Solving the Inverse Kinematic Robotics Problem: A Comparison Study of the Denavit-Hartenberg Matrix and Groebner Basis Theory

Except where reference is made to the work of others, the work described in this dissertation is my own or was done in collaboration with my advisory committee. This dissertation does not include proprietary or classified information.

Kimberly Kendricks

Certificate of Approval:

Michel Smith Professor and Chairman Mathematics and Statistics	Overtoun Jenda, Chair Professor Mathematics and Statistics
Richard Zalik Professor Mathematics and Statistics	Joe F. Pittman Interim Dean Graduate School

Solving the Inverse Kinematic Robotics Problem: A Comparison Study of the Denavit-Hartenberg Matrix and Groebner Basis Theory

Kimberly Kendricks

A Dissertation

Submitted to

the Graduate Faculty of

Auburn University

in Partial Fulfillment of the

Requirements for the

Degree of

Doctor of Philosophy

Auburn, Alabama
August 4, 2007

Solving the Inverse Kinematic Robotics Problem: A Comparison Study of the Denavit-Hartenberg Matrix and Groebner Basis Theory

Kimberly Kendricks

____

Signature of Author

____

Date of Graduation

Vita

Kimberly D. Kendricks was born on August 27, 1980 in Dayton, Ohio. After completing high school, she attended the University of Pittsburgh in Pittsburgh, Pennsylvania, where she received a Dual B.S. in Mathematics and Business. Then, she continued her passion for mathematics at Auburn University in Auburn, Alabama where she received her Master's in Applied Mathematics, and since then, has served as a graduate teaching assistant in the Department of Mathematics.

Dissertation Abstract

Solving the Inverse Kinematic Robotics Problem: A Comparison Study of the Denavit-Hartenberg Matrix and Groebner Basis Theory

Kimberly Kendricks

Doctor of Philosophy, August 4, 2007
(M.A., Auburn University, 2006)
(Dual B.S., University of Pittsburgh, 2003/2004)

90 Typed Pages

Directed by Overtoun Jenda

The aim is to analyze two very different methods for solving the inverse kinematic robotics problem. The first method, called the Engineering Approach, uses the Denavit-Hartenberg Matrix, and the second method, called the Mathematician's Approach, uses Groebner Basis Theory and MAGMA, a mathematics computer software package. With each approach, we will solve the inverse kinematics robotics problem for various robot manipulators. By comparing each method, this paper will demonstrate that Groebner Basis Theory is more advantageous, and furthermore, more beneficial to the field of mathematics and robotics engineering.

Style manual or journal used Journal of Approximation Theory (together with the style known as "auphd"). Bibliography follows van Leunen's *A Handbook for Scholars*.

Computer software used The document preparation package TEX (specifically LATEX) together with the departmental style-file `auphd.sty`.

Table of Contents

List of Figures

Chapter 1

Introduction

Robots have become an integral part of everyday life. Automobiles, household products, and canned or boxed foods, are all prepared by assembly line robots. In this paper, we highlight the importance of studying robotic motion to perform these tasks. In particular, we are concerned with finding all of the possible movements needed in order to move the robot hand to a desired location where it may pick up, pull, or push an object. By determining a robot's capabilities, we can better classify the robot for specific tasks improving efficiency and productivity. We call the problem of finding all possible movements the inverse kinematics robotics problem.

The inverse kinematic robotics problem has proved to be of great significance because the solutions found provide control over the position and orientation of the robot hand. Studies have shown that the inverse kinematic robotics problem can be solved using matrix algebra, iterative procedures, or geometric applications. Paul [18] used a matrix algebra technique called the Denavit-Hartenberg Matrix, preferred by engineers because of its simplicity and repetition, to obtain an inverse kinematic solution for a robot manipulator. Huang [7], Manseur and Doty [15] have developed iterative procedures to obtain a solution. These iterative methods require rigorous calculations that do not necessarily converge to a correct solution. Lastly, Lee [14], Yih and Youm [23] applied a geometric approach to solving the inverse kinematic robotics problem. The works of these authors are profound, but it is unclear which method to use. Is one method more efficient in calculating real solutions in real-time than another? This is debatable, but what is overwhelmingly evident is the popularity of the Denavit-Hartenberg Matrix.

Derived by authors J. Denavit and R. S. Hartenberg, the Denavit-Hartenberg Matrix (The Engineering Approach) is the most widely used technique for solving the inverse kinematic robotics problem for several reasons. One, specific intervals can be formed describing the angle of rotation or translation of each joint of a robot manipulator. These intervals define the potential movement of each joint. Two, with these intervals of movement, certain parameters can be used in a matrix to describe the movement of the $i-1^{th}$ joint in terms of the i^{th} joint. Therefore, each joint can be described by a matrix. Three, the product of all of these joint matrices describes the final position of the robot hand in space. In summary, the Denavit-Hartenberg Matrix is easy to use and easy to understand. However, when using this technique, we encounter the following difficulties: the need for additional methods and algorithms to solve the problem, matrix computations may be too large or too complex, some solutions cannot be found in real-time, and some solutions are not easily found.

Given these difficulties, and those associated with other engineering approaches, we look for an alternate method that can resolve them. Hence, Groebner Basis Theory (The Mathematician's Approach). Developed by Bruno Buchberger, an algebraic algorithm can be applied to a given set of non-zero polynomials, producing a basis set, such that every polynomial in the original set is a linear combination of those polynomials in the basis set. By finding solutions to this basis set, we thereby find solutions to the entire set. With the aid of MAGMA, an algebraic computer software package, we can calculate these solutions in real-time, and consider cases not as easily solved using the Denavit-Hartenberg Matrix, thereby finding all possible solutions.

To familiarize the reader with each approach, we begin by solving the inverse kinematic robotics problem for the GMF Robotics A-510 Robot. By demonstrating each approach on a robot with a simple construction, we are able to reveal where potential problems in the Engineering Approach can arise. We then extend our knowledge and apply each approach to a different class of robot manipulators. We will analyze each method and compare their corresponding solutions. The aim is to present to engineers and mathematicians an application of Groebner Basis Theory that can solve the inverse kinematics robotics problem while resolving the difficulties engineers face using the Denavit-Hartenberg Matrix.

In the past, solving the inverse kinematic robotics problem for various robot manipulators has advanced the field of robotics engineering, but this comparative study will contribute to the advancement of the field of mathematics, as well, and more importantly, continue to satisfy the needs of a growing industry dependent upon robots.

We begin by introducing the GMF Robotics A-510 Robot in Chapter 2, where we discuss the robot's specifications, definitions, and important properties. Next is Chapter 3 which focuses on kinematics. Here, the inverse kinematic robotics problem is explained in more detail. Chapter 4 is devoted to the Engineering Approach of the Denavit-Hartenberg Matrix. A derivation of the Denavit-Hartenberg Matrix is given as well as a solution to the inverse kinematics robotics problem. Chapter 5 is devoted to the Mathematicians' Approach of Groebner Basis Theory. The algorithm for finding a Groebner basis is provided as well as a solution to the inverse kinematic robotics problem. At the end of chapters four and five, there is a small summary of the solutions found. It is in Chapter 6 where we extend our knowledge and apply each approach to an entire class of robot manipulators. A brief conclusion follows summarizing the findings in this paper.

Chapter 2

The GMF Robotics A-510 Robot Manipulator

2.1 Introduction

The GMF Robotics A-510 Robot is manufactured by GMF FANUC Robotics and is typically used to perform assembly line tasks. The GMF Robotics A-510 Robot that is used in this study presently sits in the Electrical Engineering Robotics Laboratory at Auburn University in Auburn, Alabama. This robot came from the General Motors Saturn Automotive Plant in Tennessee. Various tools such as grippers, welders, and spray guns can be attached to the robot to perform specific tasks. We explore other attributes of the robot below.

2.2 Definitions

Definition 1: A *robot manipulator* consists of links which are connected to one another by joints.

Definition 2: A *link* is a rigid body in a robot manipulator. A *link* is also called a segment.

Definition 3: There are two types of joints. A *revolute joint* which is a rotary joint that allows the rotation of one link about the joint of the preceding link, and a *prismatic joint* which is a linear joint that allows translation between joints.

Definition 4: A *serial link manipulator* is a robot manipulator in which links and joints are arranged in an alternating fashion.

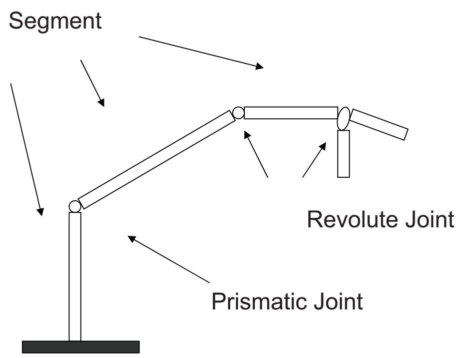

Figure 2.1: Robot Manipulator identifying segments and joints.

Definition 5: An *end effector* is the last joint in a *serial link manipulator*. The *end effector* is typically called the robot hand since various tools can be attached to the *end effector* to have the robot perform different tasks.

Definition 6: A *degree of freedom* represents a joint-link pair.

2.3 Properties

The GMF Robotics A-510 robot has four degrees of freedom. The first joint is a prismatic joint and the second, third, and fourth are revolute joints. The motion of each joint axis is described in the table below.

- The Z axis has a vertical motion of 300mm.(Prismatic Joint)
- The Theta axis has a horizontal rotation of 300 degrees. (Revolute Joint)
- The U axis has a horizontal rotation of 300 degrees. (Revolute Joint)
- The α axis (end effector) has a rotation of 540 degrees. (We are only concerned with where this joint will be placed in space, so we will not incorporate this axis in future calculations.)

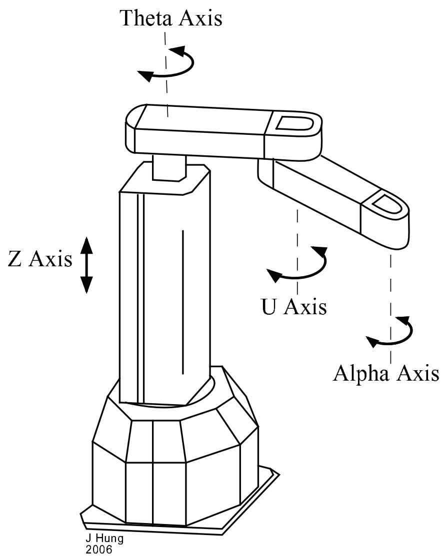

Figure 2.2: GMF A-510 Robot, ref. Hung [8].

ITEM		SPECIFICATIONS	REMARKS
Coordinate System		Horizontal Articulated Type	
Controlled Axes		Theta. Z. U. Alpha.	
Motion Range (Maximum Speed)	Theta Axis (rotation)	–150° to +150° (300°/second)	
	Z Axis (up/down)	300 mm (700 mm/second)	
	U Axis (rotation)	–150° to +150° 300°/second)	
	Alpha Axis (faceplate rotation)	540° (540°/second)	

Figure 2.3: Axis Motion for GMF A-510 Robot, ref. Hung [8].

Chapter 3

The Inverse Kinematic Robotics Problem

3.1 Definitions

Definition 1: *Kinematics* is the study of motion, ignoring such factors as mass, gravity, and torques.

Definition 2: A *Kinematic Chain* is a robot manipulator in which each joint has at most two links. Moreover, a *Kinematic Chain* is a serial link manipulator whose first link is connected to the base and whose final joint is connected to just one link, leaving the remaining end open. Thus, in a *Kinematic Chain* no closed loops can be formed.

There are two types of kinematic problems: The Forward Kinematics Robotics Problem and The Inverse Kinematics Robotics Problem.

3.2 Forward Kinematic Robotics Problem

The Forward Kinematic Robotics Problem: *Given a robot manipulator with specific joint angles, we can find the exact position and orientation of the robot hand (end effector).*

Because of its simplicity, the Forward Kinematic Robotics Problem gains little interest.

3.3 Inverse Kinematic Robotics Problem

The Inverse Kinematic Robotics Problem: *Given an orientation and position for a robotic arm, we want to show that by finding all possible combinations of joint settings, we can place the hand of the robot at this exact point and orientation.*

The inverse kinematic robotics problem has been the focus of kinematic analysis for robot manipulators. In order to determine all possible formations to place the end effector of a robot manipulator at a particular point in space, we must compute the movements associated with each joint variable. In doing so, over the span of several decades, authors have faced the following difficulties:

- The complexity of the inverse kinematic robotics problem is determined by the geometry of the robot manipulator.

- Some calculations to solving the inverse kinematic problem cannot be computed in real-time.

- There can be difficulty in finding all possible solutions.

- There can be difficulty in finding real solutions.

The GMF Robotics A510 Robot is formed by a set of interconnected rigid bodies called links and these rigid bodies are connected by prismatic or revolute joints. We are interested in how these bodies move in relation to one another in order to place the robot hand, or end effector, at a desired point in space. Hence, our focus on the inverse kinematic robotics problem.

Chapter 4

The Engineering Approach

4.1 Introduction

The Engineering Approach uses the Denavit-Hartenberg Matrix to solve the inverse kinematic robotics problem. With some examples, we derive the Denavit-Hartenberg Matrix, then use the Denavit-Hartenberg Matrix to find a solution to the inverse kinematic robotics problem for the GMF Robotics A-510 Robot.

4.2 Homogeneous Coordinates and Matrices

4.2.1 Definitions

Definition 1: *Homogeneous Coordinates* are an alternate way of representing a three dimensional vector, but have the addition of a fourth component called a scaling factor, and all of the components are multiplied by this scaling factor.

Example: In matrix form, the homogeneous representation of a vector $\vec{v}$ is

$$\begin{bmatrix} x' \\ y' \\ z' \\ s \end{bmatrix}$$

where s is the scaling factor and $x' = sx$, $y' = sy$ and $z' = sz$.

4.2.2 Remarks

1. Homogeneous representations are not unique.

2. The null vector is represented as

$$\begin{bmatrix} 0 \\ 0 \\ 0 \\ s \end{bmatrix}$$

   where s is a non-zero real number.

3. A vector is undefined if all the entries are zero.

4. Vector operations such as the Dot Product and Cross Product have homogeneous representations.

4.3 Transformations

Transformations are a class of matrix operators that can perform vector operations resulting in the translation or rotation of a vector. There are two types of transformations: translational and rotational.

4.3.1 Translational Transformations

A *Translational Transformation* denoted Trans(a,b,c) moves a point defined by a vector $\vec{x}$, to a new point defined by a vector $\vec{y}$. The location of $\vec{y}$ is given by the vector addition of $\vec{x}$ with a translation vector defined by (a,b,c), where a,b,c represent the components of the vector that are added to the components of the vector $\vec{x}$. In other words, we are moving a point described by a vector $\vec{x}$ along the diagonal (a,b,c) to a new point described by the vector $\vec{y}$. Trans(a,b,c) is defined as

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

The vector components a,b,c are added to the vector $\vec{x}$ to get the point described by the vector $\vec{y}$.

Example: Let $P = (2, 1)$ be a point in the xy-plane. We wish to move P along the diagonal of $30°$ for a distance of 8 units to a new point P'. What are the coordinates of the final point $P' = (x_2, y_2)$?

Solution: The homogeneous representation of P is,

$$\vec{P} = \begin{bmatrix} 2 \\ 1 \\ 0 \\ 1 \end{bmatrix}$$

Since we are moving $\vec{P}$ along a 30° diagonal, the unit vector corresponding to this direction is

$$\vec{u} = \begin{bmatrix} \frac{\sqrt{3}}{2} \\ \frac{1}{2} \\ 0 \\ 1 \end{bmatrix}$$

The vector of length 8 along 30° is

$$\vec{u'} = \begin{bmatrix} 4\sqrt{3} \\ 4 \\ 0 \\ 1 \end{bmatrix}$$

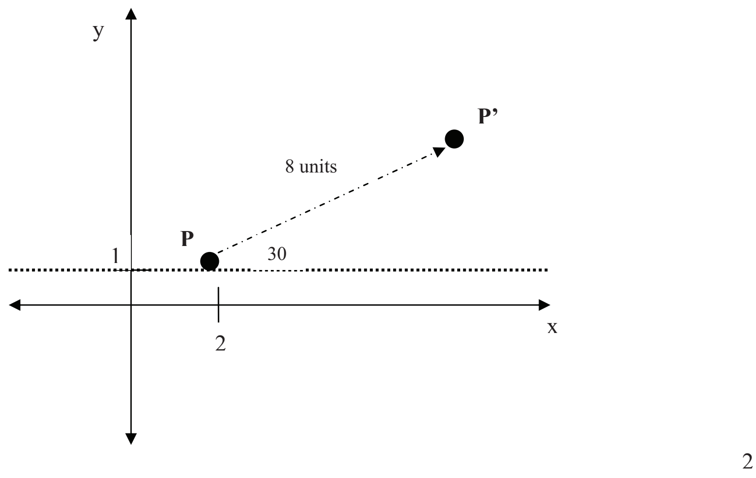

Figure 4.1: Translational Transformation Example

In homogeneous representation, then,

$$\vec{P'} = \begin{bmatrix} 1 & 0 & 0 & 4\sqrt{3} \\ 0 & 1 & 0 & 4 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 2 \\ 1 \\ 0 \\ 1 \end{bmatrix} = \begin{bmatrix} 2+4\sqrt{3} \\ 5 \\ 0 \\ 1 \end{bmatrix}$$

So,

$$P' = (2 + 4\sqrt{3}, 5)$$

Note: In the computation of $\vec{P'}$ the scaling factor is 1.

In summary, a translational transformation describes motion along a line.

4.3.2 Rotational Transformations

A *Rotational Transformation*, denoted Rot(axis,θ), moves a point defined by a vector to a new position in space by rotating the point by θ about an axis.

Using homogeneous representation, we have the following rotational transformations:

For rotation about the x-axis,

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

For rotation about the y-axis,

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

For rotation about the z-axis,

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

The angle of rotation can be positive or negative and is defined by using the right-hand rule. Rotation is positive if the cross product of two axes is in the positive direction of the third axis. That is, the cross product of the initial and final vectors is in the same direction as the axis about which the rotation is to be performed. For example, the cross product of $\vec{x} \times \vec{y} = \vec{z}$, so positive rotation will occur about the z-axis.

Example:

Let

$$\vec{P} = \begin{bmatrix} 2 \\ 2 \\ 2 \\ 1 \end{bmatrix}$$

We wish to rotate $\vec{P}$, 45° about the z-axis to the point described by the vector $\vec{P'}$. What would the homogeneous coordinate for $\vec{P'}$ be?

Solution:

First, we wish to find

$$Rot(z,45) = \begin{bmatrix} \cos 45 & -\sin 45 & 0 & 0 \\ \sin 45 & \cos 45 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Then,

$$\vec{P'} = Rot(z,45) * \vec{P} = \begin{bmatrix} \frac{\sqrt{2}}{2} & \frac{-\sqrt{2}}{2} & 0 & 0 \\ \frac{\sqrt{2}}{2} & \frac{\sqrt{2}}{2} & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 2 \\ 2 \\ 2 \\ 1 \end{bmatrix} = \begin{bmatrix} 0 \\ 2\sqrt{2} \\ 2 \\ 1 \end{bmatrix}$$

Thus,

$$\vec{P'} = (0, 2\sqrt{2}, 2, 1)$$

.

4.4 The Denavit-Hartenberg Matrix

4.4.1 Introduction

To understand the Denavit-Hartenberg Matrix, we must first understand how it is derived. The Denavit-Hartenberg Matrix is a special form of a homogeneous transformation

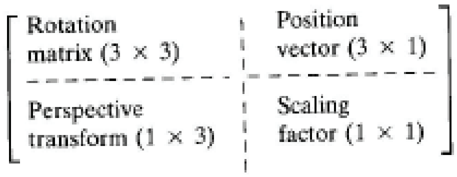

Figure 4.2: Homogeneous Transformation Matrix I, ref. Klafter [10].

matrix, a 4×4 matrix having the property of transforming a vector from one coordinate frame to another by means of a translation or rotation. For a kinematic chain with n-joints and $n-1$-links, each joint is assigned a frame of reference. We can align each frame of reference by performing a series of rotations and transformations. Thus, each joint can be represented by a homogeneous transformation matrix describing the particular rotation or translation needed to align the $i-1^{th}$ joint with i^{th} joint. The product of these matrices gives the final position of the n^{th} joint. In this case, the n^{th} joint represents the robot hand (end effector).

4.4.2 Derivation of the Denavit-Hartenberg Matrix

A homogeneous transformation matrix can be described in two different ways. One way is

For the matrix above,

1. The upper left 3×3 matrix is the *Rotation Matrix*
2. The 3×1 vector describes the *Position vector (or Translation)*
3. The 1×3 vector is called the *Perspective Transform*, and is $\{0, 0, 0\}$.

4. The 1×1 vector is the *Scaling Factor*, and will always be represented as $s = 1$.

On the other hand, we can also describe the homogeneous transformation matrix as:

$$\begin{bmatrix} n_x & o_x & a_x & p_x \\ n_y & o_y & a_y & p_y \\ n_z & o_z & a_z & p_z \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Figure 4.2 Homogeneous Transformation Matrix II.

The matrix above, also known as the *T-Variable Matrix*, describes the direction vectors for the x, y, and z axes of frame 2 in terms of the direction vectors for the x, y, and z axes of frame 1, where the normal vector $\vec{n}$ represents the x-axis, the orientation vector $\vec{o}$ represents the y-axis, and the approach vector $\vec{a}$ represents the z-axis.

1. For the first column, n_x, n_y, n_z are the components of the unit vector defining the x-axis of frame 2 in terms of the three unit vectors of the axes from frame 1.
2. For the second column, o_x, o_y, o_z are the components of the unit vector defining the y-axis of frame 2 in terms of the three unit vectors of the axes from frame 1.
3. For the third column, a_x, a_y, a_z are the components of the unit vector defining the z-axis of frame 2 in terms of the three unit vectors of the axes from frame 1.
4. The fourth column is position of frame 2 with respect to frame 1.

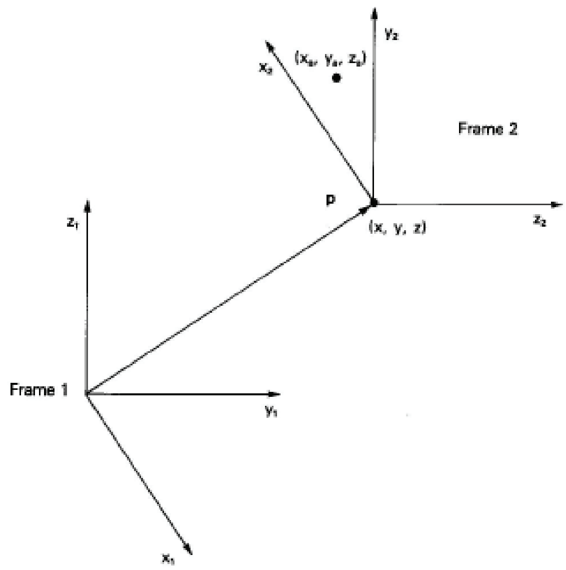

Figure 4.3: Displaced Reference Frames in the Same Space, ref. Klafter [10].

[Klafter]

In the above examples, we have looked at vectors in an orthogonal reference frame. Now, we will consider a space consisting of two or more orthogonal reference frames whose origins are displaced.

Note: We can use the previous transformation operators to align the displaced reference frames.

Notation: The symbol A_{mn} represents homogeneous transformation matrices that relate points in frame n in terms of frame m. In particular, $A_{0n} = A_{01} \times A_{12} \times A_{23} \times \cdots \times A_{(n-1)n}$.

Consider the following figure.

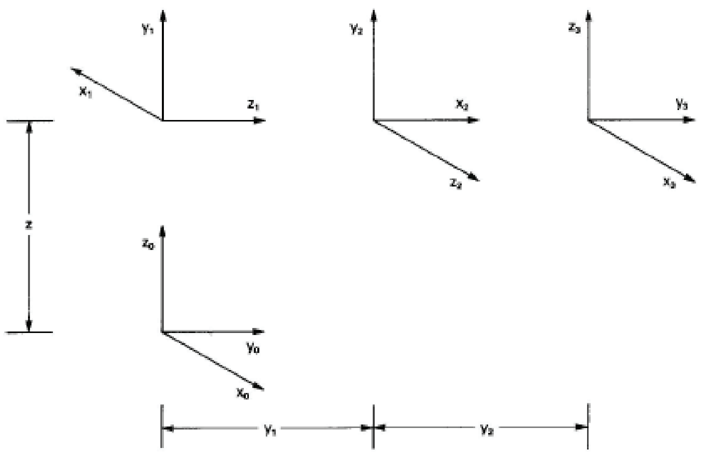

Figure 4.4: Alignment of Reference Frames Example,ref. Klafter [10].

Then,

$$A_{03} = A_{01} \times A_{12} \times A_{23} =$$

$$\begin{bmatrix} -1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & z \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 0 & 0 & -1 & 0 \\ 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & y_1 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 0 & 1 & 0 & y_2 \\ 0 & 0 & 1 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & y_1 + y_2 \\ 0 & 0 & 1 & z \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Definition: The *Inverse Homogeneous Transformation Matrix* of A_{mn} is A_{nm}. Hence, the inverse of a homogeneous transformation is defined by $(A_{mn})^{-1} = A_{nm}$. Thus, it follows that $A_{mn} \times A_{nm} = I$, the identity matrix.

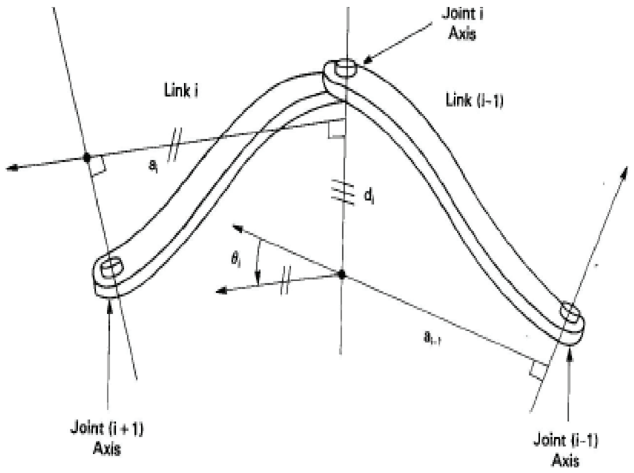

Figure 4.5: Establishing Link Coordinate Reference Frames, ref. Lee [14].

Looking at the above example, the inverse is simply taking the reverse. So, to find $(A_{03})^{-1}$, we simply walk backwards from the third reference frame and work our way towards the zero reference frame.

Similarly, a cartesian coordinate system can be assigned to each joint of a robot manipulator such that each connected link is assigned a particular coordinate axis. Each link coordinate frame is determined using the following rules:

1. The z_{i-1} axis lies along the axis of motion of the i^{th} joint.

2. The x_i axis is normal to the z_{i-1} axis, with it's positive direction towards the z_i axis.

3. The y_i axis is chosen so that the three axes form a right-handed system, ref. Lee [14].

There are four parameters associated with each link: a_i, the length of the link, α_i, the link twist, d_i, the linear distance of a prismatic joint, and θ_i, the degree rotation of a revolute joint. Each of these parameters is defined with respect to two joint axes attached to a particular link, such that

- θ_i is the joint angle from the x_{i-1} axis to the x_i about the z_{i-1} axis (using the right-hand rule)
- d_i is the distance from the origin of the $(i-1)^{th}$ coordinate frame to the intersection of the z_{i-1} axis with the x_i axis along the z_{i-1} axis.
- a_i is the distance from the intersection of the z_{i-1} axis with the x_i axis to the origin of the i^{th} frame along the x_i axis.
- α_i is the angle from the z_{i-1} axis to the z_i axis about the x_i axis.[Lee]

Observe that a_i and α_i describe the structure of a link whereas d_i and θ_i describe the position in regards to neighboring links.

Note: Assignment of reference frames is not unique.

Once the link coordinate frames are established, these four parameters are used in a transformation matrix to define the relationship between consecutive frames. This matrix is called the Denavit-Hartenberg Matrix (D-H Matrix), a homogeneous transformation matrix solely defined by the four link parameters and can be found by a series of translations and rotations.

Using the link parameters, let Rot(axis, θ_i) denote a rotational transformation about the given axis θ_i degrees, and let Trans(a,b,c) denote a translational transformation that moves a point defined by a vector $\vec{x}$, along the diagonal (a,b,c) to a new point defined by a vector $\vec{y}$. Then, the product of these transformations in this order gives the D-H Matrix:

$$A_{(i-1)i} = Rot(x_i, \alpha_i)Trans(a_i, 0, 0)Trans(0, 0, d_i)Rot(z_{i-1}, \theta_i)$$

$$= \begin{bmatrix} \cos\theta_i & -\cos\alpha_i \sin\theta_i & \sin\alpha_i \sin\theta_i & a_i \cos\theta_i \\ \sin\theta_i & \cos\alpha_i \cos\theta_i & -\sin\alpha_i \cos\theta_i & a_i \sin\theta_i \\ 0 & \sin\alpha_i & \cos\alpha_i & d_i \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Note: When solving the inverse kinematic robotics problems, a_i and α_i are constants and either d_i and θ_i are joint variables.

Thus, for an n-degree of freedom manipulator, let P be the matrix which is a function of all of the joint variables, then,

$$P = A_{0n} = A_{01} \times A_{12} \times A_{23} \times \cdots \times A_{(n-1)n}$$

where each A_{ij} matrix is of the D-H Matrix form.

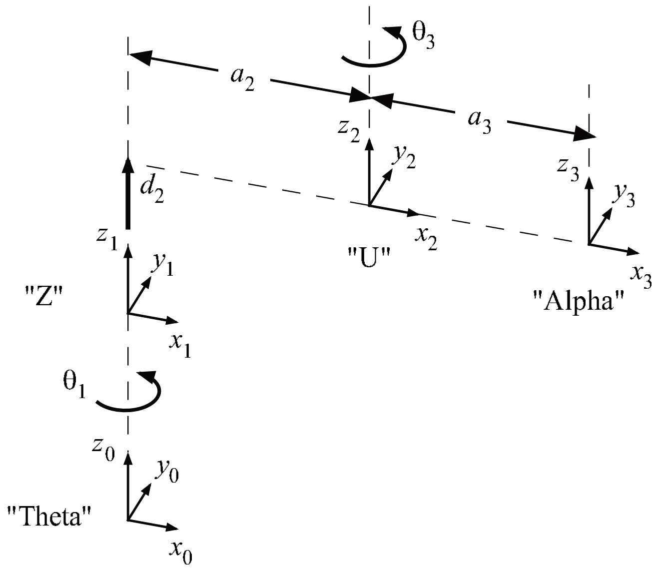

Figure 4.6: Establishing Link Coordinate Frames for the GMF A-510 Robot, ref. Hung [8].

4.5 A Kinematic Model for the GMF Robotics A510 Robot

The GMF Robotics A-510 robot has four degrees of freedom. However, the following kinematic model focuses on the first three joints. The last joint refers to the robot hand or end effector. Since we are solving the inverse kinematic robotics problem, we are only interested in where this fourth joint, the robot hand (end effector) will be placed in space. We establish the link coordinate frames in Figure 4.6.

Then describe each joint in terms of the four parameters a_i, α_i, θ_i, and d_i. Below is a table describing the joint parameters of the GMF Robotics A510 Robot.

Table 4.1: Table of A-510 link parameters, ref. Hung [8].

Joint, i	type	GMF name	location	a_i	α_i	θ_i	d_i
1	rotational	Theta	waist	0	0	θ_1	980 mm
2	prismatic	Z	-	410 mm	0	0	d_2
3	rotational	U	elbow	330 mm	0	θ_3	0

The information in the table allows us to form the following transformation matrices.

$$A_{01} = \begin{bmatrix} \cos\theta_1 & -\sin\theta_1 & 0 & 0 \\ \sin\theta_1 & \cos\theta_1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$A_{12} = \begin{bmatrix} 1 & 0 & 0 & 410 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & d_2 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$A_{23} = \begin{bmatrix} \cos\theta_3 & -\sin\theta_3 & 0 & 330 * \cos\theta_3 \\ \sin\theta_3 & \cos\theta_3 & 0 & 330 * \sin\theta_3 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

4.6 Solving the Inverse Kinematic Robotics Problem

To solve the inverse kinematic robotics problem for the GMF Robotics A510 Robot, let $P = A_{01} \times A_{12} \times A_{23}$. Using MAPLE, a mathematics computer package, we calculate P.

P=

$$\begin{bmatrix} \cos\theta_1\cos\theta_3 - \sin\theta_1\sin\theta_3 & -\cos\theta_1\sin\theta_3 - \sin\theta_1\cos\theta_3 & 0 & 330\cos\theta_1\cos\theta_3 - 330\sin\theta_1\sin\theta_3 + 410\cos\theta_1 \\ \sin\theta_1\cos\theta_3 + \cos\theta_1\sin\theta_3 & \cos\theta_1\cos\theta_3 - \sin\theta_1\sin\theta_3 & 0 & 330\sin\theta_1\cos\theta_3 + 330\cos\theta_1\sin\theta_3 + 410\sin\theta_1 \\ 0 & 0 & 1 & 980 + d_2 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

The fourth column represents the final position of the end effector. Using the Matrix Manipulation Method [8], we set the *T-Variable Matrix* equal to P, the final position matrix, and get

$$T - VariableMatrix = \begin{bmatrix} n_x & o_x & a_x & p_x \\ n_y & o_y & a_y & p_y \\ n_z & o_z & a_z & p_z \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$= \begin{bmatrix} \cos\theta_1\cos\theta_3 - \sin\theta_1\sin\theta_3 & -\cos\theta_1\sin\theta_3 - \sin\theta_1\cos\theta_3 & 0 & 330\cos\theta_1\cos\theta_3 - 330\sin\theta_1\sin\theta_3 + 410\cos\theta_1 \\ \sin\theta_1\cos\theta_3 + \cos\theta_1\sin\theta_3 & \cos\theta_1\cos\theta_3 - \sin\theta_1\sin\theta_3 & 0 & 330\sin\theta_1\cos\theta_3 + 330\cos\theta_1\sin\theta_3 + 410\sin\theta_1 \\ 0 & 0 & 1 & 980 + d_2 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

We want to find d_2, θ_1, and θ_3 that satisfy those equations in the fourth column representing p_x, p_y, and p_z. That is, we want to find those joint angles, that will allow the end effector to reach the points in space described by

$$= \begin{bmatrix} 330\cos\theta_1\cos\theta_3 - 330\sin\theta_1\sin\theta_3 + 410\cos\theta_1 \\ 330\sin\theta_1\cos\theta_3 + 330\cos\theta_1\sin\theta_3 + 410\sin\theta_1 \\ 980 + d_2 \\ 1 \end{bmatrix}$$

We multiply each side by A_1^{-1} such that

$$A_1^{-1} \times T - VariableMatrix = A_2 \times A_3$$

$$\begin{bmatrix} \cos\theta_1 & \sin\theta_1 & 0 & 0 \\ -\sin\theta_1 & \cos\theta_1 & 0 & 0 \\ 0 & 0 & 1 & -980 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} n_x & o_x & a_x & p_x \\ n_y & o_y & a_y & p_y \\ n_z & o_z & a_z & p_z \\ 0 & 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} \cos\theta_3 & -\sin\theta_3 & 0 & 330\cos\theta_3 + 410 \\ \sin\theta_3 & \cos\theta_3 & 0 & 330\sin\theta_3 \\ 0 & 0 & 1 & d_2 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Simplifying the left-hand side (LHS) further, we have

$$A_1^{-1} \times T - \mathit{VariableMatrix}$$

=

$$\begin{bmatrix} n_x\cos\theta_1 + n_y\sin\theta_1 & o_x\cos\theta_1 + o_y\sin\theta_1 & a_x\cos\theta_1 + a_y\sin\theta_1 & p_x\cos\theta_1 + p_y\sin\theta_1 \\ -n_x\sin\theta_1 + n_y\cos\theta_1 & -0_x\sin\theta_1 + o_y\cos\theta_1 & -a_x\sin\theta_1 + a_y\cos\theta_1 & -p_x\sin\theta_1 + p_y\cos\theta_1 \\ n_z & o_z & a_z & p_z - 980 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$= \begin{bmatrix} \cos\theta_3 & -\sin\theta_3 & 0 & 330\cos\theta_3 + 410 \\ \sin\theta_3 & \cos\theta_3 & 0 & 330\sin\theta_3 \\ 0 & 0 & 1 & d_2 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

=

$$A_2 \times A_3$$

We then set the elements of each matrix equal to one another. In particular, we have

$$p_x\cos\theta_1 + p_y\sin\theta_1 = 330\cos\theta_3 + 410 \tag{4.1}$$

$$-p_x \sin\theta_1 + p_y \cos\theta_1 = 330 \sin\theta_3 \tag{4.2}$$

$$d_2 = p_z - 980 \tag{4.3}$$

Thus, we have found $d_2 = p_z - 980$. So we have found one of the three unknown variables. However, we still need to find θ_1 and θ_3.

To find θ_3 , we use (4.1) and (4.2). Observe by squaring both sides of (4.1),

$$(p_x \cos\theta_1 + p_y \sin\theta_1)^2 = (330 \cos\theta_3 + 410)^2 \tag{4.4}$$

gives

$$p_x^2 (\cos\theta_1)^2 + 2p_x p_y \cos\theta_1 \sin\theta_1 + p_y^2 (\sin\theta_1)^2 = 330^2 (\cos\theta_3)^2 + 2 * 410 * 330 \cos\theta_3 + 410^2 \tag{4.5}$$

And by squaring both sides of (4.2)

$$(-p_x \sin\theta_1 + p_y \cos\theta_1)^2 = (330 \sin\theta_3)^2$$

gives

$$p_x^2(\sin\theta_1)^2 - 2p_xp_y\cos\theta_1\sin\theta_1 + p_y^2(\cos\theta_1)^2 = 330^2(\sin\theta_3)^2 \tag{4.6}$$

Then, by adding (4.5) and (4.6), we get

$$p_x^2 + p_y^2 = 330^2 + 2*410*330\cos\theta_3 + 410^2 \tag{4.7}$$

Then,

$$\frac{p_x^2 + p_y^2 - 330^2 - 410^2}{2*410*330} = \cos\theta_3 = X_3 \tag{4.8}$$

Now, we shall find $\sin\theta_3$. To do so, we substitute for $\cos\theta_3$ in (4.1), and square both sides, then add (4.6).

$$p_x^2(\cos\theta_1)^2 + 2p_xp_y\cos\theta_1\sin\theta_1 + p_y^2(\sin\theta_1)^2 = \{\frac{330(p_x^2 + p_y^2 - 330^2 - 410^2)}{2*410*330} + 410\}^2$$

$$+p_x^2(\sin\theta_1)^2 - 2p_xp_y\cos\theta_1\sin\theta_1 + p_y^2(\cos\theta_1)^2 = 330^2(\sin\theta_3)^2$$

Combining the left-hand sides and right-hand sides respectively, yields

$$p_x^2 + p_y^2 = \{\frac{330 * (p_x^2 + p_y^2 - 330^2 - 410^2)}{2 * 410 * 330} + 410\}^2 + 330^2(\sin\theta_3)^2$$

Then,

$$\pm\sqrt{\frac{p_x^2 + p_y^2 - \{330 * \frac{(p_x^2+p_y^2-330^2-410^2)}{(2*410*330)} + 410^2\}}{330}} = \sin\theta_3 = Y_3 \tag{4.9}$$

Thus,

$$\theta_3 = \tan^{-1}(\frac{Y_3}{X_3}), \qquad for \qquad \frac{-\pi}{2} \le \theta_3 \le \frac{\pi}{2} \tag{4.10}$$

Now that we have found θ_3, we look to find θ_1. Using (4.1) and (4.2), we have two equations in two unknowns

$$p_x \cos\theta_1 + p_y \sin\theta_1 = 330\cos\theta_3 + 410$$

$$-p_x \sin\theta_1 + p_y \cos\theta_1 = 330\sin\theta_3$$

Using elimination, we solve the system of equations to get

$$Y_1 = \sin\theta_1 = \frac{-p_y(330X_3 + 410) + p_x 330Y_3}{-p_x^2 - p_y^2} \tag{4.11}$$

and

$$X_1 = \cos\theta_1 = \frac{(330X_3 + 410)p_x + 330Y_3 p_y}{p_x^2 + p_y^2} \tag{4.12}$$

Thus,

$$\theta_1 = \tan^{-1}(\frac{Y_1}{X_1}), \qquad for \qquad \frac{-\pi}{2} \leq \theta_1 \leq \frac{\pi}{2} \tag{4.13}$$

So, we have found the joint angles (d_2, θ_1, and θ_3) that will allow the end effector to touch the points (p_x, p_y, p_z) in space. Hence, the inverse kinematic robotics problem for the GMF Robotics A-510 Robot is solved.

4.6.1 Case Examples

We verify these results in the following examples:

Note: Since p_z determines the height of (p_x, p_y), we restrict our focus to (p_x, p_y), with the

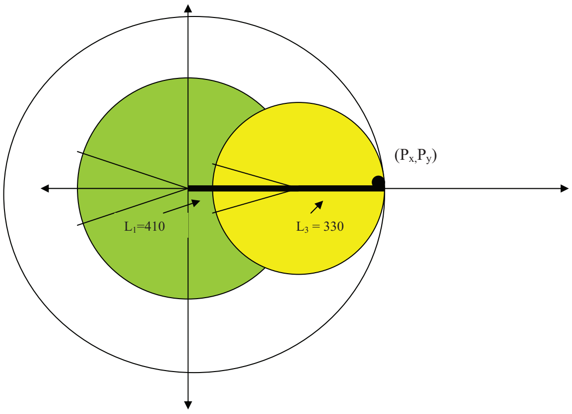

Figure 4.7: Case 1: L_1 and L_3 are Collinear

understanding that (p_x, p_y) can occur at any height p_z.

Case 1: When L_1 and L_3 are Collinear

If $L_1 = 410$ and $L_3 = 330$ are collinear, then $\theta_3 = 0°$. Thus, for θ_1 from $-150° \leq \theta_1 \leq 150°$, we have the following

$$p_x = 330\cos(\theta_1 + 0) + 410\cos\theta_1 = 740\cos\theta_1$$

$$p_y = 330\sin(\theta_1 + 0) + 410\sin\theta_1 = 740\sin\theta_1$$

Thus, there is one formation to reach the point (p_x, p_y).

Case 2: A Point Inside a Circle of Radius $r = 740$ Centered at the Origin

We wish to find (p_x, p_y) where $\theta_1 = 30°$ and $\theta_3 = 30°$ by using the Forward Kinematic Robotics Problem

$$p_x = 330\cos(60) + 410\cos(30) = 165 + 205\sqrt{3}$$

$$p_y = 330\sin(60) + 410\sin(30) = 165\sqrt{3} + 205$$

Using the Inverse Kinematic Robotics Problem, we check these results. Given the final position described by p_x and p_y, we wish to find θ_1 and θ_3. We substitute into the equations derived in the previous section,

$$X_3 = \cos\theta_3 = \frac{(165 + 205\sqrt{3})^2 + (165\sqrt{3} + 205)^2 - 330^2 - 410^2}{2 * 410 * 330} = \frac{\sqrt{3}}{2}$$

$$Y_3 = \sin\theta_3 = \pm\frac{\sqrt{(165+205\sqrt{3})^2+(165\sqrt{3}+205)^2-(330\frac{\sqrt{3}}{2}+410)^2}}{330} = \pm\frac{1}{2}$$

Thus,

$$\theta_3 = \tan^{-1}(\frac{\frac{1}{2}}{\frac{\sqrt{3}}{2}}) = 30^\circ \qquad for \qquad \frac{-\pi}{2} \le \theta_3 \le \frac{\pi}{2}$$

and,

$$\theta_3 = \tan^{-1}(\frac{\frac{-1}{2}}{\frac{\sqrt{3}}{2}}) = -30^\circ \qquad for \qquad \frac{-\pi}{2} \le \theta_3 \le \frac{\pi}{2}$$

To find θ_1, using $Y_3 = \frac{1}{2}$, we substitute these values into the equations above to get

$$Y_1 = \sin\theta_1 = \frac{-(165\sqrt{3}+205)(330\frac{\sqrt{3}}{2}+410)+(165+205\sqrt{3})330\frac{1}{2}}{-(165+205\sqrt{3})^2-(165\sqrt{3}+205)^2} = \frac{1}{2}$$

$$X_1 = \cos\theta_1 = \frac{(330\frac{\sqrt{3}}{2}+410)(165+205\sqrt{3})+330\frac{1}{2}(165\sqrt{3}+205)}{(165+205\sqrt{3})^2+(165\sqrt{3}+205)^2} = \frac{\sqrt{3}}{2}$$

Thus,

$$\theta_1 = \tan^{-1}(\frac{\frac{1}{2}}{\frac{\sqrt{3}}{2}}) = 30^\circ$$

Also, we need to find θ_1 when $Y_3 = -\frac{1}{2}$

$$Y_1 = \sin\theta_1 = \frac{-(165\sqrt{3}+205)(330\frac{\sqrt{3}}{2}+410)+(165+205\sqrt{3})330\frac{-1}{2}}{-(165+205\sqrt{3})^2-(165\sqrt{3}+205)^2} = 0.84$$

$$X_1 = \cos\theta_1 = \frac{(330\frac{\sqrt{3}}{2}+410)(165+205\sqrt{3})+330\frac{-1}{2}(165\sqrt{3}+205)}{(165+205\sqrt{3})^2+(165\sqrt{3}+205)^2} = 0.55$$

Thus,

$$\theta_1 = \tan^{-1}(\frac{0.84}{0.55}) \approx 56.83^\circ$$

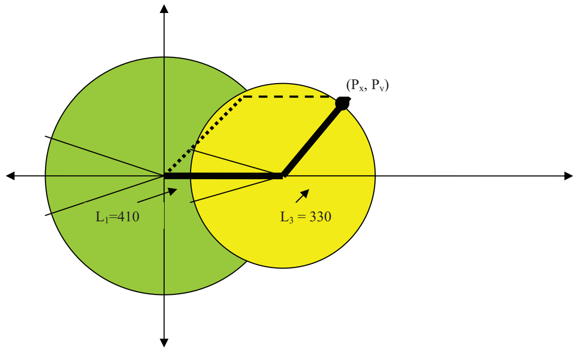

Figure 4.8: Case 2: A Point Inside a Circle Centered at the Origin of radius $r = 740$

Since we solved a quadratic equation for Y_3, we found two solutions for θ_3 and two corresponding solutions for θ_1. Hence, there are two joint settings that will place the robot hand at $(p_x, p_y) = (165 + 205\sqrt{3}, 165\sqrt{3} + 205)$. They are

$$\theta_3 = 30^\circ \qquad \theta_1 = 30^\circ$$

and

$$\theta_3 = -30^\circ \qquad \theta_1 = 56.83^\circ$$

Thus, we have found all joint settings that will place the robot hand at $(p_x, p_y) = (165 + 205\sqrt{3}, 165\sqrt{3} + 205)$.

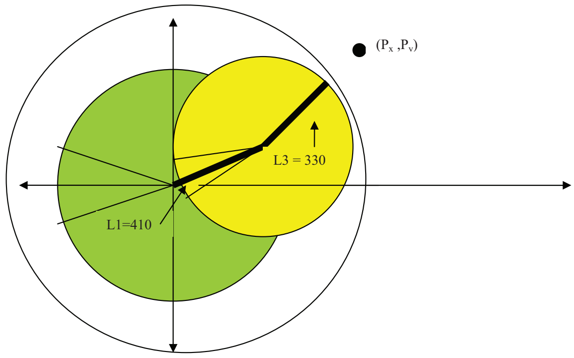

Figure 4.9: Case 3: A Point Outside a Circle Centered at the Origin of radius $r = 740$

Case 3: A Point Outside a Circle of Radius $r = 740$ Centered at the Origin

Let $(p_x, p_y) = (760, 760)$. We need to calculate θ_1 and θ_3.

For θ_3,

$$X_3 = \cos\theta_3 = \frac{760^2 + 760^2 - 330^2 - 410^2}{2 * 410 * 330} = \frac{878200}{270600} = 3.24538$$

But this is a contradiction since $\cos\theta_3$ must lie between -1 and 1.

It is clear that a solution for θ_3 will not be found. Thus, a solution for θ_1 will not be found. Hence, this example supports the fact that only those values that lie in the specified domain for each joint variable have the potential to exist as possible solutions.

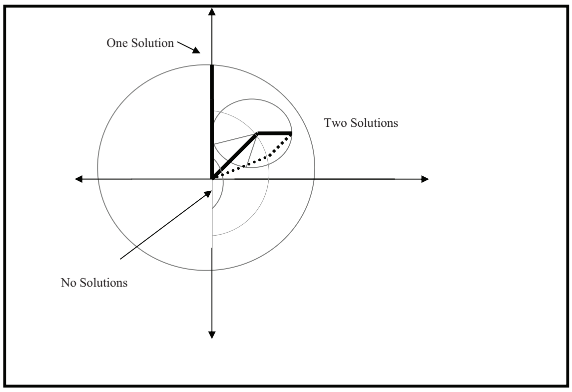

Figure 4.10: Summary of Solutions

4.7 Summary: Engineering Approach

4.7.1 Results for the GMF Robotics A510 Robot

We have found d_2, θ_1, and θ_3. In particular, we found θ_1 and θ_3 in terms of (p_x, p_y). From this representation, however, we are not able to gain enough information about each (p_x, p_y). In fact, we don't know if (p_x, p_y) is a reachable point in space unless we substitute it into the equations derived for θ_1 and θ_3. This could become tedious. From our calculations of a quadratic equation, the most we can conclude is that there are two angle configurations for each joint, to reach (p_x, p_y). It is desired to find all (p_x, p_y) that are reachable points within our workspace.

By picking three different points in space, we discovered for θ_1 and θ_3, such that $\frac{-\pi}{2} \leq \theta_1 \leq \frac{\pi}{2}$ and $\frac{-\pi}{2} \leq \theta_3 \leq \frac{\pi}{2}$, that there are at most none, one, or two formations that can place the robot hand at a particular point in space. We wish to determine the unique set of points that will produce no solutions, the unique set of points that will produce one solution, and the unique set of points that will produce two solutions.

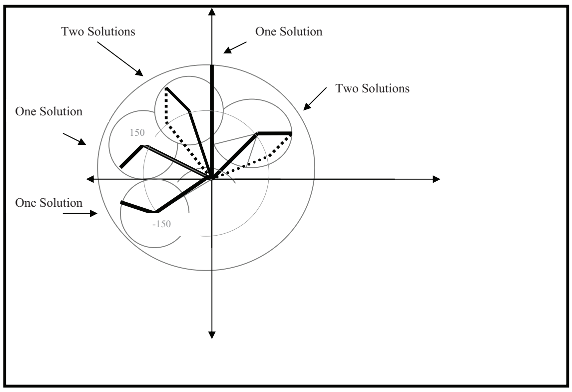

Figure 4.11: Summary of Solutions found with Manipulation

Observe further that since θ_1 and θ_3 range from $-150°$ to $150°$, and the equations we derived for them satisfy $\frac{-\pi}{2} \leq \theta_1 \leq \frac{\pi}{2}$ and $\frac{-\pi}{2} \leq \theta_3 \leq \frac{\pi}{2}$, we have not found all possible solutions. We manipulate the equations in our kinematic model to find any remaining solutions.

Using (4.1), if we let

$$\theta_3 = \cos^{-1}\left[\frac{p_x^2 + p_y^2 - 330^2 - 410^2}{2 * 410 * 330}\right]$$

where $0 \leq \theta_3 \leq \pi$, and let θ_1 range from $-150°$ to $150°$, we discover more solutions. In fact, by choosing three more test points in this new domain, there are none, one, or two formations that can place the robot hand at a particular point inside this domain. Since the geometry of our robot arm corresponds to the symmetry of the unit axis, we have also found those solutions where $-\pi \leq \theta_3 \leq 0$

By analyzing the geometry of the robot, and manipulating our kinematic model, we discovered more solutions that determined new formations. But we were only able to do so because of the simple geometry of the GMF Robotics A-510 Robot. However, if the geometry of the robot is more complex, solutions are not easily found, and in some cases, solutions are not found at all. Facing this challenge, we acknowledge other difficulties to the Engineering Approach.

4.7.2 Difficulties

1. There are instances when no solution can be found due to limitations on the robot manipulator,

2. Multiple solutions are possible since the robot manipulator can use different formations to place the robot hand at a particular point in space,

3. Careful analysis of the geometry of the robot manipulator forces a close examination and, in some cases, manipulation of the equations found to solve the inverse kinematic robotics problem in order to find all possible solutions.

4. Some calculations cannot be computed as easily by hand. Hence, the use of MAPLE, a mathematics computer software.

To resolve these difficulties and other challenges mentioned, we search for an alternate method–The Mathematician's Approach.

Chapter 5

The Mathematician's Approach

5.1 Introduction

The Mathematician's Approach uses Groebner Basis Theory to solve the inverse kinematic robotics problem. We begin with background on Groebner Basis Theory. Then, with an example, show how to develop a Groebner basis. Lastly, we use a Groebner basis to solve the inverse kinematic robotics problem for the GMF Robotics A-510 Robot.

5.2 Groebner Basis Theory

Groebner Basis Theory allows mathematicians to use a particular algorithm to solve systems of polynomial equations. More formally, this algorithm, named Buchberger's algorithm [2], after mathematician Bruno Buchberger, applies computational algebra techniques to specific polynomial ideals, producing a Groebner basis that can be used to find solutions to a set of non-zero polynomials for a given ideal.

For multivariant polynomials, the complexity of computing a Groebner basis does not depend on the number of polynomials in the set, but rather on the term order. By ordering a set of polynomials with respect to lexigraphical order, at least one element in the Groebner basis will be in terms of one variable in the set. This makes calculating a Groebner basis 'nice', because through back substitution, we can find a solution to the rest of the variables in the set. We illustrate this in the following sections.

Groebner Basis Theory is a very popular field of study, but mathematicians appreciate it most for its applications in other areas of mathematics. For example, Robbiano [20] uses Groebner bases to solve problems in Design Experiments in Statistics. Wang [22] shows how Groebner bases can be used to prove Geometric problems. Cox, Little, and O'Shea [3] have devoted a text to Groebner Basis Theory and its applications in Geometry and Graph Theory. Below, we illustrate an application in robotics.

5.2.1 Terminology and Notation

Terminology:

1. For a field k, a *Monomial Ordering* on $k[x_1, \ldots, x_n]$ is any relation on the set of monomials $x^\alpha, \alpha \in Z^n_{\geq 0}$ such that this relation is a linear ordering, and if $\alpha, \beta, \gamma \in Z^n_{\geq 0}$ such that $\alpha > \beta$, then $\alpha + \gamma > \beta + \gamma$. Moreover, this relation is also a well ordering.

2. *Lexigraphical order (lex order):* Given a monomial ordering, let $\alpha = (\alpha_1, \ldots, \alpha_n)$ and $\beta = (\beta_1, \ldots, \beta_n) \in N^n_{\geq 0}$. We say $x^\alpha > x^\beta$ if $\alpha > \beta$.

Example: Let $f = 7x^2yx + 5z^2y + 12xz + 3y^2z + 8 \in k[x, y, z]$. Then,

With respect to lex order $x > y > z$, $f = 7x^2yz + 12xz + 3y^2z + 5yz^2 + 8$

Notation:

Let k be any field and let $k \in [x_1, \ldots, x_n]$ be ordered with respect to lex order, then for all $f \in k[x_1, \ldots, x_n]$, with $f \neq 0$,

$$f = a_1 x^{\alpha_1} + a_2 x^{\alpha_2} + \ldots + a_r x^{\alpha_r},$$

where $0 \neq a_i \in k[x_1, \ldots, x_n]$ and x_i^{α} is ordered such that $x^{\alpha_1} > x^{\alpha_2} > \ldots > x^{\alpha_r}$.

- The *leading power product* of f is defined as $lp(f) = x^{\alpha_1}$

- The *leading coefficient* of f is defined as $lc(f) = a_1$

- The *leading term* of f is defined as $lt(f) = a_1 x^{\alpha_1}$

5.3 Algorithm for Computing a Groebner Basis

Introduced in 1965 by Bruno Buchberger, an algorithm can be applied to a non-zero set of polynomials for a given ideal to produce a Groebner basis for that given ideal. More formally, we have the following definition,

Definition: A set of non-zero polynomials $G = \{g_1, \ldots, g_t\}$ contained in an ideal I, is called a *Groebner Basis* for I if and only if for all $f \in I$ such that $f \neq 0$, there exists $i \in 1, \ldots, t$ such that $lp(g_i)$ divides the $lp(f)$.

5.3.1 S-Polynomials and Buchberger's Algorithm

To develop Buchberger's Algorithm, we must first understand the nature of *S-polynomials*. In the following examples, let k be any field, and let Q denote the Rational Field.

Definition: Let $0 \neq f,\, g \in k[x_1, \ldots, x_n]$. Let $L = lcm(lp(f), lp(g))$. The polynomial

$$S(f, g) = \frac{L}{lt(f)} * f - \frac{L}{lt(g)} * g$$

is called the *S-polynomial* of f and g.

Example: Let $f = 2yx - y, g = 3y^2 - x \in Q[x, y]$, with lex order $y > x$. Find $S(f, g)$.

Solution: First, we need to find $L = lcm(lp(f), lp(g)) = lcm(yx, y^2) = y^2x$. Then,

$$S(f, g) = \frac{y^2x}{2yx} f - \frac{y^2x}{3y^2} g = \frac{y}{2} f - \frac{x}{3} g$$

$$= \frac{y}{2}(2yx - y) - \frac{x}{3}(3y^2 - x) = -\frac{y^2}{2} + \frac{x^2}{3}$$

Ref. Adams [1]

Thus, an S-polynomial allows for the cancelation of leading terms.

Definition: Let $G = \{g_1, \ldots, g_t\}$ be a set of non-zero polynomials in $k[x_1, \ldots, x_n]$. Then G is a *Groebner basis* for the ideal $I = \langle g_1, \ldots, g_t \rangle$ if and only if for all $i \neq j$,

$$S(g_i, g_j) \xrightarrow{G}_{+} 0$$

. That is, $S(g_i, g_t)$ is divided by those $\{g_1, \ldots, g_t\} \in G$, such that the remainder is zero.

Example: Let $f_1 = yx - x, f_2 = -y + x^2 \in Q[x, y]$. Using lex order with $y > x$, compute a Groebner basis for the ideal.

Solution: First, let $F = \{f_1, f_2\}$ we calculate the S-polynomial, $S(f_1, f_2)$. Then, $L = lcm(lp(f_1), lp(f_2)) = lcm(xy, y) = xy$ and

$$S(f_1, f_2) = \frac{xy}{xy} f_1 - \frac{xy}{-y} f_2 = \frac{xy}{xy}(xy - x) - \frac{xy}{-y}(-y + x^2)$$

$$xy - x + x(-y + x^2) = xy - x - xy + x^3 = -x + x^3,$$

with respect to lex order $S(f_1, f_2) = x^3 - x$. Using the above definition, to have a Groebner basis, $S(f_1, f_2) \longrightarrow_{+}^{F} = 0$, but since $S(f_1, f_2) \longrightarrow_{+}^{F} \neq 0$, we take $f_3 = x^3 - x$ and form F'. That is, $F' = \{f_1, f_2, f_3\}$.So we compute a Groebner basis for F' by calculating $S(f_1, f_2)$, $S(f_2, f_3)$ and $S(f_1, f_3)$.

For $S(f_2, f_3)$,

$$L = lcm(lp(f_2), lp(f_3)) = lcm(y, x^3) = yx^3.$$

$$S(f_2, f_3) = \frac{yx^3}{-y} f_2 - \frac{yx^3}{x^3} f_3 = \frac{yx^3}{-y}(-y + x^2) - \frac{yx^3}{x^3}(x^3 - x)$$

$$yx^3 - x^5 - yx^3 + yx = yx - x^5$$

For $S(f_1, f_3)$,

$$L = lcm(lp(f_1), lp(f_3)) = lcm(xy, x^3) = yx^3.$$

$$S(f_1, f_3) = \frac{yx^3}{xy} f_1 - \frac{yx^3}{x^3} f_3 = \frac{yx^3}{xy}(xy - x) - \frac{yx^3}{x^3}(x^3 - x)$$

$$yx^3 - x^3 - yx^3 + yx = yx - x^3 = -x * f_2$$

But,

$$S(f_2, f_3) = yx - x^5 = -(yx - x) + (x^3 - x) = -yx + x^3 = x * f_2 \tag{5.1}$$

and

$$S(f_1, f_3) = -x^3 + yx = -x(-y + x^2) = -x * f_2 \tag{5.2}$$

By adding (5.1) and (5.2) we get zero. Thus, $S(f_2, f_3) \longrightarrow_{+}^{F'} = 0$ and $S(f_1, f_3) \longrightarrow_{+}^{F'} = 0$.

Hence, $F' = \{f_1, f_2, f_3\}$ is a Groebner basis for $F = \{f_1, f_2\}$, ref. Adams [1].

With the aid of MAGMA, a mathematical computer software package, we are able to compute a Groebner basis faster and for larger sets of non-zero polynomials.

5.4 MAGMA: Algebraic Computer Software

MAGMA is an algebraic computer system created to solve problems in Algebra, Geometry, and Number theory. It operates on Linux based systems, but has recently been adapted for Windows. Developed in 1993 by the Computational Algebra Group in the School of Mathematics and Statistics at the University of Sydney, MAGMA has received much praise for its advanced algorithms. Devoted to efficiency, the developers release new versions of the software every year.

With the assistance of MAGMA, we are able to calculate Groebner bases faster and, more importantly, are able to compare various alternatives yielding more accurate results that could not necessarily be calculated as easily by hand.

5.5 Solving the Inverse Kinematics Robotics Problem

5.5.1 Introduction

The inverse kinematic robotics problem is simply determining all combinations of joint settings that will place the robot arm at a given point in space. To solve this, we must do the following:

1. Derive polynomial equations modeling the motion of the robot arm at each joint setting such that the robot arm can be placed at a given point in space.
2. Determine the proper intervals which allow movement at each joint.
3. Then with these equations, find a 'nice' set of solutions that will provide us with possible movements of each joint.

Hence the use of a Groebner basis!!

5.5.2 Algebraic Model for the GMF Robotics A-510 Robot

We look to model the behavior of the GMF Robotics A-510 Robot using simple polynomial equations. By taking a top-down view of the robot, we can project the movements

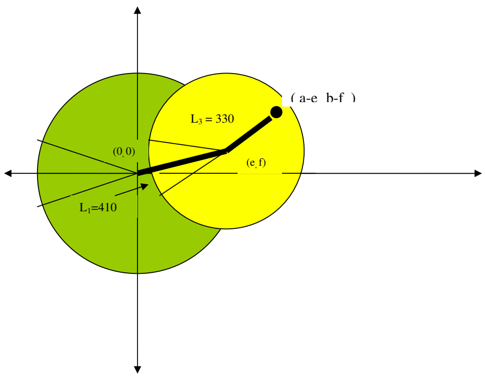

Figure 5.1: Algebraic Model: Top View of GMF A-510 Robot

of each joint variable onto the xy-plane. Let the point (a, b) represent where the robot hand (end effector) will be placed in space. To solve the inverse kinematic robotics problem, these polynomial equations need to describe the behavior of θ_1 and θ_3 in terms of the point (a, b).

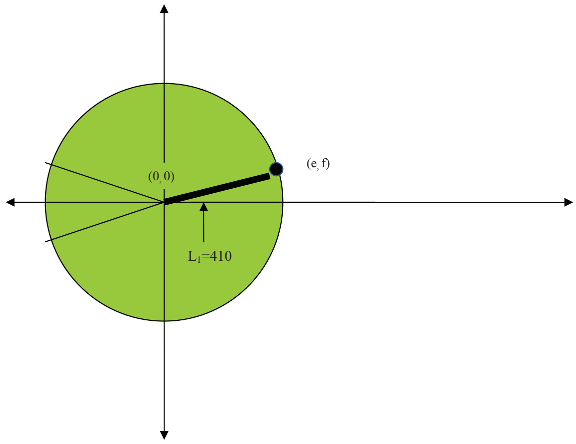

Figure 5.2: Algebraic Model: Top View of GMF A-510 Robot $L_1 = 410$

Thus,

$$(a - e, b - f) = (330\cos(\theta_1 + \theta_3), 330\sin(\theta_1 + \theta_3))$$

for

$$-150^\circ \leq \theta_1 \leq 150^\circ$$

and

$$-150^\circ \leq \theta_3 \leq 150^\circ$$

$$(e, f) = (410\cos\theta_1, 410\sin\theta_1) \qquad for \qquad -150^\circ \leq \theta_1 \leq 150^\circ$$

.

From the figures above,

- $e = 410 \cos \theta_1$
- $f = 410 \sin \theta_1$
- $a - e = 330 \cos(\theta_3 + \theta_1)$
- $b - f = 330 \sin(\theta_3 + \theta_1)$

By substituting, we get

$$a = 330 \cos(\theta_3 + \theta_1) + 410 \cos \theta_1$$

$$b = 330 \sin(\theta_3 + \theta_1) + 410 \sin \theta_1$$

By using the addition identities of trigonometric functions,

$$a = 330(\cos\theta_1 \cos\theta_3 - \sin\theta_1 \sin\theta_3) + 410\cos\theta_1$$

and

$$b = 330(\cos\theta_1 \sin\theta_3 + \cos\theta_3 \sin\theta_1) + 410\sin\theta_1$$

Recall the trigonometric identities:

$$\cos^2\theta_3 + \sin^2\theta_3 = 1$$

and

$$\cos^2\theta_1 + \sin^2\theta_1 = 1$$

Together, we have a system of four equations in four unknowns.

$$330(\cos\theta_1 \cos\theta_3 - \sin\theta_1 \sin\theta_3) + 410\cos\theta_1 - a = 0 \tag{5.3}$$

$$330(\cos\theta_1 \sin\theta_3 + \cos\theta_3 \sin\theta_1) + 410\sin\theta_1 - b = 0 \tag{5.4}$$

$$\cos^2\theta_3 + \sin^2\theta_3 - 1 = 0 \tag{5.5}$$

$$\cos^2\theta_1 + \sin^2\theta_1 - 1 = 0 \tag{5.6}$$

We use MAGMA to find a solution to the system.

5.5.3 Calculating A Groebner Basis With MAGMA

Let $c_i = \cos\theta_i$ and $s_i = \sin\theta_i$.

We have the following MAGMA code:

Q:=RationalField ();

FF $< L1, L3, a, b >$:=FunctionField(Q,4, 'lex');

P $< c_3, s_3, c_1, s_1 >$:=PolynomialRing(FF,4);

$f_1 := L3 * (c_1 * c_3 - s_1 * s_3) + L1 * c_1 - a;$

$f_2 := L3 * (c_1 * s_3 + c_3 * s_1) + L1 * s_1 - b;$

$f_3 := c_1^2 + s_1^2 - 1;$

$f_4 := c_3^2 + s_3^2 - 1;$

I:=$ideal < P | f_1, f_2, f_3, f_4 >$;

G:=GroebnerBasis(I);

Note: A *Function Field* allows some variables to act as coefficients and also allows for division. Above, c_3, s_3, c_1, s_1 act as variables and $L1, L3, a, b$ act as coefficients.

Using lex order $c_3 > s_3 > c_1 > s_1$, MAGMA produces the following basis:

$c_3 - \frac{a^2+b^2-L1^2-L3^2}{2*L1*L3}$

$s_3 + \frac{a^2+b^2}{a*L3} - a^2b + b^3 + \frac{b(L1^2+L3^2)}{2a*L1*L3}$

$c_1 + \frac{b}{a}s_1 - \frac{a^2+b^2+L1^2-L3^2}{2a*L3}$

$s_1^2 + \frac{a^2b+b^3+b(L1^2+L3^2)}{L1(a^2+b^2)}s_1 + \frac{(a^2+b^2)^2+(L1^2+L3^2)^2-a^2(L1^2+L3^2)+2b^2(L1^2-L3^2)}{4*L1^2(a^2+b^2)}$

So, we have a Groebner basis when $L_1, L_3, a \neq 0$ and $(a^2+b^2) \neq 0$. If we substitute $L_1 = 410$ and $L_3 = 330$, we have the following basis:

$c_3 - \frac{a^2+b^2-410^2-330^2}{2*410*330}$

$s_3 + \frac{a^2+b^2}{a*330} - a^2b + b^3 + \frac{b(410^2+330^2)}{2a*410*330}$

$c_1 + \frac{b}{a}s_1 - \frac{a^2+b^2+410^2-330^2}{2a*330}$

$s_1^2 + \frac{a^2b+b^3+b(410^2+330^2)}{410(a^2+b^2)}s_1 + \frac{(a^2+b^2)^2+(410^2+330^2)^2-a^2(410^2+330^2)+2b^2(410^2-330^2)}{4*410^2(a^2+b^2)}$

Looking at the last element in our basis, we see that it is a quadratic polynomial in terms of s_1. So, we can solve for s_1 and determine the values of a and b that will provide us with solutions for the GMF Robotics A-510 robot.

Notation: Let s_{ij} denote solution number j for s_i or c_i, for $i = 1, 3$, $j = 1, 2$.

By examining the discriminant, there will be two real solutions for s_1, and for each of s_{11}, and s_{12}, there will be one corresponding value for c_1, s_2, c_2, found by using back substitution. Furthermore, we conclude that there are two unique joint settings when

$$0 \leq 4\frac{(a^2+b^2)^2 + 3504640000 - 2a^2 * 27700 + 2b^2 * 59200}{1640(a^2+b^2)} \leq \frac{a^2b + b^3 + 592000}{24272000^2}$$

Now we can consider two remaining cases:

1. Case 1: $a = 0$ and $b = 0$

2. Case 2: $a = 0$ and $b \neq 0$

5.5.4 Case Examples

Case 1: $a = 0$ **and** $b = 0$

We have the following MAGMA code:

Q:=RationalField ();

FF $< L1, L3 >$:=FunctionField(Q,2);

P $< c_3, s_3, c_1, s_1 >$:=PolynomialRing(FF,4, 'lex');

$f_1 := L3 * (c_1 * c_3 - s_1 * s_3) + L1 * c_1;$

$f_2 := L3 * (c_1 * s_3 + c_3 * s_1) + L1 * s_1;$

$f_3 := c_1^2 + s_1^2 - 1;$

$f_4 := c_3^2 + s_3^2 - 1;$

I:=$ideal < P|f_1, f_2, f_3, f_4 >$;

G:=GroebnerBasis(I);

The Groebner basis for this case is 1.

Thus, there are no joint settings to place the robot arm at the point (a, b). It is obvious to see by looking at the geometry of the robot arm.

Case 2: $a = 0$ **and** $b \neq 0$

Q:=RationalField ();

FF $< L1, L3b >$:=FunctionField(Q,2);

P $< c_3, s_3, c_1, s_1 >$:=PolynomialRing(FF,4, 'lex');

$f_1 := L3 * (c_1 * c_3 - s_1 * s_3) + L1 * c_1;$

$f_2 := L3 * (c_1 * s_3 + c_3 * s_1) + L1 * s_1 - b;$

$f_3 := c_1^2 + s_1^2 - 1;$

$f_4 := c_3^2 + s_3^2 - 1;$

I:=$ideal < P|f_1, f_2, f_3, f_4 >$;

G:=GroebnerBasis(I);

The Groebner basis for this case is

$c_3 + \frac{L_3^2+L_1^2-b^2}{2L_3L_1}$

$s_3 - \frac{b}{L_3}c_1$

$c_1^2 + \frac{(L_3^4+L_1^4+b^4)-2(L_3^2+L_1^2+b^2(L_3^2+L_1^2))}{4L_1^2b^2}$

$s_1 + \frac{L_3^2-L_1^2-b^2}{2L_1b}$

So, we have a Groebner basis when $L_1, L_3, b \neq 0$. By substituting $L_1 = 410$ and $L_3 = 330$, we have the following basis:

$c_3 + \frac{330^2+410^2-b^2}{2*410*330}$

$s_3 - \frac{b}{330}c_1$

$c_1^2 + \frac{(330^4+410^4+b^4)-2(330^2+410^2+b^2(330^2+410^2))}{4*410^2b^2}$

$s_1 + \frac{330^2-410^2-b^2}{2*410b}$

We can immediately solve for s_1 and c_3. Solving for these aligns each joint link pair. Note that the third element in the basis is a quadratic polynomial in terms of c_1. By solving for c_1, we find two solutions c_{11} and c_{12}, and for each, there is a corresponding value for s_3. This means that we can rotate about θ_1, given those values of s_3 that will keep each joint link pair aligned. Thus, there is only one unique joint setting to place the robot arm at the point (a, b) when $a = 0$ and $b \neq 0$.

5.6 Summary: Mathematician's Approach

By analyzing the above, we find the following results:

1. There are at most two real solutions (joint configurations) when (a, b) satisfies

$$0 \leq \frac{4(a^2+b^2)^2 + 3504640000 - 554000a^2 + 118400b^2}{1640(a^2+b^2)} \leq (\frac{a^2b + b^3 + 59200}{24272000})^2.$$

2. From Case 1, no solutions when $a = b = 0$.

3. From Case 2, one real solution (joint configuration) when $(0, b)$ satisfies

$$0 \leq b\sqrt{554000 - b^2} \leq 3504640000.$$

4. Those points (a,b) that do not satisfy any of the above are outside of the robot's reachable workspace. These points represent no solutions.

By using Groebner Basis Theory and MAGMA, we have found real solutions (see above) to the inverse kinematic robotics problem. In fact, we have found all of the possible formations to place the robot hand at the point (a, b). These solutions are more precise because we determine the set of points that determine two solutions, the set of points that determine one solution, and the set of points that determines no solution. Thus, the inverse kinematic robotics problem is solved. We explain our results further in the next chapter.

Chapter 6

Method Analysis

6.1 GMF Robotics A-510 Robot

From the Engineering Approach, we have learned that there is the possibility of none, one, or two solutions. However, when using the Denavit-Hartenberg Matrix, we were only able to develop an equation that could find two solutions for θ_1 and θ_3 within the given domain: $-\frac{\pi}{2} \leq \theta_1 \leq \frac{\pi}{2}$ and $-\frac{\pi}{2} \leq \theta_3 \leq \frac{\pi}{2}$. Since θ_1 and θ_3 rotated beyond $\frac{-\pi}{2}$ to $\frac{\pi}{2}$, there was the possibility that more solutions existed. By manipulating the equations found using the Denavit-Hartenberg Matrix, we expanded our domain for θ_3 from 0 to π, allowing θ_1 to rotate freely. Within this new domain, we found that there are none, one, or two solutions. Since the geometry of the robot corresponds to the symmetry of the unit circle, we found similar solutions for θ_3 when $-\pi \leq \theta_3 \leq 0$. Finding these solutions required a close examination of the geometry of the robot arm, and manipulation of our kinematic model.

The GMF Robotics A-510 Robot is a 'simple' robot manipulator formed by four degrees of freedom. By focusing on the first three degrees of freedom, we were able to analyze the geometry of the robot more easily and manipulate less complicated equations. But what if we were to attempt the Engineering Approach on a robot arm with six or seven degrees of freedom? Then, finding all of the possible solutions can become quite difficult. The geometry of the robot arm can be challenging, and manipulating the kinematic model to find more solutions can be very complicated.

From the Mathematician's Approach, we learned that there are at most two real solutions for (p_x, p_y). Recall, p_z determines the height of (p_x, p_y). What makes the Mathematician's Approach most appealing is that we used one algorithm to determine the set of points that would produce two solutions, the set of points that would produce one solution, and the set of points that would produce no solution. This was achieved by incorporating the geometry of the robot arm in our algebraic model. By doing so, we immediately acknowledged the proper intervals of movement between each joint. Moreover, by using simple polynomial equations, no further manipulation of our algebraic model was needed to achieve our solutions. The difference in how these solutions are represented is profound, and the amount of work that is required for each method speaks just as loudly.

What accounts for this difference is how each approach solves the inverse kinematic robotics problem. The Engineering Approach finds θ_1, θ_3, and d_2 in terms of the point (x, y), but this approach does not identify which (x, y) will guarantee a solution without plugging the point (x, y) into each equation as shown in the three case examples, where as the Mathematician's Approach finds only those points (a, b) that are reachable points within our workspace. (Note that (x, y) and (a, b) are arbitrary points.)

Because the Denavit-Hartenberg Matrix uses trigonometric functions, the solutions are bounded by certain domains corresponding to each trigonometric function. This is why we only found none, one, or two solutions when $-150° \leq \theta_1 \leq 150°$ and $-\pi \leq \theta_3 \leq 0$. After manipulating our kinematic model and analyzing the geometry of the robot arm, more solutions existed when $0 \leq \theta_1 \leq \pi$ and $-\frac{\pi}{2} \leq \theta_3 \leq \frac{\pi}{2}$. The Mathematician's Approach avoids the trouble of finding solutions within a limiting domain. By developing an algebraic model using simple polynomial equations we incorporate the entire unit circle with the assistance of trigonometric identities. There is no need to manipulate the equations in our model to find more solutions. Calculating a Groebner basis with MAGMA finds all possible solutions. In particular, by using the Algebraic Approach, given (a, b) we can easily discover the maximum number of formations to place the robot hand at the point (a, b), and more importantly, we discover all real solutions that produce these formations. Because of these precise results, it is clear why the Mathematician's Approach is the more efficient method.

In the following sections, we introduce the Jacobian Matrix and Singularities, a supplemental method to the Denavit-Hartenberg Matrix that determines singular configurations (i.e, one solution). This section is presented to demonstrate the great lengths required by the Engineering Approach to better determine and classify all solutions.

6.2 The Jacobian Matrix and Singularities

For a robot manipulator, there are two joint configurations that require further calculations in order to solve the inverse kinematic problem. These particular configurations occur when:

- joint axes are intersecting
- joint axes are parallel (collinear)

In fact, when establishing link coordinate frames, these particular configurations are exceptions to the rules outlined by Lee (See Section 4.4). We call the points in space associated with these configurations *singularities* and use the *Jacobian Matrix* to find these points.

6.2.1 The Jacobian for a Robot Manipulator

The Jacobian discusses joint velocities and accelerations, but in this comparison study, we are not interested in either of these, but are concerned with how the Jacobian can be used to determine when there is one solution.

Definition: A *Jacobian* for a robot manipulator is a matrix of differentials. This matrix describes the differential changes in the location of the end effector caused by the differential changes in joint variables.

Using Cartesian coordinates, the displacement of the end effector can be described by a differential motion vector $\vec{D} = [d_x d_y d_z \delta_x \delta_y \delta_z]$. Similar to our derivation of the Denavit-Hartenberg matrix, we arrive at $\vec{D}$ by performing certain translational and rotational transformations that describe the change in motion from the $n-1^{th}$joint to the n^{th} joint. Also, recall that the Denavit-Hartenberg matrix is a special homogeneous matrix composed of only four parameters. Now let $\vec{D_q}$ be a special differential motion vector that describes the displacement of the end effector in terms of the joint coordinates. Then, $\vec{D_q} = [dq_1 dq_2 \ldots dq_n]$, for an n-joint manipulator, where dq_n is a differential rotation and dd_n is a differential translation at joint n.

Consider a robot manipulator with four joints, then $\vec{D}$ and $\vec{D_q}$ are related by the following equation:

$$\vec{D} = \vec{J} \times \vec{D_q}.$$

Using this, we find the Jacobian.

$$\begin{bmatrix} d_x \\ d_y \\ \delta_x \\ \delta_y \end{bmatrix} = \begin{bmatrix} J_{11} & J_{12} & J_{13} & J_{14} \\ J_{21} & J_{22} & J_{23} & J_{24} \\ J_{31} & J_{32} & J_{33} & J_{34} \\ J_{41} & J_{42} & J_{43} & J_{44} \end{bmatrix} \begin{bmatrix} dq_1 \\ dq_2 \\ dq_3 \\ dq_4 \end{bmatrix}$$

Thus,

$$J = \begin{bmatrix} J_{11} & J_{12} & J_{13} & J_{14} \\ J_{21} & J_{22} & J_{23} & J_{24} \\ J_{31} & J_{32} & J_{33} & J_{34} \\ J_{41} & J_{42} & J_{43} & J_{44} \end{bmatrix}$$

is the Jacobian.

So, the Jacobian establishes the relationship between the Cartesian velocities $\vec{D}$ and the joint velocities $\vec{D_q}$. Furthermore, McKerrow,[16] highlights the following:

- $d_x = J_{11}dq_1 + J_{12}dq_2 + J_{13}dq_3 + J_{14}dq_4$. But, d_x represents the x component of the differential motion of the end effector, so from this equation, we conclude that d_x is also a function of the differential motion of the joints of the manipulator.
- $d_y = J_{21}dq_1 + J_{22}dq_2 + J_{23}dq_3 + J_{24}dq_4$. But, d_y represents the y component of the differential motion of the end effector, so from this equation, we conclude that d_y is also a function of the differential motion of the joints of the manipulator.
- $\delta_x = J_{31}dq_1 + J_{32}dq_2 + J_{33}dq_3 + J_{34}dq_4$. But δ_x is the x component of the angular motion of the end effector, so from this equation, we conclude that δ_x is a function of

the differential motion of the joints of the manipulator. Similarly, for δ_y, we conclude that δ_y is a function of the differential motion of the joints of the manipulator.

- J_{ij} is the partial derivative with respect to joint j of the x_i component of the position of the end effector. In this case, $x_1 = x$ and $x_2 = y$. So, J_{ij} is the partial derivative with respect to joint j, where $j = 1, 2, 3, 4$.

The inverse Jacobian establishes the relationship between the Cartesian velocities of the end effector and the joint velocities, and is given by,

$$\vec{D_q} = J^{-1}\vec{D}$$

Properties of the Jacobian

For the Jacobian Matrix,

1. The number of rows is determined by the number of degrees of freedom.(The joint representing the end effector is not counted.)
2. The number of columns is determined by the number of joints in the manipulator
3. In some instances, the Jacobian is not a square matrix.

6.2.2 Finding the Jacobian for the GMF Robotics A-510 Robot

We begin by calculating the Forward Kinematic solution by taking P calculated in Chapter 3, and setting $P = T - VariableMatrix$. Then, the final position of the end effector is described by (p_x, p_y, p_z). In particular,

$$\begin{bmatrix} p_x \\ p_y \\ p_z \end{bmatrix} = \begin{bmatrix} 330\cos(\theta_1+\theta_3)+410\cos\theta_1 \\ 330\sin(\theta_1+\theta_3)+410\sin\theta_1 \\ 980+d_2 \end{bmatrix}$$

Observe that p_z represents the height of the final position of the end effector. But p_x and p_y are determined by θ_1 and θ_3. If we examine the robot from a top view, by looking down on the robot, we can project each joint configuration onto the xy- plane. With this perspective, we restrict our focus to θ_1 and θ_3. So, from the matrix above, we now have the following representation.

$$\begin{bmatrix} p_x \\ p_y \end{bmatrix} = \begin{bmatrix} 330\cos(\theta_1+\theta_3)+410\cos\theta_1 \\ 330\sin(\theta_1+\theta_3)+410\sin\theta_1 \end{bmatrix}$$

To calculate the Jacobian, we differentiate, such that

$$\begin{bmatrix} dp_x \\ dp_y \end{bmatrix} = J \begin{bmatrix} d\theta_1 \\ d\theta_2 \end{bmatrix}$$

To find the matrix J, we need to find the partial derivatives,

$$\begin{bmatrix} dp_x \\ dp_y \end{bmatrix} = \begin{bmatrix} -330\sin\theta_1\cos\theta_3 - 330\cos\theta_1\sin\theta_3 - 410\sin\theta_1 & -330\cos\theta_1\sin\theta_3 - 330\sin\theta_1\cos\theta_3 \\ 330\cos\theta_1\cos\theta_3 - 330\sin\theta_1\sin\theta_3 + 410\cos\theta_1 & -330\sin\theta_1\sin\theta_3 + 330\cos\theta_1\cos\theta_3 \end{bmatrix} \begin{bmatrix} d\theta_1 \\ d\theta_2 \end{bmatrix}$$

Next, we calculate the inverse Jacobian.

$$J^{-1} = \frac{\begin{bmatrix} J_{22} & -J_{12} \\ -J_{21} & J_{11} \end{bmatrix}}{|J|}$$

where

$$J_{11} = -330 \sin\theta_1 \cos\theta_3 - 330 \cos\theta_1 \sin\theta_3 - 410 \sin\theta_1$$

$$J_{12} = -330 \cos\theta_1 \sin\theta_3 - 330 \sin\theta_1 \cos\theta_3$$

$$J_{21} = 330 \cos\theta_1 \cos\theta_3 - 330 \sin\theta_1 \sin\theta_3 + 410 \cos\theta_1$$

$$J_{22} = -330 \sin\theta_1 \sin\theta_3 + 330 \cos\theta_1 \cos\theta_3$$

Then,

$$|J| = J_{11}J_{22} - J_{12}J_{21}$$

$$= (-330 \sin\theta_1 \cos\theta_3 - 330 \cos\theta_1 \sin\theta_3 - 410 \sin\theta_1)(-330 \sin\theta_1 \sin\theta_3 + 330 \cos\theta_1 \cos\theta_3)$$

$$-(-330 \cos\theta_1 \sin\theta_3 - 330 \sin\theta_1 \cos\theta_3)(330 \cos\theta_1 \cos\theta_3 - 330 \sin\theta_1 \sin\theta_3 + 410 \cos\theta_1)$$

After several cancelations,

$$|J| = 330 * 410 \sin^2\theta_1 \sin\theta_3 + 330 * 410 \cos^2\theta_1 \sin\theta_3 = 135300 \sin\theta_3$$

So the inverse Jacobian is defined when $\sin\theta_3 \neq 0$.

In particular,

$$\begin{bmatrix} d\theta_1 \\ d\theta_2 \end{bmatrix} = \frac{1}{135300\sin\theta_3} \begin{bmatrix} J_{22} & -J_{12} \\ -J_{21} & J_{11} \end{bmatrix} \begin{bmatrix} dp_x \\ dp_y \end{bmatrix}$$

And the Jacobian for the GMF Robotics A-510 Robot has been found. In the next section, we explain the significance of the Jacobian in more detail.

6.2.3 Singularities

Definition: A *singularity* is simply a point in space where a singular configuration results.

There are two types of singularities: *a workspace internal singularity* and a *workspace boundary singularity.*

Definition: A *workspace internal singularity* occurs within the workspace,

Definition: A *workspace boundary singularity* occurs when the manipulator is fully extended to the outer boundary or fully retracted to the inner boundary of its workspace. Ref Mckerrow, [16].

In the case where joint axes are parallel, a robot manipulator has lost one or more degrees of freedom, reducing the robot's mobility in some directions. The point in space where this occurs is called a workspace internal singularity or a workspace boundary singularity. The Jacobian helps us determine these singular configurations.

6.2.4 Finding Singularities for the GMF A-510 Robot

When the determinant of the Jacobian is zero, the robot manipulator has a workspace singularity. We determine the singularities for the GMF Robotics A-510 Robot.

From the previous section, $\sin\theta_3 = 0$ when θ_3 is either 0 or π. But θ_3 ranges from $-150^\circ \leq \theta_3 \leq 150^\circ$, so π is not a point within the workspace. However, when $\theta_3 = 0$, we get a workspace boundary singularity. We substitute $\theta_3 = 0$ into J, and get,

$$\begin{bmatrix} dp_x \\ dp_y \end{bmatrix} = \begin{bmatrix} -330\sin\theta_1 - 410\sin\theta_1 & -330\sin\theta_1 \\ 330\cos\theta_1 + 410\cos\theta_1 & 330\cos\theta_1 \end{bmatrix} \begin{bmatrix} d\theta_1 \\ d\theta_2 \end{bmatrix}$$

which reduces to,

$$\begin{bmatrix} dp_x \\ dp_y \end{bmatrix} = \begin{bmatrix} -740\sin\theta_1 & -330\sin\theta_1 \\ 740\cos\theta_1 & 330\cos\theta_1 \end{bmatrix} \begin{bmatrix} d\theta_1 \\ d\theta_2 \end{bmatrix}$$

Observe that the two column vectors in the above Jacobian matrix are parallel. Thus, only those points on the boundary, that is, those points on the circle of radius $r = 740$ centered at the origin, where θ_1 can rotate from $-150^\circ \leq \theta_1 \leq 150^\circ$, are workspace boundary singularities. Moreover, it is at these points where the inverse Jacobian is undefined. So, there is one solution (i.e., one configuration) that can reach these points.

6.3 A Procedure for an Algebraic Model for Robot Manipulators

In this section, we look to extend our results to an entire class of robot manipulators that satisfy the following conditions:

- The robot manipulator has n-degrees of freedom,
- The robot manipulator is a kinematic chain consisting of revolute or prismatic joints,
- The joints of the robot manipulator may be collinear or intersecting, but neither is required.

Each robot manipulator is different. Thus, a different algebraic model must be formed. The challenge of the Mathematician's Approach is deriving an algebraic model. The algebraic model must describe the behavior of the robot such that the relationship between neighboring joints is respected. The easiest way to achieve this is to map the movements of each joint onto a 2D-coordinate plane, then use trigonometric functions and identities to imitate each joint movement, as well as, the resulting movements that occur from related joints.

Recall the algebraic model used for the GMF Robotics A-510 Robot. Observe that the there are three joints, a prismatic joint, and two revolute joints. This prismatic joint simply moves the other two joints in an up and down motion. Because these two revolute joints are consecutive, they move up and down together. See the figure 6.1.

This picture is a side view of the GMF Robotics A-510 robot. Now consider a top/down view of the robot. With this view, we restrict our focus to each revolute joint. If we project this perspective onto the xy-plane, we can identify the movements of each joint. The point $(e, f) = (410 \cos\theta_1, 410 \sin\theta_1)$ describes the movement of revolute joint 1. The second revolute joint, however, not only describes the movement of joint 2, but also acknowledges those

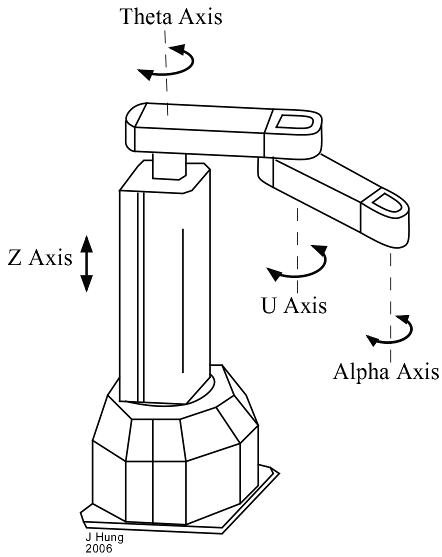

Figure 6.1: GMF A-510 Robot, ref. Hung [8]

movements of joint 2 that are influenced by the movements of joint 1. The end effector is represented by the point $(a, b) = (330\cos(\theta_1 + \theta_3) + 410\cos\theta_1, 330\sin(\theta_1 + \theta_3) + 410\sin\theta_1)$, which encompasses the movements at both joints. Using trigonometry, we are able to represent the position of the end effector with simple polynomial equations.

In order to proceed with the Mathematician's Approach, the equations in the algebraic model must be polynomial equations. Once these equations have been established, we use MAGMA to find a Groebner basis for these equations. By analyzing the Groebner basis, we are able to determine each unknown variable, as well as those points that can be reached by the end effector.

Given a robot manipulator that satisfies the conditions above, we outline the following procedure to derive an algebraic model:

1. Analyze the geometry of the robot manipulator

2. Isolate particular joint behavior and project the joint movements onto a coordinate system

3. Represent the joint movements using polynomial equations.

4. Verify that these polynomial equations preserve the movement of each joint and the relationship between neighboring joints.

By following these steps, we arrive at a system of equations, Hence, for a particular class of robot manipulators, we have provided a systematic procedure for deriving an algebraic model, and with MAGMA are able to apply Groebner Basis Theory to solve the inverse kinematic robotics problem.

Chapter 7

Conclusion

In this final chapter, we discuss the difficulties and challenges associated with the Engineering Approach, and with each difficulty, illustrate how Groebner Basis Theory solves the inverse kinematic robotics problem without these difficulties. Guez and Ahmad [5] summarize the following difficulties for an inverse kinematic solution:

1. To form the A_{ij} matrices, we need to know the forward kinematics of the manipulator.

2. There are situations where the Denavit-Hartenberg matrix will not produce equations that will find all possible solutions. Thus, an alternate method must be used to find all solutions.

3. Every robot manipulator has solutions specific to that manipulator. Thus, new algorithms and procedures have to be developed to accommodate each manipulator.

In order to use Groebner Basis Theory, we must create an algebraic model describing the behavior of the robot manipulator. There is no need to form the A_{ij} matrices. So, we address how Groebner Basis Theory can resolve the last two difficulties. However, if the previous conditions for deriving an algebraic model (in Section 6.3) are satisfied, we suggest using the Mathematician's Approach when such A_{ij} matrices are not easily formed.

Beginning with the second difficulty, there are several situations when the Denavit-Hartenberg matrix will not produce equations that will find all possible solutions. We revealed one situation in our calculations for the GMF Robotics A-510 robot. By using the

Denavit-Hartenberg matrix, we solved a quadratic equation finding two solutions(configurations) for θ_1 and θ_3 within a given domain. However, we did not find all solutions. By manipulating the kinematic model, we found more solutions. In addition, by testing three random points within the robot's workspace, we saw the possibility of no solutions or one solution. But we were uncertain of when these particular solutions would occur? Hence, the need for an alternate method called the Jacobian that could accurately determine when there would be one solution. Using this supplementary method, we found that when $L1 = 410$ and $L3 = 330$ are collinear, there would be one solution for the GMF Robotics A-510 robot. In addition, we also wish to determine when there are no solutions. Clearly, there are no solutions for those points that are outside of the reachable workspace.

The Mathematician's Approach finds all of those points that can be reached within the workspace. In fact, by calculating a Groebner basis, we are able to determine which points within the workspace have two solutions (two configurations), which points have one solution (one configuration), and which points have no solution. Through the Mathematician's Approach, we used one method that uses one algorithm. There is no need to use a supplementary method or second algorithm. Also, as a result, the number of calculations required to solve the inverse kinematic robotics problem is greatly reduced. Horn, [6], acknowledges those difficulties associated with finding all possible solutions by use of the Denavit-Hartenberg matrix, arguing that the solutions to the inverse kinematic robotics problem should be "simple to determine and easy to use." Because the Mathematician's Approach accomplishes both desires, we conclude that the Mathematician's Approach is a better method for producing equations that will find all possible solutions.

Lastly, in other situations, the Denavit-Hartenberg matrix must be coupled with other methods or algorithms to find all possible solutions. These methods and algorithms include, but are not limited to neural networking techniques, iterative algorithms, visual servoing, euler parameters, etc. In the past, each robot manipulator required its own particular method given the above mentioned difficulties associated with the Denavit-Hartenberg matrix. With the introduction of the Groebner Basis Theory, we provide an alternate method that may be used on an entire class of robot manipulators satisfying the conditions in Section 6.3. For this flexibility, we see that the Mathematician's Approach is an improved alternate. With further research we look to show that the Mathematician's Approach can be used for other classes of manipulators.

With the conclusions found in this paper, we have demonstrated how the Mathematician's Approach of Groebner Basis Theory is an alternate method to solving the inverse kinematic robotics problem. Moreover, by using Groebner Basis Theory, we used one procedure, performed less calculations, found real solutions, and established when certain solutions (none, one, two, etc.) would occur. Because of these advantages, we see how Groebner Basis Theory is more beneficial than the Denavit-Hartenberg Matrix. Also, because of these advantages, I wish to expand the applications of Groebner Basis Theory to many other classes of robot manipulators, solving the inverse kinematic robotics problem for larger classes of robot manipulators.

Bibliography

[1] William Adams, Philippe Loustaunau,"'An Introduction to Groebner bases"' American Mathematical Society, Vol.3, Providence, Rhode Island, 1994.

[2] Bruno Buchberger, "'Groebner Bases: An Algorithmic Method in Polynomial Ideal Theory,"' Multidimensional Systems Theory, Reidel Publishing Company, Dodrecht, 1985.

[3] David Cox, John Little, Donal O'Shea, "Ideals, Varieties, and Logarithms: An Introduction to Computational Algebraic Geometry and Commutative Algebra," Springer-Verlag New York Inc., 1992.

[4] J. Denavit, R. S. Hartenberg, "'A Kinematic Notation for Lower-Pair Mechanisms Based on Matrices,"' Journal of Applied Mechanics, 1955.

[5] Allon Guez, Ziauddin Ahmad,"'Solution to the Inverse Kinematic Problem in Robotics by Neural Network,"'Second International Conference of Neural Networks, March 1988.

[6] Berthold K.P. Horn, "'New Notation for Serial Kinematic Chains,"' May 1987.

[7] B. Huang, V. Milenkovic, "'Kinematics of Major Robot Linkages,"' Proceedings of the 13th International Symposium on Industrial Robots,"' April 17-19, Chicago, IL.

[8] John Hung, "Kinematic Model for the GMF Robotics A510 Robot," Department of Electrical and Computer Engineering, Auburn University, Auburn, Alabama, 2007.

[9] Kimberly D. Kendricks, "Using Groebner Bases to Solve the Inverse Kinematic Robotics Problem for a Planar Robot Arm," Master's Project, Department of Mathematics, Auburn University, Auburn, Alabama 2006.

[10] Richard D. Klafter, Thomas A. Chmielewski, Michael Negin, "Robotic Engineering An Integrated Approach," Prentice Hall, Englewood Cliffs, New Jersey, 1989.

[11] Yoram Koren, "'Robotics for Engineers"' McGraw-Hill Book Company, New York, New York, 1985.

[12] Antti J. Koivo, "'Fundamentals for Control of Robotic Manipulators,"' John Wiley and Sons, Inc., New York, 1989.

[13] C.S. George Lee, C.G. Liang, "'Displacement Analysis of the General Spatial 7-Link, 7-R Mechanism,"' Journal of Mechanisms and Machine Theory, Vol.23, Issue 3, 1988.

[14] C.S. George Lee, "'Robot Arm Kinematics, Dynamics, and Control,"' Tutorial of Robotics, University of Michigan, Ann Arbor, 1982.

[15] Rachid Manseur, Keith L. Doty, "'A Fast Algorithm for Inverse Kinematic Analysis of Robot Manipulators,"' The International Journal of Robotics Research, Vol. 7, No.3, Massachusetts Institute of Technology, 1988.

[16] Phillip John McKerrow, "'Introduction to Robotics,"'Addison-Wesley Publishing Company, New York, 1991.

[17] R. P. Paul, "'Robot Manipulators: Mathematics, Programming, and Control,"' M.I.T. Press, 1981.

[18] R. P. Paul, Bruce Shimano, Gordon E. Mayer, "' Kinematic Control Equations for Simple Manipulators,"' Tutorial on Robotics, IEEE Computer Society Press, Washington, DC, 1981.

[19] D. L. Pieper, "'The Kinematics of Manipulators Under Computer Control,"' Artificial Intelligence Project Memo No.72, Computer Science Department, Stanford University, 1968.

[20] L. Robbiano, "Groebner Bases and Statistics," Groebner Bases and Applications, Cambridge University Press, 1998.

[21] L.W. Tsai, A.P. Morgan, "'Solving the Inverse Kinematics of the Most General Six, and Five-Degree-of-Freedom Manipulators by Continuation Methods,"' American Society for Mechanical Engineers, Design Engineering Techinical Conference, Cambridge, Mass., 1984.

[22] D. Wang, "Groebner Bases Applied to Geometric Theorem Proving and Discovering," Groebner Bases and Applications, Cambridge University Press, 1998.

[23] T.C. Yih, Y.Youm, "'Geometrical Modeling of Lower-Pairs Based on Spherical-Euler Geometry,"' Trends and Developments in Mechanisms, Machines, and Robotics, Vol. 15-1, 1988.

[24] GMF Robotics A-510 Robot Descriptions Manual

Appendix

Matrix Calculations:

$$A_{12} \times A_{23} = \begin{bmatrix} 1 & 0 & 0 & 410 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & d_2 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \cos\theta_3 & -\sin\theta_3 & 0 & 330\cos\theta_3 \\ \sin\theta_3 & \cos\theta_3 & 0 & 330\sin\theta_3 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$= \begin{bmatrix} \cos\theta_3 & -\sin\theta_3 & 0 & 330\cos\theta_3 + 410 \\ \sin\theta_3 & \cos\theta_3 & 0 & 330\sin\theta_3 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$A_{01} \times A_{01}^{-1} = \begin{bmatrix} \cos\theta_1 & -\sin\theta_1 & 0 & 0 \\ \sin\theta_1 & \cos\theta_1 & 0 & 0 \\ 0 & 0 & 1 & 980 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \cos\theta_1 & \sin\theta_1 & 0 & 0 \\ -\sin\theta_1 & \cos\theta_1 & 0 & 0 \\ 0 & 0 & 1 & -980 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$= \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$