Test and Evaluation: Robot Forward Kinematics

Timothy Daly*and Rick Linger†

February 23, 2010

Abstract

In order to investigate the use of an embedded processor in practical terms we built a robot driven by a microprocessor. The robot task gave focus to the use of Function Extraction technology on the embedded processor.

This paper describes the development of the forward kinematics of the robot used for the Test and Evaluation project. The forward kinematics describes the space relationship between the base of the robot and the orientation of the hand.

The forward kinematics of the robot are described by special purpose matrices which motivates the analysis of matrix code by Function Extraction. We have used this experiment to modify the FX software with an eye toward extending our ability to generate human readable forms in FX output, in this case, matrices.

*daly@cert.org
†rlinger@sei.cmu.edu

Contents

1 Overview of the robot

1.1 The robot arm

The robot is a 5 axis arm with a gripper assembly. It is driven by a general purpose development board containing a PIC16 processor.

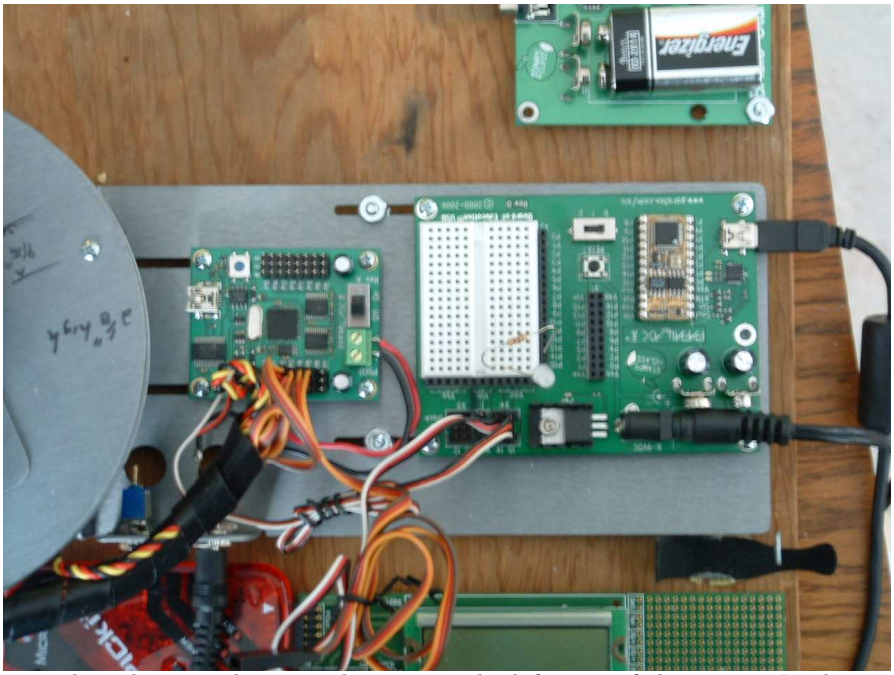
The main programming for the PIC processors is done on a connected computer. Programs are dumped over a serial line and stored on the chip.

The development environment is the Microchip MPLAB software. This software is used for all of the development efforts in this project.

Additionally we purchased a PIC 18 processor from Microchip and a PIC 18 simulator IDE from OshonSoft. This allows us to do assembler language programs.

1.2 The processor and servo boards

This is an image of the processor control board and the servo control board.



The robot arm base can be seen in the left part of the image. In the center is a small servo control board. To the right is a larger development board. The PIC 16 processor is the small black chip on the daughter board in just between the slide switch and the USB connector.

The servo control board handles 16 channels of servo control. Commands to the control board select the channel, each of which is connected to an individual servo motor. The command specifies the angle of rotation of the servos, varying from 0° to 180°.

The servo motors respond to a pulse-width modulation varying from 1ms, corresponding to 0° to 2ms, corresponding to 180°, which is managed by the board.

Commands to the servo board are passed through a serial link from the processor board.

The processor board contains a PIC 16 processor, a widely used and very popular microcontroller. It has 35 8-bit instructions. The processor is a harvard architecture, which means that the program memory and data memory spaces

are separate. This PIC 16 has Basic burned into the chip making the robot control somewhat easier. The PIC-Basic interpreter which has special instructions to drive the servo control hardware.

2 How to construct a robot description

Robot motion can be described by coordinate matrices.[1, 3]

These matrices describe the position and orientation, sometimes called the POSE, of one section of the robot with respect to another section. The position and orientation of one part of the robot is defined with respect to another part of the robot. For instance, the position of the arm is defined with respect to the shoulder. Each portion of the robot has a matrix associated with it that defines the X, Y, Z, Roll, Pitch, and Yaw. These matrices are called Frames.

In theory you can associate a frame with any part of a link of a robot. If you choose to associate a frame with the center of mass of a link, say a forearm, then that frame can be rotated and translated 6 different ways, changing the X, Y, Z, Roll, Pitch, and Yaw with respect to the shoulder. These two frames are said to have 6 degrees of freedom.

However, if we know that there is a simple joint, say a rotating joint, that connects the two robot parts then we know that certain motions cannot occur.

We also know that if we associate frames with the centers of rotation or translation that the description of motion is greatly simplified. Instead of changing many different things at once during a rotation we only need to consider the change in angle (for rotary motion) or length (for sliding motion).

Joints that rotate are called *revolute* joints. Joints that slide are called *prismatic* joints.

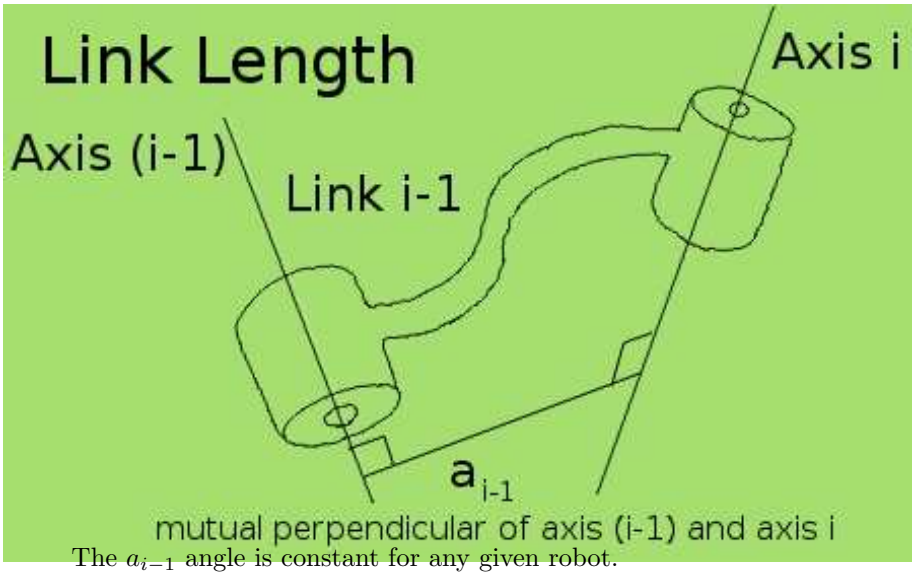
Richard Paul's [3] Ph.D. thesis introduces these optimizations. He introduces a special kind of matrix, called a Denavit-Hartenberg[1] matrix, which describes the relationship of links in a robot arm. These are called D-H Matrices. They generate a minimal description of the relationship between joints. We would like this minimal description to simplify the mathematics.

Certain joint parameters are constant, such as the length, offset, or tilt of one axis from another.

We follow the Stanford description[2], given in the open course lecture series, which differs slightly from Paul's derivation but is more common. The results are equivalent.

2.1 Defining the link length a_{i-1}

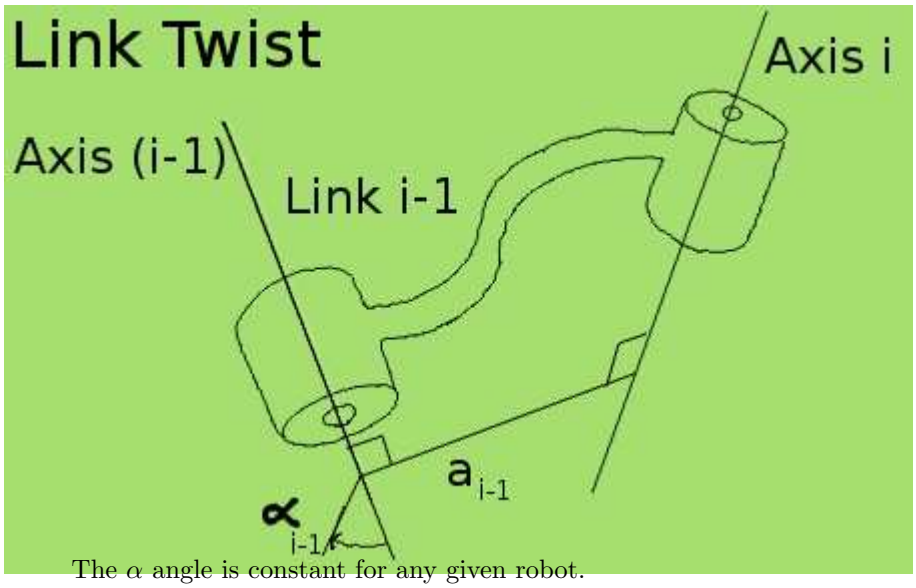
There is a direction which describes the center of each axis. We need to describe the offset distance. To do this we find the common perpendicular between axis $i-1$ (say, the shoulder joint) and axis i (say, the arm). We call this a_{i-1}.



The a_{i-1} angle is constant for any given robot.

2.2 Defining the link twist α_{i-1}

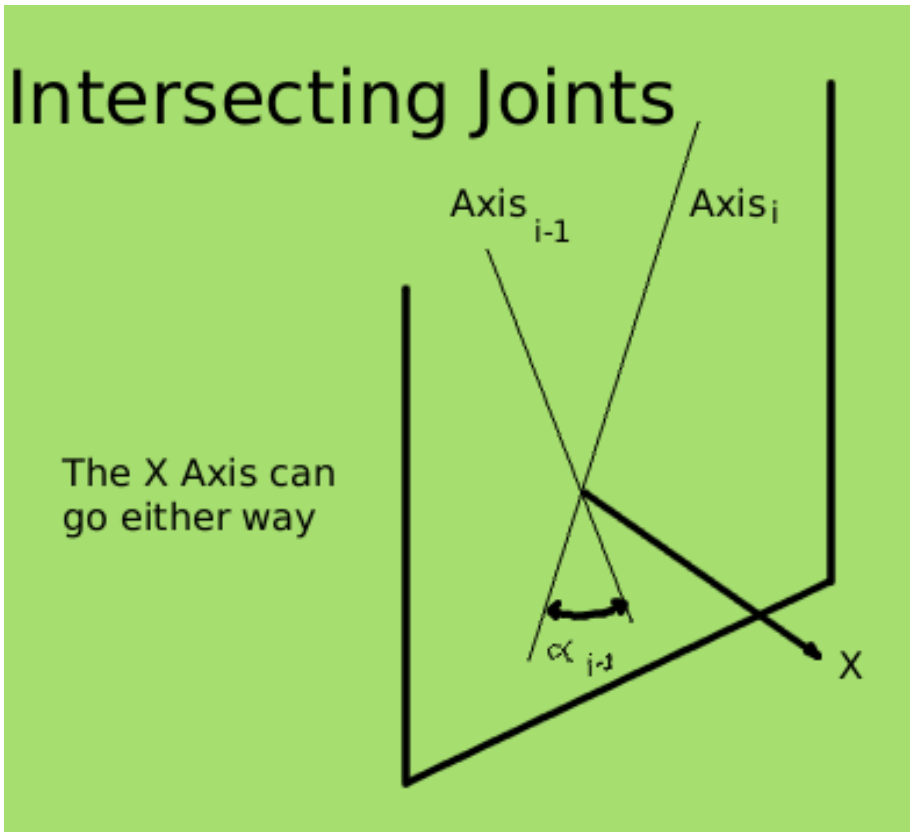
If we slide the axis i along the vector a_{i-1} until the axis $i-1$ and axis i intersect we find there is an angle between them. This angle is defined as the link twist α_{i-1}. This angle is measured in a right-hand sense about the vector a_{i-1}.



The α angle is constant for any given robot.

Note that if the two axes intersect they have no common perpendicular. In this case we have two choices for a_{i-1}, along the cross product of the two axis

vectors $i-1$ and i. We generally choose the direction that points toward the robot hand.



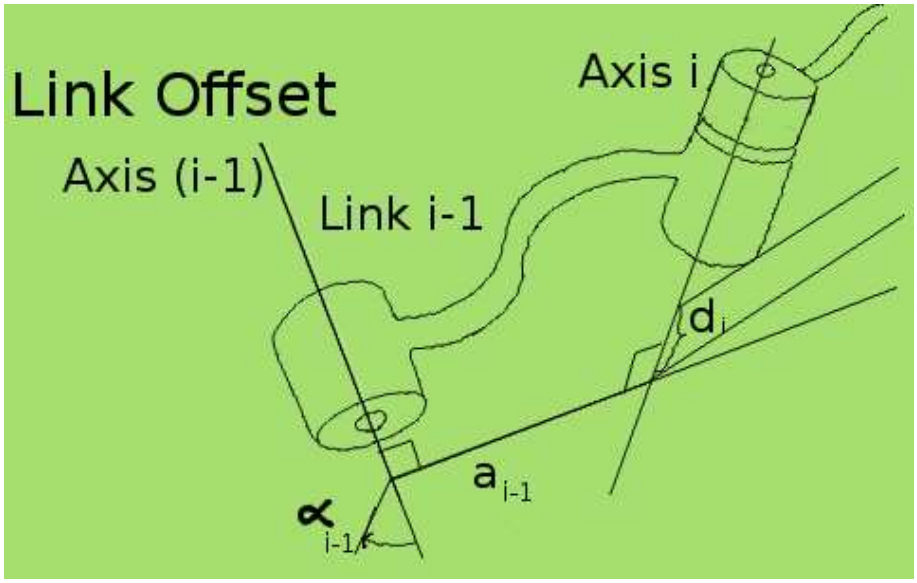
2.3 Defining the link offset

Now that we have defined the relationship between the prior link (e.g. the shoulder) and the current link (e.g. the arm) we need to describe the action of the joint when it moves or slides.

If we move to the relationship between axis i and $i+1$ we again find that we have the two paramaters, link length, this time as a_i and link twist, this time as α_i.

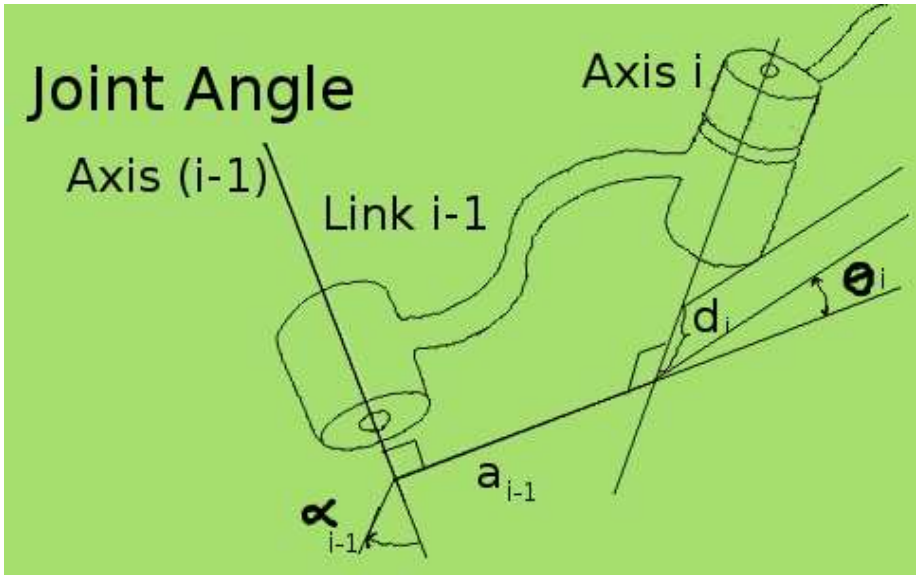
There is a point where the vector a_{i-1} intersects the axis i, call this point M. There is a point where the vector a_i intersects the axis i. call this point N. From these two points we get two more parameters.

The distances between M and N is called the link offset, d_i. This is constant for revolute joints. It is the axis along which a prismatic joint will vary.



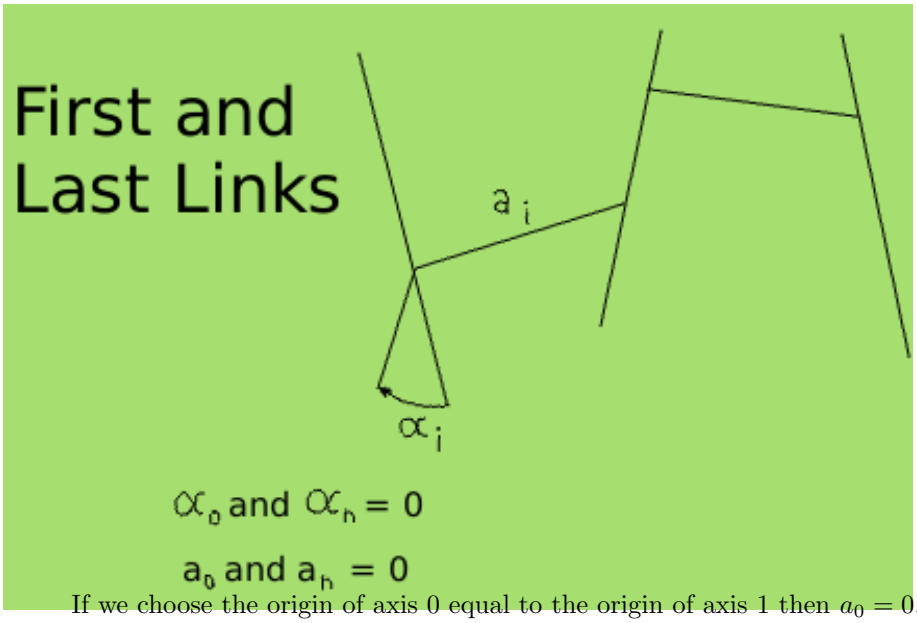
2.4 Defining the joint angle

The angle between a_{i-1} and a_i is called the joint angle, θ_i. It is variable for revolute joints and fixed for prismatic joints.



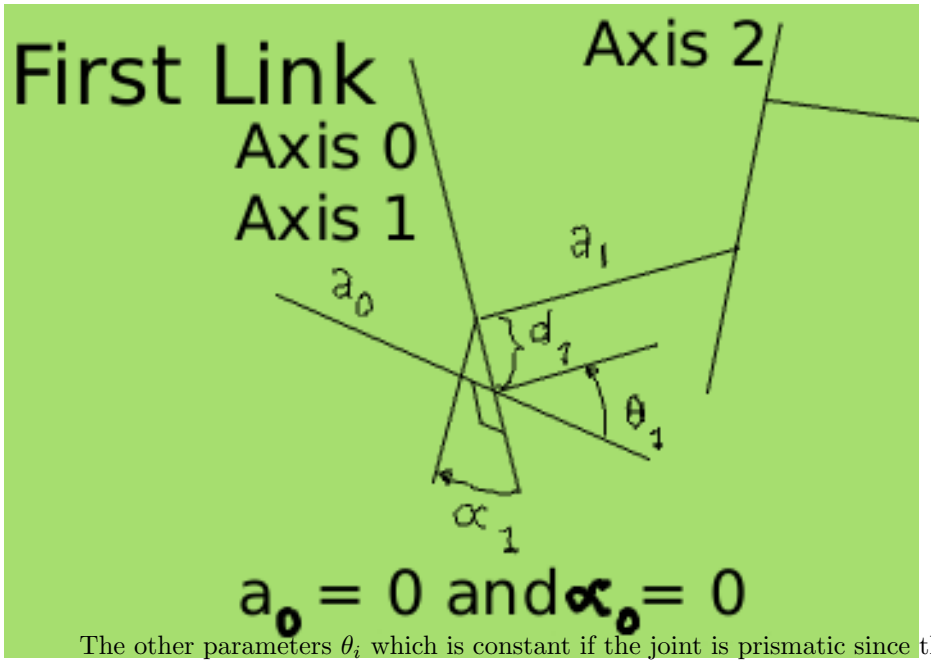
2.5 First frame assignments

The first and last frame values are underdetermined. There are many possible choices of origin and axes. Since we are trying to minimize the required mathemetics we choose values these frames for simplicity. This happens when we have the maximum number of 0 parameters.



If we choose the origin of axis 0 equal to the origin of axis 1 then $a_0 = 0$.

If we choose the X, Y, and Z axes so that they correspond to those of the first joint then $\alpha_0 = 0$.



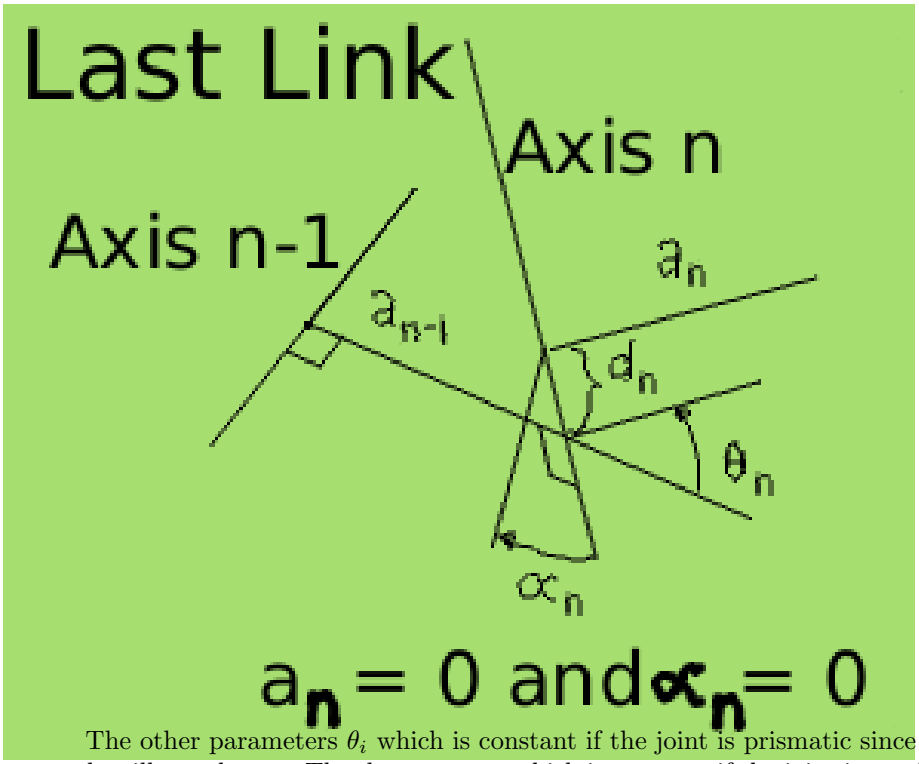
The other parameters θ_i which is constant if the joint is prismatic since the angle will not change. The d_i parameter which is constant if the joint is revolute since the length will not change.

2.6 Last frame assignments

We follow the same idea with the last frame, again choosing to minimize the mathematics.

If we choose the origin of axis n equal to the origin of axis $n-1$ then $a_n = 0$.

If we choose the X, Y, and Z axes so that they correspond to those of the last joint then $\alpha_n = 0$.



The other parameters θ_i which is constant if the joint is prismatic since the angle will not change. The d_i parameter which is constant if the joint is revolute since the length will not change.

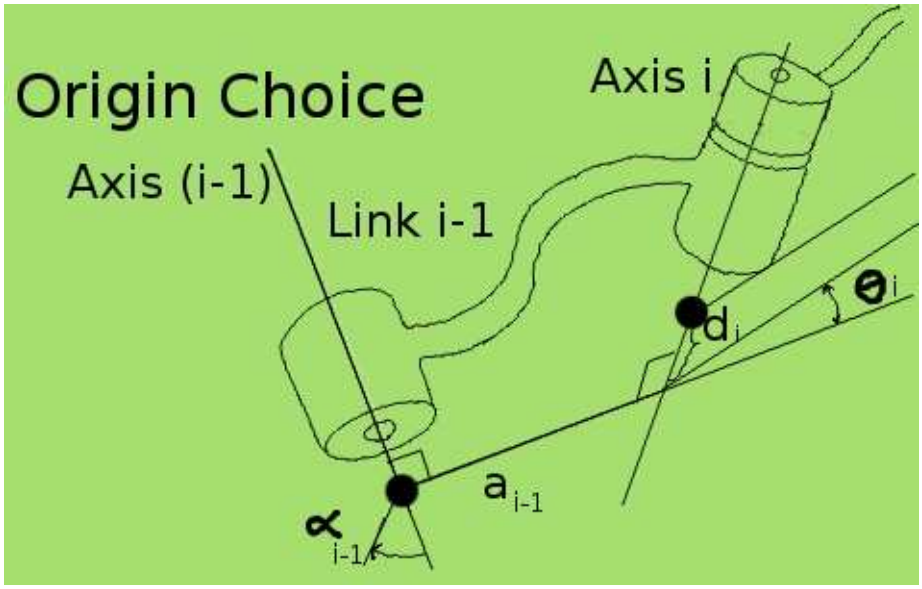
2.7 The Denavit-Hartenberg Parameters

- There are 4 DH parameters, $(\alpha_i, a_i, d_i, \theta_i)$
- revolute joints have θ_i variable
- d_i is variable for prismatic joints.
- For any choice, 3 of the 4 parameters are constant.
- α_i and a_i describe the link i structure
- d_i and θ_i describe the link connection

2.8 Selecting the origin

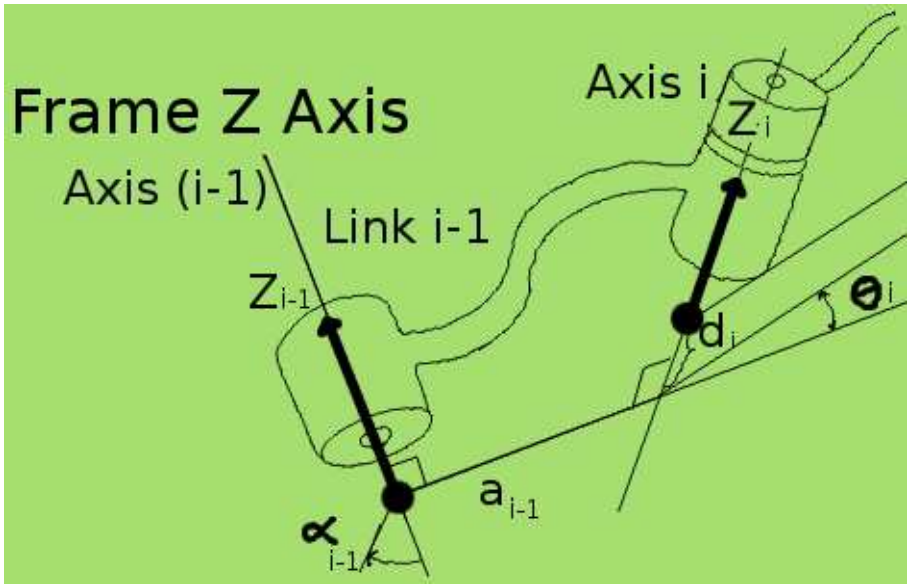
We want our frames to be aligned with the DH parameters.

The origin is set to the a_{i-1} intersection along the common normal. The next axis origin will be set at the a_i intersection along the common normal with axis $i + 1$.



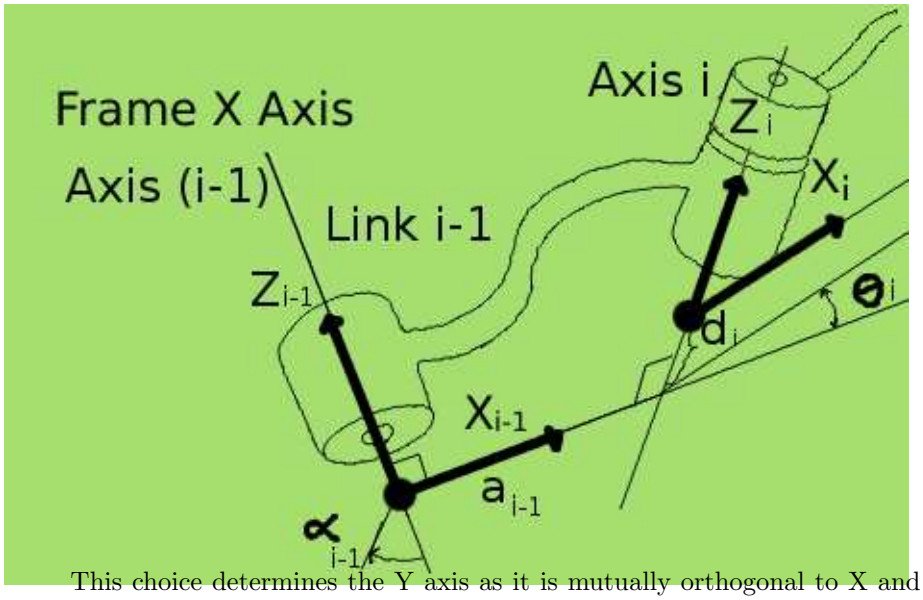
2.9 Selecting the Z axes

To do this we choose Z of each frame to be along the axis of the joint. This makes sure that the rotation or transformation is measured along the joint.



2.10 Selecting the X axes

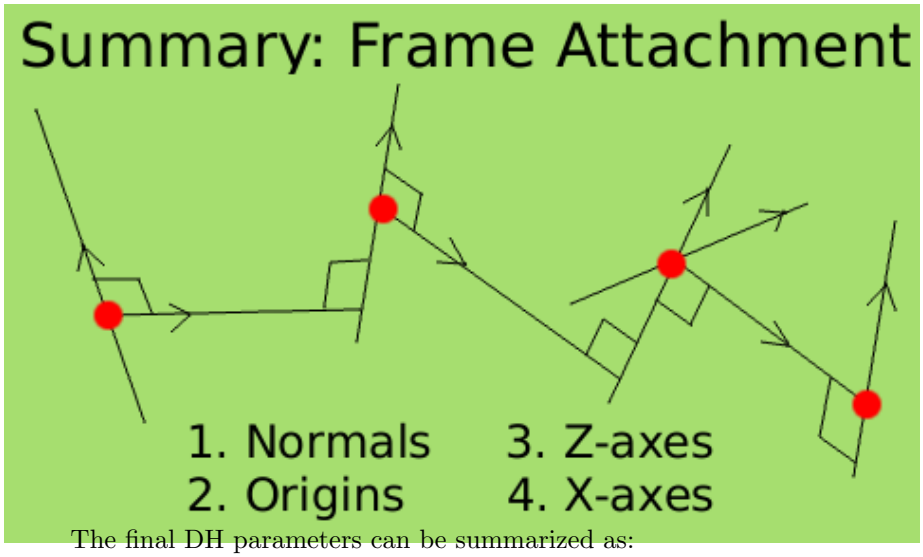
The X axis is the only choice left and we set it along the common normal to the joint axes. This is perpendicular to Z.



This choice determines the Y axis as it is mutually orthogonal to X and Z and falls along the positive direction of right-hand rule rotation.

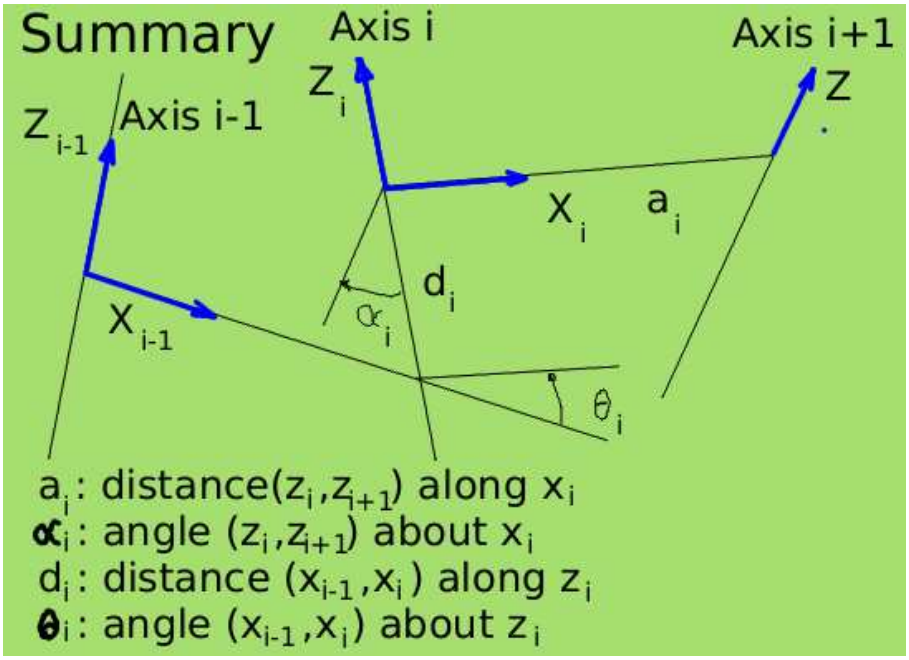
2.11 Frame creation summary

- choose normals to joint axis
- compute $a_i, \alpha_i, d_i, \theta_i$
- choose origins at normal intersections
- choose Z axis along joint axis
- define X axis along a_i
- define Y axis using right-hand rule



The final DH parameters can be summarized as:

- a_i is the distance (Z_i, Z_{i+1}) along X_i
- α_i is the angle (Z_i, Z_{i+1}) about X_i
- d_i is the distance (X_{i-1}, X_i) along Z_i
- θ_i is the angle (X_{i-1}, X_i) about Z_i

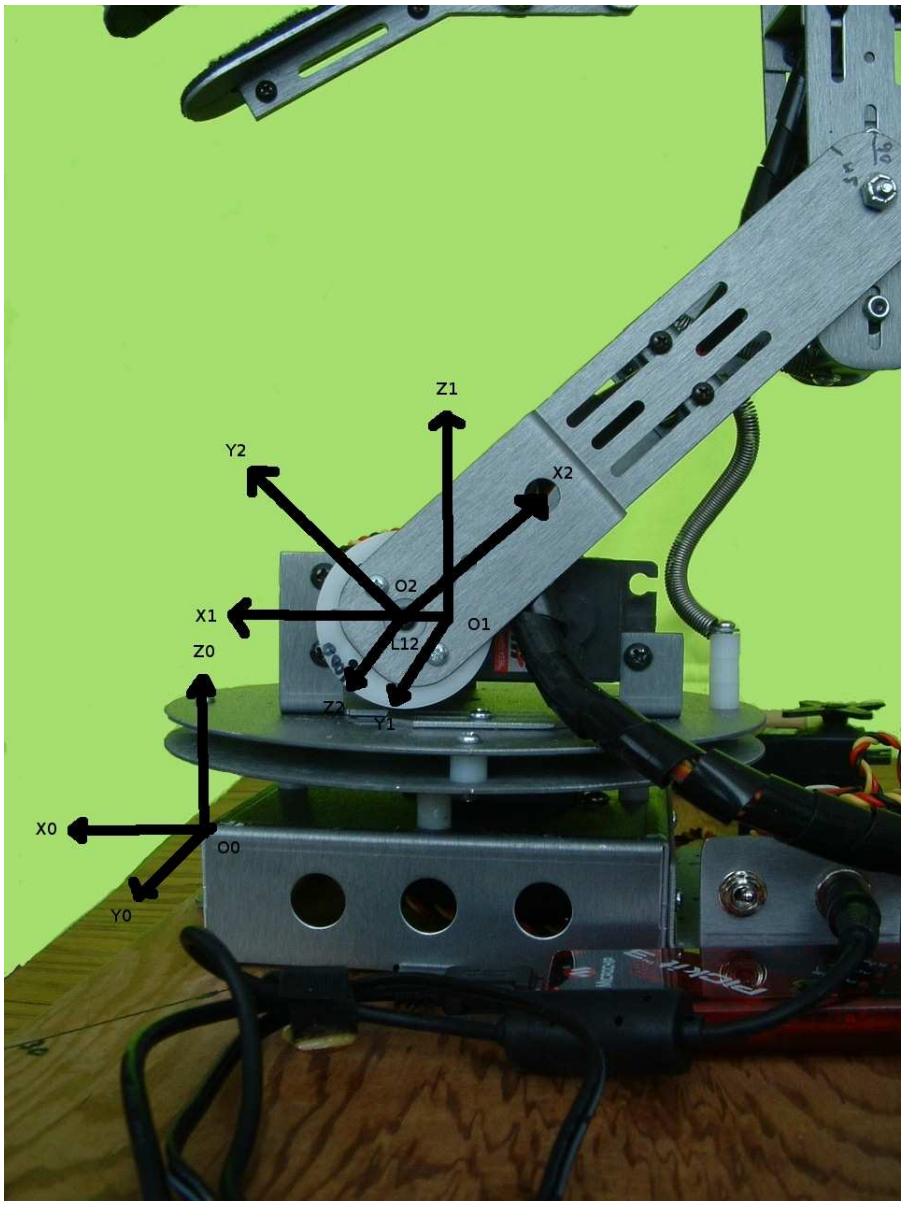


There are 2 distances (a, d) and 2 angles (α, θ). The distance a measures distance between the Z axes. The distance d measures distance between the X axes.

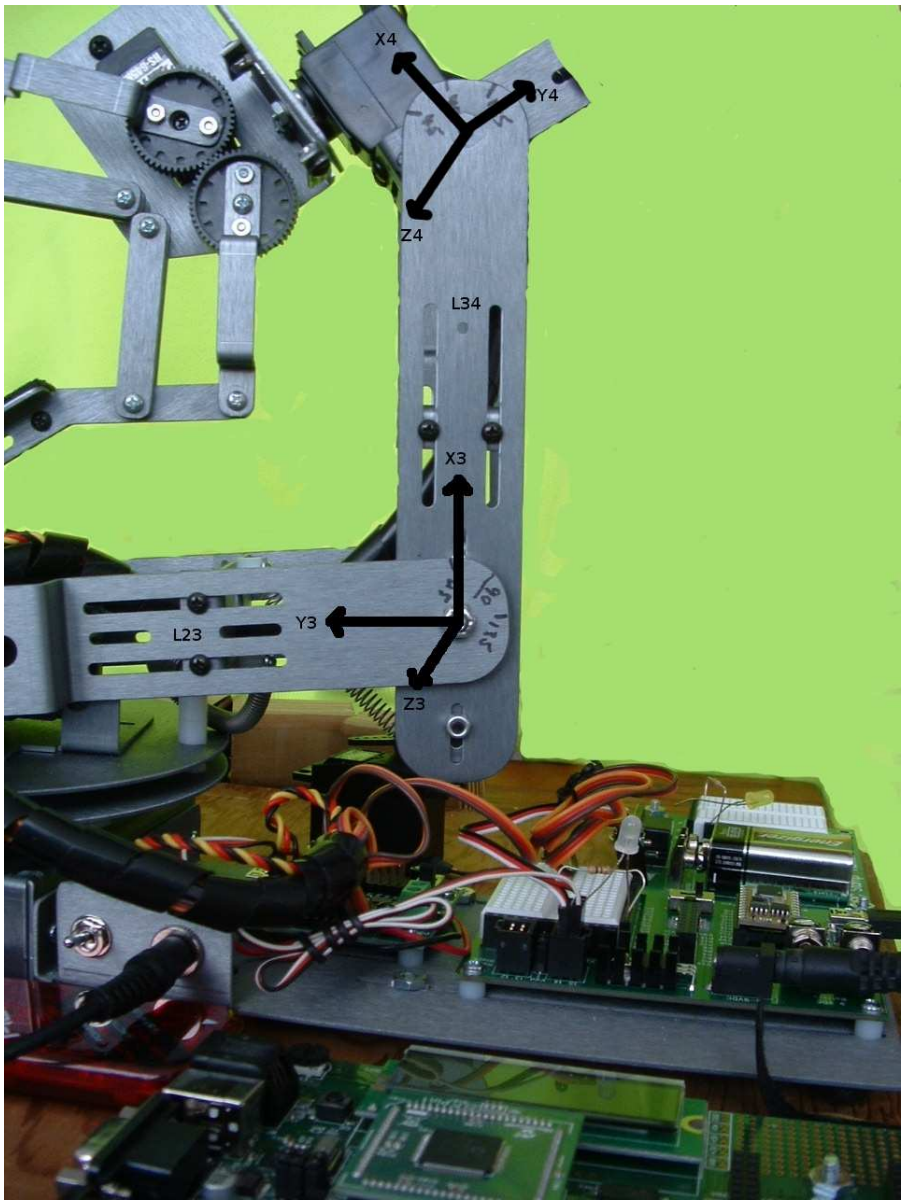
The angle α measures the angle between the Z axes. The angle θ measures the angle between the X axes.

3 Test and Evaluation Robot

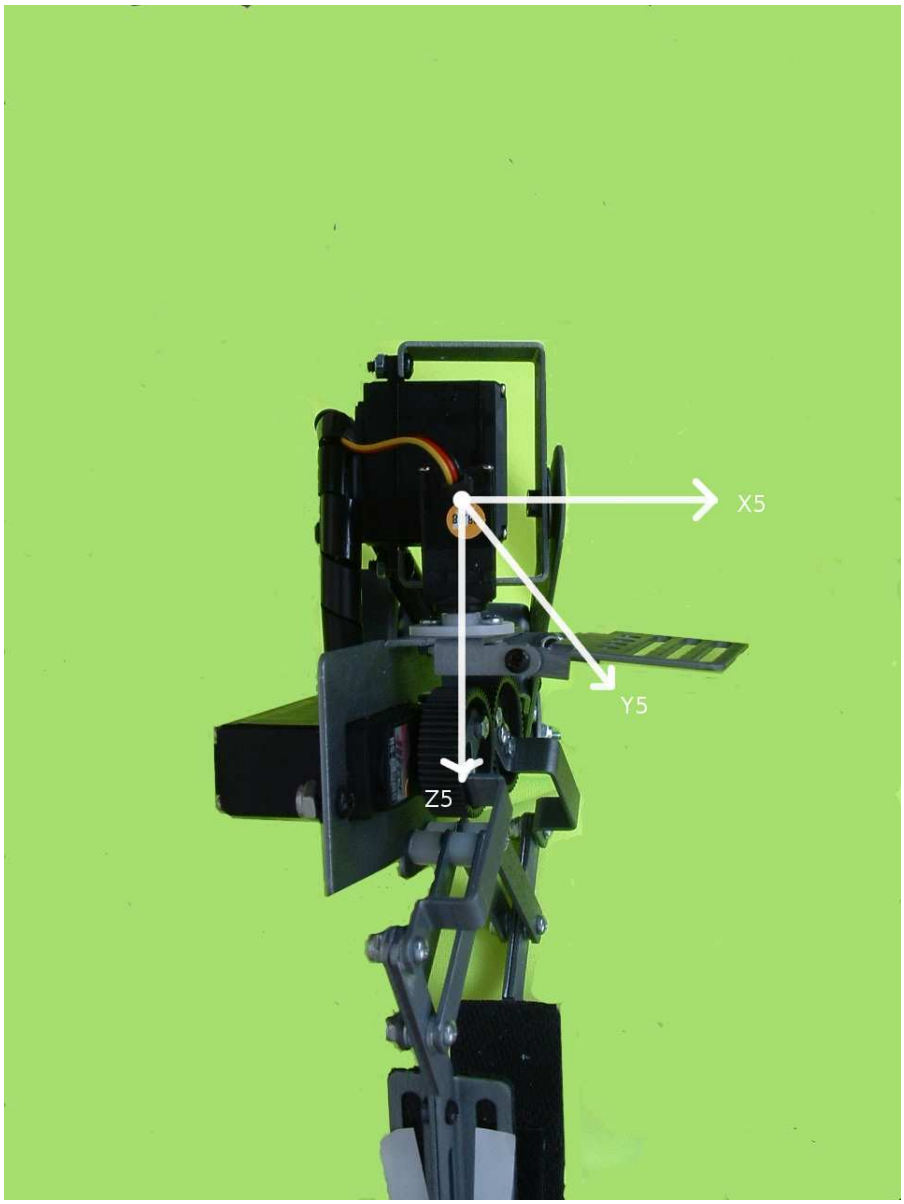
3.1 Choosing joint axes 0, 1, and 2



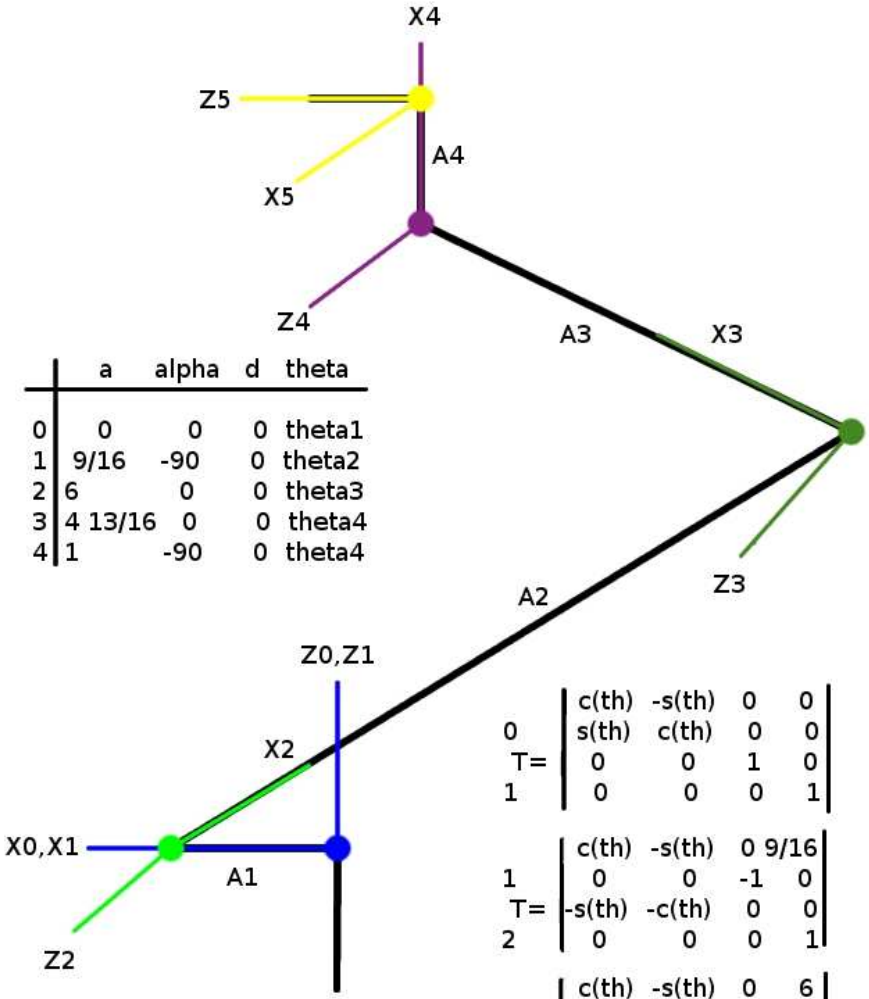
3.2 Choosing joint axes 3 and 4



3.3 Choosing joint axis 5



3.4 Constructing the skeleton diagram



	a	alpha	d	theta
0	0	0	0	theta1
1	9/16	-90	0	theta2
2	6	0	0	theta3
3	4 13/16	0	0	theta4
4	1	-90	0	theta4

$$ {}^{0}_{1}T = \begin{vmatrix} c(th) & -s(th) & 0 & 0 \\ s(th) & c(th) & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{vmatrix} $$

$$ {}^{1}_{2}T = \begin{vmatrix} c(th) & -s(th) & 0 & 9/16 \\ 0 & 0 & -1 & 0 \\ -s(th) & -c(th) & 0 & 0 \\ 0 & 0 & 0 & 1 \end{vmatrix} $$

$$ {}^{2}_{3}T = \begin{vmatrix} c(th) & -s(th) & 0 & 6 \\ s(th) & c(th) & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{vmatrix} $$

$$ {}^{3}_{4}T = \begin{vmatrix} c(th) & -s(th) & 0 & 77/16 \\ s(th) & c(th) & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{vmatrix} $$

$$ {}^{4}_{5}T = \begin{vmatrix} c(th) & -s(th) & 0 & 1 \\ 0 & 0 & -1 & 0 \\ -s(th) & -c(th) & 0 & 0 \\ 0 & 0 & 0 & 1 \end{vmatrix} $$

3.5 Constructing the DH matrix parameter table

origin	a_i	α_i	d_i	θ_i
0	0	0	0	θ_1
1	$\frac{9}{16}$ ¨	-90	0	θ_2
2	6¨	0	0	θ_3
3	$\frac{77}{16}$ ¨	0	0	θ_4
4	1¨	-90	0	θ_5

The process of constructing the DH matrices involves 4 steps:

1. rotate α around X
2. translate a along X
3. rotate θ around Z
4. translate d along Z

These rotation and translation operations are independent. We can form a matrix which represents each primitive rotation or translation and then multiply these matrices together to construct the DH matrices we use as frames.

The general form of a rotation around the X by α degrees is

$$rotate(\alpha) = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & cos(\alpha) & -sin(\alpha) & 0 \\ 0 & sin(\alpha) & cos(\alpha) & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

The general form of a translation in x, y, and z is

$$translate(x, y, z) = \begin{bmatrix} 1 & 0 & 0 & x \\ 0 & 1 & 0 & y \\ 0 & 0 & 1 & z \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

The general form of a rotation around Z by θ degrees is:

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

3.6 Constructing the robot frames

Thus

$${}^{0}_{1}T = rotatex(0) * translate(0, 0, 0) * rotatez(\theta) * translate(0, 0, 0)$$

with the resulting matrix:

$$ {}^{0}_{1}T = \begin{bmatrix} cos(\theta) & -sin(\theta) & 0 & 0 \\ sin(\theta) & cos(\theta) & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} $$

$${}^{1}_{2}T = rotatex(-90) * translate(9/16, 0, 0) * rotatez(\theta) * translate(0, 0, 0)$$

with the resulting matrix:

$$ {}^{1}_{2}T = \begin{bmatrix} cos(\theta) & -sin(\theta) & 0 & \frac{9}{16} \\ 0 & 0 & -1 & 0 \\ -sin(\theta) & -cos(\theta) & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} $$

$${}^{2}_{3}T = rotatex(0) * translate(6, 0, 0) * rotatez(\theta) * translate(0, 0, 0)$$

with the resulting matrix:

$$ {}^{2}_{3}T = \begin{bmatrix} cos(\theta) & -sin(\theta) & 0 & 6 \\ sin(\theta) & cos(\theta) & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} $$

$${}^{3}_{4}T = rotatex(0) * translate(77/16, 0, 0) * rotatez(\theta) * translate(0, 0, 0)$$

with the resulting matrix:

$$ {}^{3}_{4}T = \begin{bmatrix} cos(\theta) & -sin(\theta) & 0 & \frac{77}{16} \\ sin(\theta) & cos(\theta) & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} $$

$${}^{4}_{5}T = rotatex(-90) * translate(1, 0, 0) * rotatez(\theta) * translate(0, 0, 0)$$

with the resulting matrix:

$$ {}^{4}_{5}T = \begin{bmatrix} cos(\theta) & -sin(\theta) & 0 & 1 \\ 0 & 0 & -1 & 0 \\ -sin(\theta) & -cos(\theta) & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} $$

3.7 The 0 to 5 transformation

The position of the final joint 5 in the initial frame 0 is found by the matrix multiplication:

$$ {}^{0}_{5}T = {}^{0}_{1}T * {}^{1}_{2}T * {}^{2}_{3}T * {}^{3}_{4}T * {}^{4}_{5}T $$

This gives:

$$ {}^{0}_{5}T = \begin{bmatrix} Xx & Yx & Zx & x \\ Xy & Yy & Zy & y \\ Xz & Yz & Zz & z \\ 0 & 0 & 0 & 1 \end{bmatrix} $$

where

$$
\begin{aligned}
Xx &= sin(\theta_1)sin(\theta_5)+ \\
&\quad (-cos(\theta_1)cos(\theta_2)cos(\theta_5)sin(\theta_3)- \\
&\quad cos(\theta_1)cos(\theta_3)cos(\theta_5)sin(\theta_2))sin(\theta_4)- \\
&\quad cos(\theta_1)cos(\theta_4)cos(\theta_5)sin(\theta_2)sin(\theta_3)+ \\
&\quad cos(\theta_1)cos(\theta_2)cos(\theta_3)cos(\theta_4)cos(\theta_5)
\end{aligned}
$$

$$
\begin{aligned}
Yx &= (cos(\theta_1)cos(\theta_2)sin(\theta_3)+ \\
&\quad cos(\theta_1)cos(\theta_3)sin(\theta_2))sin(\theta_4)+ \\
&\quad cos(\theta_1)cos(\theta_4)sin(\theta_2)sin(\theta_3)- \\
&\quad cos(\theta_1)cos(\theta_2)cos(\theta_3)cos(\theta_4)sin(\theta_5)+ \\
&\quad cos(\theta_5)sin(\theta_1)
\end{aligned}
$$

$$
\begin{aligned}
Zx &= (cos(\theta_1)sin(\theta_2)sin(\theta_3)- \\
&\quad cos(\theta_1)cos(\theta_2)cos(\theta_3))sin(\theta_4)- \\
&\quad cos(\theta_1)cos(\theta_2)cos(\theta_4)sin(\theta_3)- \\
&\quad cos(\theta_1)cos(\theta_3)cos(\theta_4)sin(\theta_2)
\end{aligned}
$$

$$
\begin{aligned}
x &= ((-16cos(\theta_1)cos(\theta_2)sin(\theta_3)- \\
&\quad 16cos(\theta_1)cos(\theta_3)sin(\theta_2))sin(\theta_4)+ \\
&\quad (-16cos(\theta_1)cos(\theta_4) - 77cos(\theta_1))sin(\theta_2)sin(\theta_3)+ \\
&\quad 16cos(\theta_1)cos(\theta_2)cos(\theta_3)cos(\theta_4)+ \\
&\quad 77cos(\theta_1)cos(\theta_2)cos(\theta_3)+ \\
&\quad 96cos(\theta_1)cos(\theta_2) + 9cos(\theta_1))/16
\end{aligned}
$$

$$
\begin{aligned}
Xy &= -cos(\theta_1)sin(\theta_5)+ \\
&\quad (-cos(\theta_2)cos(\theta_5)sin(\theta_1)sin(\theta_3)- \\
&\quad cos(\theta_3)cos(\theta_5)sin(\theta_1)sin(\theta_2))sin(\theta_4)- \\
&\quad cos(\theta_4)cos(\theta_5)sin(\theta_1)sin(\theta_2)sin(\theta_3)+ \\
&\quad cos(\theta_2)cos(\theta_3)cos(\theta_4)cos(\theta_5)sin(\theta_1)
\end{aligned}
$$

$$
\begin{aligned}
Yy &= (cos(\theta_2)sin(\theta_1)sin(\theta_3) + cos(\theta_3)sin(\theta_1)sin(\theta_2))sin(\theta_4)+ \\
&\quad cos(\theta_4)sin(\theta_1)sin(\theta_2)sin(\theta_3)- \\
&\quad cos(\theta_2)cos(\theta_3)cos(\theta_4)sin(\theta_1)sin(\theta_5)- \\
&\quad cos(\theta_1)cos(\theta_5)
\end{aligned}
$$

$$
\begin{aligned}
Zy &= (sin(\theta_1)sin(\theta_2)sin(\theta_3) - cos(\theta_2)cos(\theta_3)sin(\theta_1))sin(\theta_4)- \\
&\quad cos(\theta_2)cos(\theta_4)sin(\theta_1)sin(\theta_3)- \\
&\quad cos(\theta_3)cos(\theta_4)sin(\theta_1)sin(\theta_2)
\end{aligned}
$$

$$
\begin{aligned}
y &= ((-16cos(\theta_2)sin(\theta_1)sin(\theta_3) - 16cos(\theta_3)sin(\theta_1)sin(\theta_2))sin(\theta_4)+ \\
&\quad (-16cos(\theta_4) - 77)sin(\theta_1)sin(\theta_2)sin(\theta_3)+ \\
&\quad (16cos(\theta_2)cos(\theta_3)cos(\theta_4)+ \\
&\quad 77cos(\theta_2)cos(\theta_3) + 96cos(\theta_2) + 9)sin(\theta_1))/16
\end{aligned}
$$

$$
\begin{aligned}
Xz &= (cos(\theta_5)sin(\theta_2)sin(\theta_3) - cos(\theta_2)cos(\theta_3)cos(\theta_5))sin(\theta_4)- \\
&\quad cos(\theta_2)cos(\theta_4)cos(\theta_5)sin(\theta_3)- \\
&\quad cos(\theta_3)cos(\theta_4)cos(\theta_5)sin(\theta_2)
\end{aligned}
$$

$$
\begin{aligned}
Yz &= (-sin(\theta_2)sin(\theta_3) + cos(\theta_2)cos(\theta_3))sin(\theta_4)+ \\
&\quad cos(\theta_2)cos(\theta_4)sin(\theta_3)+ \\
&\quad cos(\theta_3)cos(\theta_4)sin(\theta_2)sin(\theta_5)
\end{aligned}
$$

$$
\begin{aligned}
Zz &= (cos(\theta_2)sin(\theta_3) + cos(\theta_3)sin(\theta_2))sin(\theta_4) + cos(\theta_4)sin(\theta_2)sin(\theta_3)- \\
&\quad cos(\theta_2)cos(\theta_3)cos(\theta_4)
\end{aligned}
$$

$$
\begin{aligned}
z &= ((16sin(\theta_2)sin(\theta_3) - 16cos(\theta_2)cos(\theta_3))sin(\theta_4)+ \\
&\quad (-16cos(\theta_2)cos(\theta_4) - 77cos(\theta_2))sin(\theta_3)+ \\
&\quad (-16cos(\theta_3)cos(\theta_4) - 77cos(\theta_3) - 96)sin(\theta_2))/16
\end{aligned}
$$

4 Robot Dynamics

If we consider small differential changes in θ, that is $\Delta\theta$ the question is, what is the change in x, Δx?

$\delta\theta$ is connected to δx. There is a matrix representation, called the Jacobian, that describes this connection. The derivative gives a linear change in angle and its relationship to linear changes in position.

The joint coordinate C_i is θ_i for revolute joints or d_i for prismatic joints. The joint coordinate is given by

$$q_i = \bar{\epsilon}\theta_i + \epsilon d_i$$

where

$$\epsilon_i = \begin{cases} 0 & revolute \\ 1 & prismatic \end{cases}$$

The joint coordinate vector is

$$q = (q_1 q_1 \dots q_n)^T$$

We can then compute the differentiations using the direct differentiaion

$$x = f(q)$$

which is a vector differentiation given by:

$$\begin{pmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{pmatrix} = \begin{pmatrix} f_1(q) \\ f_2(q) \\ \vdots \\ f_n(q) \end{pmatrix}$$

where x_1 is the coordinate X, x_2 is the coordinate Y, etc. so we get the matrix form:

$$\delta x = \begin{bmatrix} \frac{\delta f_1}{\delta q_1} & \dots & \frac{\delta f_1}{\delta q_n} \\ \vdots & \vdots & \vdots \\ \frac{\delta f_m}{\delta q_1} & \dots & \frac{\delta f_m}{\delta q_n} \end{bmatrix} \delta q$$

which is

$$\delta x_{(m\times 1)} = J_{(m\times n)}(q)\delta q_{(m\times 1)}$$

where J is known as the Jacobian matrix, n is the number of joints and m is the number of variables.

This computation is the same as computing

$$\dot{x}_{(m\times 1)} = J_{(m\times n)}(q)\dot{q}_{(m\times 1)}$$

Note that

$$J_{ij}(q) = \frac{\delta}{\delta q_j} f_i(q)$$

4.1 Differential Motion

4.2 Linear and Angular Motion

4.3 Velocity Propagation

4.4 Explicit Form

4.5 Static Forces

The relationship between torques and forces are given by the Jacobian.

References

[1] Denavit, J. and Hartenberg, R.S. "A Kinematic Notation for Lower-Pair Mechanisms Based on Matrices" ASME Journal of Applied Mechanics (June 1955) 215-221

[2] Khatib, Oussama "Introduction to Robotics" Stanford University July 22, 2008 `http://www.youtube.com`

[3] Paul, Richard "Robot Manipulators" MIT Press 1982 ISBN 0-262-16082-X