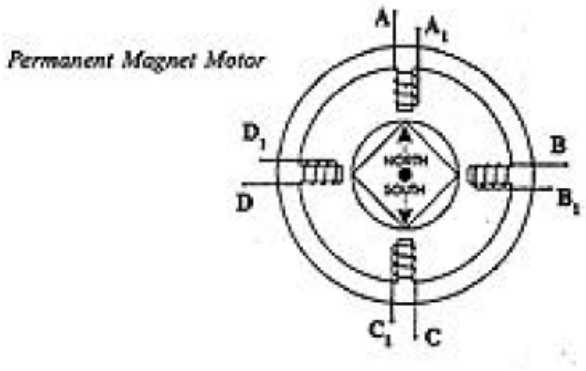


Figure 7: Permanent Magnet Motor[1]

Unlike the other stepper motors, the PM motor rotor has no teeth and is designed to be magnetized at a right angle to its axis. The above illustration shows a simple, 90° PM motor with four phases (A-D). Applying current to each phase in sequence will cause the rotor to rotate by adjusting to the changing magnetic fields. A 45° angle can be obtained by energizing two adjacent poles simultaneously.

Variable Reluctance Steppers

The variable reluctance motor does not use a permanent magnet. As a result, the motor rotor can move without constraint or detent torque. This type of construction is good in non-industrial applications that do not require a high degree of motor torque, such as the positioning of a micro slide.

The variable reluctance motor shown in figure 8 has four stator pole pairs set 15° apart. Current applied to pole A through the motor winding causes a magnetic attraction that aligns the rotor to pole A. Energizing stator pole B causes a 15° rotation to align with pole B. This process will continue with pole C and back to A in a clockwise direction. Reversing this procedure (C to A) would result in a counterclockwise rotation.

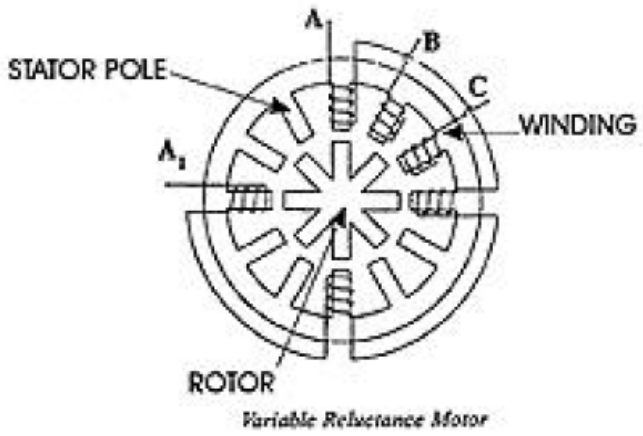


Figure 8: Variable Reluctance Motor[1]

Hybrid Steppers

Hybrid motors combine the best characteristics of the variable reluctance and permanent magnet motors. They are constructed with multi-toothed stator poles and a permanent magnet rotor. Standard hybrid motors have 200 rotor teeth and rotate at 1.8º step angles. Other hybrid motors are available in 0.9º and 3.6º step angle configurations. Because they exhibit high static and dynamic torque and run at very high step rates, hybrid motors are used in a wide variety of industrial applications. Figure 9 shows the configuration of a hybrid stepper motor.

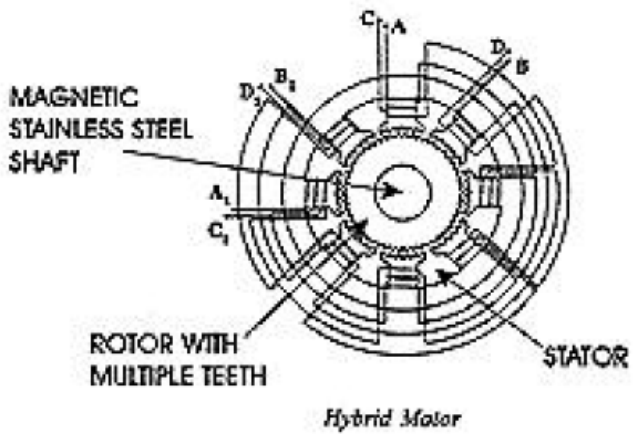


Figure 9: Hybrid Stepper Motor[1]

Motor Windings

Hybrid stepper motor stators can be wound in two ways, unifilar and bifilar. The windings affect how the current flows through the motor and, in turn how the motor performs.

Unifilar Windings

Unifilar has only one winding per stator pole. Stepper motors with a unifilar winding will have 4 lead wires. Figure 10 is a wiring diagram that illustrates a typical unifilar motor:

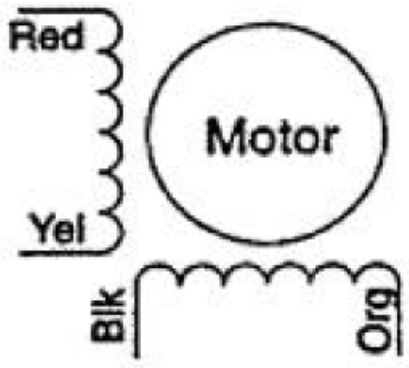


Figure 10: Unifilar Motor Winding [1]

Bifilar Windings

Bifilar wound motors means that there are two identical sets of windings on each stator pole. This type of winding configuration simplifies operation in that transferring current from one coil to another wound in the opposite direction, will reverse the rotation of the motor shaft. To accomplish this feat in a unifilar application would require the current to physically reverse its direction along the same winding. The most common wiring configuration for bifilar wound stepper motors is 8 leads because it offers the flexibility of either a series or parallel connection. There are, however, many 6-lead stepping motors available for series connection applications. Figure 11 demonstrates both the 6 and 8 lead wiring diagrams.

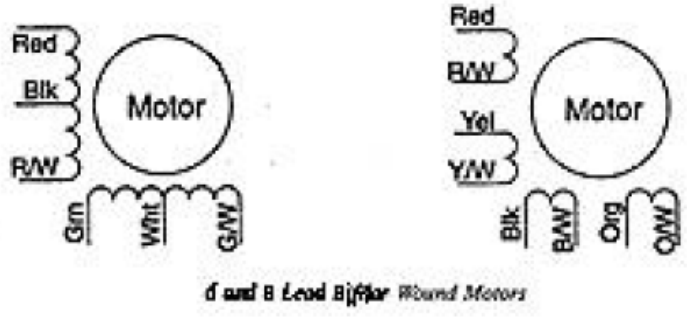


Figure 11: Bifilar Motor Windings. Left: 6 Lead Motor. Right: 8 Lead Motor [1]

Step Modes

Steppers may be stepped one of three ways depending on how and when the stators are energized. The step modes are:

- Full step mode
- Half step mode

- Microstep mode

The capabilities of the driver determine what step modes are available to the user.

Full Stepping

Standard hybrid stepper motors have 200 rotor teeth, or 200 full steps per revolution of the motor shaft. Dividing the 200 steps into the 360º rotation equals a 1.8º full step angle. Normally, full step mode is achieved by energizing both windings while reversing the current alternately. Essentially, one digital input from the driver is equivalent to one step.

Half Stepping

Half step simply means that a motor with 200 teeth is rotating at 400 steps per revolution. In this mode, one winding is energized and then two windings are energized alternately, causing the rotor to rotate at half the distance, or 0.9º. Half stepping is a more practical solution however, in industrial applications. Although it provides slightly less torque (about 70% of the motor's rated holding torque), half step mode increases the amount of resolution over a full step mode and increases positional accuracy. The torque reduction can be countered by over sizing the motor accordingly for the application.

Microstepping

Microstepping is a relatively new stepper motor technology that controls the current in the motor winding to a degree that further subdivides the number of positions between poles. Currently, microstep drivers are capable of rotating at 1/256 of a step (per step), or over 50,000 steps per revolution. This provides for a very smooth motor operation, with reduced mechanical noise and system resonance. The tradeoff with this enhanced performance is decreased motor torque. Figure 12 charts the torque decrease as

a single step is divided. At 256 steps per step the motor is only producing 0.61% of the full holding torque. In some cases, this may not even be enough torque to rotate the shaft, which affects the motor's accuracy.

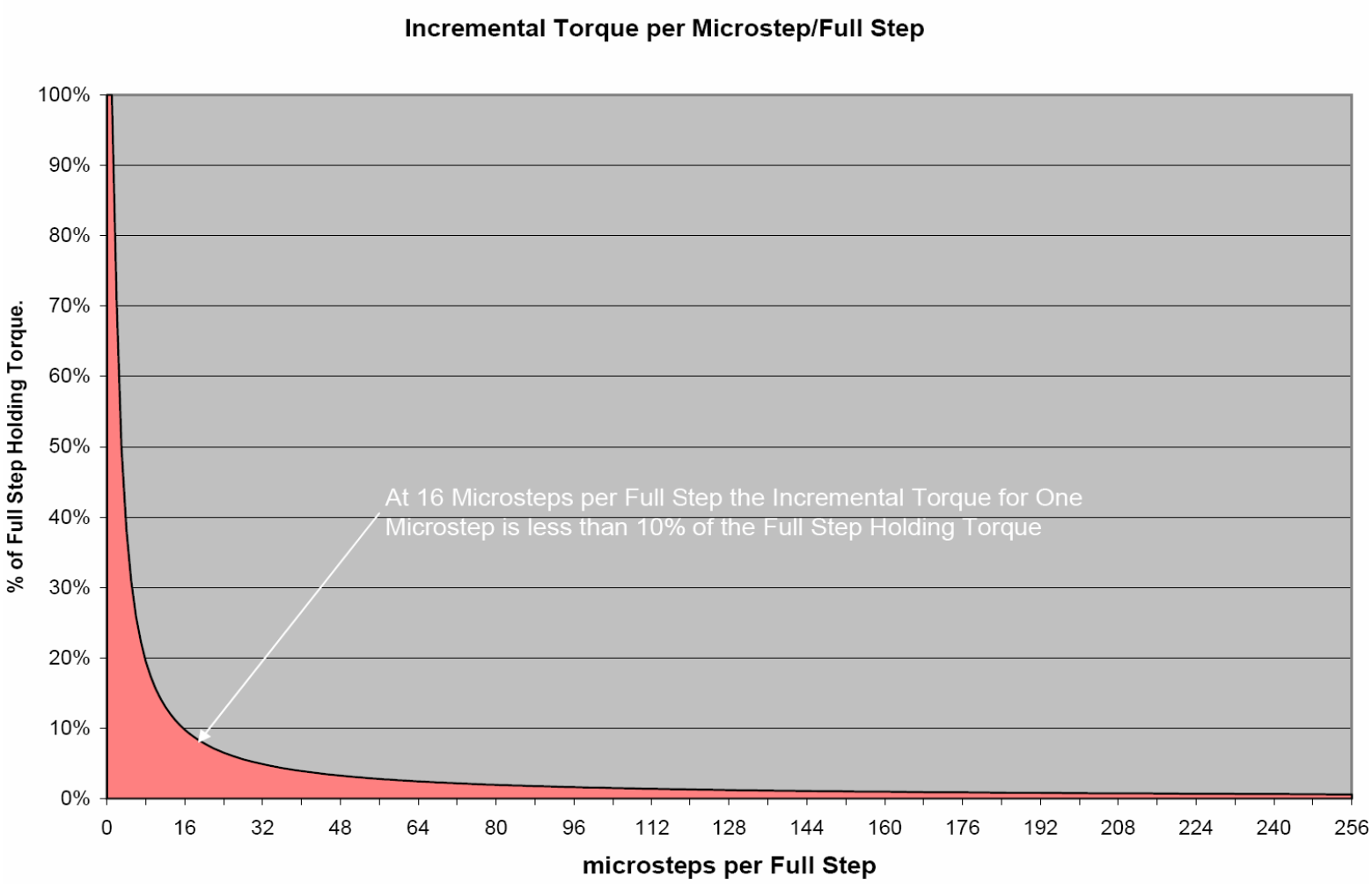


Figure 12: Torque Degradation with Step Division [2]

Servomotors

The term “servomotor” does not refer to one single kind of motor. Instead it refers any type of motor that receives a command signal from a controller. In this same respect, any closed loop system can be referred to as a servo system. Figure 13 diagrams the operation of a typical servo system.

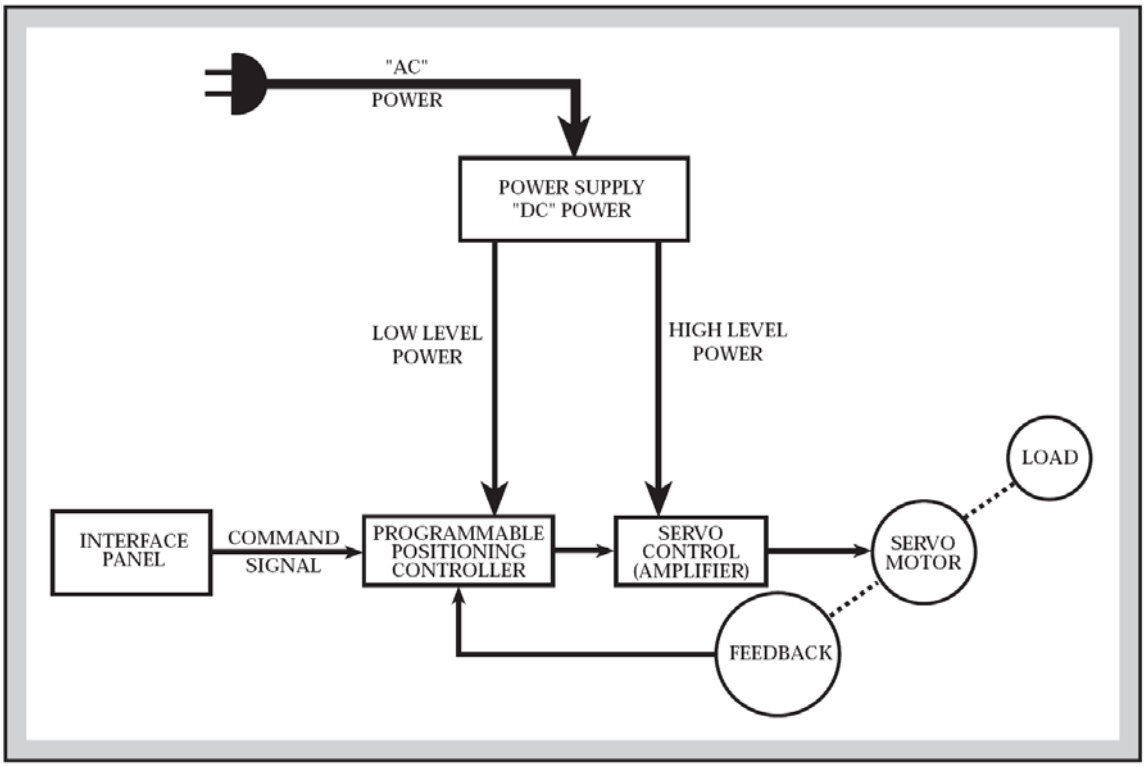


Figure 13: Example of a Servo System

This flexibility allows for several suitable types of electric motors to be used in servo systems. These electric motors include:

- Permanent Magnet DC Motor
- Brushless DC Motor

- Induction AC Motor

Electromagnetic motors operate based on the principle that the magnetic force on an electrical conductor in a magnetic field is perpendicular to that field. This force is defined by:

$$\vec{\mathbf{F}} = q\vec{\mathbf{v}} \times \vec{\mathbf{B}} \quad (1)$$

Where:

- $\vec{\mathbf{F}}$ is the vector describing the magnetic force
- q is the magnitude of the electrical charge
- $\vec{\mathbf{v}}$ is the vector magnitude of the charged particle's velocity
- $\vec{\mathbf{B}}$ is the vector describing the magnetic field

However, in the case of an electric motor the force can be quantified as a scalar:

$$F = I * L * B \quad (2)$$

Where:

- F is the magnetic force
- I is the electric current in the coil
- L is the length of the coil contained within the magnetic field
- B is the strength of the magnetic field

Permanent Magnet DC Motor

The DC permanent magnet motor is based on a similar concept to permanent magnet stepper motors, but it is the mechanical inverse. Whereas the PM stepper relies on stationary coils and a movable magnet attached to the rotor, a DC PM motor has a stationary electromagnet. The coil is wrapped around the rotor and is coupled via brushes to a commutator, which can switch the direction of the current and cause the motor to

rotate clockwise or counterclockwise (figure 14). Since the motor shaft will rotate freely while an electric current is present, an encoder must be used to provide feedback to a controller.

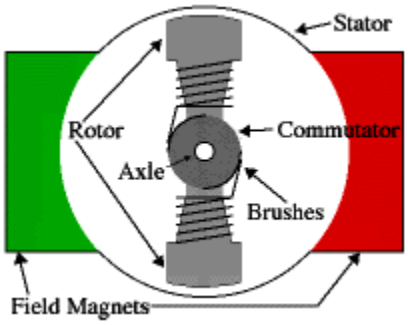


Figure 14: Permanent Magnet DC Motor

DC PM motors are common and can be very cost effective, however many of the motor's problems are related to the interface between the brushes and commuter. Contact between the two components causes friction and can be disrupted at higher speeds. A brushless DC motor addresses these issues.

Brushless DC Motor [17]

A brushless DC motor replaces the commutator and brushes with an electronic controller. This controller maintains the proper current in the stationary coils. Figure 15 shows a basic diagram of a brushless DC motor.

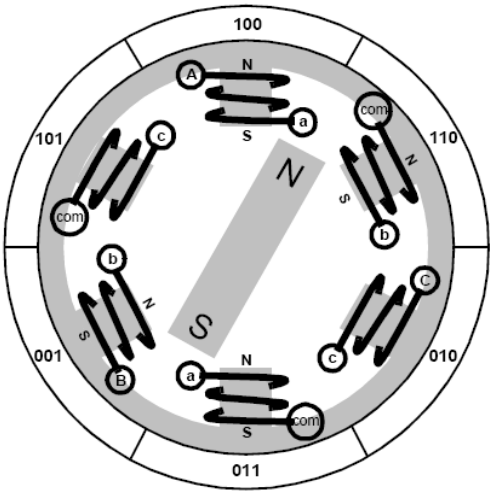


Figure 15: Brushless DC Motor

It should be noted that the internal layout of a brushless DC motor looks very similar to a permanent magnet stepper, yet a brushless motor relies on a feedback device such as a Hall Effect sensor to keep track of the position of the rotor. This provides for precise speed control. The brushless DC motor has a much higher initial cost than a conventional DC motor, but these costs can usually be justified by the increased performance and elimination of the maintenance needed to replace the brush contacts.

Induction AC Motor

AC induction motors rely on a minimum of two alternating current signals applied out of phase to cause the motor shaft to rotate. Figure 16 shows how the north pole of the motor shaft is attracted to the active south pole of the magnet.

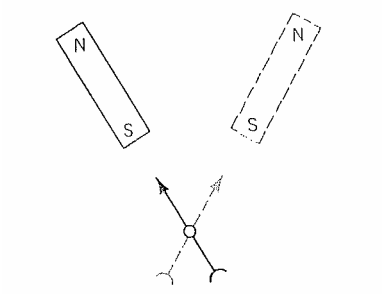


Figure 16: Magnet Follows Rotating Field

Applying the current out of phase (as shown in figure 17) ensures that when one pole is at its peak current, the opposing signal is equal to zero. This is called a rotating magnetic field since in reality the shaft is not rotating but following the shifting magnetic force.

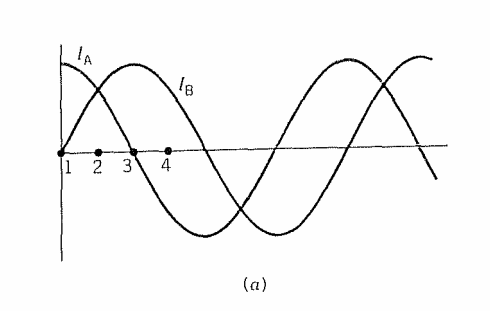


Figure 17: Applying the Current out of Phase

The AC signal regulates the motor speed based on the frequency of the sine wave and the number of stator poles. The maximum speed the motor could theoretically achieve is called the synchronous speed, and is defined as:

$$N_S = \frac{120f}{P} \tag{3}$$

Where:

- N_S is the synchronous speed of the motor
- f is the frequency of the AC signal
- P is the number of stator poles

However, in order to create torque, the motor must rotate slower than the synchronous speed. The difference between the two speeds is called slip and is represented as a percentage.

$$s = \frac{100(N_S - N_a)}{N_S} \tag{4}$$

Where:

- s is the percentage of slip
- N_S is the synchronous speed of the motor
- N_a is the actual speed of the motor

As the speed of the motor shaft decreases to generate torque, the slip percentage will increase.

System Types

Systems, in general, can be one of two types: open loop or closed loop. Figure 18 diagrams the two systems with respect to the information contained in this thesis.

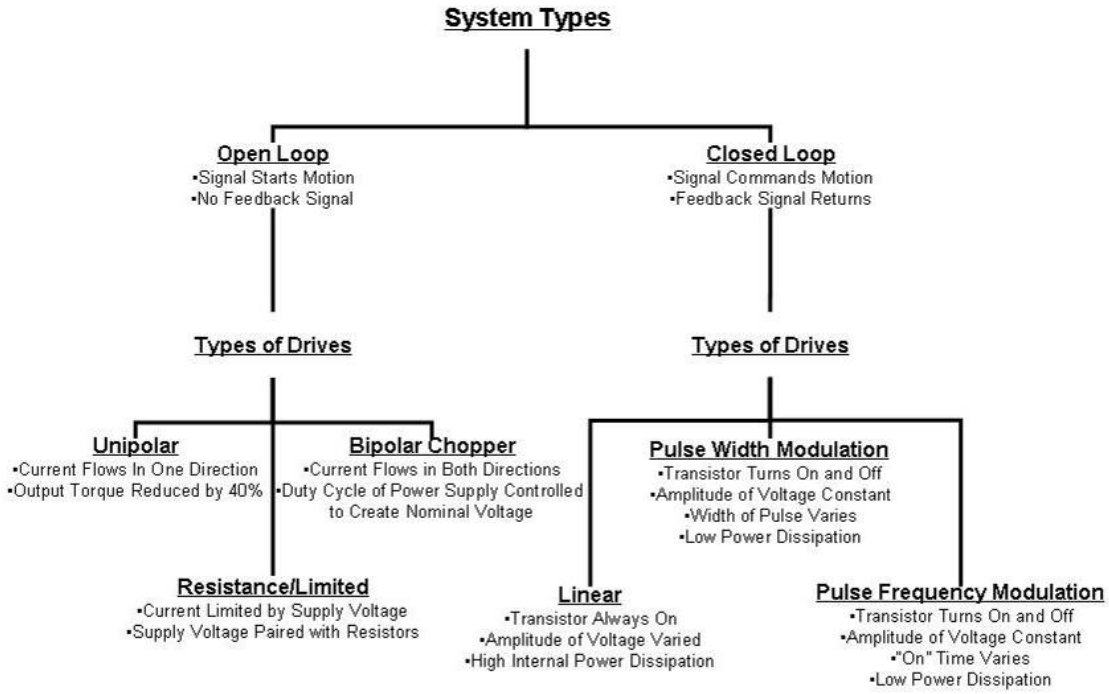


Figure 18: Open Loop vs. Closed Loop System Comparison

Open Loop

The premise of an open loop system is that the controller sends a signal to the driver based on the set point. The driver in turn will move the actuator the appropriate amount. No further action is taken unless the set point changes. Figure 19 diagrams this system.

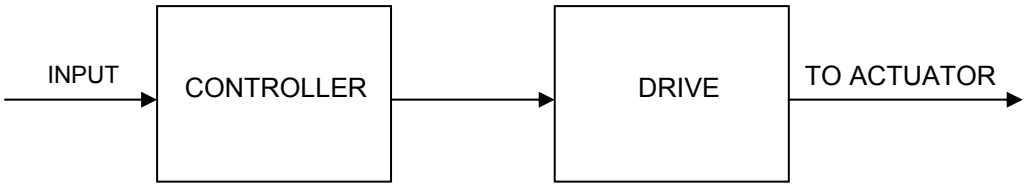


Figure 19: Open Loop System Diagram

Open loop systems are most effective when it is desirable for an actuator to move constantly until commanded to stop or when the actuation is extremely accurate. The latter case is true of stepper motors, since the steps are a constant increment.

Closed Loop [3]

A closed loop system operates as shown in figure 20. A command signal is sent to the motor and a feedback signal is returned to the controller indicating the motor's current state. The controller compares the command and feedback signals and adjusts its output accordingly.

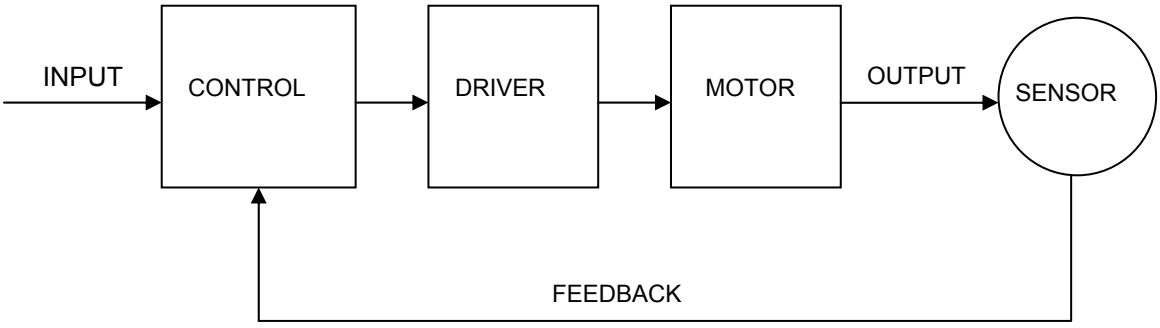


Figure 20: Closed Loop System

Closed loop systems are effective when the process demands control over a variety of complex motion profiles. Closed loop allows for precise control over speed and position through the use of feedback devices such as tachometers, encoders, or resolvers. Because of additional components required, a closed loop system is more complex and can cost more initially; but these issues are offset by the added degrees of control.

The low power command signal is amplified by the servo controller to produce movement of the motor and load. As the motor moves the load, an appropriate feedback signal is generated and returned to the positioning controller. The controller in turn evaluates this signal and determines whether or not the motor is operating properly. If the command signal and feedback signal are not equal, the controller will correct the position signal until the difference between the two is zero.

Motor Driver Circuits

The electric current produced by the control circuit is normally not high enough to induce the necessary torque for motor rotation. For this reason, driver circuits are employed. They manage the higher currents required by the motor and convert a digital control signal from the controller into a movement by the motor. Drivers also manage the direction that the current is flowing to produce clockwise or counterclockwise motion.

Types of Stepper Motor Drivers

Generally, there are three basic types of stepper driver technologies, they are:

- Unipolar
- Resistance/limited
- Bipolar chopper.

All drivers utilize a "translator" to convert the step and direction signals from the indexer into electrical pulses to the motor. The essential difference between driver options is the “switch set,” or the circuit that energizes the motor windings. Figure 21 shows the flow of information from controller to stepper motor.

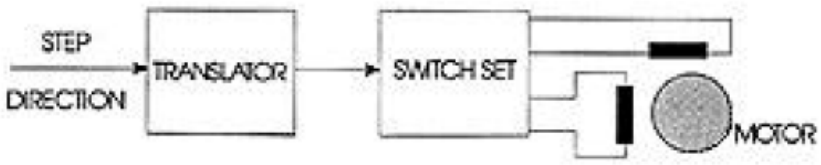


Figure 21: Information Flow Between Driver and Motor[1]

Unipolar Drive

A unipolar drive consists of a winding with a center-tap, or two separate windings per phase, which limits the current flow to one direction. The direction is reversed by moving the current from one half of the winding to the other half using two switches per phase, as shown in figure 22. Therefore, the switch set of a unipolar drive is simple and inexpensive. However, the unipolar drive utilizes only half the available conducting wire volume on the winding. As a result, the torque output of a unipolar driver is reduced by nearly 40% when compared to other technologies. Unipolar drivers are useful in applications that operate at relatively low step rates.

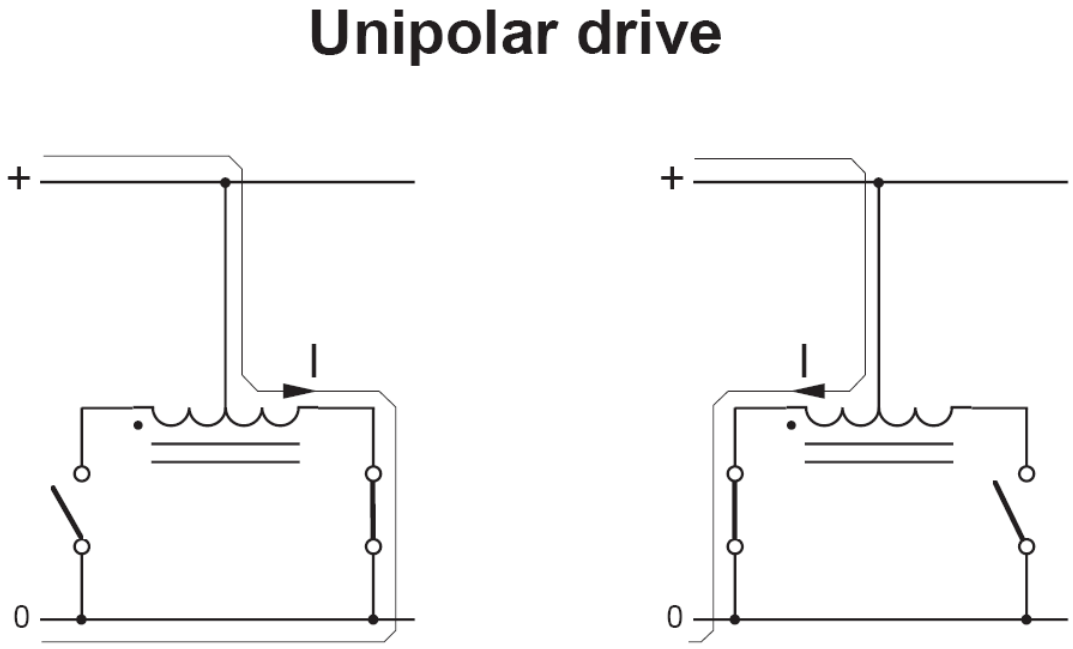


Figure 22: Current Flow Through Unipolar Drive[5]

Resistance/Limited Drive

Resistance/Limited (R/L) drivers are simple and inexpensive. The driver limits the current by supply voltage and the resistance of the winding. High speed performance is improved by increasing the supply voltage. This increased supply voltage in the R/L drive must be accompanied by an additional resistor in series with the winding to limit the current to the previous level (figure 23). This resistor, called a dropping resistor, is added to maintain a useful increase in speed. The drawback of this method is the power loss in the dropping resistors. This process also produces an excessive amount of heat and must rely on a DC power supply for its current source.

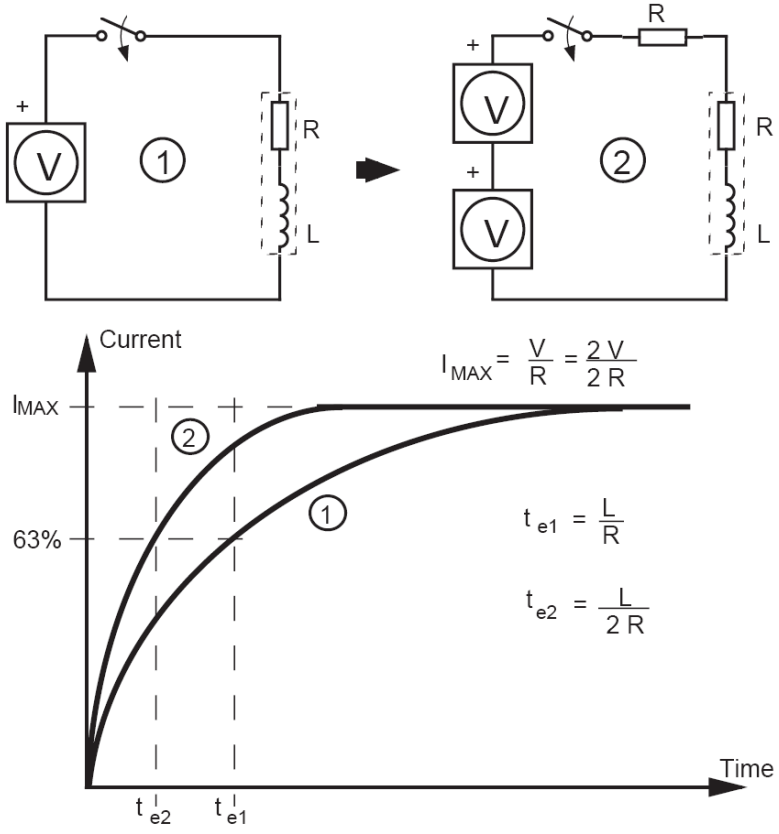


Figure 23: R/L Driver Circuit [5]

Bipolar Chopper Drive

Bipolar chopper drivers are by far the most widely used drivers for industrial applications. Although they are typically more expensive to design, they offer high performance and high efficiency. This driver employs two different principles to control the current flow to the motor windings: a bipolar switch set and current chopping. Both are explained in this section.

A Bipolar drive, as the name implies, switches the current direction on a single winding by shifting the voltage polarity across the terminals. The polarity switch is accomplished using four switches configured as shown in figure 24. This configuration is called an H-bridge.

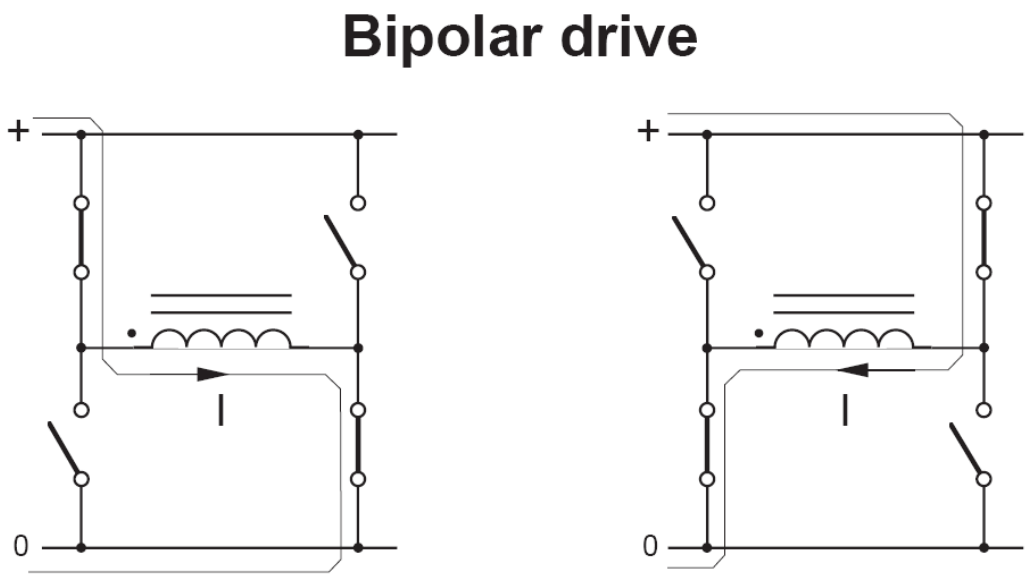


Figure 24: H-bridge [5]

The method behind a chopper driver is to use a supply voltage which is several times higher than the nominal voltage of the motor causing a significant increase in the current. By controlling the duty cycle of the chopper, an average voltage and an average current equal to the nominal motor voltage and current are created. The advantage of this constant current control is management of the developed torque, regardless of power supply voltage variations. It also provides the shortest possible current build-up and reversal time.

Additionally, these drivers use a four transistor H-bridge with recirculating diodes and a sense resistor that maintains a feedback voltage proportional to the motor current. Motor windings, using a bipolar chopper driver, are energized to the full supply level by turning on one set of the switching transistors. The sensing resistor develops a voltage with the linear rise in current, which is monitored by a comparator, until the required level is reached. At this point the top switch opens and the current in the motor coil is maintained via the bottom switch and the diode. Current decay occurs until a preset position is reached and the process starts over. This "chopping" effect of the supply is what maintains the correct current voltage to the motor at all times (figure 25).

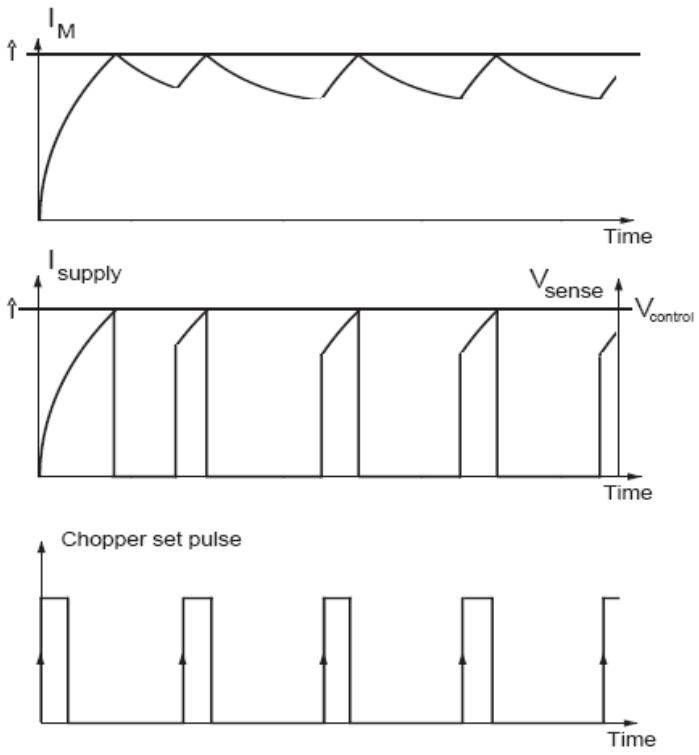


Figure 25: Chopper Pulse Chart [5]

Figure 26 illustrates an H-bridge configured as a constant current chopper. Depending on how the H-bridge is switched during the turn-off period, the current will either recirculate through one transistor and one diode (path 2), giving the slow current decay, or recirculate back through the power supply (path 3).

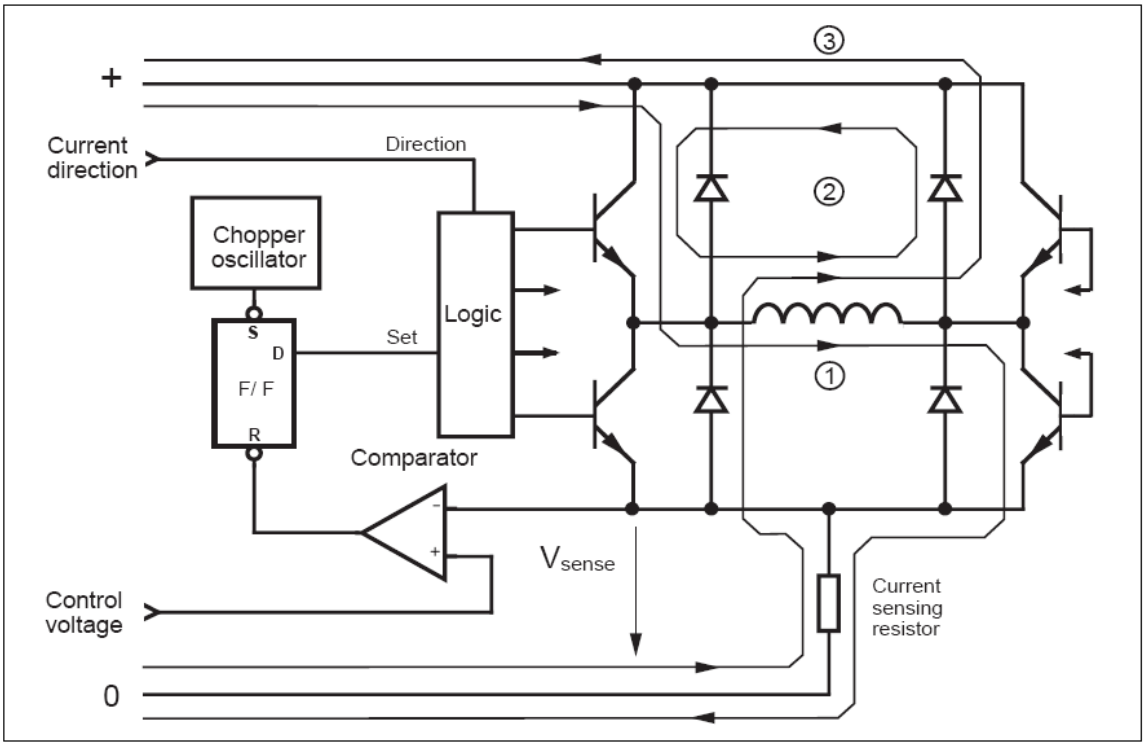


Figure 26: Bipolar Chopper Schematic [5]

Servo Drivers [6]

Servo motors are controlled using electronic pulses. Normally, transistors are employed to regulate the pulses. There are three basic transistor circuits used for servomotor control; linear, pulse width modulated and pulse frequency modulated.

Linear Driver

Linear drivers operate using transistors that are always active, but regulate the amount of power supplied. Transistors act as a valve which, based on the input voltage, controls the current it draws from a connected voltage source. In this way the controller functions as a faucet. For example, if a transistor receives half of the full range input voltage, then the motor runs at half power. Linear drive provides for a stable motor speed and control.

Pulse Width Modulation Driver

Pulse width modulation, or PWM, drivers regulate power by varying the time it is applied to the motor. This way, the average power is controlled through the duty cycle of the pulse. If the pulse is wide (i.e., the power is applied longer) the average power sent to motor is high causing it to turn faster, and so on. Figure 27 illustrates how the pulse width determines the average voltage. The advantage of this approach is that the amount power dissipated by the transistor is low because it is always fully on or off. Requiring less power means the transistors can be smaller, which leads to smaller packaging for the driver.

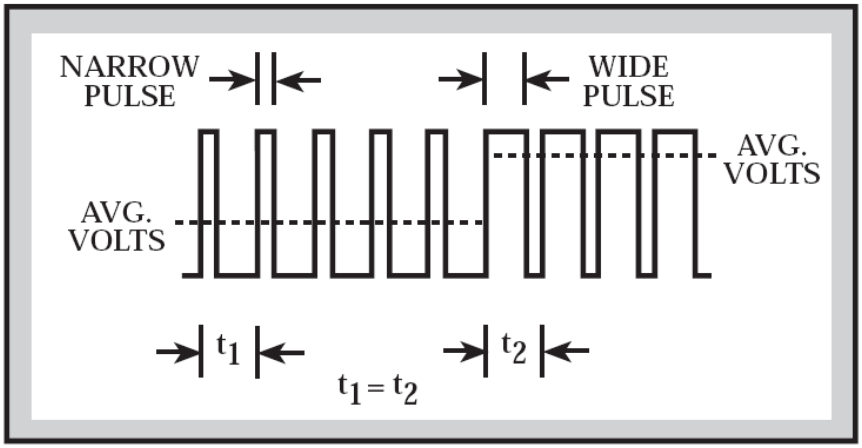


Figure 27: Average Voltage Produced by Pulse Width

Pulse Frequency Modulation Driver

Instead of varying the duty cycle of a pulse, the width of the pulse itself could be controlled. This technique is called pulse frequency modulation, or PFM. A PFM driver operates by applying few pulses to in a given time period to generate a high average voltage and many pulses to generate a low average voltage. Figure 28 demonstrates the concept of PFM. As with PWM, the transistors are either fully on or off. Due to the complicated system required to modulate the pulse frequency, PFM drivers are not commonly used.

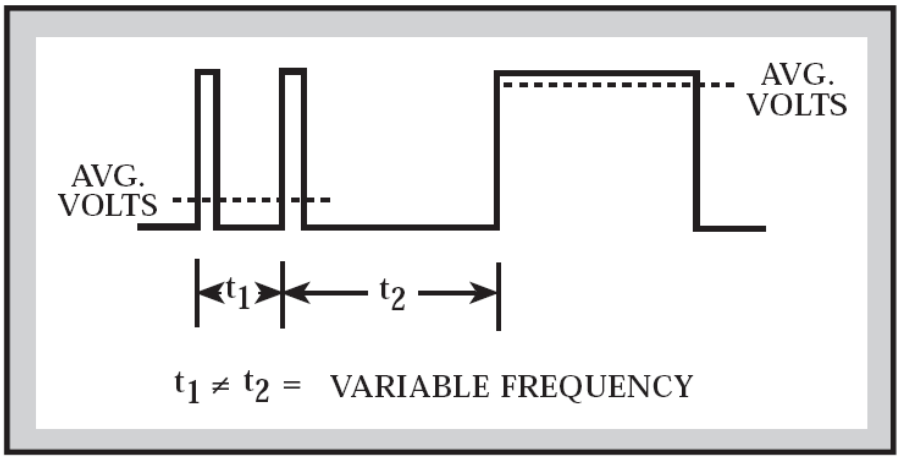


Figure 28: Average Voltage Produced by Pulse Frequency

Computer Interface

Normally, a robots controller is housed with a computer. The computer can translate motion commands into data pulses that are processed by the motor driver. This section outlines one of the most common interfaces, the parallel port, and details how the data is transferred across it.

Parallel Port [7] [8]

When a PC sends data to a printer or other device using a parallel port, it can send 8 bits of data (1 byte) simultaneously. These 8 bits are being transmitted in parallel to each other, as opposed to the same eight bits being transmitted serially, 1 bit at a time, through a serial port. The standard parallel port is capable of sending 50 to 100 kilobytes of data per second. Figure 29 shows the layout of a common configuration of parallel port, DB25, which has 25 pins.

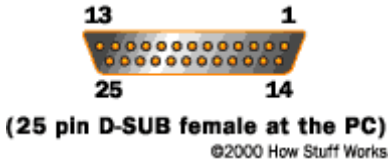


Figure 29: Layout of DB25 Parallel Port[1]

Following is a description of the pins:

- Pin 1 – maintains a voltage between 2.8 and 5 volts, called the strobe signal. When data is sent to the printer the voltage drops below 0.5 volts as the computer sends a byte of data.
- Pins 2 through 9 – used to exchange data between the PC and receiving entity. A simple method is used to indicate whether a bit has a value of 1 or 0. A charge of 5 volts is sent through means a particular pin has a bit value of 1. No charge indicates a value of 0.
- Pin 10 – operates in a similar fashion to Pin 1. A voltage drop indicates to the computer to confirm that the data has been received. This is called the acknowledge signal.

- Pin 11 – normally used as the printer busy signal when charged. The voltage drops below 0.5 volts when the printer is ready to receive more data.
- Pin 12 – a 5 volt signal that is charged when the printer is out of paper.
- Pin 13 – the printer online signal. It is constantly charged to indicate that the printer is active and ready to receive information.
- Pin 14 – an auto feed signal to the printer to move paper through the system.
- Pin 15 – error signal to let the computer know if there are any problems. It uses a voltage drop below 0.5 volts, similar to the other pins, to indicate an error.
- Pin 16 – A voltage drop below 0.5 volts initializes the printer.
- Pin 17 – Charging will take the printer offline. This charge is maintained until it is desirable to bring the printer online.
- Pins 18-25 – these are the ground pins to be used as a reference signal for the low (below 0.5 volts) charges.

Hexadecimal Conversion

When writing to or reading from a port, the status of an entire byte (8 bits) is defined at once. When the state is updated, a binary number is written to (or read from) the port, with either a 1 or a 0 for each bit. For example, if pins 2 and 7 are to be moved to logic state 1 the binary input should be 0,0,1,0,0,0,0,1. To expedite programming, this number is entered as its decimal equivalent, 33. Table 2 shows the convention for converting hexadecimal to binary and decimal numbers. While the table is truncated here, counting can easily be continued up to 255 (the largest number that can be written to a byte) using this method. For example, 16 on the decimal scale corresponds to 10000 on the binary scale. Also, it should be noted that adding leading zeros to a binary number will not increase its decimal value. In other words, if the logic state is set to 1 on pin 2 and 0 on the other pins the binary input would be 00000001 but it would still be entered in decimal form as 1.

Table 2: Binary to Hexadecimal Conversion

Binary	Hex	Decimal (Base-10)
0	0	0
1	1	1
10	2	2
11	3	3
100	4	4
101	5	5
110	6	6
111	7	7
1000	8	8
1001	9	9
1010	A	10
1011	B	11
1100	C	12
1101	D	13
1110	E	14
1111	F	15

CHAPTER III

ROBOT DYNAMICS

The challenge of designing a robot stems from anticipating and controlling the system dynamics. This chapter outlines the dynamic behavior of robotic arms and spatial transformations used to determine position. Path planning, curve tracking, and inverse transformation solution methods are also discussed.

Definition of Link Coordinate Frames

Each point of actuation between two sections of a robot arm is called a link. These links are numbered starting from the base (link 0) and move outward toward the tool (link n). Each link is assigned its own local coordinate system, or frame. In order to remain consistent between problems, the following conventions are used to establish frame coordinates.

Rules for assigning frame coordinates:

The coordinate system is defined for each link following the convention as follows:

- Z_n is directed along the axis of rotation of the link n+1
- X_n is directed along the common normal a_n
- Y_n assigned by right hand rule

Figure 30 demonstrates this convention further.

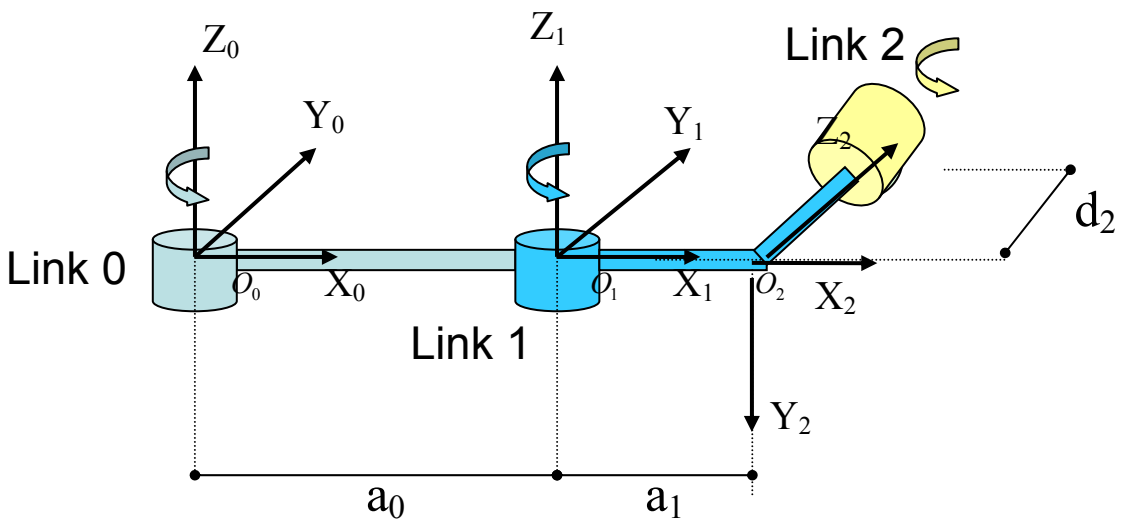


Figure 30: Establishing links

Coordinate Frame Transformations

The orientation of the tool point can be expressed in terms of the world coordinate system through the use of coordinate frame transformations. These transformations are used to rotate or translate each link coordinate system and mate it with the next coordinate system closer to the robot base.

Coordinate frame transformations are performed using the following conventions.

Rotational Transformations

If a commanded move requires rotation, the transformation is performed as follows:

Rotation θ degrees about the x axis:

$$Rot(x,\theta) = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & \cos\theta & -\sin\theta & 0 \\ 0 & \sin\theta & \cos\theta & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{5}$$

Rotation θ degrees about the y axis:

$$Rot(y,\theta) = \begin{bmatrix} \cos\theta & 0 & -\sin\theta & 0 \\ 0 & 1 & 0 & 0 \\ \sin\theta & 0 & \cos\theta & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{6}$$

Rotation θ degrees about the z axis:

$$Rot(z,\theta) = \begin{bmatrix} \cos\theta & -\sin\theta & 0 & 0 \\ \sin\theta & \cos\theta & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{7}$$

Translational Transformations

If the tool moves from one point to another, the transformation occurs along a vector with components a, b, and c. This transformation is performed as follows:

$$Trans(a,b,c) = \begin{bmatrix} 1 & 0 & 0 & a \\ 0 & 1 & 0 & b \\ 0 & 0 & 1 & c \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{8}$$

Denavit-Hartenberg Convention

The Denavit-Hartenberg (DH) convention allows for consistent frame assignments by creating a set of rules used to bring two link coordinate frames into coincidence with one another. The rules are as follows:

- Rotate around the new z axis by an angle θ_n
- Translate along the new z axis by a distance d_n
- Translate along the x_i-1 axis by a distance a_n
- Rotate around the x_i-1 axis by an angle α_n

Figure 31 gives an example of link coordinate frames established for 3 axes.

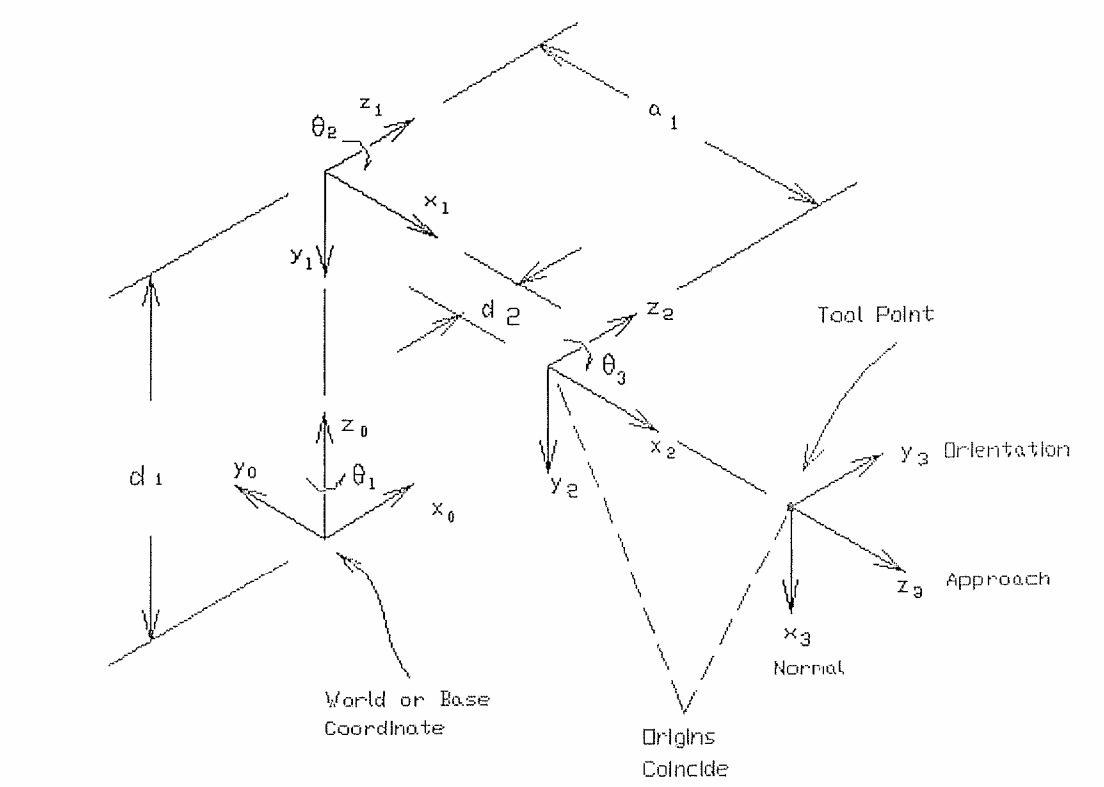


Figure 31: Example of link coordinate frames

Using these conventions, the position of the nth coordinate frame can be obtained in terms of the (n-1)th frame by performing the following four transformations.

$$^{n-1}T_n = Rot(\hat{z},\theta_n)Trans(0,0,d_n)Trans(a_n,0,0)Rot(\hat{x},\alpha_n) \qquad (9)$$

Or in terms of the full matrices:

$$
{}^{n-1}T_n = \begin{bmatrix} \cos\theta_n & -\sin\theta_n & 0 & 0 \\ \sin\theta_n & \cos\theta_n & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & d_n \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & a_n \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & \cos\alpha_n & -\sin\alpha_n & 0 \\ 0 & \sin\alpha_n & \cos\alpha_n & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{10}
$$

When evaluated, this matrix takes the form of the DH matrix:

$$
{}^{n-1}A_n = \begin{bmatrix} \cos\theta_n & -\cos\alpha_n \sin\theta_n & \sin\alpha_n \sin\theta_n & a_n \cos\theta_n \\ \sin\theta_n & \cos\alpha_n \cos\theta_n & -\sin\alpha_n \cos\theta_n & a_n \sin\theta_n \\ 0 & \sin\alpha_n & \cos\alpha_n & d_n \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{11}
$$

The kinematic equation becomes the chain product of the successive coordinate transformation matrices, taking the form:

$$
{}^0T_n = {}^0A_1 {}^1A_2 \ldots {}^{n-1}A_n \tag{12}
$$

Where ${}^0T_n$ is the location of the nth coordinate frame with respect to the base coordinate system.

Definition of Link Kinetic Parameters

The ultimate goal if the Kinetic transformation is to convert the joint space matrix, in terms of the joint rotation angles:

$$
\begin{bmatrix} \theta_1 \\ \theta_2 \\ \theta_3 \\ \vdots \\ \theta_n \end{bmatrix} \tag{13}
$$

To the task space matrix, in terms of the coordinate system transformation:

$$\begin{bmatrix} x \\ y \\ z \\ \phi \\ \theta \\ \varphi \end{bmatrix} \tag{14}$$

Dynamic Equations

This section details the frame transformations involved in determining the dynamic model of a link, including the kinetic and potential energy calculations.

Velocity

The velocity parameter is determined by assuming ${}^{n}r_{n}$ is a point fixed on link n and expressed with respect to frame n by:

$${}^{n}r_{n} = \begin{bmatrix} x_n \\ y_n \\ z_n \\ 1 \end{bmatrix} \tag{15}$$

Or with respect to the base frame as:

$${}^{0}r_{n} = {}^{0}T_{n}\,{}^{n}r_{n} = \left({}^{0}T_{1}\cdots{}^{n-1}T_{n}\right){}^{n}r_{n} \tag{16}$$

This transformation is illustrated in figure 32:

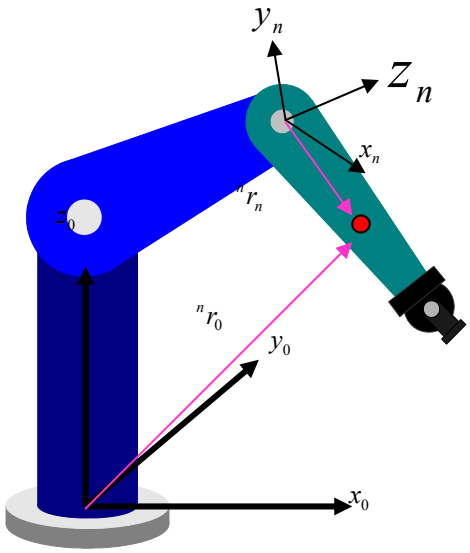


Figure 32: Relating link velocity to the base frame

The velocity of point ${}^n r_n$ with respect to the n[th] frame is 0, but velocity with respect to the base frame is V_n, which is defined as:

$$V_n \equiv {}^0V_n = \frac{d}{dt}{}^0r_n = \frac{d}{dt}\left({}^0T_1\,{}^1T_2 \cdots {}^{n-1}T_n\right){}^n r_n \tag{17}$$

$$= {}^0\dot{T}_1\,{}^1T_2 \cdots {}^{n-1}T_n\,{}^n r_n + {}^0T_1\,{}^1\dot{T}_2 \cdots {}^{n-1}T_n\,{}^n r_n + {}^0T_1\,{}^1T_2 \cdots {}^{n-1}\dot{T}_n\,{}^n r_n + {}^0T_n\,{}^n\dot{r}_n = \left(\sum_{j=1}^{n} \frac{\partial {}^0T_n}{\partial q_j}\dot{q}_j\right){}^n r_n \tag{18}$$

Where q_j is the joint variable of the J[th] joint. $\frac{\partial {}^0T_n}{\partial q_j}$ is defined as U_{nj} hereafter for simplicity.

To obtain the link velocity, first the link variable must be defined. For a rotary joint, the joint variable $q_n = \theta_n$ which has a partial derivative defined as Q_n:

$$\frac{\partial}{\partial \theta_n} = Q_n = \begin{bmatrix} 0 & -1 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} \tag{19}$$

And a transformation matrix:

$$^{n-1}T_n = \begin{bmatrix} \cos\theta_n & -\cos\alpha_n \sin\theta_n & \sin\alpha_n \sin\theta_n & \alpha_n \cos\theta_n \\ \sin\theta_n & \cos\alpha_n \cos\theta_n & -\sin\alpha_n \sin\theta_n & \alpha_n \sin\theta_n \\ 0 & \sin\alpha_n & \cos\alpha_n & d_n \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{20}$$

Therefore the velocity transformation is:

$$\frac{\partial^{n-1}T_n}{\partial q_n} = Q_n {}^{n-1}T_n = \begin{bmatrix} 0 & -1 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} \cos\theta_n & -\cos\alpha_n \sin\theta_n & \sin\alpha_n \sin\theta_n & \alpha_n \cos\theta_n \\ \sin\theta_n & \cos\alpha_n \cos\theta_n & -\sin\alpha_n \sin\theta_n & \alpha_n \sin\theta_n \\ 0 & \sin\alpha_n & \cos\alpha_n & d_n \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{21}$$

Whose solution is:

$$\frac{\partial^{n-1}T_n}{\partial q_n} = \begin{bmatrix} \cos\theta_n & -\cos\alpha_n \sin\theta_n & \sin\alpha_n \sin\theta_n & \alpha_n \cos\theta_n \\ \sin\theta_n & \cos\alpha_n \cos\theta_n & -\sin\alpha_n \sin\theta_n & \alpha_n \sin\theta_n \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{22}$$

Similarly, for a prismatic joint the joint variable $q_n = d_n$ and the partial would be:

$$\frac{\partial}{\partial d_n} = Q_n = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 \end{bmatrix} \tag{23}$$

And the velocity transformation would be:

$$\frac{\partial^{n-1}T_n}{\partial q_n} = Q_n{}^{n-1}T_n = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} \cos\theta_n & -\cos\alpha_n \sin\theta_n & \sin\alpha_n \sin\theta_n & \alpha_n \cos\theta_n \\ \sin\theta_n & \cos\alpha_n \cos\theta_n & -\sin\alpha_n \sin\theta_n & \alpha_n \sin\theta_n \\ 0 & \sin\alpha_n & \cos\alpha_n & d_n \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{24}$$

Whose solution is:

$$\frac{\partial^{n-1}T_n}{\partial q_n} = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 \end{bmatrix} \tag{25}$$

The effect of the motion of joint j for all points on link n is as follows:

$$\frac{\partial^0 T_n}{\partial q_j} \equiv U_{nj} = \begin{cases} {}^0T_{j-1} Q_j {}^{j-1}T_n & for \quad j \le n \\ 0 & for \quad j > n \end{cases} \tag{26}$$

The velocity of the link would then be defined as

$$V_n \equiv {}^0V_n = \frac{d}{dt}{}^0r_i = \frac{d}{dt}({}^0T_1 {}^1T_2 \cdots {}^{n-1}T_n)^n r_n = (\sum_{j=1}^{n} \frac{\partial^0 T_n}{\partial q_j} \dot{q}_j)^n r_n = (\sum_{j=1}^{n} U_{nj} \dot{q}_j)^n r_n \tag{27}$$

Jacobian

An important element of modeling the velocity of the robot arm is the Jacobian matrix. It is used to relate the joint space velocity to the task space velocity as follows:

$$\begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{z} \\ \omega_x \\ \omega_y \\ \omega_z \end{bmatrix} = \left[\frac{dh(q)}{dq} \right]_{6\times n} \begin{bmatrix} \dot{q}_1 \\ \dot{q}_2 \\ \vdots \\ \dot{q}_n \end{bmatrix}_{n\times 1} \tag{28}$$

$$J = \left(\frac{dh(q)}{dq} \right)_{6\times n} = \begin{bmatrix} \frac{\partial h_1}{\partial q_1} & \frac{\partial h_1}{\partial q_2} & \cdots & \frac{\partial h_1}{\partial q_n} \\ \frac{\partial h_2}{\partial q_1} & \frac{\partial h_2}{\partial q_2} & \cdots & \frac{\partial h_2}{\partial q_n} \\ \vdots & \vdots & \vdots & \vdots \\ \frac{\partial h_6}{\partial q_1} & \frac{\partial h_6}{\partial q_2} & \cdots & \frac{\partial h_6}{\partial q_n} \end{bmatrix}_{6\times n} \tag{29}$$

Kinetic Energy

Once the link velocity has been determined, the kinetic energy can be calculated. For a rigid body in 3-D space with a linear velocity (V) and angular velocity (ω) about the center of mass, the kinetic energy is:

$$K = \frac{1}{2} m v^2 + \frac{1}{2} I \omega^2 \quad \text{or} \quad K = \frac{1}{2} m V^T V + \frac{1}{2} I \omega^T \omega \tag{30}$$

Where I is the inertia tensor:

$$I \equiv \begin{bmatrix} \int (y^2 + z^2) dm & -\int xy dm & -\int xz dm \\ -\int xy dm & \int (x^2 + z^2) dm & -\int yz dm \\ -\int xz dm & -\int yz dm & \int (x^2 + y^2) dm \end{bmatrix} \tag{31}$$

It should be noted that the diagonal terms represent the moments of inertia and the off diagonal terms represent the products of inertia. The kinetic energy of a particle with differential mass dm in link n is:

$$dK_n = \frac{1}{2} \left(\dot{x}_n^2 + \dot{y}_n^2 + \dot{z}_n^2 \right) dm = \frac{1}{2} trace\left(V_n V_n^T \right) dm \tag{32}$$

As stated in (27) V_n can be defined as $(\sum_{j=1}^{n} U_{nj} \dot{q}_j)^n r_n$ therefore:

$$dK_n = \frac{1}{2} Tr \left[\sum_{p=1}^{n} U_{np} \dot{q}_p r_n^n \left(\sum_{r=1}^{n} U_{nr} \dot{q}_r r_n^n \right)^T \right] dm \tag{33}$$

Which can be arranged as:

$$dK_n = \frac{1}{2} Tr\left[\sum_{p=1}^{n} \sum_{r=1}^{n} U_{np} (r_n^n r_n^{n^T} dm) U_{nr}{}^T \dot{q}_p \dot{q}_r \right] \tag{34}$$

The kinetic energy of the link is then calculated by integrating kinetic energy of the differential mass elements across the link:

$$K_n = \int dK_n = \frac{1}{2} Tr\left[\sum_{p=1}^{n} \sum_{r=1}^{n} U_{np} (\int r_n^n r_n^{n^T} dm) U_{nr}{}^T \dot{q}_p \dot{q}_r \right] \tag{35}$$

The term $\int r_n^n r_n^{n^T} dm$ is defined as the pseudo-inertia matrix for link n and is represented by J_n:

$$J_n = \int r_n^n r_n^{n^T} dm = \begin{bmatrix} \int x_n^2 dm & \int x_n y_n dm & \int x_n z_n dm & \int x_n dm \\ \int x_n y_n dm & \int y_n^2 dm & \int y_n z_n dm & \int y_n dm \\ \int x_n z_n dm & \int y_n z_n dm & \int z_n^2 dm & \int z_n dm \\ \int x_n dm & \int y_n dm & \int z_n dm & \int dm \end{bmatrix} \tag{36}$$

Finally, the total kinetic energy for the robot arm is the sum of the kinetic energies of the links:

$$K = \sum_{n=1}^{i} K_n = \frac{1}{2} \sum_{n=1}^{i} Tr\left[\sum_{p=1}^{n} \sum_{r=1}^{n} U_{np} (\int r_n^n r_n^{n^T} dm) U_{nr}{}^T \dot{q}_p \dot{q}_r \right] = \frac{1}{2} \sum_{n=1}^{i} \sum_{p=1}^{n} \sum_{r=1}^{n} \left[Tr(U_{np} J_n U_{nr}{}^T) \dot{q}_p \dot{q}_r \right] \tag{37}$$

Potential Energy

The potential energy of link n is defined as:

$$P_n = -m_n g^n \bar{r}_0 = -m_n g ({}^n T_0 {}^n \bar{r}_n) \tag{38}$$

Where:

- ${}^n \bar{r}_0$ center of mass wrt base frame

- ${}^{n}\bar{r}_{n}$ center of mass wrt to n[th] frame
- $g = (g_x, g_y, g_z, 0)$ is the gravity row vector expressed in base frame

The potential energy of the robot arm is then:

$$P = \sum_{n=0}^{n} P_n = \sum_{n=0}^{n} [-m_n g({}^{n}T_0 \, {}^{n}\bar{r}_n)] \tag{39}$$

Which is a function of q_n.

Lagrange Function

In order to simplify the solution, the dynamic equations can be determined using Lagrangian mechanics. The Lagrange function allows the forces to be determined using the system's energy. This eliminates the need to solve for acceleration components. Since energy is a scalar quantity, it also makes the equations easier to manage. The Lagrangian function takes the form of:

$$L = K - P = \frac{1}{2} \sum_{n=1}^{i} \sum_{j=1}^{n} \sum_{k=1}^{n} \left[Tr(U_{nj} J_n U_{nk}{}^{T}) \dot{q}_j \dot{q}_k \right] + \sum_{n=1}^{i} m_n g({}^{0}T_n \, {}^{n}\bar{r}_n) \tag{40}$$

Where L is the Lagrange – Euler equation:

$$\frac{d}{dt}\left(\frac{\partial L}{\partial \dot{q}_n}\right) - \frac{\partial L}{\partial q_n} = \tau_n \tag{41}$$

$$= \sum_{j=n}^{i} \sum_{k=1}^{j} Tr(U_{jk} J_j U_{jn}{}^{T}) \ddot{q}_k + \sum_{j=n}^{i} \sum_{k=1}^{j} \sum_{m=1}^{j} Tr\left(\frac{\partial U_{jk}}{\partial q_m} J_j U_{jn}{}^{T}\right) \dot{q}_k \dot{q}_m - \sum_{j=n}^{i} m_j g U_{jn}{}^{j} \bar{r}_j \tag{42}$$

And:

- K: Total kinetic energy of robot
- P: Total potential energy of robot
- q_n : Joint variable of n[th] joint
- $\dot{q}_n$: First time derivative of q_n

- τ_n : Summation of torques about link n

Recalling equation (26) and expanding it to incorporate the effects of joints j and k on link n results in:

$$U_{njk} \equiv \frac{\partial U_{nj}}{\partial q_k} = \begin{cases} {}^0T_{j-1}Q_j{}^{j-1}T_{k-1}Q_k{}^{k-1}T_n & n \geq k \geq j \\ {}^0T_{k-1}Q_k{}^{k-1}T_{j-1}Q_j{}^{j-1}T_n & n \geq j \geq k \\ 0 & n < j \quad or \quad n < k \end{cases} \tag{43}$$

The dynamic model of an n-link arm is then presented as:

$$\tau_n = \sum_{k=1}^{i} D_{nk}\ddot{q}_k + \sum_{k=1}^{i}\sum_{m=1}^{i} h_{nkm}\dot{q}_k\dot{q}_m + C_n \tag{44}$$

Where:

- $D_{nk} = \sum_{j=\max(n,k)}^{i} Tr(U_{jk}J_jU_{jn}{}^T)$
- $h_{nkm} = \sum_{j=\max(n,k,m)}^{i} Tr(U_{jkm}J_jU_{jn}{}^T)$
- $C_n = -\sum_{j=n}^{i} m_j g U_{jn}{}^j \bar{r}_j$

Or in a simplified form as:

$$\tau = D(q)\ddot{q} + h(q,\dot{q}) + C(q) \tag{45}$$

In terms of a matrix representation of this model, each component refers to:

- $D = \begin{bmatrix} D_{11} & \cdots & D_{1n} \\ & \vdots & \\ D_{n1} & \cdots & D_{nn} \end{bmatrix}$ Inertia matrix terms
- $h(q,\dot{q}) = \begin{bmatrix} h_1 \\ \vdots \\ h_n \end{bmatrix}$ Coriolis and centrifugal terms

- $C(q) = \begin{bmatrix} C_1 \\ \vdots \\ C_n \end{bmatrix}$ Gravity terms

- $\tau = \begin{bmatrix} \tau_1 \\ \vdots \\ \tau_n \end{bmatrix}$ Driving torques for each link

Forward Solution

A forward solution calculates the location of the end effector based on the given combination of joint angles and displacements. The movement is described using equations (5) through (10) and the resulting transformation takes the form of (11). Using simulation software, a forward solution is demonstrated in figures 33 and 34. In figure 33, the robot arm is shown at a relative zero point while figure 34 shows the arm's position after a rotation of 20, 45, and 25 degrees for links 0, 1, and 2, respectively.

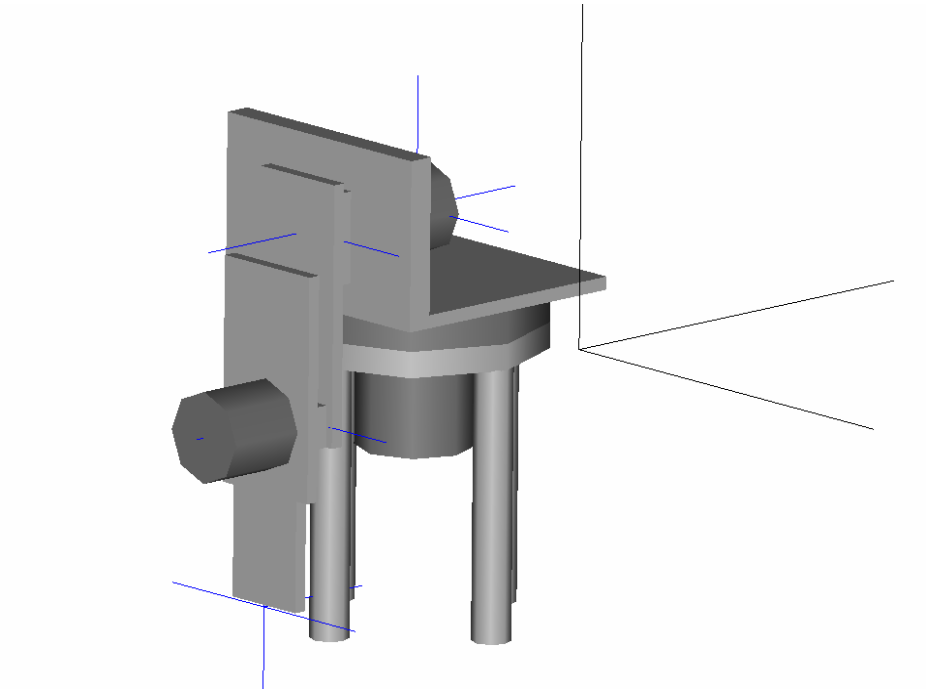


Figure 33: Robot arm before forward solution

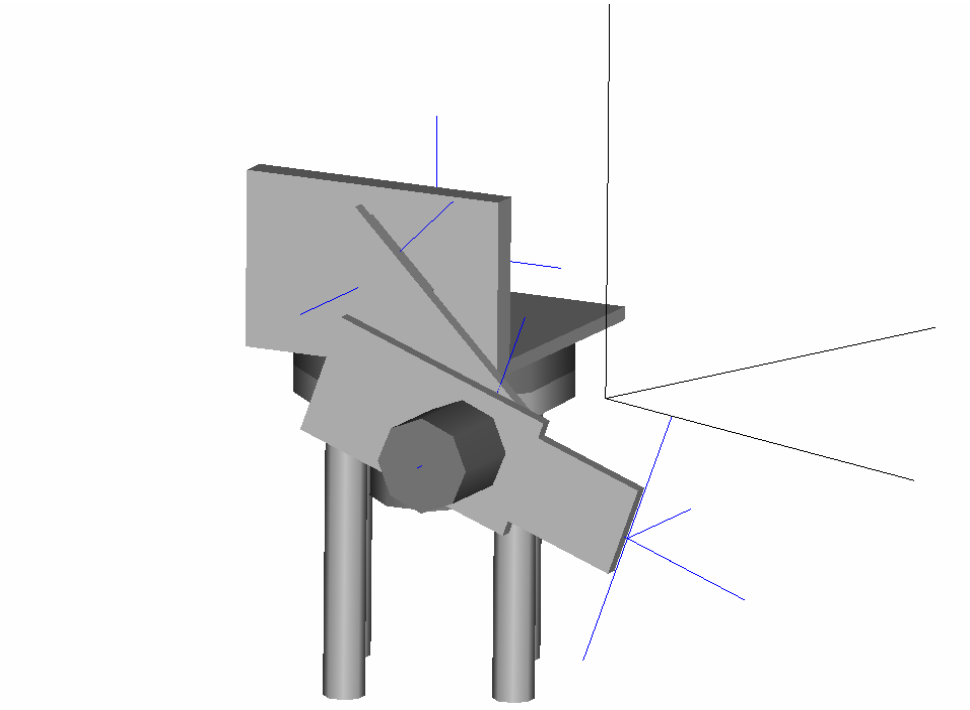


Figure 34: Links 0, 1, and 2 rotated 20, 45, and 25 degrees respectively

Inverse Solution

The inverse solution requires determining the necessary link rotations and transformations needed to achieve the desired location and orientation of the end effector. In some cases this can lead to multiple solutions for a movement. There have been numerous studies performed on determining the most efficient orientation of the joints, some of which will be referenced in this section. An example of an inverse solution is shown below, using a proprietary algorithm from New River Kinematics. Figure 35 shows the robot arm at [0,0,0] and the desired end point in space at [7, -1, 3]. Figure 36 shows the robot arm translated to the point. A plot of the joint motions during the inverse solution is included in Appendix F.

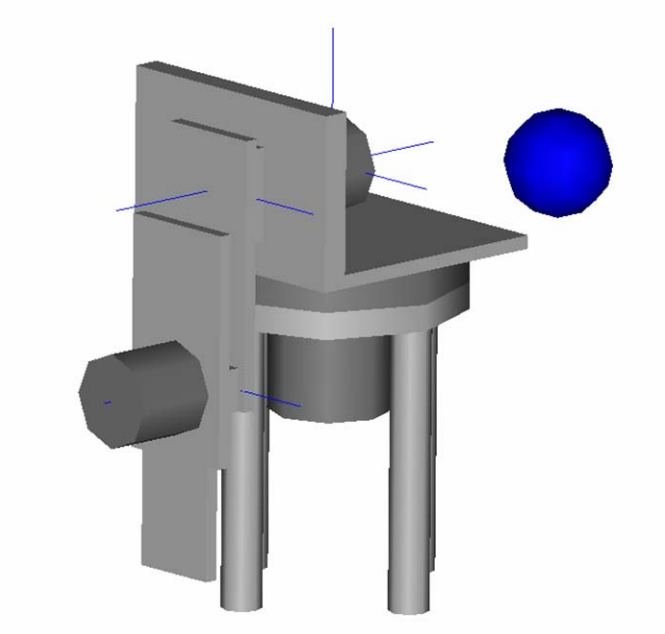


Figure 35: Robot arm before inverse solution

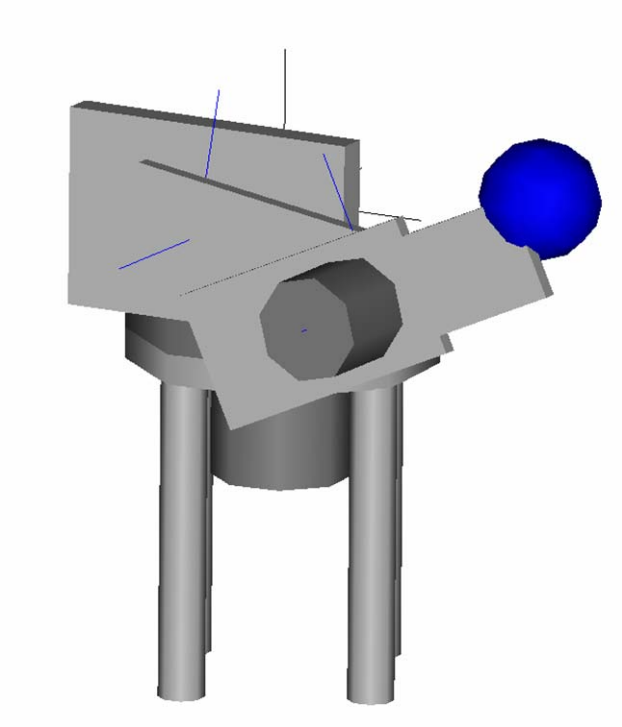


Figure 36: Robot arm translated to end point

Elements of the inverse solution include path planning, curve tracking and the configuration space. A brief commentary on these concepts follows.

Path Planning

The goal of path planning algorithms is to generate the most efficient arrangement of the link frames, taking into account any physical obstacles and system limitations.

Configuration Space

Problems that arise in the motion planning process can be related to the physical design of the robot of the restrictions of the workspace [11]. For example, as the number of available degrees of freedom increase, so do the possible solutions to the inverse transformation. Also, obstacles in the workspace, which includes links of the robot arm, can add difficulty to the solution.

Curve Tracking

When a robot has to follow a complicated path or if the reference point is moving with time, an open-loop controller is sometimes not sufficient. Curve tracking and path planning methods are a vast area of study. This section describes some closed-loop control methods for tracking a robots movement.

Potential Energy Method

[12] outlines using a potential energy field approach to guiding the robot. Sometimes it is desirable for the robot to stay on a certain curve instead of following a reference point. The exact position at a given time is not important, as long as it is somewhere on this desired curve. To solve this problem, a virtual potential field is constructed around the desired curve, such that the potential energy is minimal everywhere on the desired curve, and increases with the deviation from the desired curve. The system can then follow the path of steepest descent to find its commanded position.

[13] extends the use of this method to add control terms that are power-continuous but change the distribution of kinetic energy over the various directions to obtain asymptotic convergence.

Orthogonal Decomposition Control

[14] describes using a vision system to guide the robot. The automatic input is accomplished using a method called Orthogonal Decomposition Control. The center of the camera is defined by the vector **r**. At each sample point, the $\vec{\mathbf{i}} - \vec{\mathbf{j}}$ vector coordinate frame is established, where $\vec{\mathbf{i}}$ is a unit vector tangent to the curve and $\vec{\mathbf{j}}$ is the normal unit vector. The objective is to control the velocity error signal based on the vector tangent to the curve and the position error signal based on the normal vector.

CHAPTER IV

DESIGN OF ROBOT ARM

This section discusses the specific assumptions and design parameters as they relate to the robotic arm constructed for the purposes of this research.

Speed

A triangular velocity profile was assumed for this research. The acceleration and deceleration were both assumed to be 5 seconds. ω_{peak} and α_{peak} are the peak angular velocity and acceleration, respectively and were defined:

$\omega_{peak} = 0.3142\, rad/s$

$\alpha_{peak} = 0.0628\, rad/s^2$

Kinematic Design

The robot was constructed with 3 totally revolute joints. The degree of mobility is three.

Controller

The controller used for this design was a three-axis, open loop resistance limited (R/L) type controller.

Actuator

Stepper Motors were selected as the actuators for this robot arm.

Material

6061-T6 aluminum was used for the majority of the components for this robot arm for its cost, corrosion resistance, and workability. Only the legs were constructed from 321 stainless steel to add weight to the base and increase stability. Table 3 lists the mechanical properties for both materials.

Table 3: Material Properties

Material	Alloy	Tensile Strength (ksi)	Modulus of Elasticity (ksi)	Density (lbs/in^3)
Aluminum	6061-T6	36 Yield 42 Ultimate	9.9×10^3	0.098
Stainless Steel	321	26 Yield 73 Ultimate	29.0×10^3	0.286

Robot Arm Design

The assembly of the 3 link robot arm designed for this thesis was modeled using Autodesk Inventor software. Figure 37 shows the assembly while figure 38 gives an exploded view of the components.

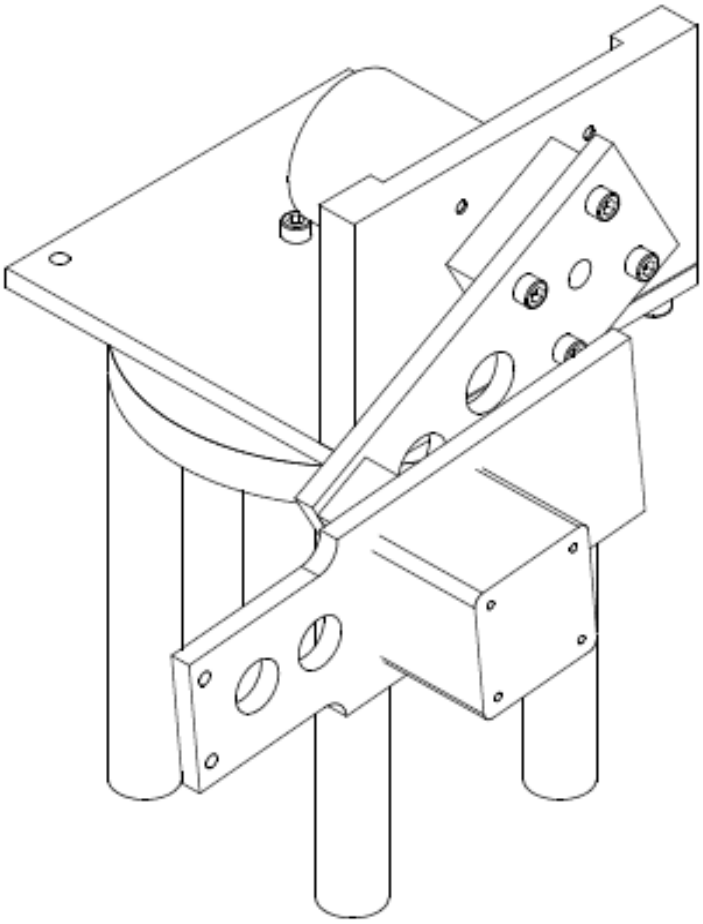


Figure 37: Assembled Robot Arm

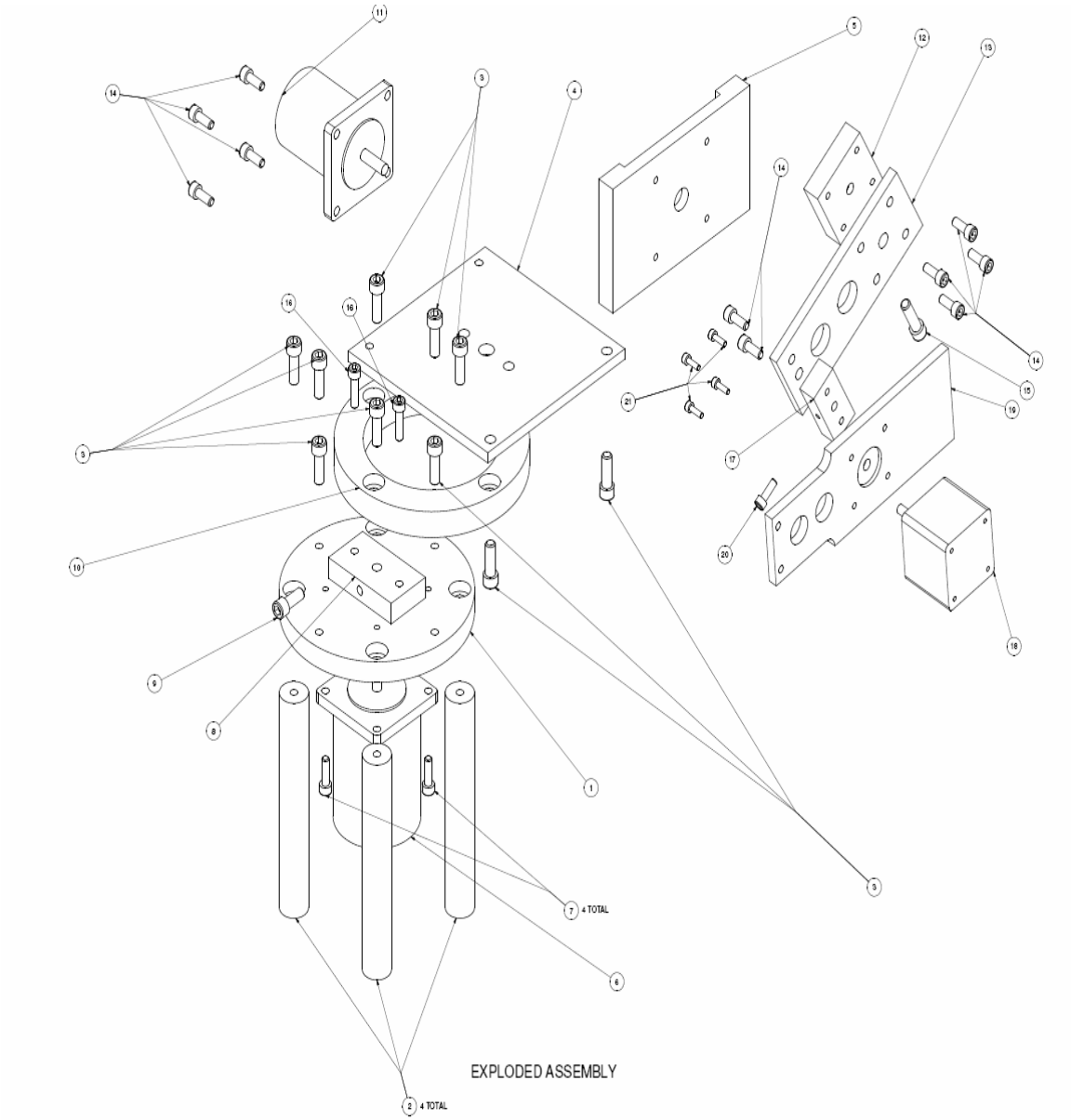


Figure 38: Exploded Assembly View

Including the stepper motors, there are 16 major components to the assembly (table 4). Link 0 is made from parts 4 through 9, parts 11 through 17 make link 1, and link 2 is made up of parts 18 through 21. A detailed drawing package is in included in Appendix A.

Table 4: Assembly Parts List

Item #	Quantity	Description
1	1	Base
2	4	Leg
3	10	1/4-20 x 3/4” Screws
4	1	Waist
5	1	Shoulder Mount
6	1	Waist Motor (5618L-23)
7	4	#10-24 x 5/8” Screws
8	1	Waist Clamp
9	1	1/4-20 x 5/8” Screw
10	1	Waist Bearing
11	1	Shoulder Motor (5618M-06)
12	1	Shoulder Clamp
13	1	Upper Arm
14	10	#10-24 x 1/2” Screws
15	1	1/4-20 x 7/8” Screw
16	2	#10-24 x 3/4” Screws
17	1	Elbow Clamp
18	1	Elbow Motor (4218M-06)
19	1	Lower Arm
20	1	#8-32 x 1/2” Screw
21	4	M3 x 10 Screws

Robot Workspace

The workspace is defined in figures 39 through 41. The reaching length is 4.5 inches, the height is 15.0 inches and the total rotation at the waist (link 0) is 90°, defined by a range of [0,90] (Figure 40). Link 1 is permitted to have a range of [-40,70] (Figure 41 (a)), for a total rotation of 110° and link 2 is allowed a range of [-80,90] (Figure 41 (b)), for a total rotation of 170°.

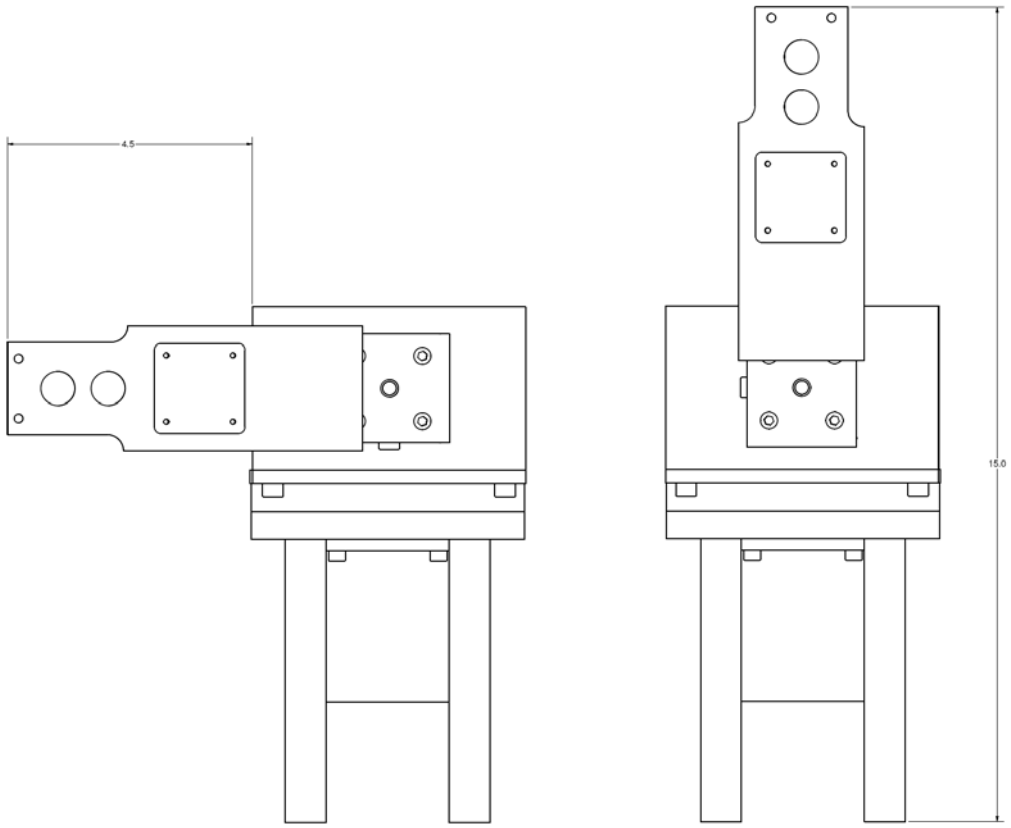


Figure 39: Robot Arm Reach Dimensions

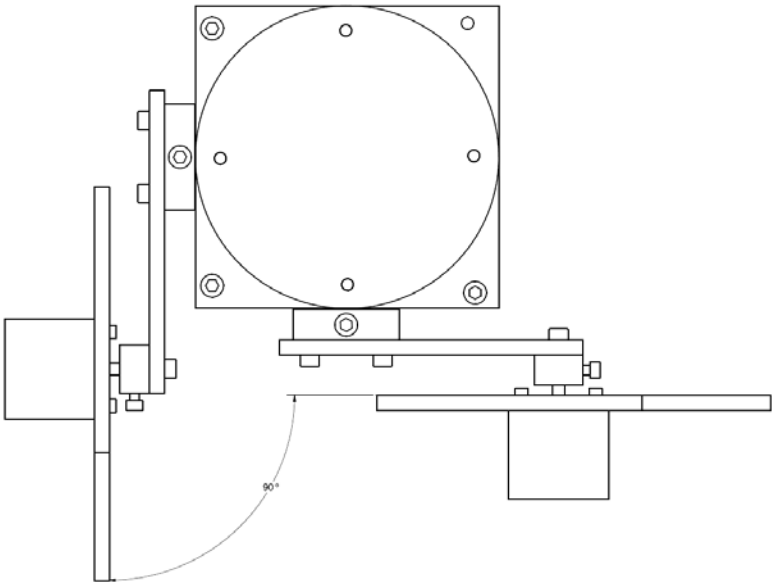
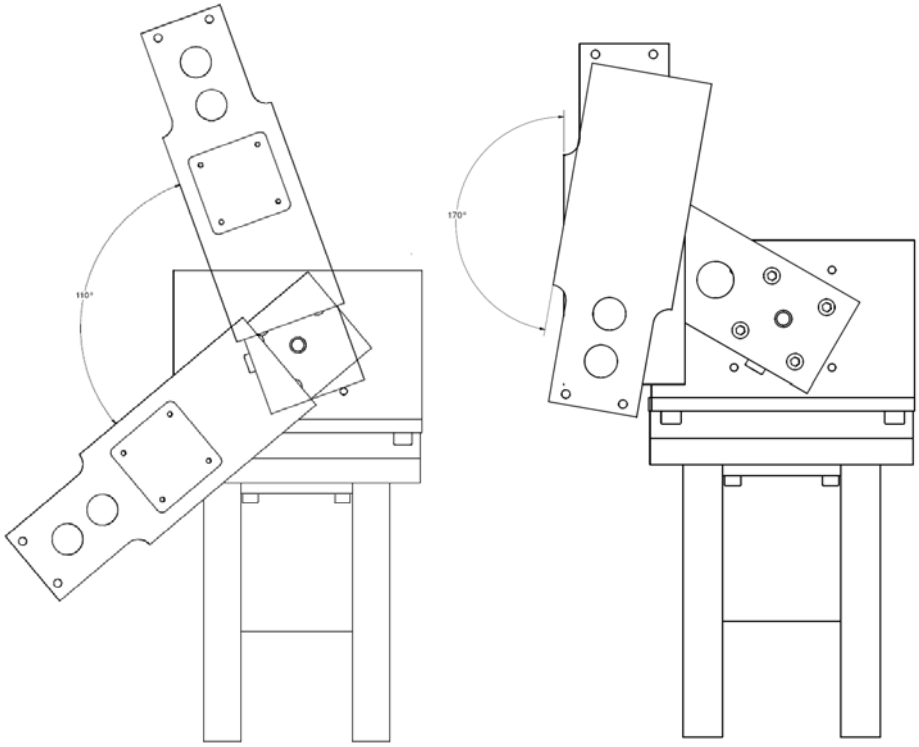


Figure 40: Link 0 Rotation



(a) (b)

Figure 41: Rotation Range for Link 1 (a) and Link 2 (b)

Static Link Analysis

Figure 42 shows a jointed manipulator structure in two dimensional space. The tip of link 2 is positioned by means of revolute joints 1 and 2.

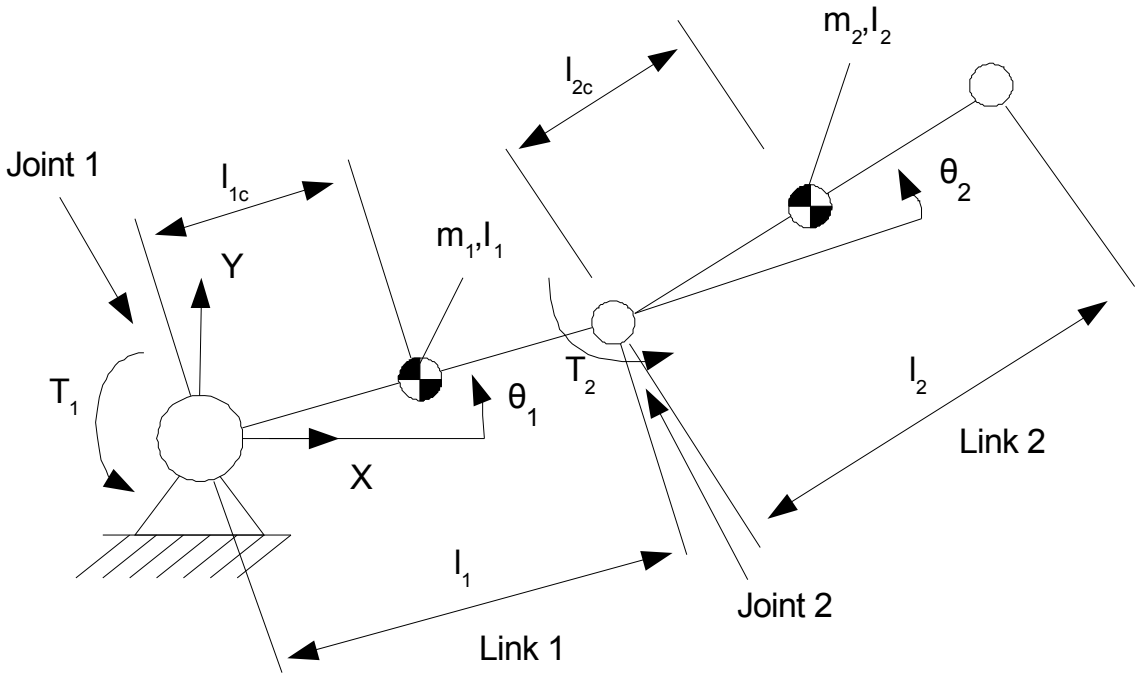


Figure 42: Jointed Manipulator Structure

m_1, m_2, I_1, and I_2 define the mass and moment of inertia about the centers of gravity of links 1 and 2, respectively. The location of the gravity centers is denoted by l_{1c} and l_{2c}, taken from the axes of the respective joints. Angles θ_1 and θ_2 are the rotations of links 1 and 2.

For the purpose of this research, it is assumed that the robot arm is not subjected to any forces relating to tasks or payloads at the tip of link 2. Therefore the only forces exerted are those needed to overcome gravity. Equilibrium with reaction forces and joint

torques must be established. Figure 43 displays the free body diagrams for the two links of the jointed manipulator structure.

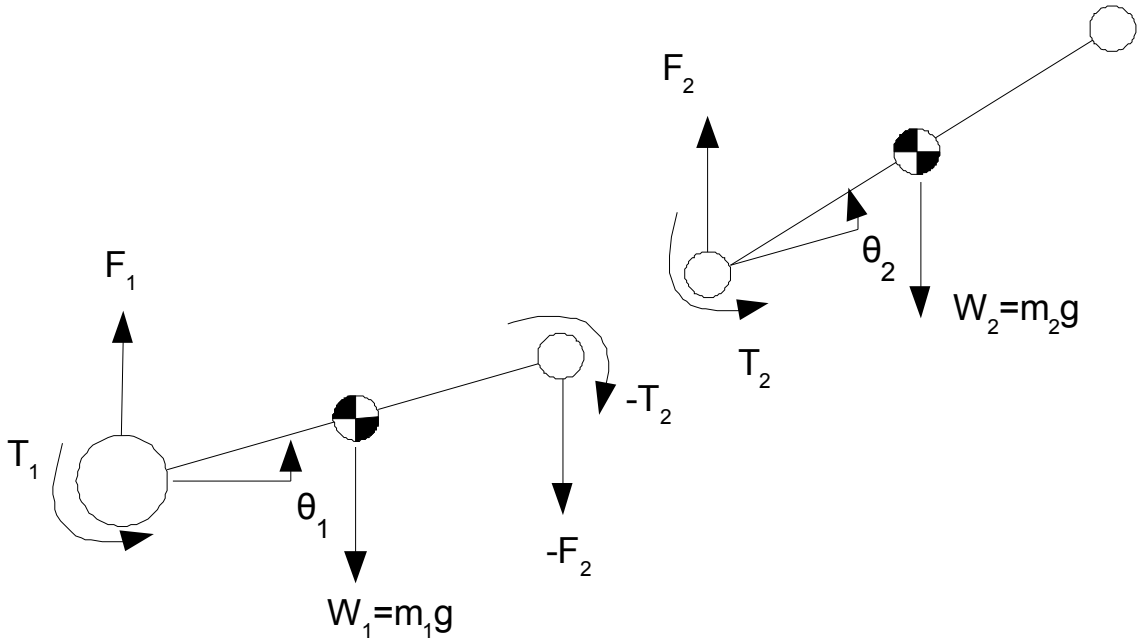


Figure 43: Forces and Moments Imposed on Manipulator Structure

The gravity forces acting through the center of gravity for links 1 and 2 are W_1 and W_2, respectively. Forces F_1 and F_2 and torques T_1 and T_2 must be generated by the joint motors to overcome W_1 and W_2. The magnitudes of these forces are:

Link 1

$$F_1 - F_2 - W_1 = 0 \tag{46}$$

$$T_1 - W_1 l_{1c} \cos\theta_1 - T_2 - F_2 l_1 \cos\theta_1 = 0 \tag{47}$$

Link 2

$$F_2 - W_2 = 0 \tag{48}$$

$$T_2 - W_2 l_{2c} \cos(\theta_1 + \theta_2) = 0 \tag{49}$$

Where:

- $F_1 = (m_1 + m_2)g$
- $F_2 = m_2 g$
- $T_1 = m_1 g l_{1c} \cos\theta_1 + m_2 g[l_{2c} \cos(\theta_1 + \theta_2) + l_1 \cos\theta_1]$
- $T_2 = m_2 g l_{2c} \cos(\theta_1 + \theta_2)$
- $m_1 =$ 1.116 lbs. = 17.86 oz.
- $m_2 =$ 0.217 lbs. = 3.470 oz

Mass m_1 includes the weight of the shoulder clamp (item 12), the elbow clamp (item 17), and the elbow motor (item 18). The torques, T_1 and T_2, required to overcome gravity forces, can become very large in magnitude. However, these magnitudes can be controlled during the design phase by performing actions such as choosing low mass materials, shaping the link to move the center of gravity closer to the joint axis of rotation, or adding a counterbalance to the link.

System Stability

In order to prevent the robot from tipping over when the arms are outstretched, the torque generated by links 1 and 2 at the maximum reaching distance must be less than the counterbalance torque from the robots base.

In figure 44, T_l represents the torque generated by links 1 and 2 through the gravity center of the two components. Torque T_b results from the weights of the base, legs, waist motor, waist, waist clamp, shoulder, and shoulder motor of the robot arm.

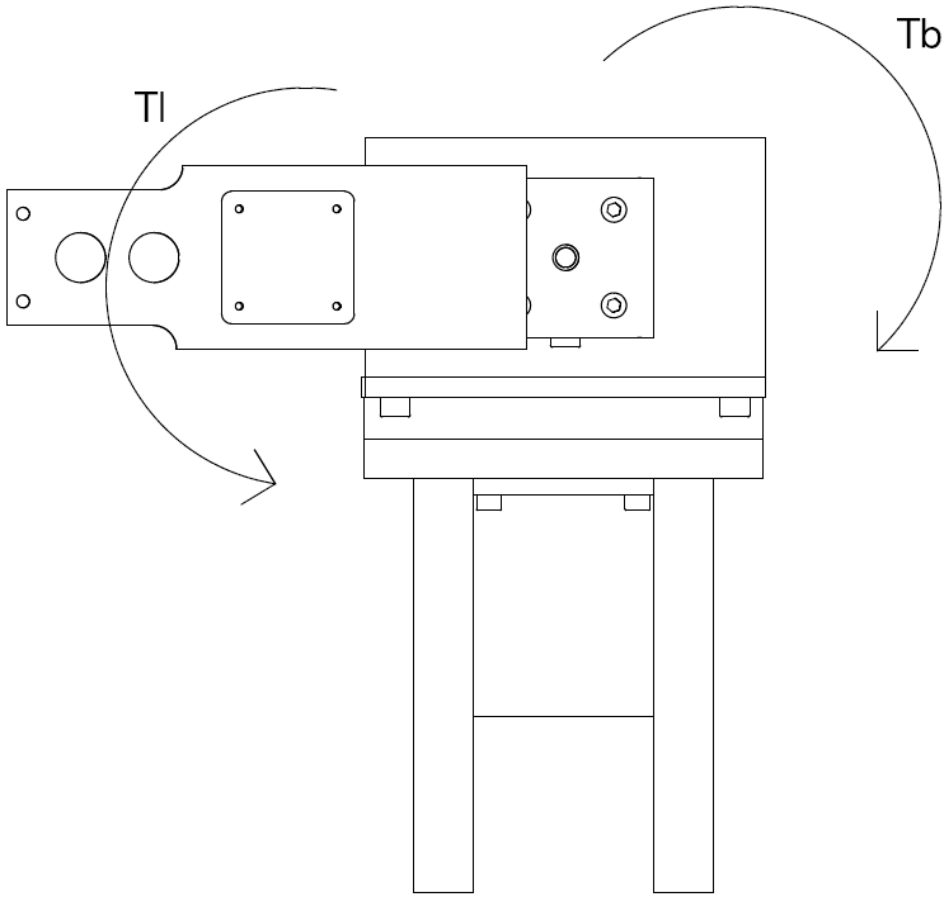


Figure 44: Torques on the Robot Arm Assembly

$T_l = 2.145$ in. lbs.

$T_b = 5.748$ in. lbs.

Since $T_b > T_l$, the robot arm will not tip over.

Dynamic Torque Analysis

In order to calculate the forces required to drive the links at the desired velocity and acceleration profiles, dynamic equations must be used. These equations are related to the Lagrange equations previously described.

Dynamic Torque Link 0

For link 0 the kinetic and potential energy are expressed as

$$K_0 = I_0 \frac{\dot{\theta}_0^{\,2}}{2} \qquad V_0 = 0 \tag{50}$$

The torque T_{00}, which has to be developed by the waist joint motor to support the link 0 motion, is determined by the Lagrange equation

$$T_{00} = \frac{d}{dt}\left(\frac{\partial K_0}{\partial \dot{\theta}_0}\right) - \frac{\partial K_0}{\partial \theta_0} \qquad \text{or} \qquad T_{00} = I_0 \ddot{\theta}_0 \tag{51}$$

Dynamic Torque Link 1

For link 1, the position vector r_{1g}, the origin of which is at the center O_1, and the velocity v_{1g} of the center of gravity of link 1 are

$$r_{1g} = l_{1c}\left[(\cos\theta_0 \cos\theta_1)i + (\sin\theta_0 \sin\theta_1)j + (\sin\theta_1)k\right] \tag{52}$$

$$\begin{aligned} v_{1g} = \dot{r}_{1g} = l_{1c}\{&-[(\sin\theta_0 \cos\theta_1)\dot{\theta}_0 + (\cos\theta_0 \sin\theta_1)\dot{\theta}_1]i \\ &+ [(\cos\theta_0 \cos\theta_1)\dot{\theta}_0 + (\sin\theta_0 \sin\theta_1)\dot{\theta}_1]j \\ &+ [(\cos\theta_1)\dot{\theta}_0]k\} \end{aligned} \tag{53}$$

The kinetic energy of link 1 is:

$$K_1 = m_1 \frac{{v_{1g}}^2}{2} + I_1 \frac{{\omega_1}^2}{2} \tag{54}$$

And the magnitude of the angular velocity is defined as:

$$\omega_1 = \frac{v_{1g}}{l_{1c}} \tag{55}$$

After substituting equations. (52) and (54) into equation (53) one would obtain:

$$K_1 = \frac{m_1 {l_{1c}}^2 + I_1}{2}\left(\dot{\theta}_0{}^2 \cos^2\theta_1 + \dot{\theta}_1{}^2\right) \tag{56}$$

To support the motion of link 1, torques T_{01} and T_{11} are developed by the waist joint motor and by the shoulder joint motor, respectively.

$$T_{01} = \frac{d}{dt}\left(\frac{\partial K_1}{\partial \dot{\theta}_0}\right) - \frac{\partial K_1}{\partial \theta_0} = \left(m_1 {l_{1c}}^2 + I_1\right)\left(\ddot{\theta}_0 \cos^2\theta_1 - 2\dot{\theta}_0\,\dot{\theta}_1 \sin\theta_1 \cos\theta_1\right) \tag{57}$$

$$T_{11} = \frac{d}{dt}\left(\frac{\partial K_1}{\partial \dot{\theta}_1}\right) - \frac{\partial K_1}{\partial \theta_1} = \left(m_1 {l_{1c}}^2 + I_1\right)\left(\ddot{\theta}_1 + \dot{\theta}_0{}^2 \sin\theta_1 \cos\theta_1\right) \tag{58}$$

In equation (56), the expression $\left(m_1 {l_{1c}}^2 + I_1\right)\cos^2\theta_1$ represents the inertia of link 1 about vertical axis Z. This inertia multiplied by the angular acceleration of link 0 gives the torque required to produce that acceleration. The second term computes a coriolis force. In equation (57), the first term is the torque required to accelerate link 1 having inertia $\left(m_1 {l_{1c}}^2 + I_1\right)$ relative to joint 1 (shoulder). The second term is a centrifugal force.

Dynamic Torque Link 2

For link 2, the position vector of center of gravity of link 2 relative to O_1 is

$$r_{2g} = \cos\theta_0[l_1\cos\theta_1 + l_{2c}\cos(\theta_1+\theta_2)]\,i$$
$$+\sin\theta_0[l_1\cos\theta_1 + l_{2c}\cos(\theta_1+\theta_2)]\,j \quad (59)$$
$$+[l_1\sin\theta_1 + l_{2c}\cos(\theta_1+\theta_2)]k$$

The velocity center of gravity of link 2 is

$$v_{2g} = \dot{r}_{2g} = \{-[l_1\cos\theta_1 + l_{2c}\cos(\theta_1+\theta_2)]\dot{\theta}_0\sin\theta_0$$
$$-[l_1\dot{\theta}_1\sin\theta_1 + l_{2c}\left(\dot{\theta}_1+\dot{\theta}_2\right)\sin(\theta_1+\theta_2)]\cos\theta_0\}i$$
$$+\{[l_1\cos\theta_1 + l_{2c}\cos(\theta_1+\theta_2)]\dot{\theta}_0\cos\theta_0 \quad (60)$$
$$-[l_1\dot{\theta}_1\sin\theta_1 + l_{2c}\left(\dot{\theta}_1+\dot{\theta}_2\right)\sin(\theta_1+\theta_2)]\sin\theta_0\}\,j$$
$$+[l_1\dot{\theta}_1\cos\theta_1 + l_{2c}\left(\dot{\theta}_1+\dot{\theta}_2\right)\cos(\theta_1+\theta_2)]k$$

$$v_{2g}{}^2 = v_{2g}\cdot v_{2g} = [l_1{}^2\cos^2\theta_1 + 2l_1l_{2c}\cos\theta_1\cos(\theta_1+\theta_2)$$
$$+l_{2c}{}^2\cos^2(\theta_1+\theta_2)]\dot{\theta}_0{}^2 + l_1{}^2\dot{\theta}_1{}^2 \quad (61)$$
$$+2l_1l_{2c}\dot{\theta}_1\left(\dot{\theta}_1+\dot{\theta}_2\right)\cos\theta_2 + l_{2c}{}^2\left(\dot{\theta}_1+\dot{\theta}_2\right)^2$$

The magnitude of angular velocity ω_2 of link 2 is

$$\omega_2 = \left| v_{2g} - \frac{l_1}{l_{1c}} v_{1g} \right| \Big/ l_{2c} \tag{62}$$

Where $(l_1/l_{1c})v_{1g}$ is the velocity of the joint between links 1 and 2. Inserting equations (52) and (59) produces

$$\begin{aligned} \omega_2{}^2 &= \left(v_{2g} - \frac{l_1}{l_{1c}} v_{1g} \right)\left(v_{2g} - \frac{l_1}{l_{1c}} v_{1g} \right) \Big/ l_{2c}{}^2 \\ &= \dot{\theta}_0 \cos^2(\theta_1 + \theta_2) + (\dot{\theta}_1 + \dot{\theta}_2)^2 \end{aligned} \tag{63}$$

Thus the kinetic energy of link 2 is

$$\begin{aligned} K_2 = \frac{1}{2}\{ & m_2[l_1{}^2 \cos^2\theta_1 + 2l_1 l_{2c} \cos\theta_1 \cos(\theta_1 + \theta_2) \\ & + l_{2c}{}^2 \cos^2(\theta_1 + \theta_2)]\dot{\theta}_0{}^2 + m_2 l_1{}^2 \dot{\theta}_1{}^2 \\ & + 2m_2 l_1 l_{2c} \dot{\theta}_1(\dot{\theta}_1 + \dot{\theta}_2)\cos\theta_2 + (m_2 l_{2c}{}^2 + I_2)(\dot{\theta}_1 + \dot{\theta}_2)^2 \\ & + I_2 \dot{\theta}_0{}^2 \cos^2(\theta_1 + \theta_2)\} \end{aligned} \tag{64}$$

And the torques in the motors of joints 0, 1, and 2, respectively, required to support motion of link 2 are

$$T_{02} = \frac{d}{dt}\left(\frac{\partial K_2}{\partial \dot{\theta}_0}\right) - \frac{\partial K_2}{\partial \theta_0} = \{m_2[l_1^2 \cos^2\theta_1 + 2l_1 l_{2c}\cos\theta_1 \cos(\theta_1+\theta_2)]$$

$$+ \left(m_2 l_{2c}^2 + I_2\right)\cos^2(\theta_1+\theta_2)\}\ddot{\theta}_0 - 2\{m_2[l_1^2 \sin\theta_1\cos\theta_1$$

$$+ l_1 l_{2c}[\sin\theta_1\cos(\theta_1+\theta_2) + \cos\theta_1\sin(\theta_1+\theta_2)]$$

$$+ \left(m_2 l_{2c}^2 + I_2\right)\sin(\theta_1+\theta_2)\cos(\theta_1+\theta_2)\}\dot{\theta}_0\dot{\theta}_1 \qquad (65)$$

$$- 2[m_2 l_1 l_{2c}\cos\theta_1\sin(\theta_1+\theta_2)$$

$$+ \left(m_2 l_{2c}^2 + I_2\right)\sin(\theta_1+\theta_2)\cos(\theta_1+\theta_2)]\dot{\theta}_0\dot{\theta}_2$$

$$T_{12} = \frac{d}{dt}\left(\frac{\partial K_2}{\partial \dot{\theta}_1}\right) - \frac{\partial K_2}{\partial \theta_1} = (m_2 l_1^2 + 2m_2 l_1 l_{2c}\cos\theta_2 + m_2 l_{2c}^2 + I_2)\ddot{\theta}_1$$

$$+ \left(m_2 l_1 l_{2c}\cos\theta_2 + m_2 l_{2c}^2 + I_2\right)\ddot{\theta}_2 + \{m_2[l_1^2\sin\theta_1\cos\theta_1$$

$$+ l_1 l_{2c}\sin\theta_1\cos(\theta_1+\theta_2) + l_1 l_{2c}\cos\theta_1\sin(\theta_1+\theta_2) \qquad (66)$$

$$+ l_{2c}^2\sin(\theta_1+\theta_2)\cos(\theta_1+\theta_2)] + I_2\sin(\theta_1+\theta_2)\cos(\theta_1+\theta_2)\}\dot{\theta}_0^2$$

$$- (m_2 l_1 l_{2c}\sin\theta_2)\dot{\theta}_2^2 - \left(2m_2 l_1 l_{2c}\sin\theta_2\right)\dot{\theta}_1\dot{\theta}_2$$

$$T_{22} = \frac{d}{dt}\left(\frac{\partial K_2}{\partial \dot{\theta}_2}\right) - \frac{\partial K_2}{\partial \theta_2} = \left(m_2 l_1 l_{2c} \cos\theta_2 + m_2 {l_{2c}}^2 + I_2\right)\ddot{\theta}_1 + (m_2 {l_{2c}}^2 + I_2)\ddot{\theta}_2$$

$$+ \{m_2[l_1 l_{2c} \cos\theta_1 \sin(\theta_1 + \theta_2) + {l_{2c}}^2 \sin(\theta_1 + \theta_2)\cos(\theta_1 + \theta_2)] \tag{67}$$

$$+ I_2 \sin(\theta_1 + \theta_2)\cos(\theta_1 + \theta_2)\}\dot{\theta}_0^2 + (m_2 l_1 l_{2c} \sin\theta_2)\dot{\theta}_1^2$$

Assuming link 0 is vertical, the potential energy is

$$V = g\{m_1 l_{1c} \sin\theta_1 + m_2[l_1 \sin\theta_1 + l_{2c} \sin(\theta_1 + \theta_2)]\} \tag{68}$$

Accordingly, the gravity-related components of the torque in the actuators of joints 1 and 2 are as follows

$$T_{1g} = -\frac{\partial V}{\partial \theta_1} = -g\{m_1 l_{1c} \cos\theta_1 + m_2[l_1 \cos\theta_1 + l_{2c} \cos(\theta_1 + \theta_2)]\} \tag{69}$$

$$T_{2g} = -\frac{\partial V}{\partial \theta_2} = -g[m_2 l_{2c} \cos(\theta_1 + \theta_2)] \tag{70}$$

These torque components are added to the motion-related components from equations (57), (65), and (66). The overall torques to be developed by actuators in joints 0, 1, and 2 (T_0, T_1, and T_2) are calculated as follows

$$T_0 = T_{00} + T_{01} + T_{02} = \{I_0 + \left(m_1 l_{1c}{}^2 + I_1\right)\cos^2\theta_1 + m_2 l_1[l_1\cos^2\theta_1$$

$$+ 2l_{2c}\cos\theta_1\cos(\theta_1+\theta_2)] + \left(m_2 l_{2c}{}^2 + I_2\right)\cos^2(\theta_1+\theta_2)\}\ddot{\theta}_0$$

$$- 2\{\left(m_1 l_{1c}{}^2 + I_1\right)\sin\theta_1\cos\theta_1 + m_2 l_1[l_1\sin\theta_1\cos\theta_1 + l_{2c}\sin\theta_1\cos(\theta_1+\theta_2) \quad (71)$$

$$+ l_{2c}\cos\theta_1\sin(\theta_1+\theta_2) + \left(m_2 l_{2c}{}^2 + I_2\right)\sin(\theta_1+\theta_2)\cos(\theta_1+\theta_2)]\}\dot{\theta}_0\dot{\theta}_1$$

$$- 2[m_2 l_1 l_{2c}\cos\theta_1\sin(\theta_1+\theta_2) + \left(m_2 l_{2c}{}^2 + I_2\right)\sin(\theta_1+\theta_2)\cos(\theta_1+\theta_2)]\dot{\theta}_0\dot{\theta}_2$$

$$T_1 = T_{11} + T_{12} + T_{1g} = (m_1 l_{1c}{}^2 + I_1 + m_2 l_1{}^2 + 2m_2 l_1 l_{2c}\cos\theta_2 + m_2 l_{2c}{}^2 + I_2)\ddot{\theta}_1$$

$$+ (m_2 l_1 l_{2c}\cos\theta_2 + m_2 l_{2c}{}^2 + I_2)\ddot{\theta}_2 + \{\left(m_1 l_{1c}{}^2 + I_1\right)\sin\theta_1\cos\theta_1 + m_2[l_1{}^2\sin\theta_1\cos\theta_1$$

$$+ l_1 l_{2c}\sin\theta_1\cos(\theta_1+\theta_2) + l_1 l_{2c}\cos\theta_1\sin(\theta_1+\theta_2) + l_{2c}{}^2\sin(\theta_1+\theta_2)\cos(\theta_1+\theta_2) \quad (72)$$

$$+ I_2\sin(\theta_1+\theta_2)\cos(\theta_1+\theta_2)]\}\dot{\theta}_0{}^2 - (m_2 l_1 l_{2c}\sin\theta_2)\dot{\theta}_2{}^2 - (2m_2 l_1 l_{2c}\sin\theta_2)\dot{\theta}_1\dot{\theta}_2$$

$$+ g\{m_1 l_{1c}\cos\theta_1 + m_2[l_1\cos\theta_1 + l_{2c}\cos(\theta_1+\theta_2)]\}$$

$$T_2 = T_{22} + T_{2g} = (m_2 l_1 l_{2c}\cos\theta_2 + m_2 l_{2c}{}^2 + I_2)\ddot{\theta}_1 + (m_2 l_{2c}{}^2 + I_2)\ddot{\theta}_2$$

$$+ \{m_2[l_1 l_{2c}\cos\theta_1\sin(\theta_1+\theta_2) + l_{2c}{}^2\sin(\theta_1+\theta_2)\cos(\theta_1+\theta_2)] \quad (73)$$

$$+ I_2\sin(\theta_1+\theta_2)\cos(\theta_1+\theta_2)\}\dot{\theta}_0{}^2 + (m_2 l_1 l_{2c}\sin\theta_2)\dot{\theta}_1{}^2$$

$$+ g\{m_2 l_{2c}\cos(\theta_1+\theta_2)]$$

Dynamic Force Analysis Observations

The high degree of nonlinearity in the equations along with interaxial coupling could cause vibration to occur in the manipulator structure and affect the systems performance; these terms are especially dominant for link 2. Therefore, this link is the most critical aspect of the design in terms of mass and moment of inertia. Counterbalancing the mass or distributing it around the center of gravity can greatly reduce the coupling terms.

Stepper Motor Selection

The motors used for this project were hybrid stepper motors with a 1.8° step angle. Detailed specifications for the motors are shown in table 5. The motors selected for each link must be powerful enough to support the desired speed and acceleration profiles as well as any motor half-stepping that may be required.

Table 5: Stepper Mechanical Properties

Location	NEMA Size	Amps/ Phase	Torque (oz-in)	Resistance (Ohm/Phase)	Inductance (mH/Phase)	Inertia ($oz\text{-}in^2$)	Weight (Lbs)	# of Leads
Elbow	17	0.9	45	5.7	5	0.27	0.6	6
Shoulder	23	1.2	84	5	8	0.74	1.2	6
Waist	23	0.6	125	17.1	30	1.2	1.9	6

From the torque curves shown in Appendix B, it can be seen that as the rotation speed of the motor increases, the torque output decreases. This has an affect on the maximum movement rate of the robot arm. It should be noted that the no-load torque of the motors is not sufficient to prevent collapse of the robot arm in the event of a power loss. A braking system or battery backup would be needed to protect the system in this event.

Dynamic Torque Calculation

Using equations developed in equations (49) through (72), it is possible to calculate the dynamic torque requirements of the robot arm.

Torque Calculation Link 2

As shown in figure 45, known variables in equation (48) for the arm section of the robot, or link 2, are as follows:

- $l_{2c} = 0$ in.
- $m_2 = 0.217$ lbs. = 3.47 oz.
- $I_2 = 1.158$ lb in^2
- $l_1 = 3.50$ in.

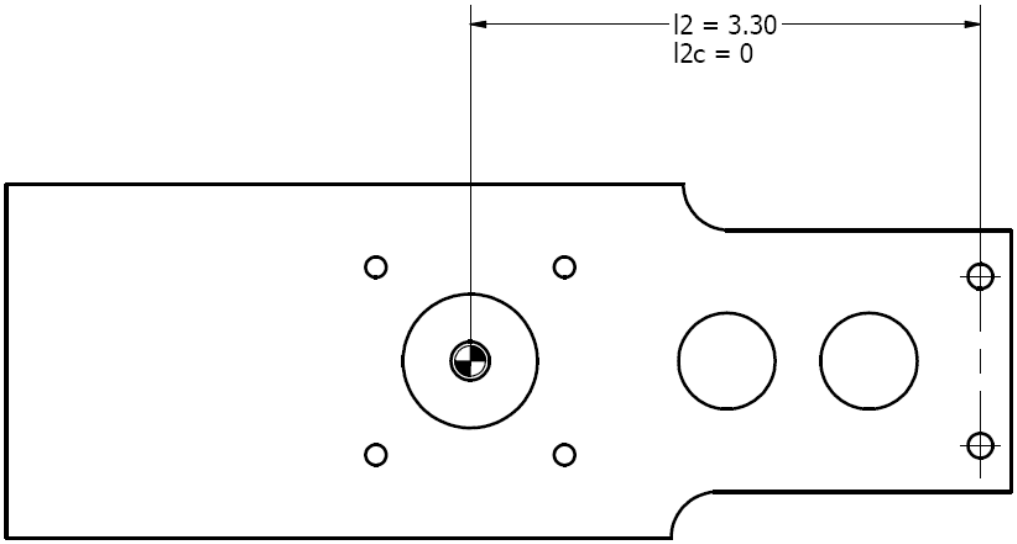


Figure 45: Design Variables Link 2

Figure 41 previously defined the range of rotation angles for link 1 and 2 (θ_1 and θ_2, respectively) as [-40, 70] for link 1 and [-80, 90] for link 2. The maximum static

torque for link 2 occurs when both θ_1 and θ_2 are 0. Substituting the known variables into equations (47) and (48) and solving for T_2 yields 11.46 oz-in. Since 45 > 11.46, the arm motor is acceptable.

The maximum dynamic torque for link 2, defined by equation (72), was solved using Matlab and is shown in figure 46 (see Appendix C for code).

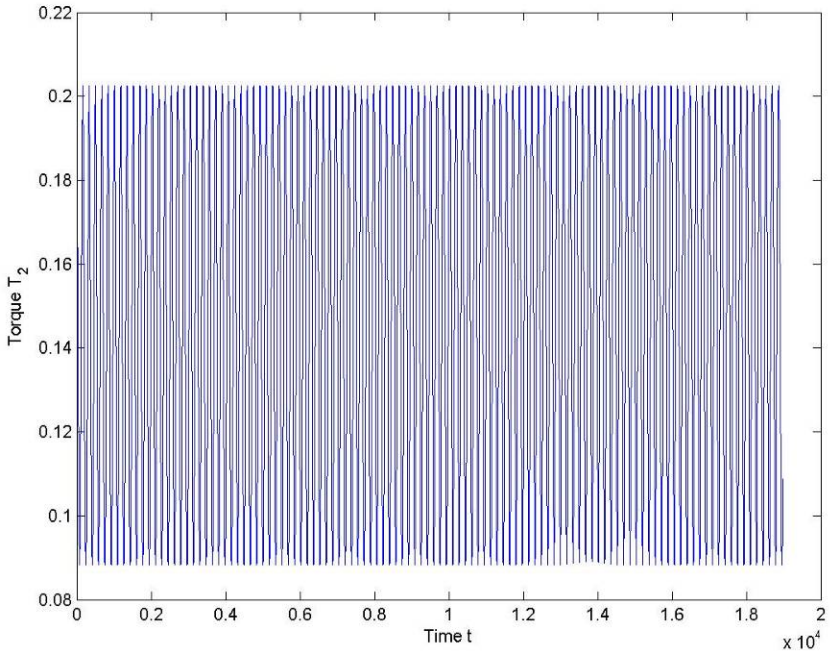


Figure 46: Link 2 Dynamic Torque Curve

Torque Calculation Link 1

As shown in figure 47, known variables for the shoulder section of the robot, or link 1, are as follows:

- $l_{1c} = 1.24$ in.
- $m_1 = 1.116$ lbs. = 17.86 oz. (includes the weight of items 12, 17, and 18)
- $I_1 = 4.246$ lb in^2

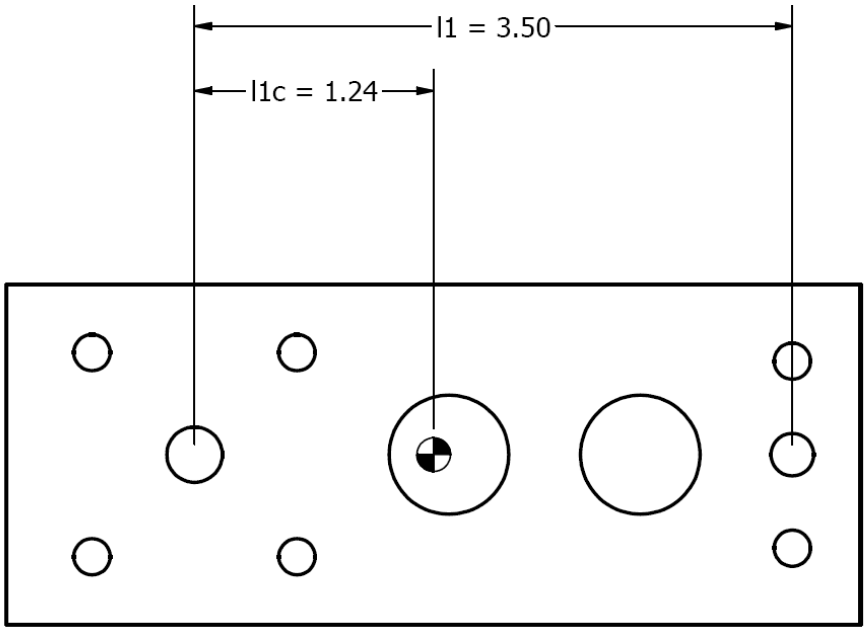


Figure 47: Design Variables Link 1

The maximum dynamic torque for link 1, defined by equation (71), was solved using Matlab and is shown in figure 48 (see Appendix D for code).

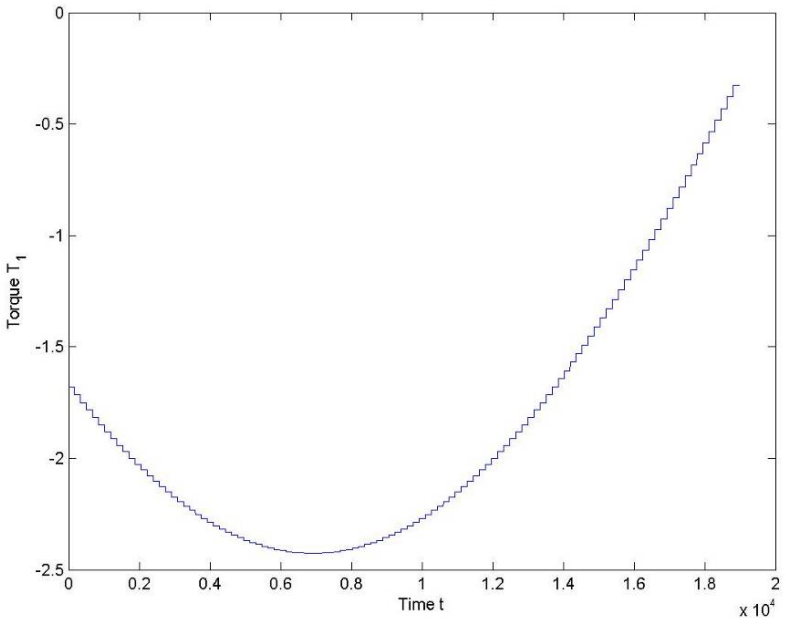


Figure 48: Link 1 Dynamic Torque Curve

The maximum static torque for link 1 also occurs when both θ_1 and θ_2 are 0. Solving for T_1 in equation 46 yields 45.742 oz-in. Since 84 > 45.742, the arm motor is acceptable.

Torque Calculation Link 0

Since the other coordinate systems are based off of link 0, the only variable is the inertia factor, listed below:

$I_0 = 26.494 \text{ lb in}^2$

The maximum dynamic torque for link 0, defined by equation 70, was solved using Matlab and is shown in figure 49 (see Appendix E for code).

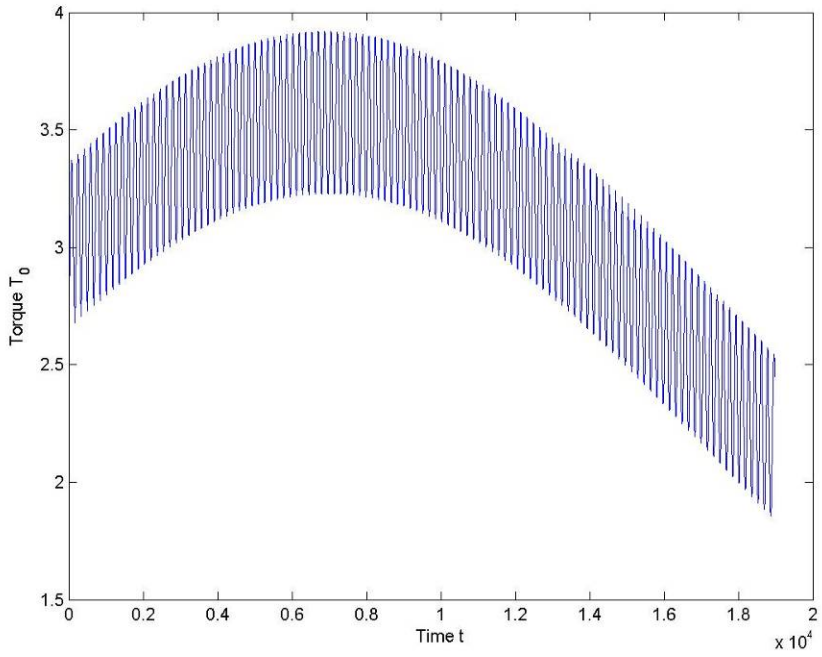


Figure 49: Link 0 Dynamic Torque Curve

Kinetic Link Parameters

The kinetic link parameters for the robot arm are listed in table 6.

Table 6: Link Parameters

Link	α_i	a_i	d_i	θ_i
1	0	0	1.5	θ_1
2	90	0	2.8	θ_2
3	0	3.5	.25	θ_3

The coordinate transformation matrices for the individual links are as follows:

$$ {}^{0}A_1 = \begin{bmatrix} \cos\theta_1 & -\sin\theta_1 & 0 & 0 \\ \sin\theta_1 & \cos\theta_1 & 0 & 0 \\ 0 & 0 & 1 & 1.5 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{74} $$

$$ {}^{1}A_2 = \begin{bmatrix} \cos\theta_2 & 0 & \sin\theta_2 & 0 \\ \sin\theta_2 & 0 & -\cos\theta_2 & 0 \\ 0 & 1 & 0 & 2.8 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{75} $$

$$ {}^{2}A_3 = \begin{bmatrix} \cos\theta_3 & -\sin\theta_3 & 0 & 3.5\cos\theta_3 \\ \sin\theta_3 & \cos\theta_3 & 0 & 3.5\sin\theta_3 \\ 0 & 0 & 1 & 0.25 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{76} $$

Multiplying the matrices produces the kinematic equation *T*:

$$ {}^{0}T_3 = \begin{bmatrix} \cos\theta_3\cos(\theta_1+\theta_2) & -\sin\theta_3\cos(\theta_1+\theta_2) & \sin(\theta_1+\theta_2) & 3.5\cos\theta_3\cos(\theta_1+\theta_2)+0.25\sin(\theta_1+\theta_2) \\ \cos\theta_3\sin(\theta_1+\theta_2) & -\sin\theta_3\sin(\theta_1+\theta_2) & \cos(\theta_1+\theta_2) & 3.5\cos\theta_3\sin(\theta_1+\theta_2)-0.25\cos(\theta_1+\theta_2) \\ \sin\theta_3 & \cos\theta_3 & 0 & 3.5\sin\theta_3+4.3 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{77} $$

Open Loop Control

Based on the transformation matrix derived in (77), a database can be constructed which contains all of the possible end effector locations in the task space. These locations can then be translated into a number of steps that the motors need to take in order to rotate each joint into position.

Selection of Driver

The purpose of a stepper motor driver is to supply the current to the motor windings through a separate power source since the required current is too high to be supplied by a PC.

Stepper Motor Drivers

A stepper motor driver is selected according to the type of stepper motor (ie., unipolar or bipolar) and current/voltage requirements (refer back to table 5 for motor info). For this application, the FET3 system, supplied by StepperWorld, was chosen. The FET3 is a low cost driver for hobby use that proved to be very versatile to the needs of this research. The features of the FET3 board are as follows [15]:

- 3 Axis Unipolar (L/R Type Driver)
- High Current Capability
- Step and Direction Interface (compatible with 3rd party software)
- Dip Switch Mode Selection for each axis - Wave drive, Hi-Torque, Half Step
- 3 Switch Inputs for Limit Detection
- Connects to Parallel Port with Standard DB-25 Cable
- Optional Separate Power Supply Input for Z axis
- 50 VDC Maximum Motor Supply Voltage

- 36 Amp Per Phase Max
- 140 Watts Per Phase Max

A diagram of the control board is shown in figure 50.

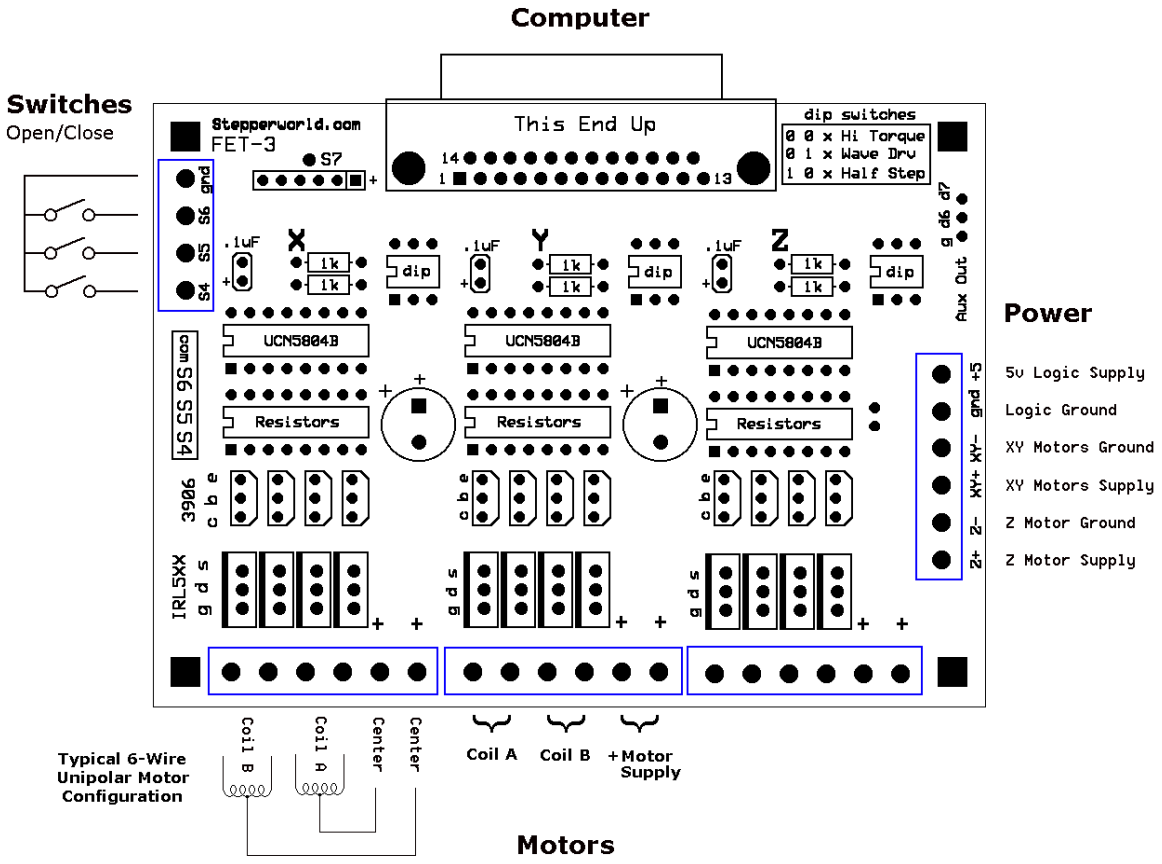


Figure 50: FET-3 Controller Board Diagram

The FET-3 board operates with a step/direction interface, which makes it compatible with most 3rd party CNC software systems. Figure 51 shows the proper configuration of the LPT port for CNC software operation.

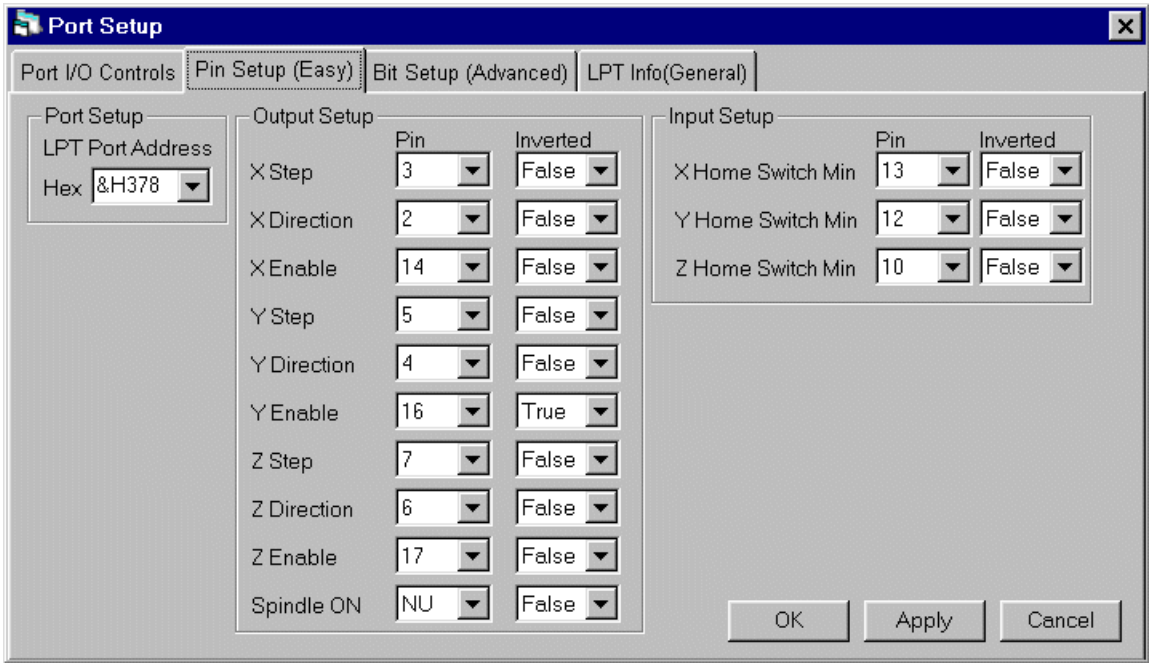


Figure 51: Parallel Port Setup for FET-3 Controller Interface

The purpose of CNC code is to position the machine bed or tool in a manufacturing environment accurately and repeatedly. The code provides speeds and coordinates to actuators that then move load accordingly. This existing technology provides an easy control interface for the arm used in this research. Once the final position of each joint is determined, the command can be entered as a single line of code in the CNC software and the move will be carried out. The command line takes the format shown in figure 52.

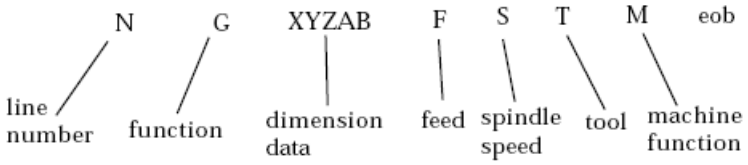


Figure 52: CNC Command Line Format

The function code, designated by the letter "G" in the previous figure, represents the action the machine is to take using the remaining inputs as its setting. Table 7 lists a number of common CNC machine codes or "G-words:"

Table 7: Common CNC G Words

G-Number	Function
G00	Rapid traverse movement
G01	Linear interpolation movement
G02	Circular interpolation movement, CW
G03	Circular interpolation movement, CCW
G04	Dwell
G10	Input cutter offset data
G17	Selection of X-Y plane in milling
G18	Selection of X-Z plane in milling
G19	Selection of Y-Z plane in milling
G20	Input values specified in inches
G21	Input values specified in millimeters
G28	Return to reference point
G32	Thread cutting in turning
G40	Cancel cutter offset
G41	Cutter offset compensation, left of part surface
G42	Cutter offset compensation, right of part surface
G90	Programming in absolute coordinates
G91	Programming in incremental coordinates
G92	Reset coordinate system origin relative to starting location of cutting tool
G94	Specify feed per minute in milling and drilling
G95	Specify feed per revolution in milling and drilling
G98	Specify feed per minute in turning
G99	Specify feed per revolution in turning

In the case of this research, the CNC software will be used only to actuate the arm joints a given angle, therefore the only G function that will be used is G00, the rapid traverse movement. The software will be configured to accept an input defining the number of steps that each motor must take. Figure 53 shows the software interface.

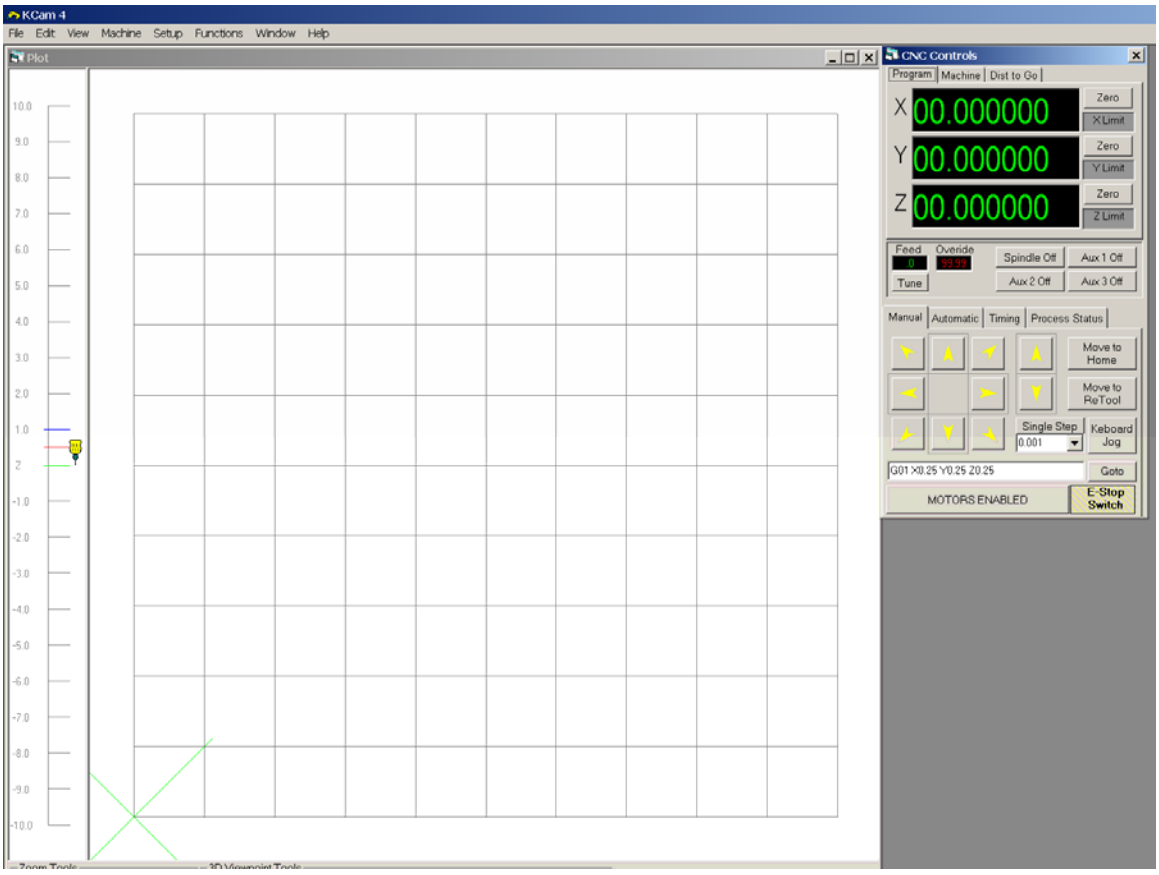


Figure 53: FET-3 Controller Interface

The control board uses the parallel port logic described above to power the stepper motors. Figure 54 shows the pin out of the connector and table 8 describes the function of the pins.

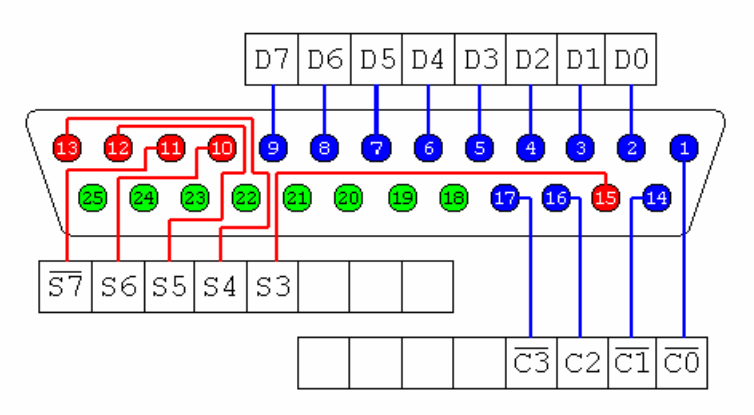


Figure 54: Standard Parallel Port Pinout [2]

Table 8: SP-3/FET-3 Stepper Parallel Port Pin Assignments (DB25 Connector)

Pin	[Bit]	Address	I/O	Function
2-	[D0]	bs	OUTPUT	X axis direction
3-	[D1]	bs	OUTPUT	X axis step
4-	[D2]	bs	OUTPUT	Y axis direction
5-	[D3]	bs	OUTPUT	Y axis step
6-	[D4]	bs	OUTPUT	Z axis direction
7-	[D5]	bs	OUTPUT	Z axis step
8-	[D6]	bs	OUTPUT	Aux Out (d6)
9-	[D7]	bs	OUTPUT	Aux Out (d7)
10-	[S6]	bs+1	OUTPUT	Z switch detect
12-	[S5]	bs+1	OUTPUT	Y switch detect
13-	[S4]	bs+1	OUTPUT	X switch detect
14-	[C1]	bs+2	OUTPUT	X axis *Enable (active LO)
16-	[C2]	bs+2	OUTPUT	Y axis *Enable (active LO)
17-	[C3]	bs+2	OUTPUT	Z axis *Enable (active LO)
18-	Ground			
25-	Ground			

Note that since pins 14 and 17 (data bits C1 and C3) are recognized as outputs from the computer, the signal must be inverted to become an output from the control board. The bit descriptions D, S, and C describe the function assigned to a particular pin. "D" designates a pin as a data output operating on the port base address, or data port. The base address of a parallel port is normally Hex 378, 278, or 3BC. Normally, the data port is a "write only" port (the exception is the case of a bi-directional ports, which are not discussed in this paper). "S" corresponds to a switch input operating on the status port or base address +1, which would be Hex 379, 279, or 3BD respectively. The status port is a "read only" port, meaning any data written to this port is ignored. Finally, "C" coincides with a control output from the PC. In the example of the robot arm, this control output enables the motor on a particular axis. A "C" signal operates on the control port defined as the base address +2, or Hex 37A, 27A, or 3BE. The control port is a "write only" port.

Robot Mechanical Assembly Instructions

Following is a detailed description of how the robot arm used in this research was constructed.

Step 1

Secure Elbow Motor (#18) to Lower Arm (#19) using four M3 x 10 screws.

Step 2

Attach Waist Motor (#6) to Base (#1) using four #10-24 x 5/8" screws.

Step 3

Attach Legs (#2) to Base using four 1/4-20 x 3/4" screws.

Step 4

Secure Waist Clamp (#8) to Waist Motor using one 1/4-20 x 5/8” screw. Ensure the set screw is tight against the flat on the motor shaft.

Step 5

Attach Waist Bearing (#10) to Base using four 1/4-20 x 3/4” screws.

Step 6

Line up through holes on Waist (#4) with Waist Clamp. Attach using two #10-24 x 3/4” Screws.

Step 7

Insert Shoulder Motor (#11) into Shoulder Mount (#5). Fasten in place with four #10-24 x 1/2” Screws.

Step 8

Affix Shoulder Mount to Waist using two 1/4-20 x 3/4” screws.

Step 9

Using four #10-24 x 1/2” screws, attach Shoulder Clamp (#12) to Upper Arm (#13).

Step 10

Assemble Elbow Clamp (#17) to Upper Arm with two #10-24 x 1/2” screws.

Step 11

Insert Shoulder Motor shaft into Shoulder Clamp and secure with 1/4-20 x 7/8” screw. Ensure the set screw is tight against the flat on the motor shaft.

Step 12

Insert Elbow Motor shaft into Elbow Clamp and secure with #8-32 x 1/2” screw. Ensure the set screw is tight against the flat on the motor shaft.

Robot Electrical Assembly Instructions

The wiring of the robot involves coupling the motors and power supplies with the motor controller and computer interface. A description of the steps involved in the electrical connections follows.

Step 1

Attach wire harness to the motor inputs on the driver board. Refer to figure 49 for the locations. Actual connection is shown in figure 55.

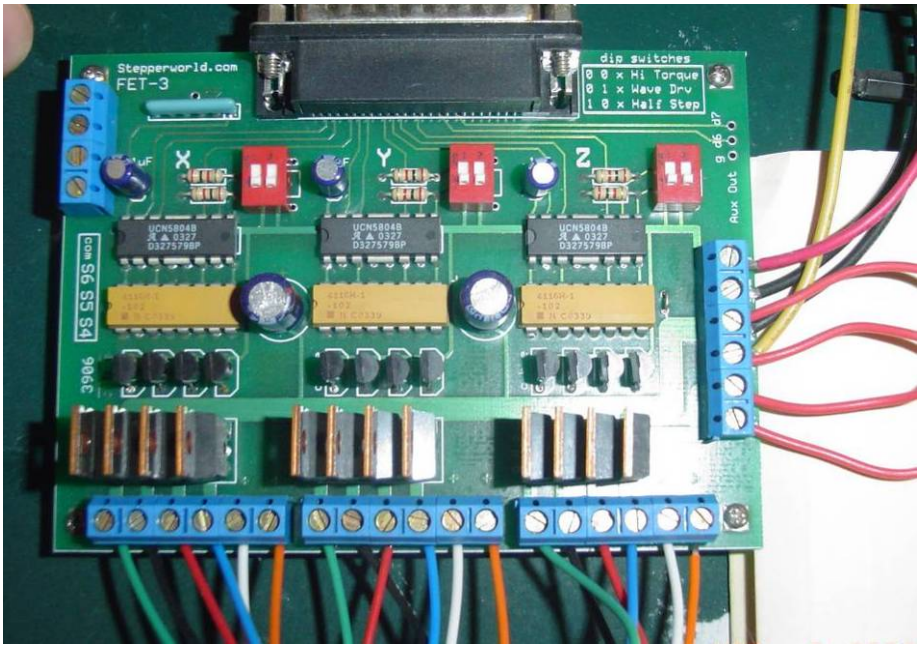


Figure 55: Connecting Wire Harness to Board

Step 2

The motors are 6 lead unipolar motors. They are attached to the driver as previously shown in figure 55, with A and C referring to coil A and B and D referring to coil B. Table 9 lists the wiring requirements for the different motor configurations.

Table 9: 6 Lead Wire Motor Wiring

Color Code 1	Red	White	Blue	Green	Yellow	Black
Color Code 2	Brown	Black	Orange	Red	White	Yellow
Color Code 3	Red	Black	Red/ White	Green	White	Green/ White
Wire	A	A/C Common	C	B	B/D Common	D

The robot has a terminal built onto the waist to allow for a wire harness extension. This allows the robot to be far away from the control board. The junction terminals are in groups of six for each motor, as shown in figure 56. Using the corresponding color code in table 9, attach the motors to the junction block as required. The motors for links 1 and 2 are wired per color code 1 while link 0 follows color code 3.

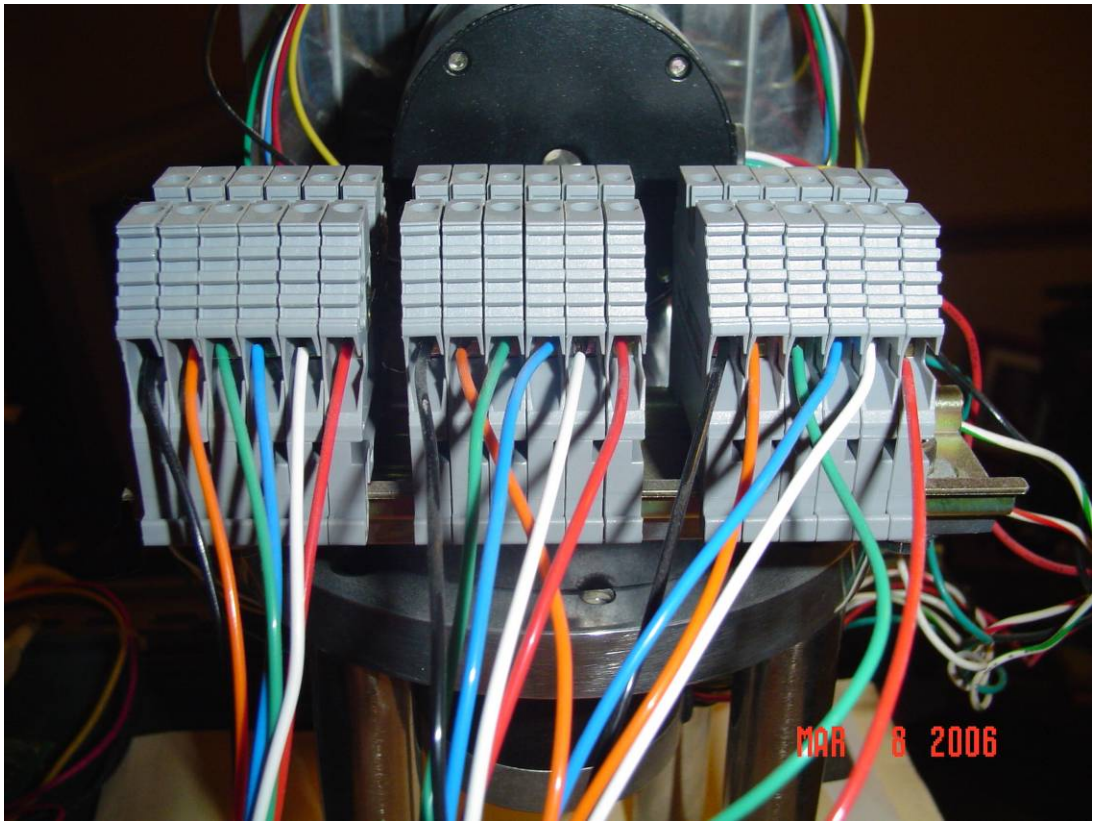


Figure 56: Motor Junction Terminals

Step 3

The driver board allows for the use of two separate power supplies, one for the X and Y axes and a separate supply for the Z axis. This research only used a single, high current power supply so the XY+ terminal was connected to the Z+ terminal as shown in figure 50. The XY- is similarly connected to the Z- terminal to complete the wiring.

CHAPTER V

SUMMARY

Robotic arm systems were covered in detail in this research. The actuators, controllers, and driver hardware systems were introduced. Mathematical properties such as system statics and dynamics, path planning, and coordinate transformations were also covered in detail.

A three link revolute jointed robot arm was developed to demonstrate some of the principles discussed in this research. It was able to actuate to any point in its workspace through the use of a CNC driver board and software. It's simple construction and basic open loop control system made it ideal to demonstrate the concepts brought up during this research.

Further research could be performed on this subject using the robot arm. Some possible improvements on the system could be as follows:

1) Replace stepper motors with servo motors. This would give a greater degree of precision and accuracy to the system. However, servos are more difficult to implement and would require closed loop control.
2) Switch to a closed loop controller. A controller that employs sensors would lead to move accurate movements.
3) Build a dedicated controller. A stand alone controller instead of a PC would allow the controls to be customized for the application.

4) Add a gripper function to the end effector. This could take the form of a simple open/closed gripper or utilize an additional joint to allow for rotation of the object.

BIBLIOGRAPHY

References:

[4] McKerrow, Phillip John: Introduction to Robotics, Addison-Wesley Publishing, 1991

[1] Shelton, Bob: "Types of Robots. NASA ROVer Ranch Website." 2003. <http://prime.jsc.nasa.gov/ROV/types.html>

[2] Beauchemin, George A.: "Stepper Motor Technical Note: Microstepping Myths and Realities." 2003, MicroMo Electronics, Inc. <http://www.micro-drives.com>

[3] Baldor Electric Company: "Servo Control Facts: A Handbook Explaining the Basics of Motion" 1994, Baldor Electric Company. <http://www.baldor.com/pdf/manuals/1205-394.pdf>

[5] New Japan Radio Company: "Drive Circuit Basics." New Japan Radio Company Ltd. <www.njr.co.jp/pdf/ee/ee05002.pdf>

[6] Chernousko, Falex L.: Manipulation Robots, CRC Press, 1993

[7] Peacock, Craig: "Interfacing the Standard Parallel Port." 2005 Beyond Logic. <http://www.beyondlogic.org/spp/parallel.htm>

[8] Tyson, Jeff: "How Parallel Ports Work." 2005 How Stuff Works. <http://computer.howstuffworks.com/parallel-port1.htm>

[9] Kelly, Alonzo: "Introduction to Mobile Robots." 1996, Carnegie Mellon University <https://www.andrew.cmu.edu/course/16-761/Documentation/>

[10] Schwartz, Jacob T, Sharir, Micha: "On the Piano Movers' Problem." 1981, New York University, Dept. of Computer Science.

[11] Amato, Nancy: "A Brief Introduction: The Motion Planning Problem, Configuration Space, Basic Path Planning Methods." 2004, University of Padova. <http://parasol.tamu.edu/~amato/Courses/padova04/lectures/L1.intro.pdf>

[12] Duindam, Vincent, Stramigioli, Stefano: “Passive Asymptotic Curve Tracking.” IFAC Workshop on Lagr. and Hamilt. Methods for Nonlinear Control, pages 229 – 234 April 2003

[13] Vincent Duindam, Stefano Stramigioli: “Port-Based Asymptotic Curve Tracking for Mechanical Systems.” European Journal of Control, Volume 10(5), pages 411 – 420 December 2004

[14] Jing Lin, Hui-Tang Chen, Ping Jiang, Yue-Juan Wang, Peng-Yung Woo. Curve Tracking and Reproduction by a Robot with a Vision System

[15] Laidman Russell. “Interfacing the FET-3 Stepper Controller”. 2000,StepperWorld <http://www.stepperworld.com>

[16] Benson, Harris. University Physics. 1996 Rand McNally NewYork ISBN 0-471-00689-0

[17] Brown, Ward. Brushless DC Motor Control Made Easy. 2002, Microchip Technology Inc.

[18] Laidman Russell. “Unipolar Stepper Motors and Control”. 1999 StepperWorld <http://www.stepperworld.com>

[19] Malcom, Douglas R. Robotics, an Introduction. 1988. Delmar Publishers. Albany, New York

[20] Advanced Micro Systems: “Stepper Motor System Basics.” 2008, Advanced Micro Systems. <http://www.ams2000.com/stepping101.html>

[21] Jones, Douglas W. “Control of stepping motors,” 1998, University of Iowa. <http://www.cs.uiowa.edu/~jones/step/>

[22] Liu, Yuhang. Design and Control Simulation of a Robot Arm. Master’s Thesis 2004, University of Akron.

[23] Lin Engineering. Data sheets 1.8° NEMA size 17 standard step motor.

[24] Lin Engineering. Data sheets 1.8° NEMA size 23 standard step motor.

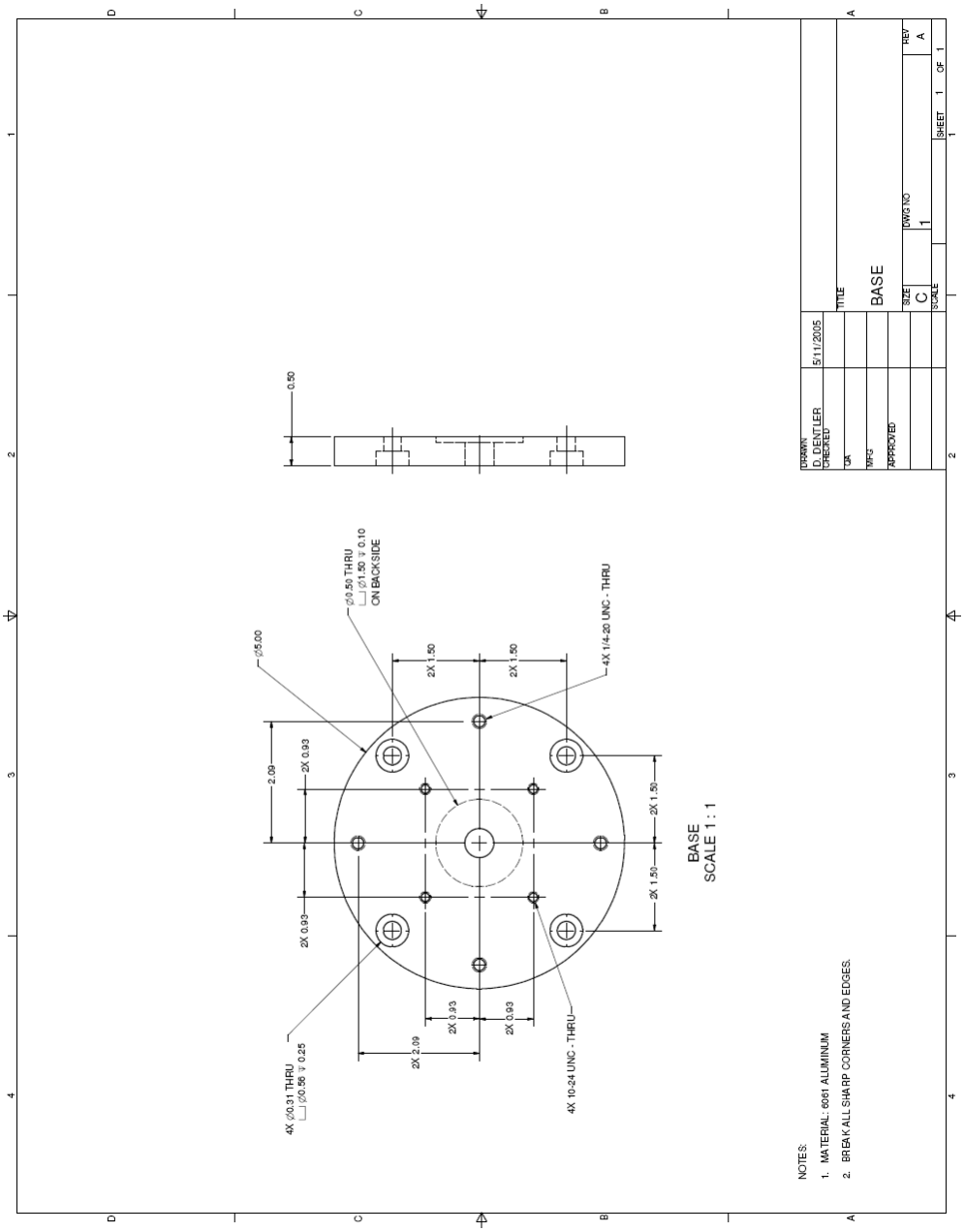
[25] Lin Engineering. Data sheets Wiring diagram 6 lead standard step motor.

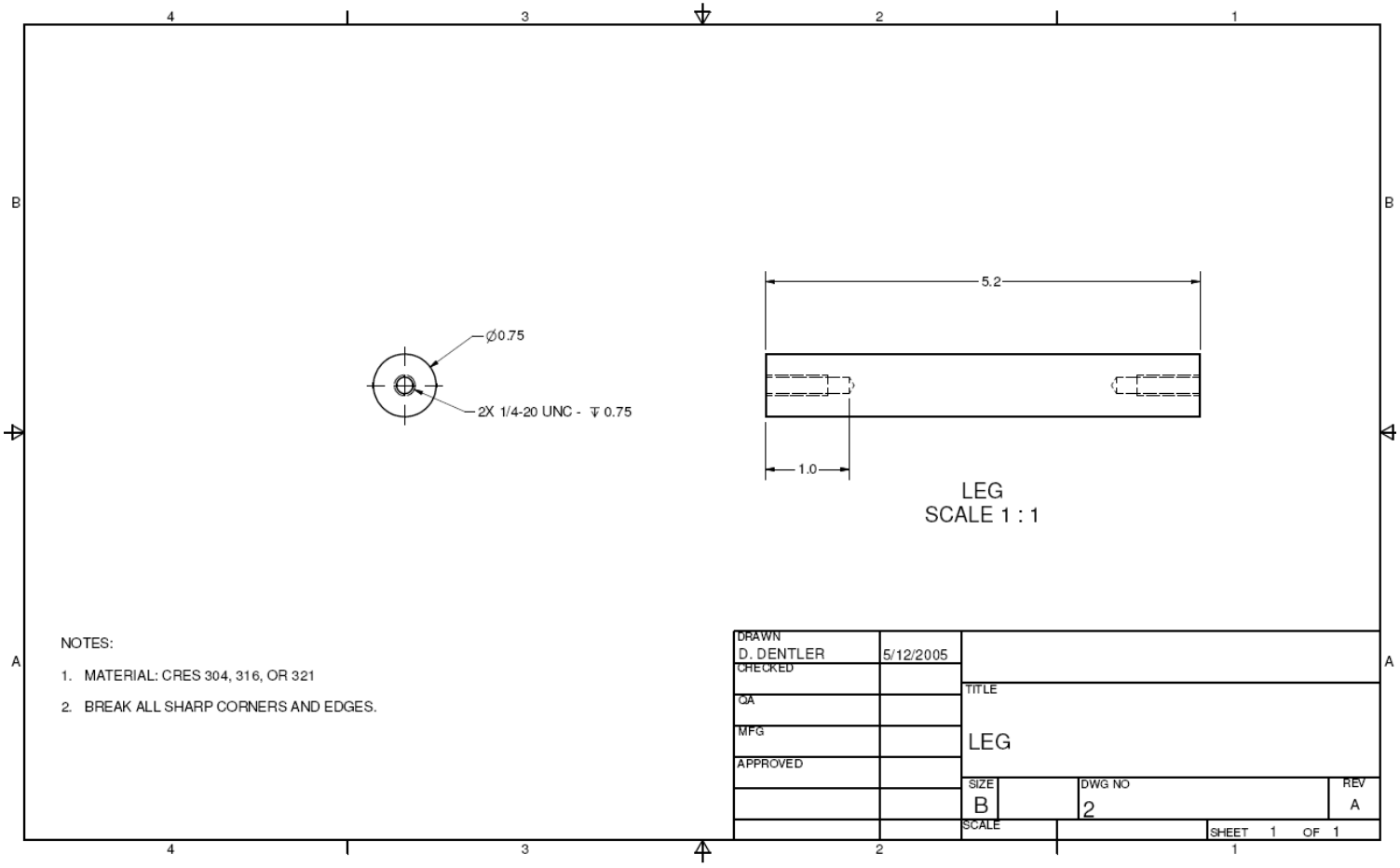
[26] Laidman, Russell. Data sheet FET-3 stepper motor controller. 2000, StepperWorld <http://www.stepperworld.com>

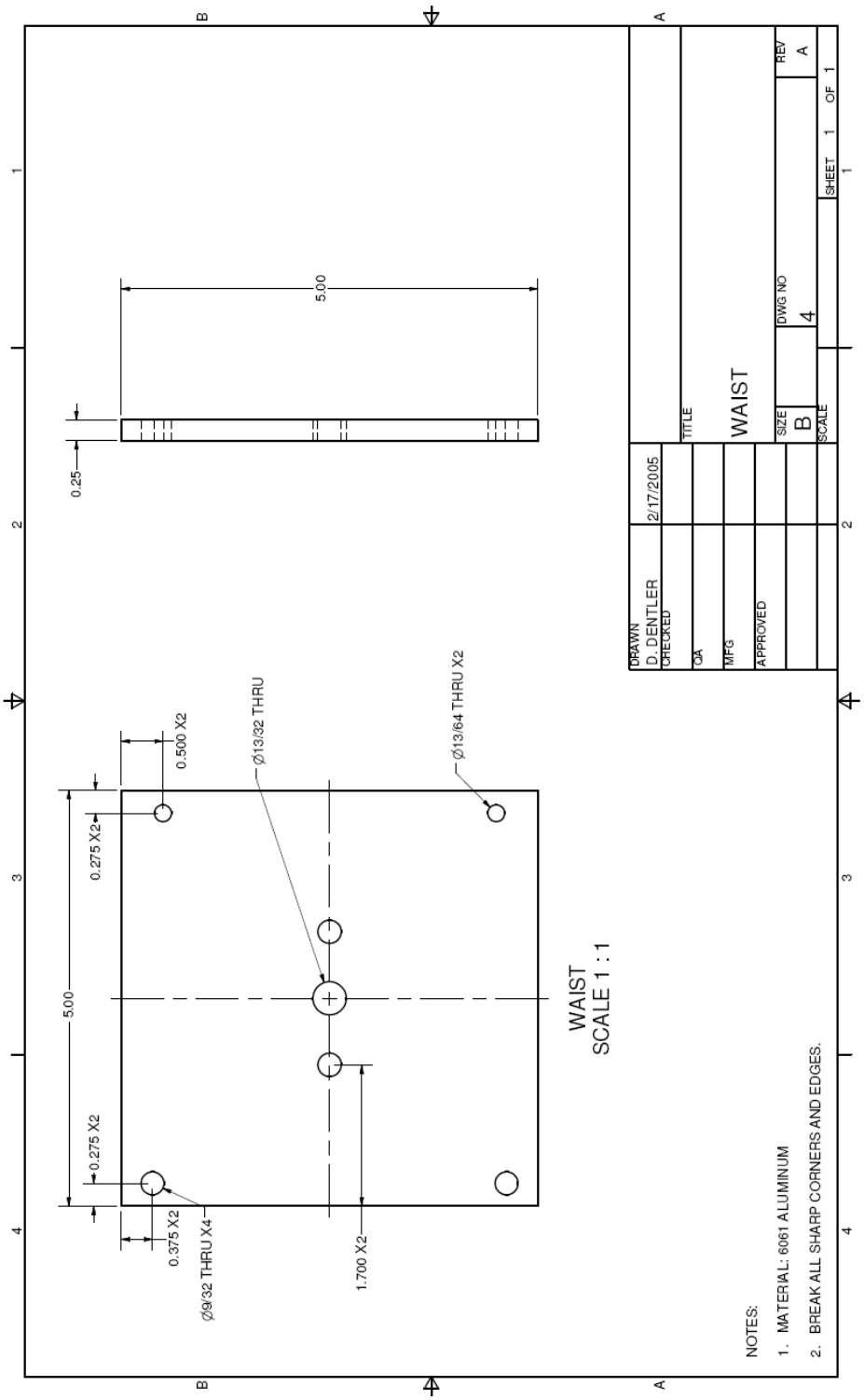
APPENDICES

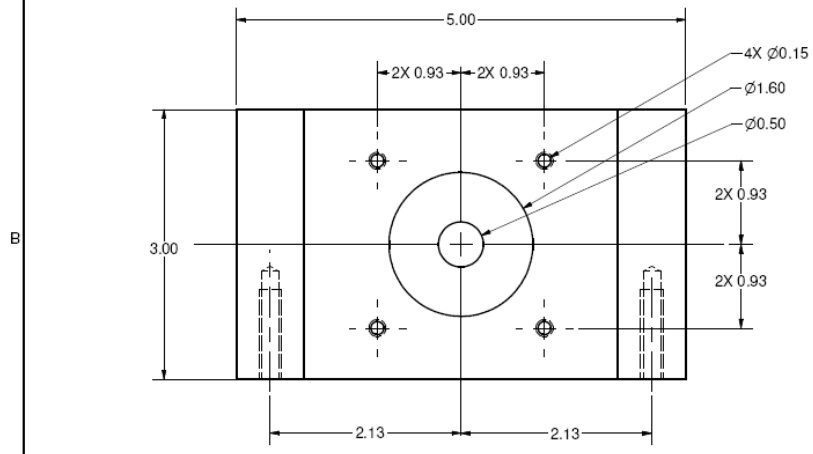
APPENDIX A

DETAILED ROBOT DRAWING PACKAGE

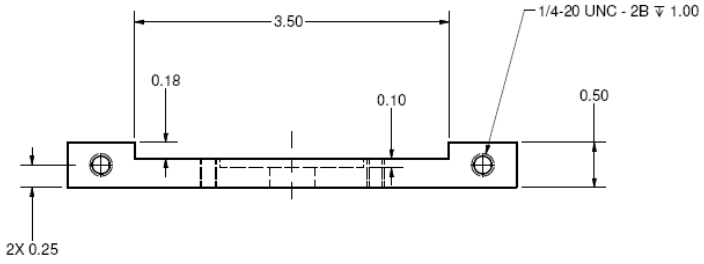



Ø0.75
2X 1/4-20 UNC - ↧ 0.75
5.2
1.0
LEG
SCALE 1 : 1
NOTES:
1. MATERIAL: CRES 304, 316, OR 321
2. BREAK ALL SHARP CORNERS AND EDGES.
DRAWN
D. DENTLER
5/12/2005
CHECKED
QA
MFG
APPROVED
TITLE
LEG
SIZE
B
DWG NO
2
REV
A
SCALE
SHEET 1 OF 1


5.00
0.25
0.500 X2
Ø13/32 THRU
Ø13/64 THRU X2
0.275 X2
0.275 X2
5.00
0.375 X2
Ø9/32 THRU X4
1.700 X2
WAIST
SCALE 1 : 1
NOTES:
1. MATERIAL: 6061 ALUMINUM
2. BREAK ALL SHARP CORNERS AND EDGES.
DRAWN
D. DENTLER
2/17/2005
CHECKED
QA
MFG
APPROVED
TITLE
WAIST
SIZE
B
DWG NO
4
REV
A
SCALE
SHEET 1 OF 1



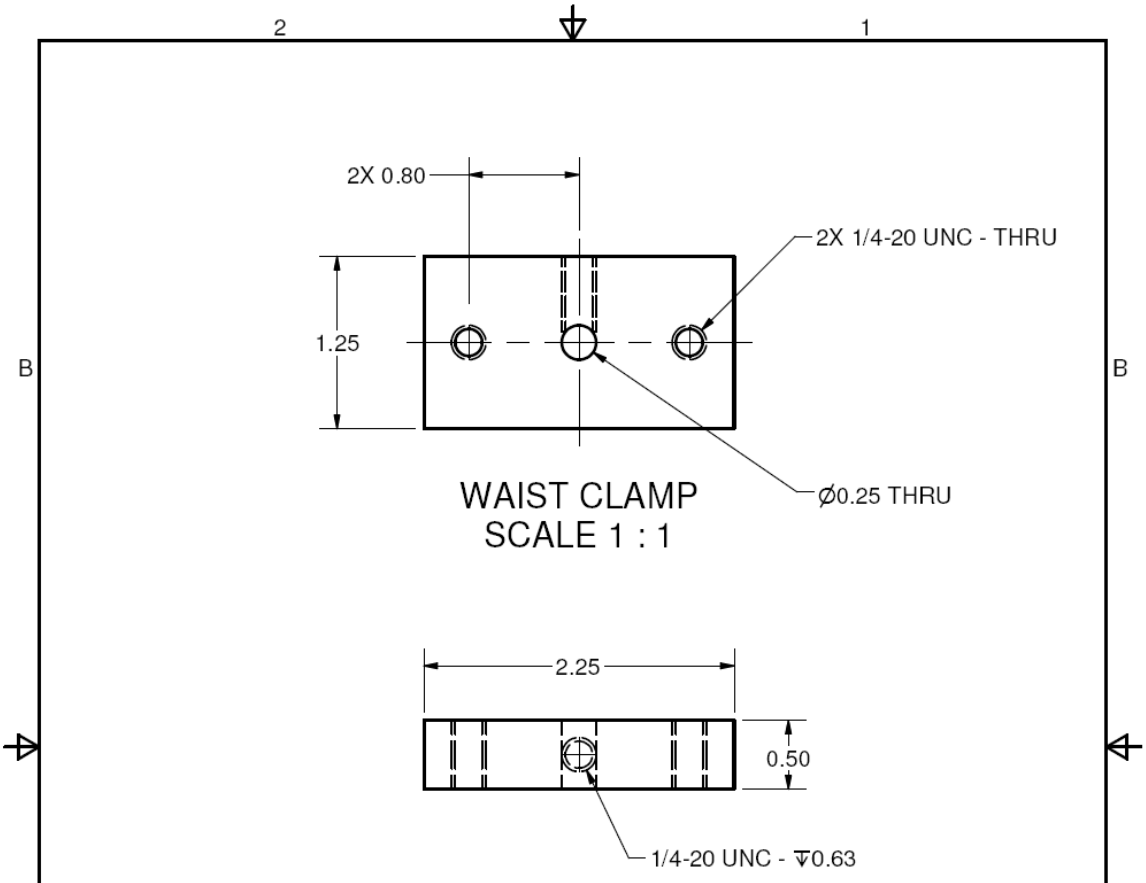
SHOULDER MOUNT
SCALE 1 : 1



NOTES:

1. MATERIAL: 6061 ALUMINUM
2. BREAK ALL SHARP CORNERS AND EDGES.

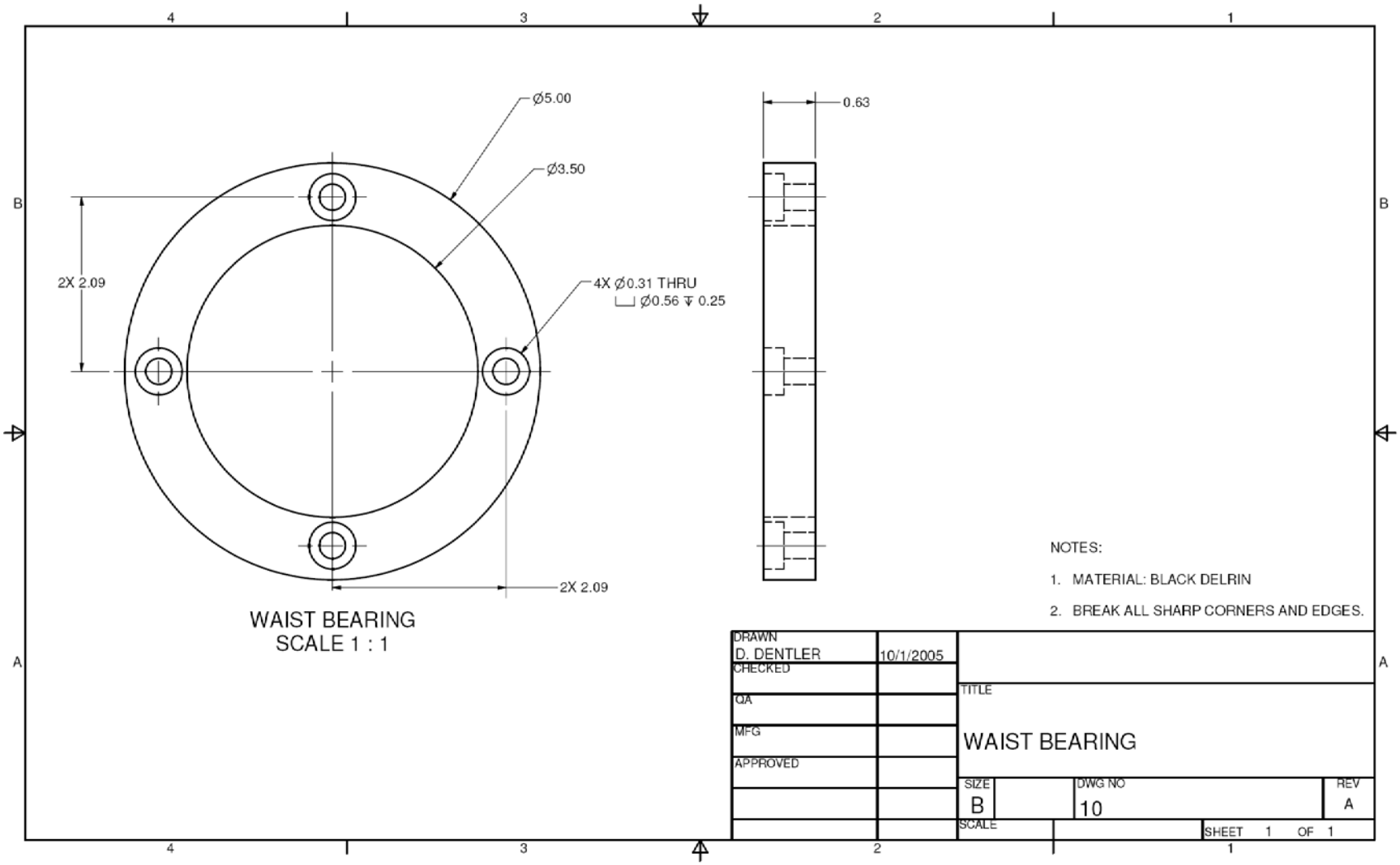
DRAWN D. DENTLER	4/26/2005	
CHECKED		TITLE: SHOULDER MOUNT
QA		
MFG		
APPROVED		SIZE B / DWG NO 5 / REV A
		SCALE / SHEET 1 OF 1

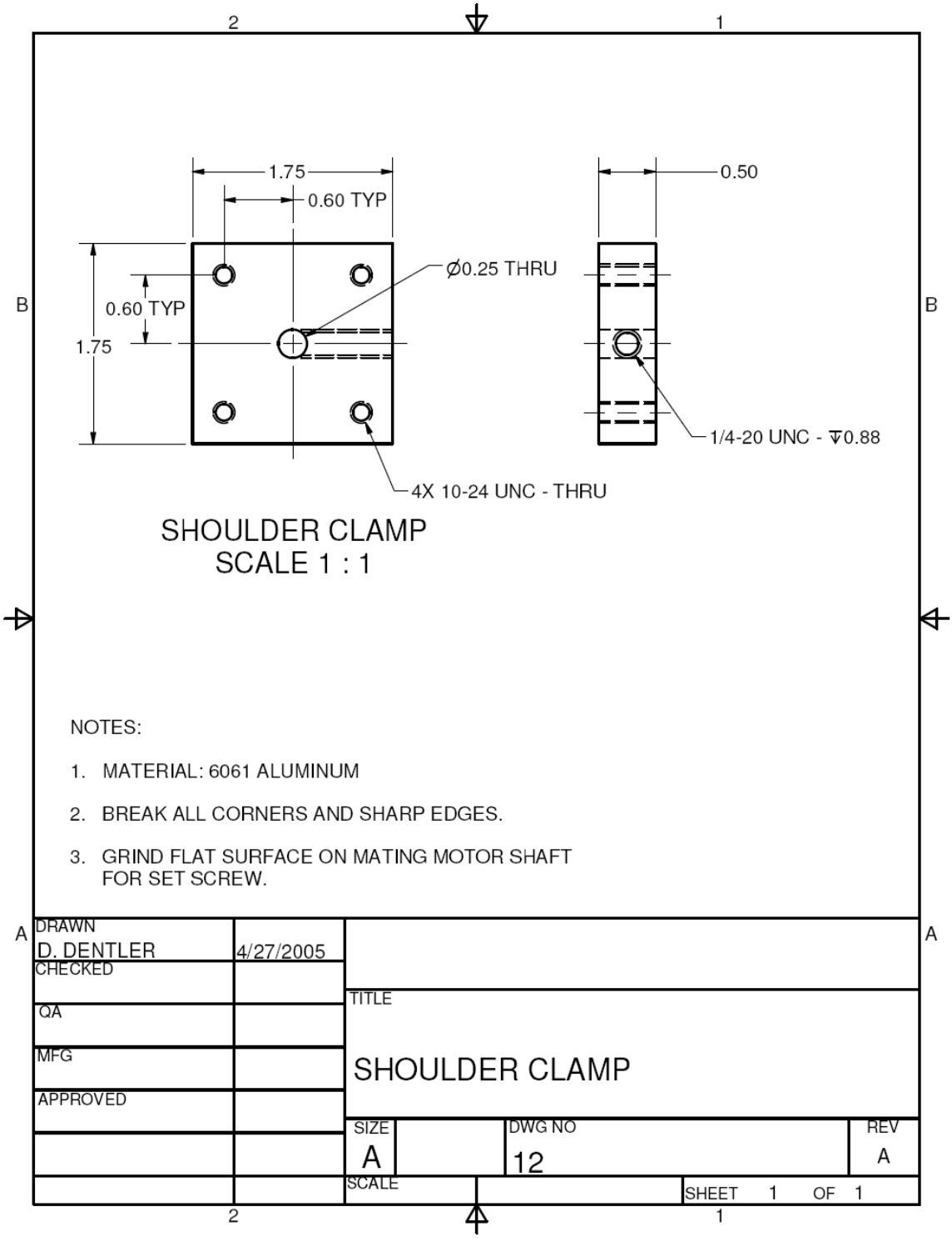


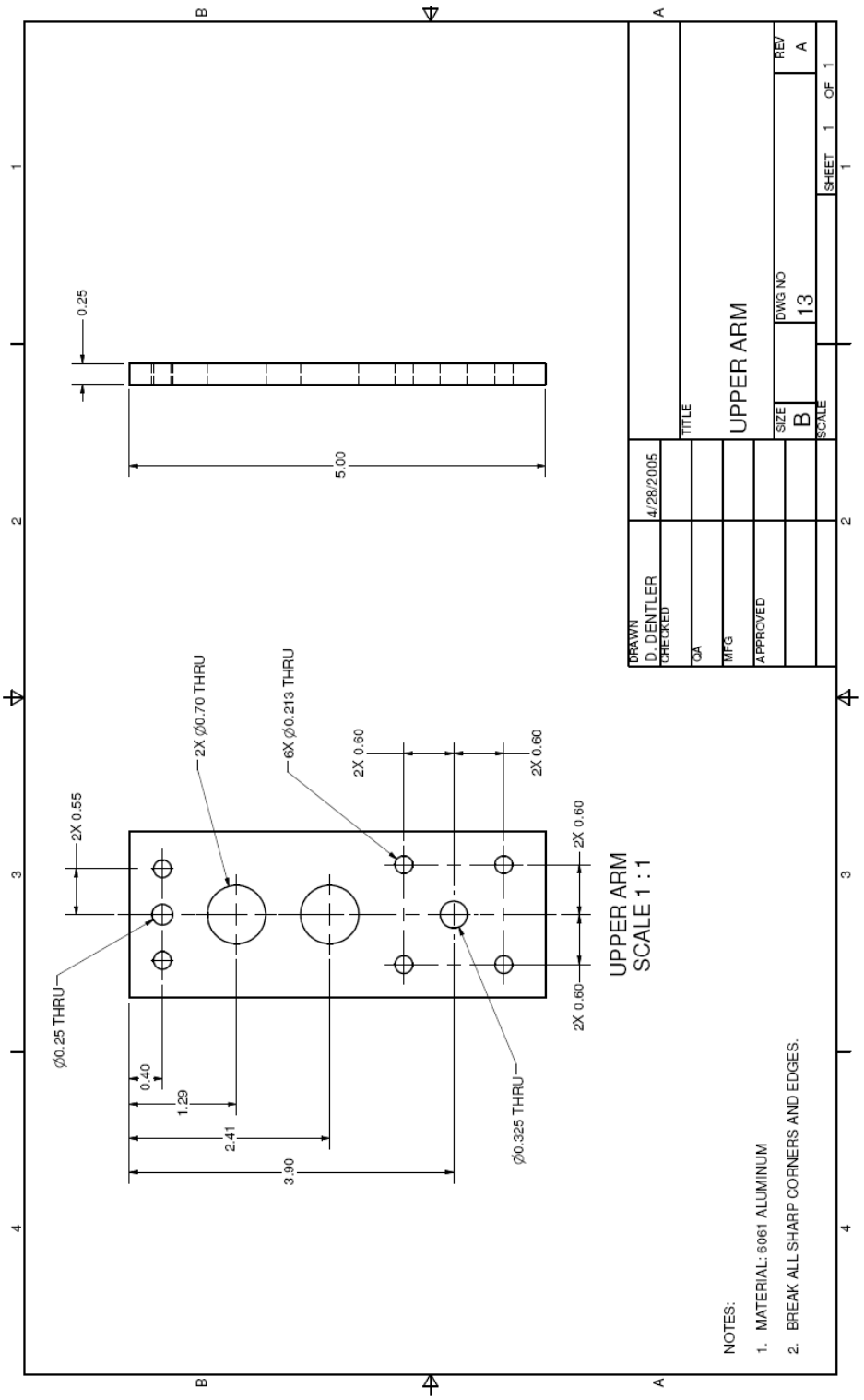
NOTES:

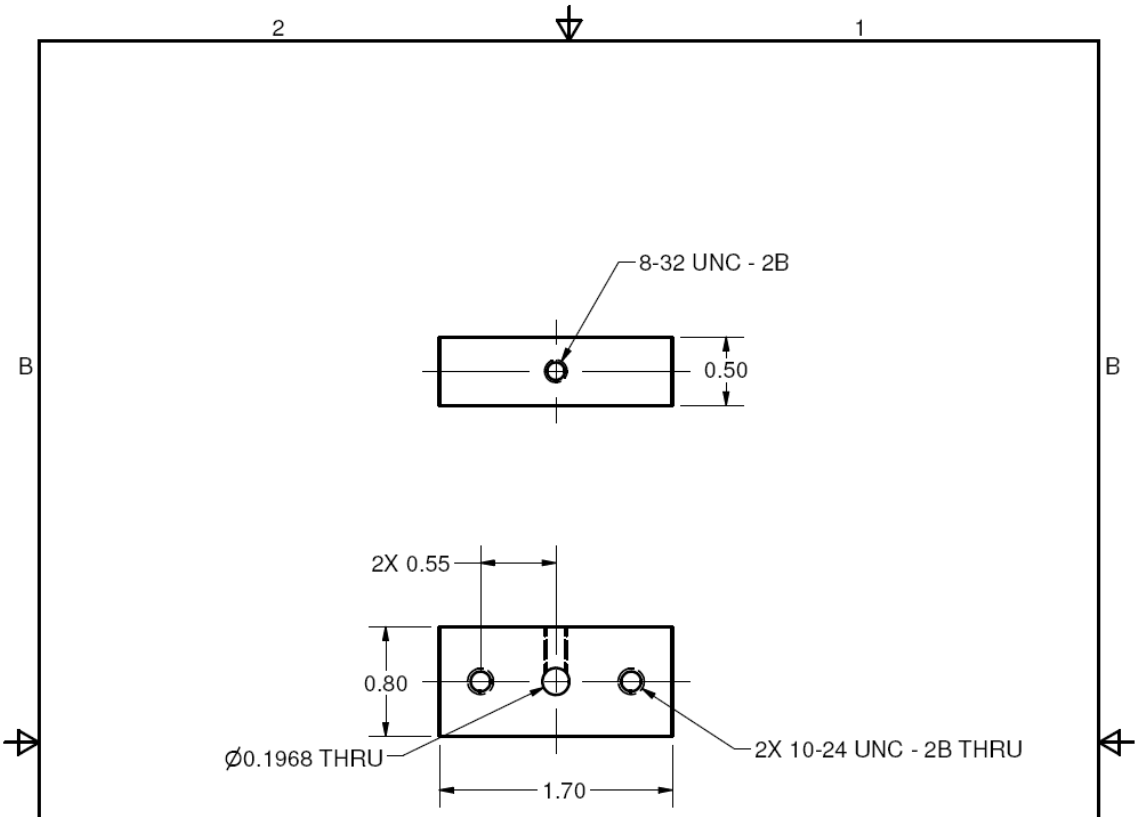
1. MATERIAL: 6061 ALUMINUM
2. BREAK ALL SHARP CORNERS AND EDGES.
3. GRIND FLAT SURFACE ON MATING MOTOR SHAFT FOR SET SCREW.

DRAWN D. Dentler	4/21/2005		
CHECKED		TITLE	
QA		WAIST CLAMP	
MFG			
APPROVED		SIZE A	DWG NO 8 / REV A
		SCALE	SHEET 1 OF 1


Ø5.00
Ø3.50
2X 2.09
4X Ø0.31 THRU
⌴ Ø0.56 ↧ 0.25
2X 2.09
0.63
WAIST BEARING
SCALE 1 : 1
NOTES:
1. MATERIAL: BLACK DELRIN
2. BREAK ALL SHARP CORNERS AND EDGES.
DRAWN
D. DENTLER
10/1/2005
CHECKED
QA
MFG
APPROVED
TITLE
WAIST BEARING
SIZE
B
DWG NO
10
REV
A
SCALE
SHEET 1 OF 1


2
1
B
A
1.75
0.60 TYP
0.60 TYP
1.75
Ø0.25 THRU
0.50
1/4-20 UNC - ↧0.88
4X 10-24 UNC - THRU
SHOULDER CLAMP
SCALE 1 : 1
NOTES:
1. MATERIAL: 6061 ALUMINUM
2. BREAK ALL CORNERS AND SHARP EDGES.
3. GRIND FLAT SURFACE ON MATING MOTOR SHAFT FOR SET SCREW.
DRAWN
D. DENTLER
4/27/2005
CHECKED
QA
MFG
APPROVED
TITLE
SHOULDER CLAMP
SIZE
A
DWG NO
12
REV
A
SCALE
SHEET 1 OF 1


0.25
5.00
UPPER ARM
SCALE 1 : 1
2X 0.55
2X Ø0.70 THRU
6X Ø0.213 THRU
2X 0.60
2X 0.60
2X 0.60
2X 0.60
Ø0.25 THRU
Ø0.325 THRU
0.40
1.29
2.41
3.90
NOTES:
1. MATERIAL: 6061 ALUMINUM
2. BREAK ALL SHARP CORNERS AND EDGES.
DRAWN
D. DENTLER
4/28/2005
CHECKED
QA
MFG
APPROVED
TITLE
UPPER ARM
SIZE
B
DWG NO
13
REV
A
SCALE
SHEET 1 OF 1

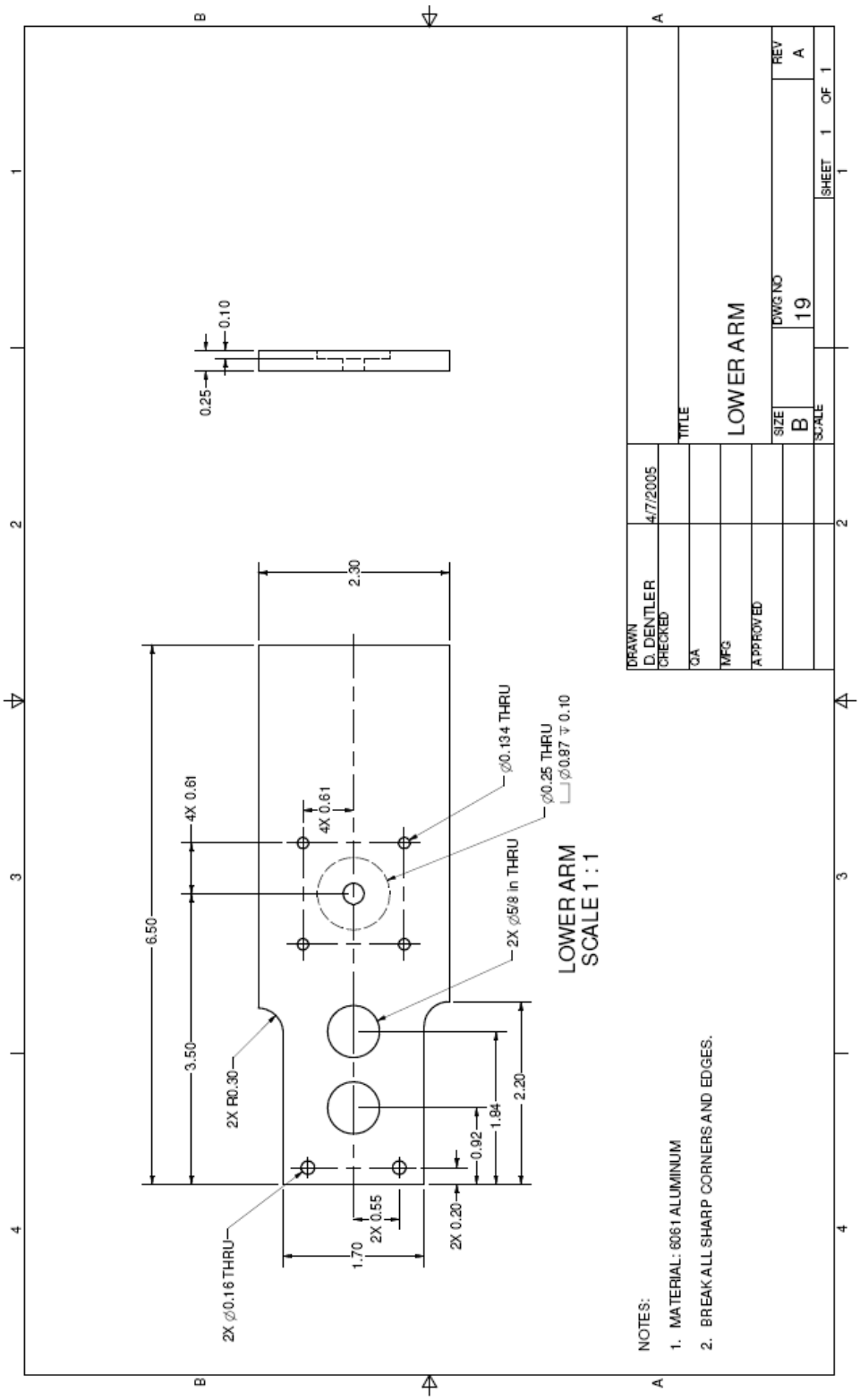


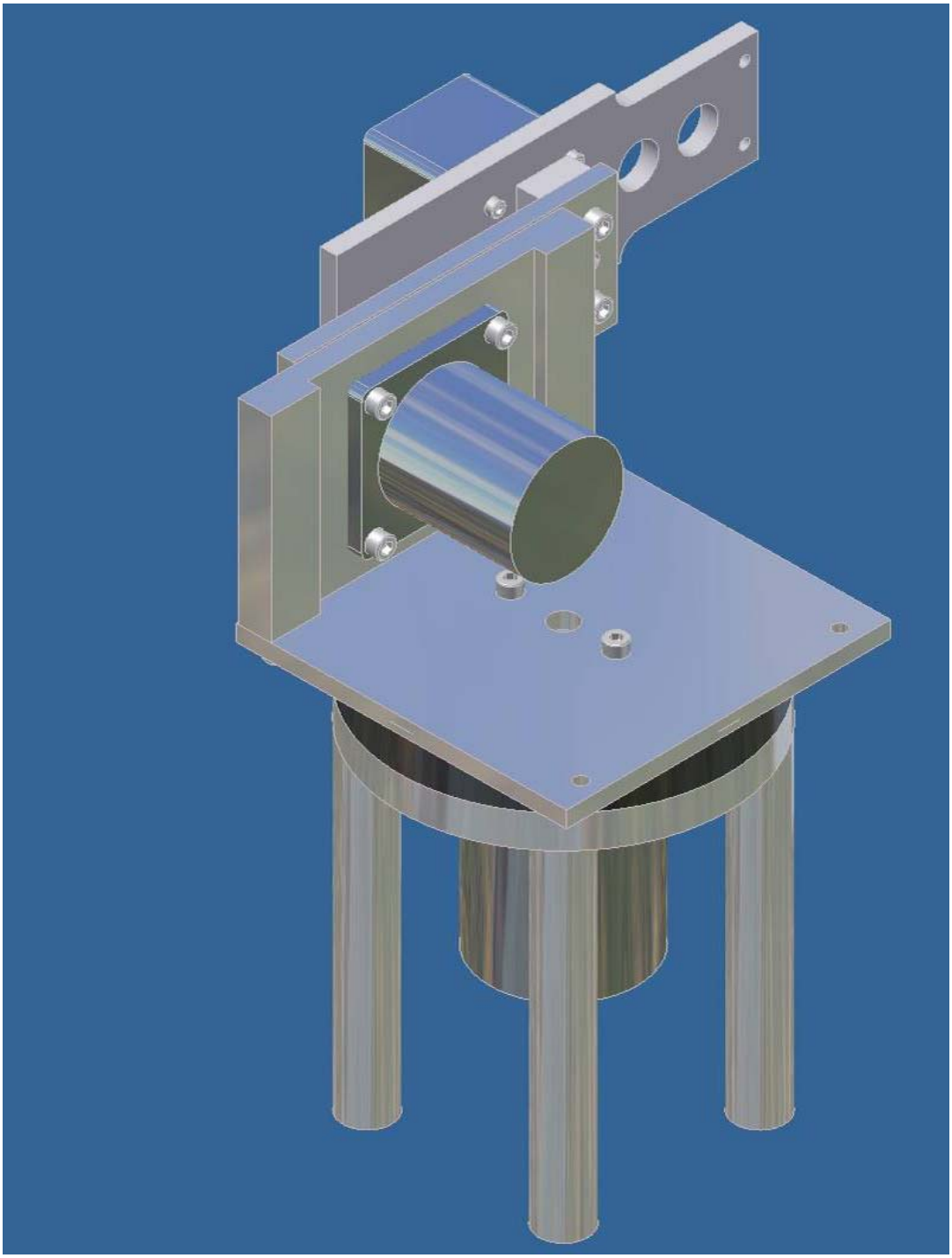
ELBOW CLAMP
SCALE 1 : 1

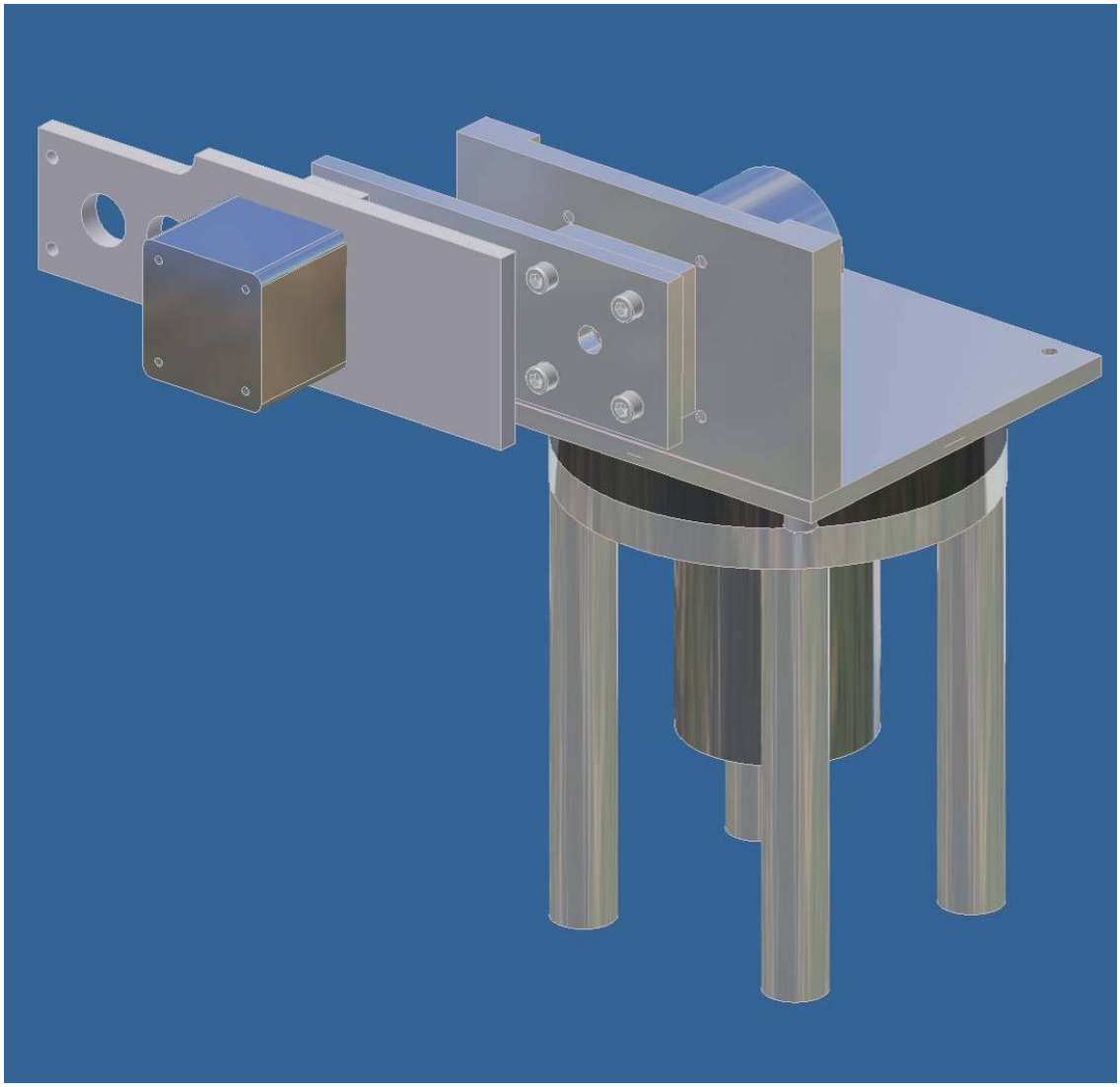
NOTES:

1. MATERIAL: 6061 ALUMINUM
2. BREAK ALL SHARP CORNERS AND EDGES.
3. GRIND FLAT SURFACE ON MATING MOTOR SHAFT FOR SET SCREW.

DRAWN D. Dentler	4/11/2005			
CHECKED		TITLE		
QA		ELBOW CLAMP		
MFG				
APPROVED		SIZE A	DWG NO 17	REV A
		SCALE	SHEET 1 OF 1	

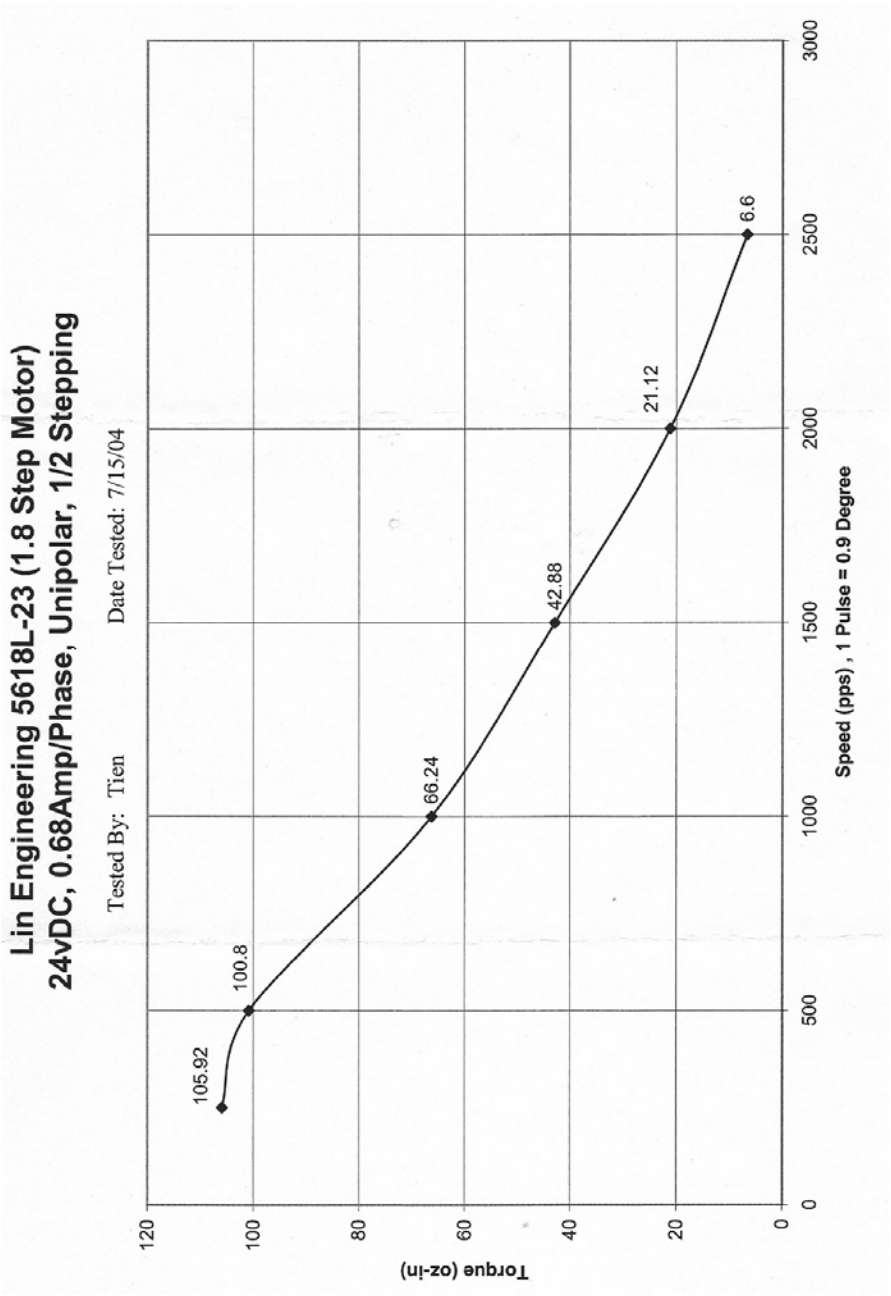

LOWER ARM
SCALE 1 : 1
NOTES:
1. MATERIAL: 6061 ALUMINUM
2. BREAK ALL SHARP CORNERS AND EDGES.
2X ⌀0.16 THRU
2X R0.30
2X ⌀5/8 in THRU
⌀0.134 THRU
⌀0.25 THRU
⌴ ⌀0.87 ∓ 0.10
4X 0.61
4X 0.61
6.50
3.50
2.30
2.20
1.84
0.92
2X 0.20
2X 0.55
1.70
0.25
0.10
DRAWN
D. DENTLER
4/7/2005
CHECKED
QA
MFG
APPROVED
TITLE
LOWER ARM
SIZE
B
DWG NO
19
REV
A
SCALE
SHEET 1 OF 1

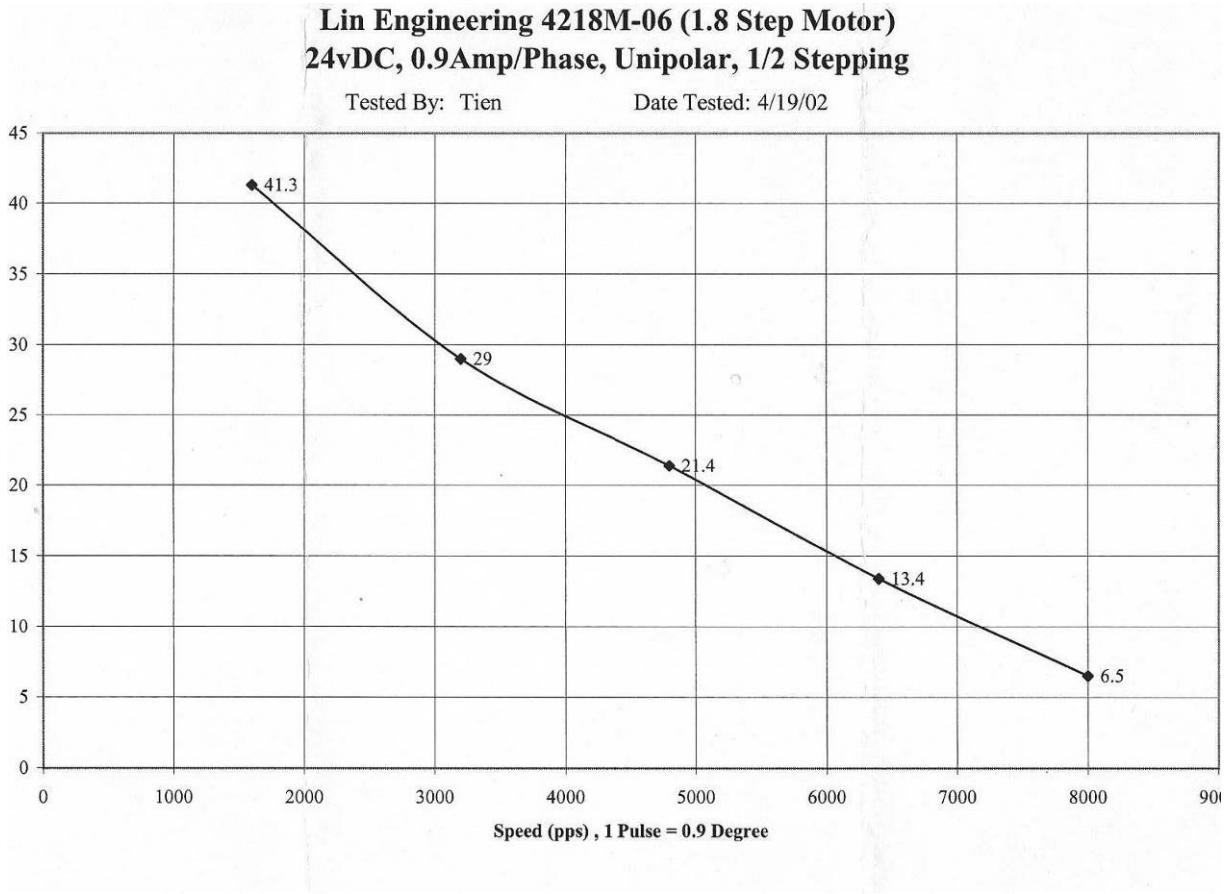


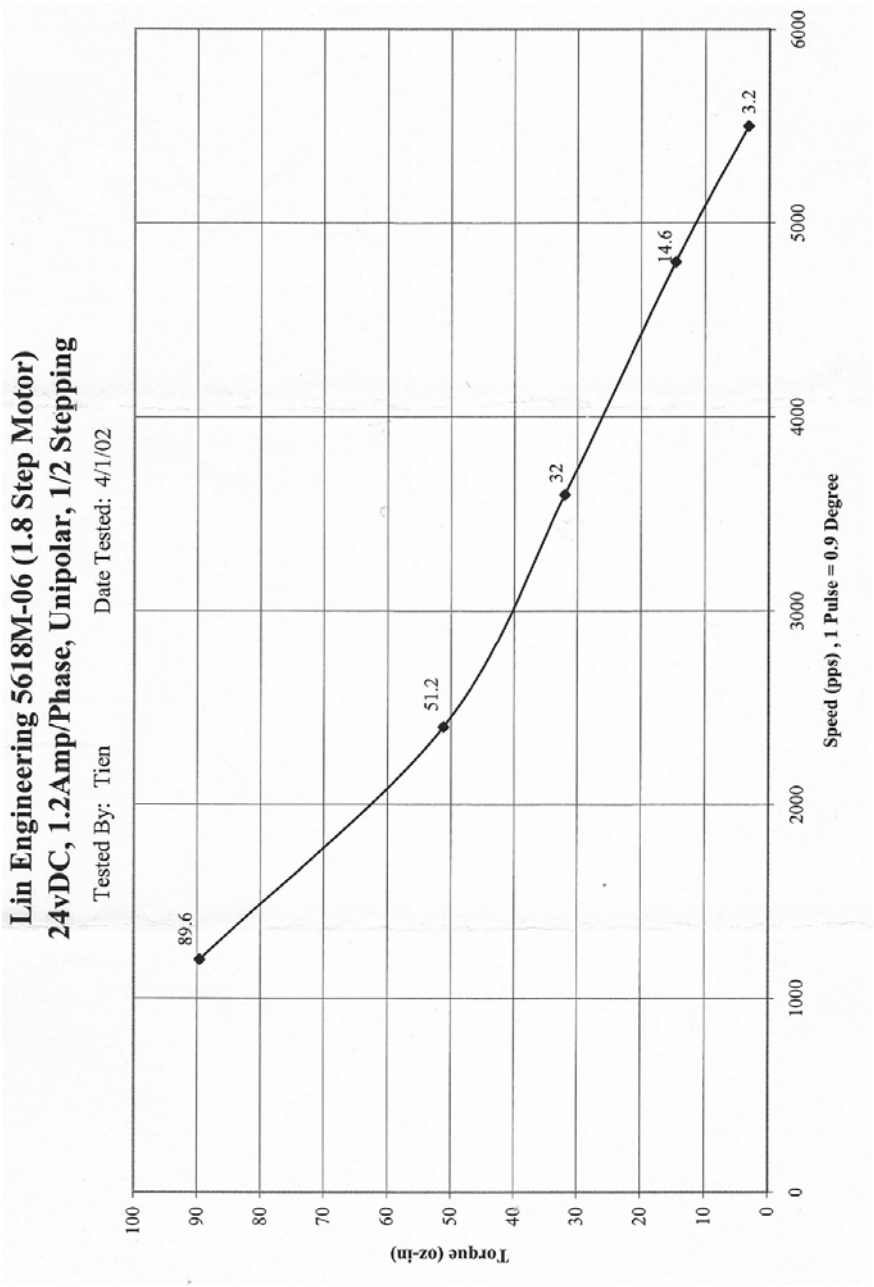


APPENDIX B

STEPPER MOTOR TORQUE CURVES




Lin Engineering 4218M-06 (1.8 Step Motor)
24vDC, 0.9Amp/Phase, Unipolar, 1/2 Stepping
Tested By: Tien
Date Tested: 4/19/02
45
40
35
30
25
20
15
10
5
0
41.3
29
21.4
13.4
6.5
0
1000
2000
3000
4000
5000
6000
7000
8000
Speed (pps) , 1 Pulse = 0.9 Degree


Lin Engineering 5618M-06 (1.8 Step Motor)
24vDC, 1.2Amp/Phase, Unipolar, 1/2 Stepping
Tested By: Tien
Date Tested: 4/1/02
Torque (oz-in)
Speed (pps) , 1 Pulse = 0.9 Degree
89.6
51.2
32
14.6
3.2
0
10
20
30
40
50
60
70
80
90
100
0
1000
2000
3000
4000
5000
6000

APPENDIX C

MAXIMUM DYNAMIC TORQUE FOR LINK 2

```
%PROGRAM CALCULATES THE MAXIMUM DYNAMIC TORQUE REQUIRED
TO ACTUATE LINK 2

clear;

%Initial Conditions

%Mass of link 2
m2 =0.217;
%Length of link 1
l1=3.5;
%Inertia of link 2
I2=1.158;
%Length between CG of link 2 and elbow joint
lc2=0;
%Velocity and acceleration
v=0.3142;
a=0.0628;

%Max Torque Calculation

%Theta 1 and 2 describe the angle of link 1 and 2, respectively
theta1=(-40:70)*pi/180;
theta2=(-80:90)*pi/180;

cosine1=cos(theta1(1));
cosine2=cos(theta2(1));
sine2=sin(theta2(1));
sine12=sin(theta1(1)+theta2(1));
cosine12=cos(theta1(1)+theta2(1));
sinecosine12=sine12*cosine12;

%Torque?
torque=(m2*l1*lc2*cosine2+2*(m2*lc2^2+I2))*a+...
(m2*(l1*lc2*cosine1*sine12+lc2^2*sinecosine12)+...
```

```
I2*sinecosine12+m2*l1*lc2*sine2)*(v^2)+m2*lc2*cosine12;

angle1=theta1(1);
angle2=theta2(1);
tt2=[];

%Find solution
for i=1:length(theta1)
        for j=1:length(theta2)
                sine12=sin(theta1(i)+theta2(j));
                cosine12=cos(theta1(i)+theta2(j));
                sinecosine12=sine12*cosine12;
                cosine1=cos(theta1(i));
                cosine2=cos(theta2(j));
                sine2=sin(theta2(j));
                torque2=(m2*l1*lc2*cosine2+2*(m2*lc2^2+I2))*a+...
                (m2*(l1*lc2*cosine1*sine12+lc2^2*sinecosine12)+...
                I2*sinecosine12+m2*l1*lc2*sine2)*(v^2)+m2*lc2*cosine12;
                tt2=[tt2 torque2];
                if abs(torque2)>abs(torque);
                        torque=torque2;
                        angle1=theta1(i);
                        angle2=theta2(j);
                end
        end
end

%Display max torque in lbs. and oz.
torque
torque*16

%Display and of link 1 and link 2 where the maximum torque occurs
angle1*180/pi
angle2*180/pi

%Plot the torque curve
k=1:(70+41)*(90+81);
plot(k,tt2)
xlabel('Time t')
ylabel('Torque T_2')
```

APPENDIX D

MAXIMUM DYNAMIC TORQUE FOR LINK 1

```
%PROGRAM CALCULATES THE MAXIMUM DYNAMIC TORQUE REQUIRED
TO ACTUATE LINK 1

clear;

%Initial Conditions

%Mass of link 1 and link 2
m1= 1.933;
m2= .217;
%Length of link 1
l1= 3.5;
%Inertia of link 1 and link 2
I1= 4.246;
I2= 1.158;
%Length between CG of link1 and shoulder joint
lc1= 1.24;
%Length between CG of link 2 and elbow joint
lc2= 0;
%Velocity and acceleration
v= .3142;
a= .0628;

%Max Torque Calculation

%Theta 1 and 2 describe the angle of link 1 and 2, respectively
theta1= (-40:70)*pi/180;
theta2= (-80:90)*pi/180;

cosine1=cos(theta1(1));
cosine2=cos(theta2(1));
sine1=sin(theta1(1));
sine2=sin(theta2(1));
sine12=sin(theta1(1)+theta2(1));
cosine12=cos(theta1(1)+theta2(1));
```

```
sinecosine12=sine12*cosine12;
sine1cosine1=sine1*cosine1;
sine1cosine12=sine1*cosine12;
cosine1sine12=cosine1*sine12;

%Torque?
torque=(m1*lc1^2+I1+m2*l1^2+2*m2*l1*lc2*cosine2+m2*lc2^2+I2)*a+...
(m2*l1*lc2*cosine2+m2*lc2^2+I2)*a+...
((m1*lc1^2+I1)*sine1cosine1+m2*(l1^2*sine1cosine1+l1*lc2*(sine1cosine12+cosine1si
ne12)+lc2^2*sinecosine12)+...
m1*lc1*cosine1+m2*(l1*cosine1+lc2*cosine12));

angle1=theta1(1);
angle2=theta2(1);
tt1=[];

%Find solution
for i=1:length(theta1)
        for j=1:length(theta2)
                sine12=sin(theta1(i)+theta2(j));
                cosine12=cos(theta1(i)+theta2(j));
                sinecosine12=sine12*cosine12;
                cosine1=cos(theta1(i));
                cosine2=cos(theta2(j));
                sine1=sin(theta1(1));
                sine2=sin(theta2(j));
                sine1cosine1=sine1*cosine1;
                sine1cosine12=sine1*cosine12;
                cosine1sine12=cosine1*sine12;

        torque1=(m1*lc1^2+I1+m2*l1^2+2*m2*l1*lc2*cosine2+m2*lc2^2+I2)*a+...
                (m2*l1*lc2*cosine2+m2*lc2^2+I2)*a+...

        ((m1*lc1^2+I1)*sine1cosine1+m2*(l1^2*sine1cosine1+l1*lc2*(sine1cosine12+c
osine1sine12)+lc2^2*sinecosine12)+...
                m1*lc1*cosine1+m2*(l1*cosine1+lc2*cosine12));
                tt1=[tt1 torque1];
                if abs(torque1)>abs(torque);
                        torque=torque1;
                        angle1=theta1(i);
                        angle2=theta2(j);
                end
        end
end
```

```
%Display max torque in lbs. and oz.
torque
torque*16

%Display and of link 1 and link 2 where the maximum torque occurs
angle1*180/pi
angle2*180/pi

%Plot the torque curve
k=1:(70+41)*(90+81);
plot(k,tt1)
xlabel('Time t')
ylabel('Torque T_1')
```

APPENDIX E

MAXIMUM DYNAMIC TORQUE FOR LINK 0

```
%PROGRAM CALCULATES THE MAXIMUM DYNAMIC TORQUE REQUIRED
TO ACTUATE LINK 0

clear;

%Initial Conditions

%Mass of link 1 and link 2
m1= 1.933;
m2= .217;
%Length of link 1
l1= 3.5;
%Inertia of link 0, link 1, and link 2
I0= 26.494;
I1= 4.246;
I2= 1.158;
%Length between CG of link1 and shoulder joint
lc1= 1.24;
%Length between CG of link 2 and elbow joint
lc2= 0;
%Velocity and acceleration
v= .3142;
a= .0628;

%Max Torque Calculation

%Theta 1 and 2 describe the angle of link 1 and 2, respectively
theta1= (-40:70)*pi/180;
theta2= (-80:90)*pi/180;

cosine1=cos(theta1(1));
%cosine2=cos(theta2(1));
sine1=sin(theta1(1));
%sine2=sin(theta2(1));
sine12=sin(theta1(1)+theta2(1));
```

```
cosine12=cos(theta1(1)+theta2(1));
sinecosine12=sine12*cosine12;
sine1cosine1=sine1*cosine1;
sine1cosine12=sine1*cosine12;
cosine1cosine12=cosine1*cosine12;
cosine1sine12=cosine1*sine12;

%Torque?
torque=(I0+(m1*lc1^2+I1)*cosine1^2+m2*l1*(l1*cosine1^2+2*lc2*cosine1cosine12)+(
m2*lc2^2+I2)*cosine12^2)...
*a-
2*((m1*lc1^2+I1)*sine1cosine1+m2*l1*(l1*sine1cosine1+lc2*sine1cosine12+lc2*cosin
e1sine12)+(m2*lc2^2+I2)*sinecosine12)...
*v^2-2*(m2*l1*lc2*cosine1sine12+(m2*lc2^2+I2)*sinecosine12)*v^2-
2*(m2*l1*lc2*cosine1sine12+(m2*lc2^2+I2)*sinecosine12)*v^2;

angle1=theta1(1);
angle2=theta2(1);
tt0=[];

%Find solution
for i=1:length(theta1)
        for j=1:length(theta2)
                sine12=sin(theta1(i)+theta2(j));
                cosine12=cos(theta1(i)+theta2(j));
                sinecosine12=sine12*cosine12;
                cosine1=cos(theta1(i));
                %cosine2=cos(theta2(j));
                sine1=sin(theta1(1));
                %sine2=sin(theta2(j));
                sine1cosine1=sine1*cosine1;
                sine1cosine12=sine1*cosine12;
                cosine1sine12=cosine1*sine12;

        torque0=(I0+(m1*lc1^2+I1)*cosine1^2+m2*l1*(l1*cosine1^2+2*lc2*cosine1cos
ine12)+(m2*lc2^2+I2)*cosine12^2)...
                *a-
2*((m1*lc1^2+I1)*sine1cosine1+m2*l1*(l1*sine1cosine1+lc2*sine1cosine12+lc2*cosin
e1sine12)+(m2*lc2^2+I2)*sinecosine12)...
                *v^2-2*(m2*l1*lc2*cosine1sine12+(m2*lc2^2+I2)*sinecosine12)*v^2-
2*(m2*l1*lc2*cosine1sine12+(m2*lc2^2+I2)*sinecosine12)*v^2;
                tt0=[tt0 torque0];
                if abs(torque0)>abs(torque);
                        torque=torque0;
                        angle1=theta1(i);
```

```
                    angle2=theta2(j);
                end
            end
end

%Display max torque in lbs. and oz.
torque
torque*16

%Display and of link 1 and link 2 where the maximum torque occurs
angle1*180/pi
angle2*180/pi

%Plot the torque curve
k=1:(70+41)*(90+81);
plot(k,tt0)
xlabel('Time t')
ylabel('Torque T_0')
```

APPENDIX F

PLOT OF INVERSE SOLUTION