OHIO UNIVERSITY | RUSS COLLEGE OF ENGINEERING AND TECHNOLOGY

Date: November 4th, 2020

To: Dr. Bob

From: Jonathan Bowman

Subject: Mini Project #7

Dr. Bob,

This Mini Project focuses on the simulated resolved rate control for the Adept 550 SCARA robot. The kinematic diagram and the Denavit-Hartenberg (DH) parameters were derived in a previous exercise and can be seen on Pages 2 and 3, respectively. To begin the user will need to begin with the solution for the homogeneous transformation matrix the base to the hand $[{}^{B}_{4}T]$. This solution begins on Page 2.

The analytical solution for the translational and rotational velocities of the joints are provided on Pages 7 and 8. Additionally, the Jacobian Matrix ${}^{B}[J]$ is solved analytically and is shown on Page 8. The only singularity condition within the joint limits is $\theta_2 = 0°$ and the derivation is provided on Page 8.

The four active joint rates $\{\dot{\Theta}\}$ are developed of the trajectory and are provided in Figure 3 on Page 11. The four active joint values $\{\Theta\}$ are determined over the entire trajectory and are shown in Figure 4 on Page 12. The four Cartesian components of the end effector $\{{}^{B}X_4\}$ are plotted over the entire trajectory are provided in Figure 5 on Page 13. The Jacobian Matrix determinant ${}^{B}[J]$ over the trajectory are shown on Page 13 with a discussion on how this relates to the singularities. The four active joints torques and force $\{T\}$ are plotted over the trajectory in Figure 7 on Page 14.

The animation for the SCARA robot is developed using the information from the simulated resolved rate solution and is provided in Figure 10. This animation matches closely with that of the previous Mini Projects involving the SCARA robot and provides validation of an accurate result.

Thank you for your time,

Jonathan (Jon) Bowman

Jonathan Bowman

Jb146214@ohio.edu

MP7 Introduction and Background Information

This Mini-Project (MP5) asks the student to examine the joint-space trajectory for the Adept 550 SCARA robot. A prior Mini-Project (MP4) explored the forward pose kinematic solution for the Adept 550 SCARA robot. A kinematic diagram and coordinate frame locations were developed during MP4 and can be found in Figure 1.

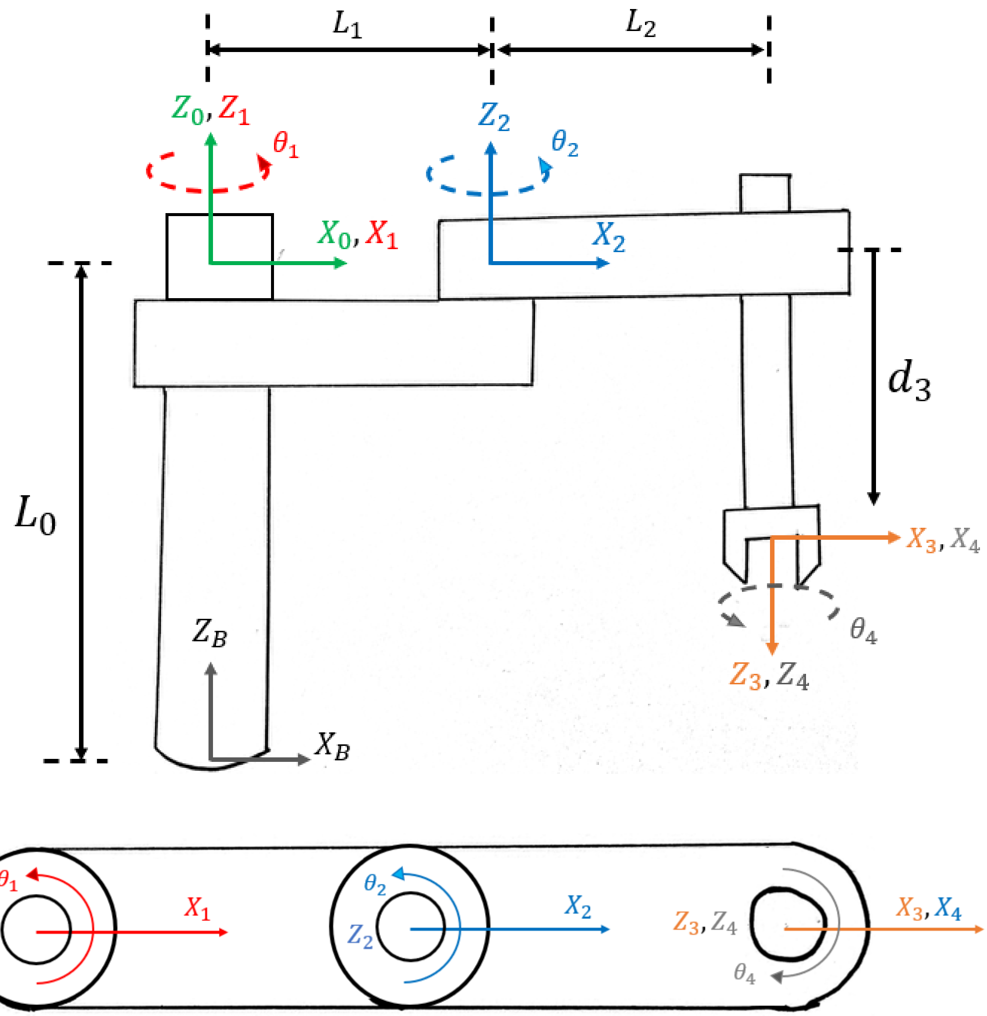

Figure 1. The Kinematic Diagram from the side view (Top) and the top view (Bottom).

From this kinematic diagram the Denavit-Hartenberg results were determined. There is only one variable per row of the associated table. Each revolute joint will have a variable θ and the prismatic joint will have a variable d_3. These results are shown in Table 1.

Table 1. The Denavit-Hartenberg (DH) Parameters

i	α_{i-1}	a_{i-1}	d_i	θ_i
1	0°	0	0	θ_1
2	0°	L_1	0	θ_2
3	180°	L_2	d_3	0°
4	0°	0	0	θ_4

The MP5 memo also includes the length lengths. The Length between the base and joint 1, L_0, is 0.552 m. The length between joint 1 and joint 2, L_1, is 0.300 m. Lastly, the length between Joint 2 and Joint 3, L_2, is 0.250 m. These results are shown in Table 2.

Table 2. The dimensions of the links in the Adept 550 SCARA Robot.

Length	**Value (m)**
L_0	0.552
L_1	0.300
L_2	0.250

The forward pose kinematics solution $[{}^B_4T]$ was developed symbolically in MP5. To simplify the resulting solutions the trigonometric sum-of-angles formula was used. This is possible since the Z axis in the Adept 550 robot are always parallel. The general form of the sum-of-angles follows:

$$\cos(a \pm b) = cacb \mp sasb$$
$$\sin(a \pm b) = sacb \pm casb$$

However, the sum of angles can be done twice back-to-back and this results in:

$$(c_1c_2 - s_1s_2)c_3 - (c_1s_2 + s_1c_2)s_3$$
$$c_{12}c_3 - s_{12}s_3$$
$$c_{123}$$

Finally, this can be written as:

$$c_{12} = \cos(\theta_1 + \theta_2) \qquad c_{123} = \cos(\theta_1 + \theta_2 + \theta_3)$$
$$s_{12} = \sin(\theta_1 + \theta_2) \qquad s_{123} = \sin(\theta_1 + \theta_2 + \theta_3)$$

The matrix multiplication provided a very large amount of results in many of the terms. The solution will take the form of:

$$[{}^{B}_{4}T] = \begin{bmatrix} TB4_1 & TB4_2 & TB4_3 & TB4_4 \\ TB4_5 & TB4_6 & TB4_7 & TB4_8 \\ TB4_9 & TB4_{10} & TB4_{11} & TB4_{12} \\ TB4_{13} & TB4_{14} & TB4_{15} & TB4_{16} \end{bmatrix}$$

Each of these terms in this matrix, in their simplified form are:

$$TB4_1 = \cos(\theta_1 + \theta_2 - \theta_4)$$
$$TB4_2 = \sin(\theta_1 + \theta_2 - \theta_4)$$
$$TB4_3 = 0$$
$$TB4_4 = L_2 * \cos(\theta_1 + \theta_2) + L_1 * \cos(\theta_1)$$
$$TB4_5 = \sin(\theta_1 + \theta_2 - \theta_4)$$
$$TB4_6 = -\cos(\theta_1 + \theta_2 - \theta_4)$$
$$TB4_7 = 0$$
$$TB4_8 = L_2 * \sin(\theta_1 + \theta_2) + L_1 * \sin(\theta_1)$$
$$TB4_9 = 0$$
$$TB4_{10} = 0$$
$$TB4_{11} = -1$$
$$TB4_{12} = L_0 - d_3$$
$$TB4_{13} = 0$$
$$TB4_{14} = 0$$
$$TB4_{15} = 0$$
$$TB4_{16} = 1$$

To begin an analytical solution of the inverse pose kinematics is determined. To develop this the student is given a valid end-effector pose $[{}^{B}_{4}T]$ or $\{{}^{B}X_{A}\} = \{\, {}^{B}x_{A} \quad {}^{B}y_{A} \quad {}^{B}z_{A} \quad \varphi\}^{T}$ and is asked to determine the four joint variables $\{\theta_1 \ \ \theta_2 \ \ d_3 \ \ \theta_4\}$. The form of the homogeneous transform is known as mentioned previously and takes the form of:

$$[{}^{B}_{4}T] = \begin{bmatrix} \cos(\theta_1 + \theta_2 - \theta_4) & \sin(\theta_1 + \theta_2 - \theta_4) & 0 & L_2 * \cos(\theta_1 + \theta_2) + L_1 * \cos(\theta_1) \\ \sin(\theta_1 + \theta_2 - \theta_4) & -\cos(\theta_1 + \theta_2 - \theta_4) & 0 & L_2 * \sin(\theta_1 + \theta_2) + L_1 * \sin(\theta_1) \\ 0 & 0 & -1 & L_0 - d_3 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

The rotation matrix is described in the 3x3 matrix in the upper left of the homogeneous transformation matrix. This matrix takes the form of:

$$[{}^{B}_{4}R] = \begin{bmatrix} \cos(\theta_1 + \theta_2 - \theta_4) & \sin(\theta_1 + \theta_2 - \theta_4) & 0 \\ \sin(\theta_1 + \theta_2 - \theta_4) & -\cos(\theta_1 + \theta_2 - \theta_4) & 0 \\ 0 & 0 & -1 \end{bmatrix}$$

The 3x1 position vector of joint 4 from the base, is in the right most column of the homogeneous transformation matrix. This position matrix describes the position in the x, y and z axes. This matrix takes the symbolic form of:

$$\{{}^{B}P_4\} = \begin{Bmatrix} {}^{B}P_{4_x} \\ {}^{B}P_{4_y} \\ {}^{B}P_{4_z} \end{Bmatrix} = \begin{Bmatrix} L_2 * \cos(\theta_1 + \theta_2) + L_1 * \cos(\theta_1) \\ L_2 * \sin(\theta_1 + \theta_2) + L_1 * \sin(\theta_1) \\ L_0 - d_3 \end{Bmatrix}$$

This position information will be given when developing the inverse pose kinematic solution. Therefore, this can be rewritten as:

$$\{{}^{B}X_4\} = \begin{Bmatrix} {}^{B}P_{4_x} \\ {}^{B}P_{4_y} \\ {}^{B}P_{4_z} \end{Bmatrix} = \begin{Bmatrix} L_2 * \cos(\theta_1 + \theta_2) + L_1 * \cos(\theta_1) \\ L_2 * \sin(\theta_1 + \theta_2) + L_1 * \sin(\theta_1) \\ L_0 - d_3 \end{Bmatrix}$$

From this position information three independent translational equations can be developed. These are:

$$x_4 = L_2 * c_{12} + L_1 * c_1$$
$$y_4 = L_2 * s_{12} + L_1 * s_1$$
$$z_4 = L_0 - d_3$$

Additionally, the angle of the end-effector φ is known in terms of the unknown joint angles as:

$$\varphi = \theta_1 + \theta_2 - \theta_4$$

Simulated Resolved Rate Control

Part A

The given joint limits are provided in the MP7 memo and are shown in Table 3.

Table 3. The given joint limits for the Adept SCARA Robot.

Joint Limits	$\boldsymbol{\theta_1}$	$\boldsymbol{\theta_2}$	$\boldsymbol{d_3}$ (m)	$\boldsymbol{\theta_4}$
Minimum	-100°	-140°	0.178	-360°
Maximum	+100°	+140°	0.378	+360°

To begin, first recall the forward pose kinematics results that:

$$[{}^{B}_{4}T] = \begin{bmatrix} \cos(\theta_1 + \theta_2 - \theta_4) & \sin(\theta_1 + \theta_2 - \theta_4) & 0 & L_2 * \cos(\theta_1 + \theta_2) + L_1 * \cos(\theta_1) \\ \sin(\theta_1 + \theta_2 - \theta_4) & -\cos(\theta_1 + \theta_2 - \theta_4) & 0 & L_2 * \sin(\theta_1 + \theta_2) + L_1 * \sin(\theta_1) \\ 0 & 0 & -1 & L_0 - d_3 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

This transformation matrix describes the four independent pose kinematics equations in the last column. When written explicitly these equations are:

$$\begin{aligned} {}^{B}X_4 &= L_2 * \cos(\theta_1 + \theta_2) + L_1 * \cos(\theta_1) \\ {}^{B}Y_4 &= L_2 * \sin(\theta_1 + \theta_2) + L_1 * \sin(\theta_1) \\ {}^{B}Z_4 &= L_0 - d_3 \\ \varphi &= \theta_1 + \theta_2 - \theta_4 \end{aligned}$$

The translational velocity equations (${}^{B}\dot{X}_4, {}^{B}\dot{Y}_4, {}^{B}\dot{Z}_4$) can be found by take the first time derivative of the position equations. Additionally, let $\theta_1 + \theta_2 = \theta_{12}$. These equations become:

$$\begin{aligned} {}^{B}\dot{X}_4 &= \frac{d}{dt}({}^{B}X_4) = -L_2 * \sin(\theta_{12}) * (\dot{\theta}_1 + \dot{\theta}_2) - L_1 * \sin(\theta_1)\,\dot{\theta}_1 \\ {}^{B}\dot{Y}_4 &= \frac{d}{dt}({}^{B}Y_4) = L_2 * \cos(\theta_{12}) * (\dot{\theta}_1 + \dot{\theta}_2) + L_1 * \cos(\theta_1)\,\dot{\theta}_1 \\ {}^{B}\dot{Z}_4 &= \frac{d}{dt}({}^{0}Z_4) = -\dot{d}_3 \end{aligned}$$

To determine the rotational velocity equations, first note that the rotations are all in the X-Y plane. Additionally, all the Z axes in the robot are parallel. Therefore ${}^{B}\omega_{4Z}$ is the time derivative of φ.

$$
\begin{gathered}
{}^{B}\omega_{4X} = 0 \\
{}^{B}\omega_{4Y} = 0 \\
{}^{B}\omega_{4Z} = \dot{\varphi} = \dot{\theta}_1 + \dot{\theta}_2 - \dot{\theta}_4
\end{gathered}
$$

These equations can be rearranged and written into a matrix vector system of equations:

$$
\begin{bmatrix} -L_2 \sin(\theta_{12}) - L_1 \sin(\theta_1) & -L_2 \sin(\theta_{12}) & 0 & 0 \\ L_2 \cos(\theta_{12}) + L_1 \cos(\theta_1) & L_2 \cos(\theta_{12}) & 0 & 0 \\ 0 & 0 & -1 & 0 \\ 1 & 1 & 0 & -1 \end{bmatrix} \begin{Bmatrix} \dot{\theta}_1 \\ \dot{\theta}_2 \\ \dot{d}_3 \\ \dot{\theta}_4 \end{Bmatrix} = \begin{Bmatrix} {}^{B}\dot{X}_4 \\ {}^{B}Y_4 \\ {}^{B}\dot{Z}_4 \\ {}^{B}\omega_{4Z} \end{Bmatrix}
$$

Part B

The matrix vector system of equation follows the form ${}^{B}[J]\{\dot{\Theta}\} = \{{}^{B}\dot{X}\}$, therefore the Jacobian Matrix ${}^{B}[J]$ is:

$$
{}^{B}[J] = \begin{bmatrix} -L_2 \sin(\theta_{12}) - L_1 \sin(\theta_1) & -L_2 \sin(\theta_{12}) & 0 & 0 \\ L_2 \cos(\theta_{12}) + L_1 \cos(\theta_1) & L_2 \cos(\theta_{12}) & 0 & 0 \\ 0 & 0 & -1 & 0 \\ 1 & 1 & 0 & -1 \end{bmatrix}
$$

It is important to consider the possible singularities that might arise with the SCARA. Robot singularities occur when the derived Jacobian Matrix Looses full rank. The rank in a matrix refers to the number of linearly independent rows (or equations). To retain full rank the rank must match the number of rows, in the case of the SCARA robot that means rank must be 4. Another way to detect when singularities occur is when the determinant of the Jacobian Matrix is approaches zero. The determinant of the SCARA Jacobian Matrix is:

$$
|{}^{B}[J]| = L_1 * L_2 * \sin(\theta_{12}) * \cos(\theta_1) - L_1 * L_2 * \cos(\theta_{12}) * \sin(\theta_1)
$$

When $\theta_2 = 0°$ the equation results in:

$$
L_1 * L_2 * \sin(\theta_1) * \cos(\theta_1) - L_1 * L_2 * \cos(\theta_1) * \sin(\theta_1) = 0
$$

Therefore, this results in a singularity at $\theta_2 = 0°$ and this must be avoided.

Part C

MATLAB is used as a tool to develop the simulated resolved rate control. The flow chart for the resolved rate algorithm is provided in Figure 2 and this is the base used to develop the program.

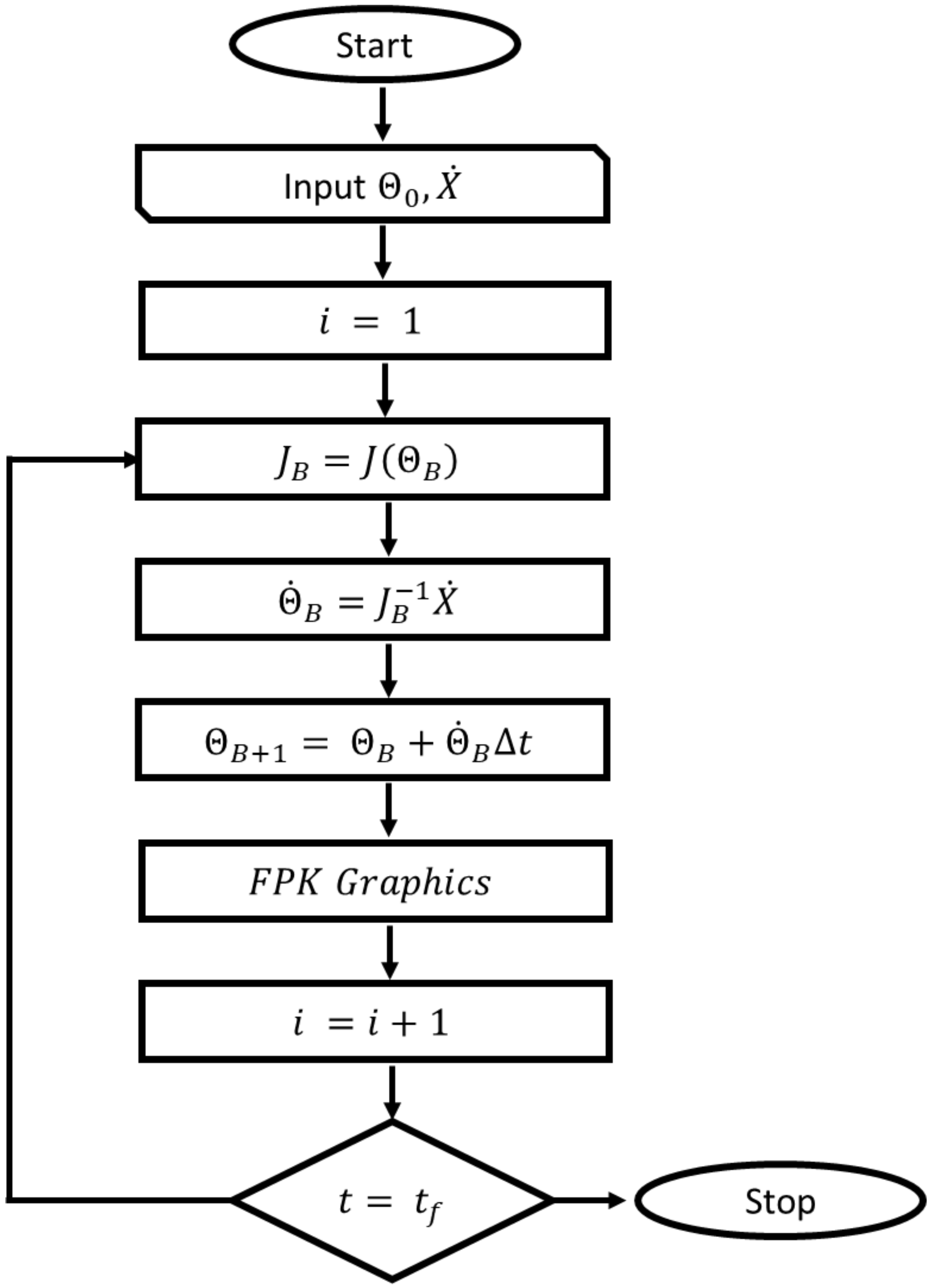

Figure 2. The flowchart for the resolved rate algorithm.

The given information for initial robot joint variables Θ_0, the constant commanded Cartesian rates $\{{}^B\dot{X}_4\}$ and the constant commanded Cartesian Wrench $\{{}^BW\}$ given for the Mini-Project 7 are:

$$\Theta_0 = \begin{Bmatrix} \theta_{1_0} \\ \theta_{2_0} \\ d_{3_0} \\ \theta_{4_0} \end{Bmatrix} = \begin{Bmatrix} -51.06° \\ 83.30° \\ 0.378\,m \\ 32.24° \end{Bmatrix} \qquad \{{}^B\dot{X}_4\} = \begin{Bmatrix} {}^B\dot{X}_4 \\ {}^B\dot{Y}_4 \\ {}^B\dot{Z}_4 \\ {}^B\omega_{4Z} \end{Bmatrix} = \begin{Bmatrix} 0 \\ 0.2 \\ 0.1 \\ 0 \end{Bmatrix} \qquad \{{}^BW\} = \begin{Bmatrix} f_X \\ f_Y \\ f_Z \\ m_Z \end{Bmatrix} = \begin{Bmatrix} 4 \\ 3 \\ 2 \\ 1 \end{Bmatrix}$$

Additionally, the overall time for the resolved-rate motion (t_f) is 2 seconds and the time steps (Δt) will be exactly 0.04 seconds. To begin the simulated resolved rate control problem, the current joint values must be used to develop the Jacobian Matrix following the form below. Remember that, for simplification, the shorthand of $\theta_1 + \theta_2 = \theta_{12}$ is used.

$${}^B[J] = \begin{bmatrix} -L_2\sin(\theta_{12}) - L_1\sin(\theta_1) & -L_2\sin(\theta_{12}) & 0 & 0 \\ L_2\cos(\theta_{12}) + L_1\cos(\theta_1) & L_2\cos(\theta_{12}) & 0 & 0 \\ 0 & 0 & -1 & 0 \\ 1 & 1 & 0 & -1 \end{bmatrix}$$

Now using the given end-effector Cartesian velocity $\{{}^B\dot{X}_4\}$ and the Jacobian Matrix at the current joint variables, the current relative joint rates $\{\dot{\Theta}\}$ can be determined by using:

$$\{\dot{\Theta}\}_i = {}^B[J(\Theta_i)]^{-1}\{{}^B\dot{X}_4\}$$

To determine the joint variables at the next time step, the current relative joint rates can be integrated as shown:

$$\Theta_{i+1} = \Theta_i + \{\dot{\Theta}\}_i\Delta t$$

To determine the active joint torques and forces at the current time the following equation can be used. This equation, derived from the principles of virtual work, uses the transpose of the current Jacobian Matrix ${}^B[J(\Theta_i)]$ multiplied with the given constant commanded Cartesian Wrench $\{{}^BW\}$.

$$\{T\} = {}^B[J(\Theta_i)]^T\{{}^BW\}$$

When these principles are applied for the entire time and each of this data is collected, they can be graphed for the entire range of motion. The following graphs are to be studied:

1. The four active joint rates $\{\dot{\Theta}\} = \{\dot{\theta}_1\ \dot{\theta}_2\ \dot{d}_3\ \dot{\theta}_4\}$ vs. time.
2. The four active joint values $\{\Theta\} = \{\theta_1\ \theta_2\ d_3\ \theta_4\}$ vs. time.
3. The four Cartesian components of $[{}^B_4T]$, $\{{}^BX_4\} = \{x_4\ y_4\ z_4\ \varphi\}^T$ vs. time.
4. The Jacobian Matrix Determinant $|{}^B[J]|$ vs. time.
5. The four active joint torques/force $\{\mathrm{T}\} = \{\tau_1\ \tau_2\ F_3\ \tau_4\}$ vs. time.

The first relationship to study is the four active joint rates over time. The active relative joint rates describe the change in the joints angles over time. Joint angle rates $(\dot{\theta})$ are measured in rad/s for revolute joints and in m/s for prismatic joints $(\dot{d})$. The plot for this SCARA Robot example is given in Figure 3. When the joint rates are negative it indicates a decreasing joint angle. The reverse is also true when a joint rate is positive it indicates an increasing joint angle. These are relationships are shown in Figure 3.

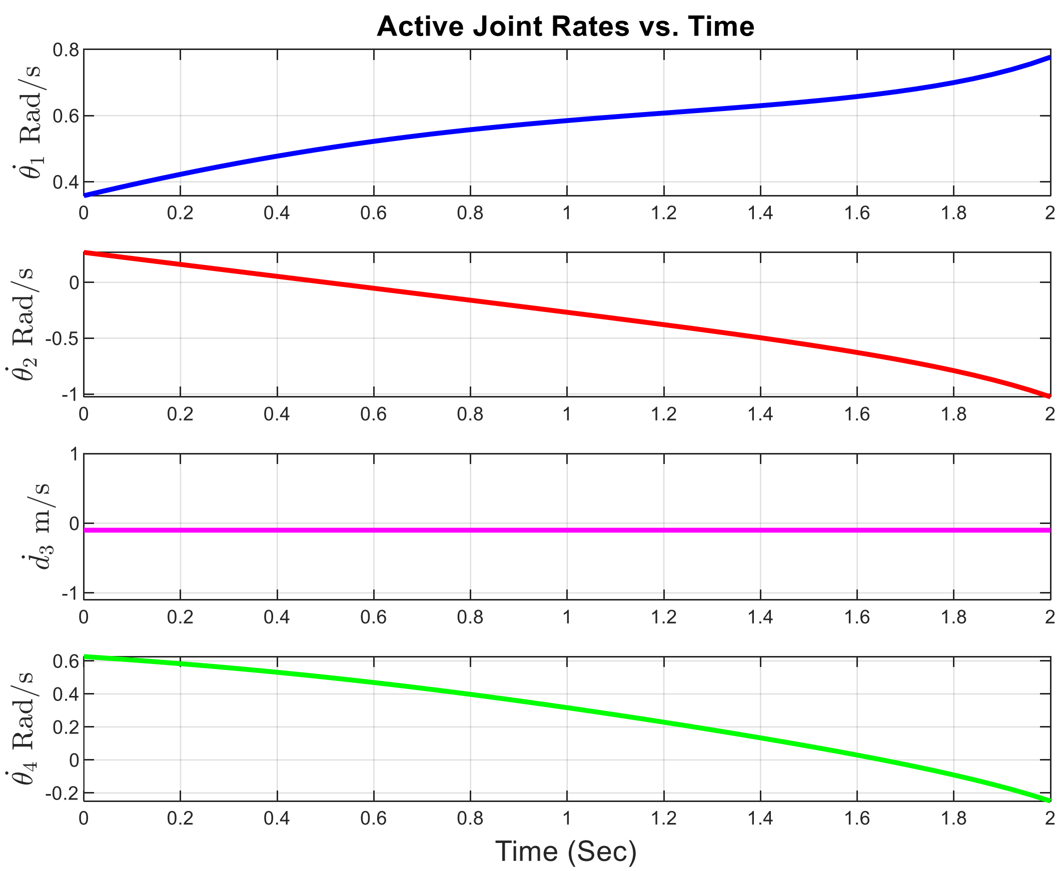

Figure 3. The active joint rates vs. time for the SCARA Robot.

The second relationship to examine is between the active joint values and time. These match almost exactly as developed in the Forward Pose Kinematics. A further discussion is provided in the next section. It is important to note than none of the provided joint angle limits are violated. The active joint values plot over time are shown in Figure 4.

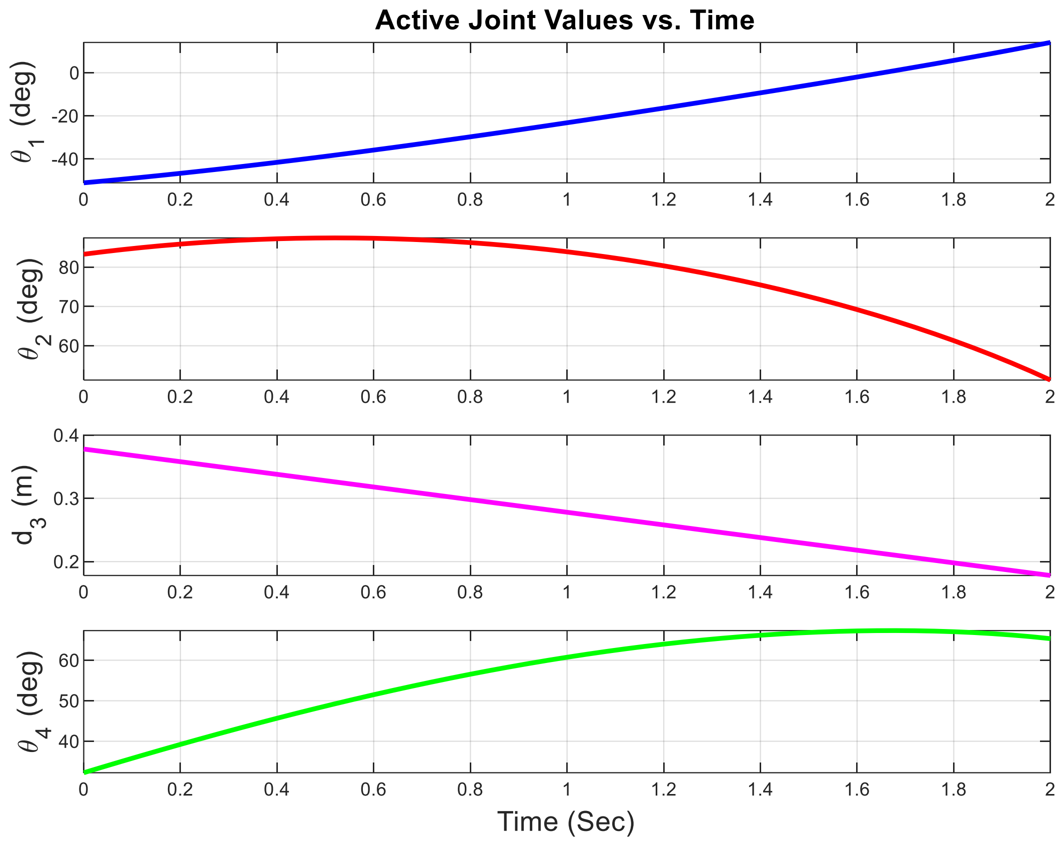

Figure 4. The active joint values vs. time for the SCARA Robot.

The next relationship to review is the four cartesian components of the transformation matrix $[{}^B_4T]$. These cartesian coordinates represent the location of the hand of the SCARA robot and can be found in Figure 5. When these are compared to the previous solutions for the Joint Trajectory solution and the Inverse Pose Kinematic solution these results match closely. However, they match more closely with the inverse solution. A discussion on these results are provided in the next section.

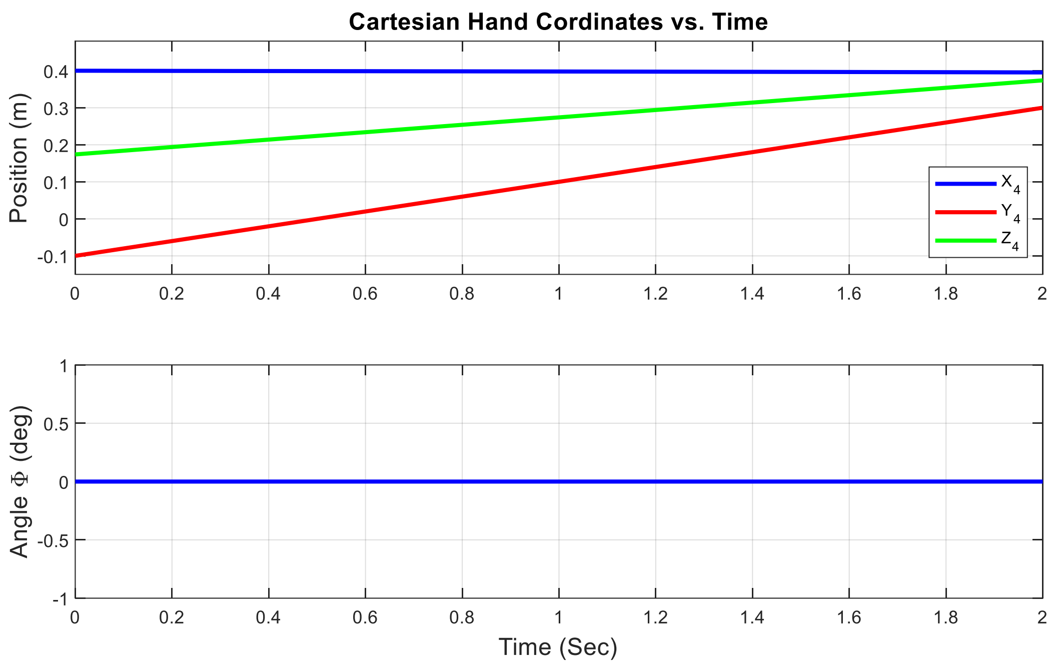

Figure 5. The cartesian components of vs. time $[{}^{B}_{4}T]$ for the SCARA Robot.

The relationship of the Jacobian Matrix determinant over time is a critical one to examine when trying to prevent robot singularities. The plot for the trajectory of the SCARA robot is provided in Figure 6. This shows that the range the robot is operating in is the optimal range, farthest away from a singularity at a determinant of 0. If this motion were to be extended the user will want to watch this graph closely to ensure no singularities are reached.

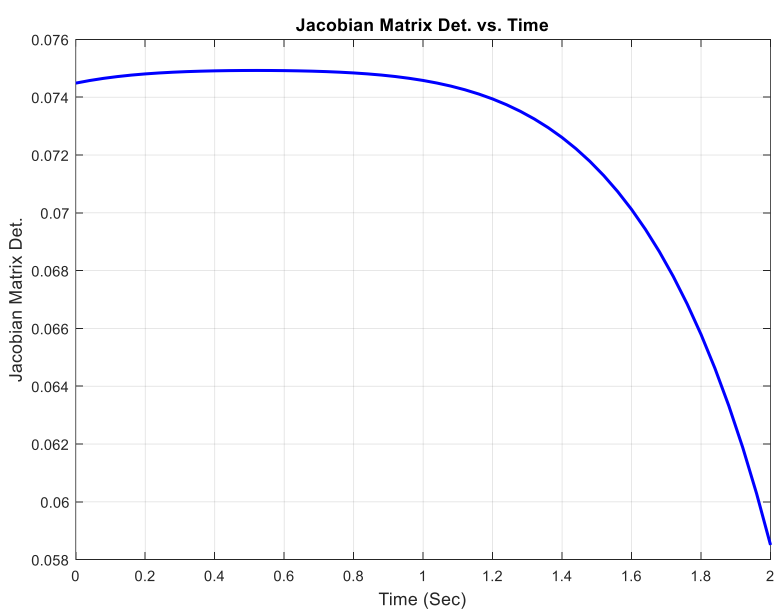

Figure 6. The Jacobian Matrix determinant $|{}^{B}[J]|$ vs. time for the SCARA Robot.

Finally, the last plot to examine is the active joint torques and forces over time is provided in Figure 7. These torques and forces indicate what is applied to the environment the given cartesian wrench. A few things to notice is that both the both the force applied by the prismatic joint d_3 and the torque at joint 4 are constant. The force will be acting in the downward direction, therefore a negative force makes sense.

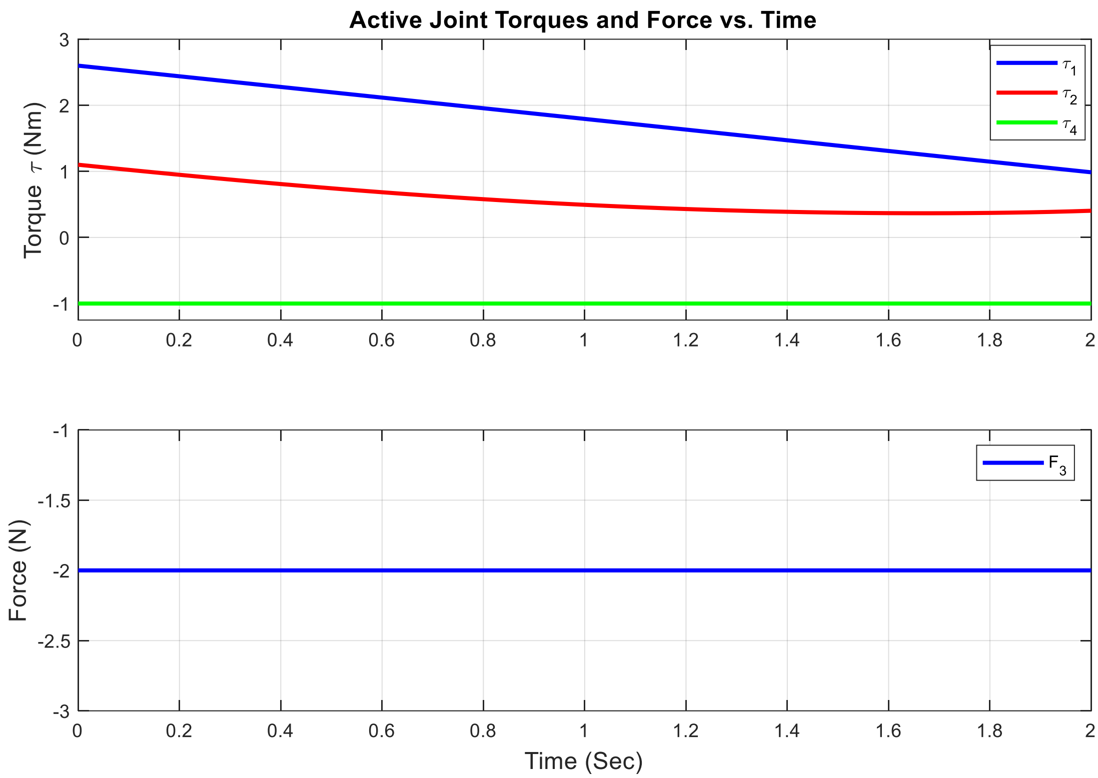

Figure 7. The active joint torques and force vs. time for the SCARA robot.

Discussion and Comparison

When the simulated resolved rate control is compared to the inverse solution result in Figure 8. Additionally, the joint angle positions in the inverse pose solution are shown compared to the resolved rate control method. This plot is shown in Figure 9. These results match nearly identically and provide strong evidence in an accurate result.

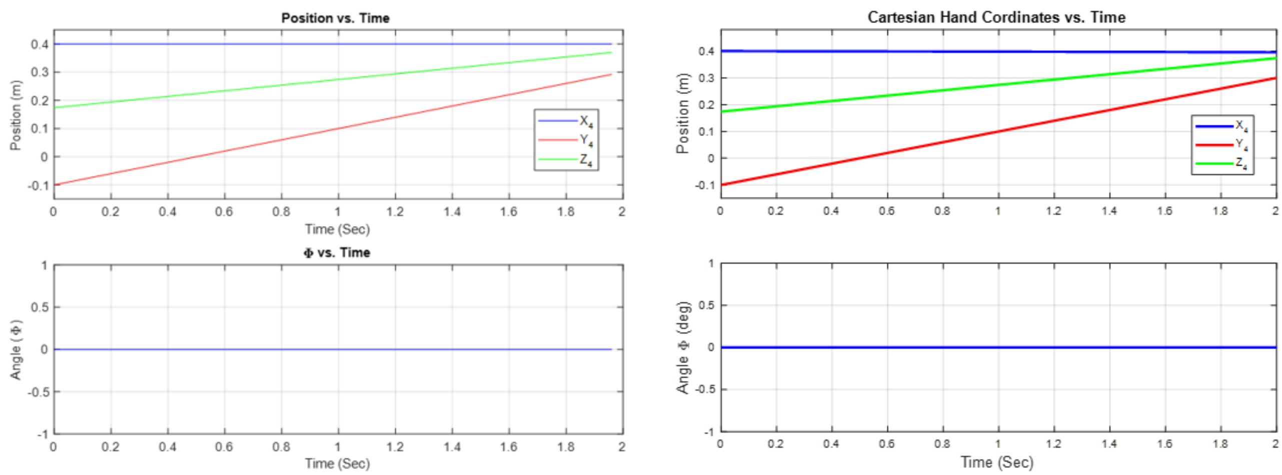

Figure 8. The cartesian hand coordinates for the inverse pose (left) and the resolved rate (right).

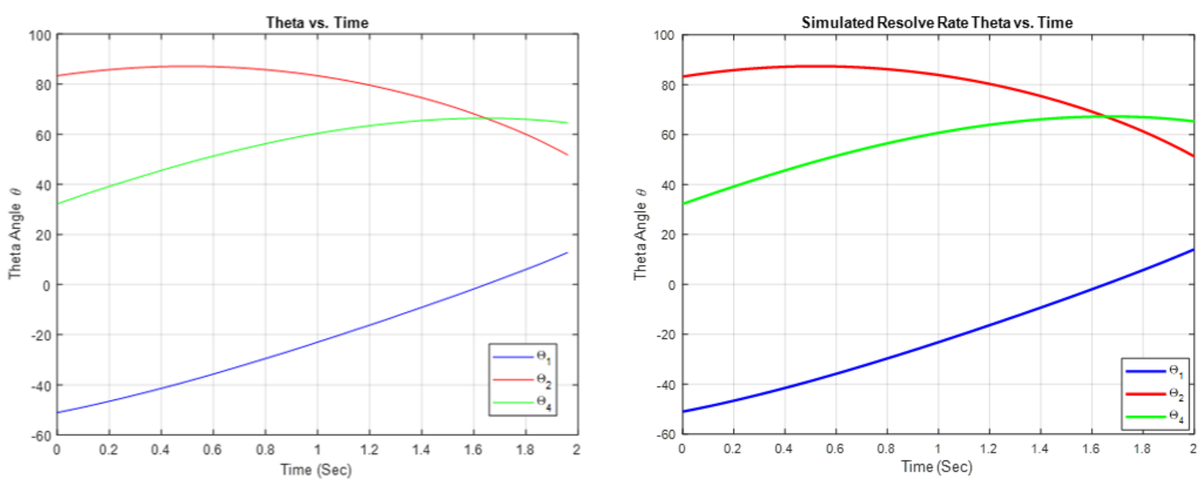

Figure 9. The joint variables for the inverse pose (left) and the resolved rate (right).

The entire trajectory for the simulated resolved rate control are shown in Figure 10. These plots show the robot motion for the two second time frame, this again matches closely with the results determined in subsequent Mini-Projects for the Adept SCARA robot. It is important to note that the singularity at $\theta_2 = 0°$ is avoided and provided additional support for plot for the determinant of the Jacobian Matrix shown in Figure 6. Lastly it is clear from this Figure and the subsequent information provided that none of the joint limits in Table 3 are violated.

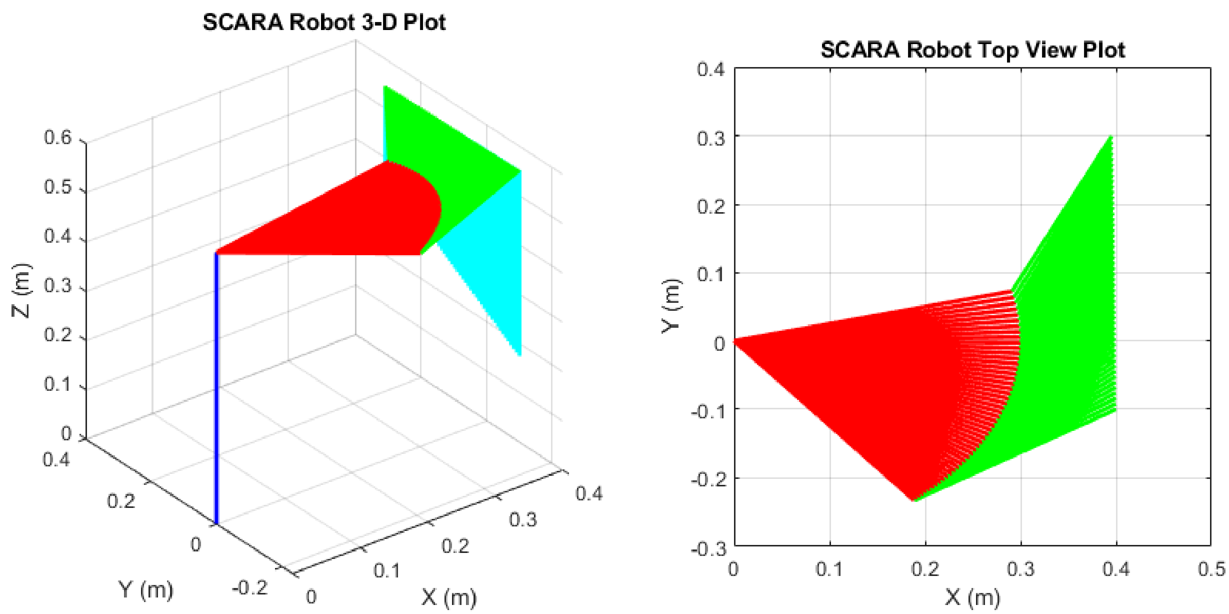

Figure 10. The 3D SCARA Robot Animation.

Simulated Resolved-Rate Control MATLAB Code

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%  Jonathan Bowman
%  ME 4290  Mini-Project 7 Part C
%  Simulated Resolved-Rate Control
%  Robot: Adept 550 SCARA Robot
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

%General Code Keeping
clear; clc; close all;

%Given Information
%Link Lengths (m)
L0 = 0.552;
L1 = 0.300;
L2 = 0.250;

%Conversion (Deg and Rad)
degToRad = (3.14159265/180);
radToDeg = (180/3.14159265);

%Time for Trajectory
tfinal = 2; %Seconds

%Starting Values (From IPK Solution)
theta1Start = -51.06; %Deg
theta2Start = 83.30; %Deg
d3Start = 0.378; %meters
theta4Start = 32.24; %Deg

%Constant Commanded Cartesian Rates
XdotB4 = 0;  %m/s
YdotB4 = 0.2; %m/s
ZdotB4 = 0.1; %m/s
omegaZ = 0; %Rad/s

Xdot = [XdotB4; YdotB4; ZdotB4; omegaZ];

%Cartesian Wrench (Given)
WB = [4;3;2;1];

%Time info.
timeStep = 0.04; %Sec
finalTime = 2.001; %Sec
startTime = 0; %Sec
currentTime = startTime; %Sec

%%%%%%%%%%%%%%%%%%%Joint Info
%Set Current Joint Values
currentTheta1 = theta1Start*degToRad;
currentTheta2 = theta2Start*degToRad;
```

(Continued on next page)

```
currentd3 = d3Start;
currentTheta4 = theta4Start*degToRad;

%Initial Joint Variables
theta1 = currentTheta1;
theta2 = currentTheta2;
d3 = currentd3;
theta4 = currentTheta4;

%Start Counter
i = 1;

while currentTime < finalTime

  %Time Information
  time(i) = currentTime;
  currentTime = currentTime+timeStep;

  %Record the Joint Variables
  theta1Record(i) = theta1*radToDeg;
  theta2Record(i) = theta2*radToDeg;
  d3Record(i) = d3;
  theta4Record(i) = theta4*radToDeg;

  %The Jacobian Matrix
  JB4 = [-L2*sin(theta1+theta2)-L1*sin(theta1) -L2*sin(theta1+theta2) 0 0;
  L2*cos(theta1+theta2)+L1*cos(theta1) L2*cos(theta1+theta2) 0 0;
  0 0 -1 0;
  1 1 0 -1];

  %The Det of JB4
  JB4det(i) = det(JB4);

  %Inverse Velcoity Solution
  thetaDot = inv(JB4)*Xdot;

  %Keep Theta Dot Values
  thetaDot1Record(i) = thetaDot(1,1);
  thetaDot2Record(i) = thetaDot(2,1);
  dDot3(i) = thetaDot(3,1);
  thetaDot4Record(i) = thetaDot(4,1);

  %Link Locations
  xx1 = [0 0];
  yy1 = [0 0];
  zz1 = [0 L0];
  xx2 = [0 (L1*cos(theta1))];
  yy2 = [0 (L1*sin(theta1))];
  zz2 = [L0 L0];
  xx3 = [(L1*cos(theta1)) ((L1*cos(theta1))+(L2*cos(theta1+theta2)))];
  yy3 = [(L1*sin(theta1)) ((L1*sin(theta1))+(L2*sin(theta1+theta2)))];
  zz3 = [L0 L0];
```

(Continued on next page)

```
xx4 = [((L1*cos(theta1))+(L2*cos(theta1+theta2)))
((L1*cos(theta1))+(L2*cos(theta1+theta2)))];
  yy4 = [((L1*sin(theta1))+(L2*sin(theta1+theta2)))
((L1*sin(theta1))+(L2*sin(theta1+theta2)))];
  zz4 = [L0 (L0-d3)];
  phiRad = theta1+theta2-theta4;

  %Plot Function in 3D Space (MAKE SQUARE)
  subplot(1,2,1)
  plot3(xx1,yy1,zz1,'b',xx2,yy2,zz2,'r',xx3,yy3,zz3,'g',xx4,yy4,zz4,'c')
  axis square; grid;
  title('SCARA Robot 3-D Plot')
  xlabel('X (m)'); ylabel('Y (m)'); zlabel('Z (m)');
  set(findall(gca, 'Type', 'Line'),'LineWidth',2);
  hold on;
  pause(1/100); %Pause to make animation visible

  %Plot Function In 2D Top View (MAKE SQUARE)
  subplot(1,2,2)
  plot(xx1,yy1,'b',xx2,yy2,'r',xx3,yy3,'g',xx4,yy4,'c')
  axis square; grid;
  title('SCARA Robot Top View Plot')
  xlabel('X (m)'); ylabel('Y (m)');
  set(findall(gca, 'Type', 'Line'),'LineWidth',2);
  hold on;
  pause(1/100); %Pause to make animation visible

  %Keep the Values
  x4(i) = ((L1*cos(theta1))+(L2*cos(theta1+theta2)));
  y4(i) = ((L1*sin(theta1))+(L2*sin(theta1+theta2)));
  z4(i) = (L0-d3);
  phi(i) = phiRad*radToDeg;

  %Integrate ThetaDot and Add to Original Matrix
  theta1 = theta1+(thetaDot(1,1)*timeStep);
  theta2 = theta2+(thetaDot(2,1)*timeStep);
  d3 = d3+(thetaDot(3,1)*timeStep);
  theta4 = theta4+(thetaDot(4,1)*timeStep);

  %Calculate the Joint Forces
  torqueForce = transpose(JB4)*WB;

  %Keep the Info
  torque1(i) = torqueForce(1,1);
  torque2(i) = torqueForce(2,1);
  force3(i) = torqueForce(3,1);
  torque4(i) = torqueForce(4,1);


  %Index Counter
  i = i+1;
end
```

(Continued on next page)

```
%Plotting Information

%Plot the Active Joint Values vs. Time
figure;
subplot(4,1,1)
plot(time,theta1Record,'b'); %Theta Changes
title('Active Joint Values vs. Time','FontSize', 12);
grid; ylabel('\theta_1 (deg)','FontSize', 12);
set(findall(gca, 'Type', 'Line'),'LineWidth',2);
xlim([0 2])

subplot(4,1,2)
plot(time,theta2Record,'r'); %Theta Changes
grid; ylabel('\theta_2 (deg)','FontSize', 12);
set(findall(gca, 'Type', 'Line'),'LineWidth',2);
xlim([0 2])

subplot(4,1,3)
plot(time,d3Record,'m'); %Theta Changes
grid; ylabel('d_3 (m)','FontSize', 12);
set(findall(gca, 'Type', 'Line'),'LineWidth',2);
xlim([0 2])

subplot(4,1,4)
plot(time,theta4Record,'g'); %Theta Changes
grid; ylabel('\theta_4 (deg)','FontSize', 12);
xlabel('Time (Sec)','FontSize', 12);
set(findall(gca, 'Type', 'Line'),'LineWidth',2);
xlim([0 2])

%Plot This Information
figure;
plot(time,theta1Record,'b',time,theta2Record,'r',time,theta4Record,'g');
%Theta Changes
grid; xlabel('Time (Sec)'); ylabel('Theta Angle \theta');
title('Simulated Resolve Rate Theta vs. Time');
legend('\Theta_1','\Theta_2','\Theta_4');
set(findall(gca, 'Type', 'Line'),'LineWidth',2);
xlim([0 2])

%Plot Cartesian Values
figure;
subplot(2,1,1)
plot(time,x4,'b',time,y4,'r',time,z4,'g'); %Theta Changes
title('Cartesian Hand Cordinates vs. Time','FontSize', 12)
grid; axis([0 2 -0.15 0.48])
ylabel('Position (m)','FontSize', 12);
legend('X_4','Y_4','Z_4');
set(findall(gca, 'Type', 'Line'),'LineWidth',2);

subplot(2,1,2)
plot(time,phi,'b'); %d3 Changes
```

(Continued on next page)

```
grid; axis([0 2 -1 1])
xlabel('Time (Sec)','FontSize', 12); ylabel('Angle \Phi (deg)','FontSize',
12);
set(findall(gca, 'Type', 'Line'),'LineWidth',2);

%Plot Active Joint Rate Values
figure;
subplot(4,1,1)
plot(time,thetaDot1Record,'b'); %Theta Changes
title('Active Joint Rates vs. Time','FontSize', 12);
grid; ylabel('$\dot{\theta_1}$ Rad/s', 'Interpreter','latex','FontSize', 12);
set(findall(gca, 'Type', 'Line'),'LineWidth',2);
xlim([0 2])

subplot(4,1,2)
plot(time,thetaDot2Record,'r'); %Theta Changes
grid; ylabel('$\dot{\theta_2}$ Rad/s', 'Interpreter','latex','FontSize', 12);
set(findall(gca, 'Type', 'Line'),'LineWidth',2);
xlim([0 2])

subplot(4,1,3)
plot(time,dDot3,'m'); %Theta Changes
grid; ylabel('$\dot{d_3}$ m/s', 'Interpreter','latex','FontSize', 12);
set(findall(gca, 'Type', 'Line'),'LineWidth',2);
xlim([0 2])

subplot(4,1,4)
plot(time,thetaDot4Record,'g'); %Theta Changes
grid;
ylabel('$\dot{\theta_4}$ Rad/s', 'Interpreter','latex','FontSize', 12);
xlabel('Time (Sec)','FontSize', 12);
set(findall(gca, 'Type', 'Line'),'LineWidth',2);
xlim([0 2])

%Plot Jacobian Matrix Determinant
figure;
plot(time,JB4det,'b');
grid; xlabel('Time (Sec)','FontSize', 12);
ylabel('Jacobian Matrix Det.','FontSize', 12);
title('Jacobian Matrix Det. vs. Time','FontSize', 12);
set(findall(gca, 'Type', 'Line'),'LineWidth',2);
xlim([0 2])

%%%%%Plot Active Joint Forces/Torques%%%%%
%Torques
figure;
subplot(2,1,1)
plot(time,torque1,'b',time,torque2,'r',time,torque4,'g');
grid; ylabel('Torque \tau (Nm)','FontSize', 12);
title('Active Joint Torques and Force vs. Time','FontSize', 12);
legend('\tau_1','\tau_2','\tau_4');
set(findall(gca, 'Type', 'Line'),'LineWidth',2);
axis([0 2 -1.25 3]);
```

(Continued on next page)

```
%Force
subplot(2,1,2)
plot(time,force3,'b');
grid; xlabel('Time (Sec)','FontSize', 12);
ylabel('Force (N)','FontSize', 12);
legend('F_3');
set(findall(gca, 'Type', 'Line'),'LineWidth',2);
xlim([0 2])
```