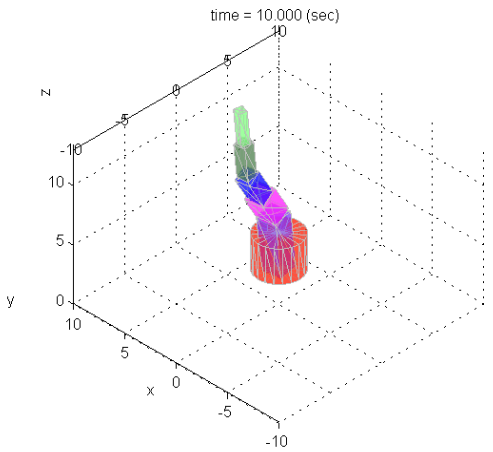
HW1 ME 739 Introduction to robotics

Spring 2015
Department of Mechanical Engineering
University of Wisconsin, Madison

Instructor: Professor Michael Zinn



By

Nasser M. Abbasi

May 3, 2022

Contents

0.1 Problem 1 3
0.1.1 problem description 3
0.1.2 solution 3
0.2 Problem 2 4
0.3 Problem 3 5
0.4 Problem 4 6
0.5 Problem 5 8
0.5.1 solution 8
0.6 Problem 6 14
0.6.1 problem description 14
0.6.2 solution 16

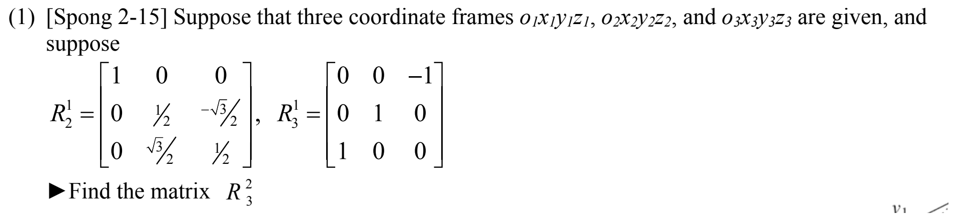
List of Figures

1 problem 2 description 4
2 problem 3 description 5
3 problem 4 description 6
4 problem 5 description 8

List of Tables

0.1 Problem 1

0.1.1 problem description



(1) [Spong 2-15] Suppose that three coordinate frames $o_1x_1y_1z_1$, $o_2x_2y_2z_2$, and $o_3x_3y_3z_3$ are given, and suppose

$$R_2^1 = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1/2 & -\sqrt{3}/2 \\ 0 & \sqrt{3}/2 & 1/2 \end{bmatrix}, \quad R_3^1 = \begin{bmatrix} 0 & 0 & -1 \\ 0 & 1 & 0 \\ 1 & 0 & 0 \end{bmatrix}$$

►Find the matrix R_3^2

0.1.2 solution

Starting with the relation

$$R_3^1 = R_2^1 R_3^2$$

Pre-multiplying both sides by $(R_2^1)^{-1}$ which exists since R is a rotation matrix and hence invertible results in

$$R_3^2 = (R_2^1)^{-1} R_3^1$$

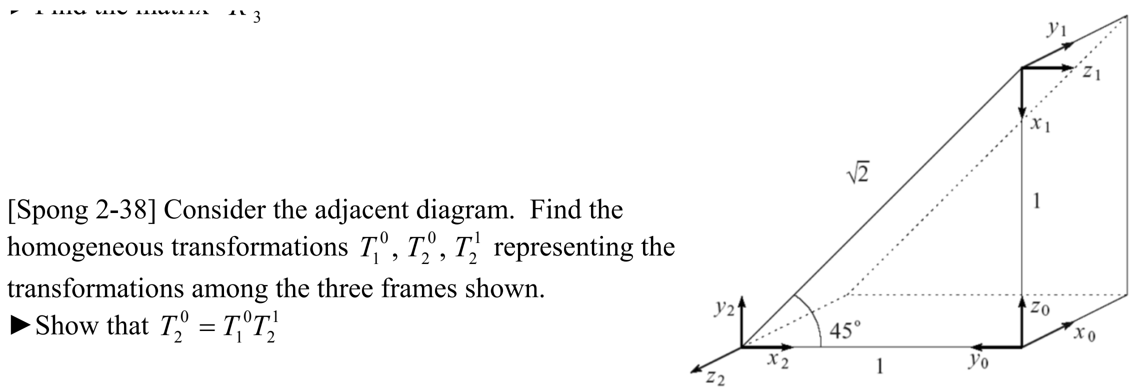
For a rotation matrix the following relation holds

$$(R_2^1)^{-1} = (R_2^1)^T$$

Therefore

$$\begin{aligned} R_3^2 &= (R_2^1)^T R_3^1 \\ &= \begin{pmatrix} 1 & 0 & 0 \\ 0 & \frac{1}{2} & \frac{-\sqrt{3}}{2} \\ 0 & \frac{\sqrt{3}}{2} & \frac{1}{2} \end{pmatrix}^T \begin{pmatrix} 0 & 0 & -1 \\ 0 & 1 & 0 \\ 1 & 0 & 0 \end{pmatrix} \\ &= \begin{pmatrix} 1 & 0 & 0 \\ 0 & \frac{1}{2} & \frac{\sqrt{3}}{2} \\ 0 & \frac{-\sqrt{3}}{2} & \frac{1}{2} \end{pmatrix} \begin{pmatrix} 0 & 0 & -1 \\ 0 & 1 & 0 \\ 1 & 0 & 0 \end{pmatrix} \\ &= \begin{pmatrix} 0 & 0 & -1 \\ \frac{\sqrt{3}}{2} & \frac{1}{2} & 0 \\ \frac{1}{2} & \frac{-\sqrt{3}}{2} & 0 \end{pmatrix} \end{aligned}$$

0.2 Problem 2


► Find the matrix R_3

(2) [Spong 2-38] Consider the adjacent diagram. Find the homogeneous transformations T_1^0, T_2^0, T_2^1 representing the transformations among the three frames shown.

► Show that $T_2^0 = T_1^0 T_2^1$

Figure 1: problem 2 description

The goal is to determine T_1^0, T_2^0, T_3^0 and T_3^2. T_i^{i-1} is the homogeneous transformation from frame $\{i-1\}$ to frame $\{i\}$ given by

$$T_i^{i-1} = \begin{pmatrix} R_i^{i-1} & d \\ 0 & 1 \end{pmatrix}$$

Where d is the position vector from the origin of frame $\{i-1\}$ to the origin of frame $\{i\}$ expressed in frame $\{i-1\}$, and R_i^{i-1} is the rotation matrix.

By direct inspection of the above diagram the following transformations are obtained

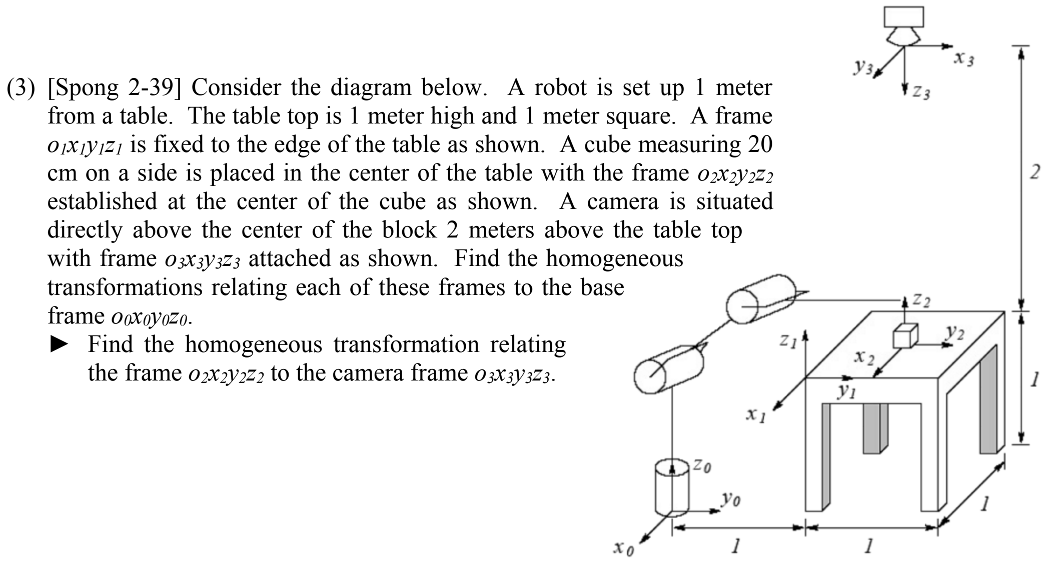
$$T_1^0 = \begin{pmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & -1 & 0 \\ -1 & 0 & 0 & 1 \\ 0 & 0 & 0 & 1 \end{pmatrix}, T_2^0 = \begin{pmatrix} 0 & 0 & -1 & 0 \\ -1 & 0 & 0 & 1 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}, T_2^1 = \begin{pmatrix} 0 & -1 & 0 & 1 \\ 0 & 0 & -1 & 0 \\ 1 & 0 & 0 & -1 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

Given the above transformations $T_1^0 T_2^1$ is found and checked to be the same as T_2^0

$$\begin{aligned} T_1^0 T_2^1 &= \begin{pmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & -1 & 0 \\ -1 & 0 & 0 & 1 \\ 0 & 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} 0 & -1 & 0 & 1 \\ 0 & 0 & -1 & 0 \\ 1 & 0 & 0 & -1 \\ 0 & 0 & 0 & 1 \end{pmatrix} \\ &= \begin{pmatrix} 0 & 0 & -1 & 0 \\ -1 & 0 & 0 & 1 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \end{aligned}$$

Which is the same as T_2^0 as expected.

0.3 Problem 3



(3) [Spong 2-39] Consider the diagram below. A robot is set up 1 meter from a table. The table top is 1 meter high and 1 meter square. A frame $o_1x_1y_1z_1$ is fixed to the edge of the table as shown. A cube measuring 20 cm on a side is placed in the center of the table with the frame $o_2x_2y_2z_2$ established at the center of the cube as shown. A camera is situated directly above the center of the block 2 meters above the table top with frame $o_3x_3y_3z_3$ attached as shown. Find the homogeneous transformations relating each of these frames to the base frame $o_0x_0y_0z_0$.

► Find the homogeneous transformation relating the frame $o_2x_2y_2z_2$ to the camera frame $o_3x_3y_3z_3$.

Figure 2: problem 3 description

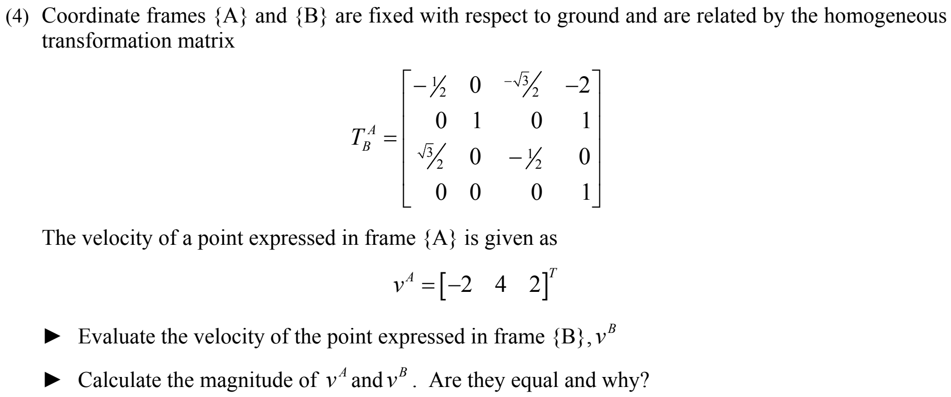
The distance from the table surface to the center of the small cube is 0.1 meter. The goal is to determine T_1^0, T_2^0, T_3^0 and T_3^2. By direct inspection of the given figure the following transformations are obtained

$$T_1^0 = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 1 \end{pmatrix}, T_2^0 = \begin{pmatrix} 1 & 0 & 0 & -0.5 \\ 0 & 1 & 0 & 1.5 \\ 0 & 0 & 1 & 1.1 \\ 0 & 0 & 0 & 1 \end{pmatrix}, T_3^0 = \begin{pmatrix} 0 & 1 & 0 & -0.5 \\ 1 & 0 & 0 & 1.5 \\ 0 & 0 & -1 & 3 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

In each of the above, the first column of T_i^{i-1} is the projection of $\hat{x}$ in frame i into frame $i-1$ and the second is the projection of $\hat{y}$ in frame i into frame $i-1$ and the third column the projection of $\hat{z}$ in frame i into frame $i-1$. The fourth column of T is the position vector of the center of frame i expressed in frame $i-1$. By inspection T_3^2 is found to be

$$T_3^2 = \begin{pmatrix} 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 0 & -1 & 1.9 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

0.4 Problem 4



(4) Coordinate frames {A} and {B} are fixed with respect to ground and are related by the homogeneous transformation matrix

$$T_B^A = \begin{bmatrix} -1/2 & 0 & -\sqrt{3}/2 & -2 \\ 0 & 1 & 0 & 1 \\ \sqrt{3}/2 & 0 & -1/2 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

The velocity of a point expressed in frame {A} is given as

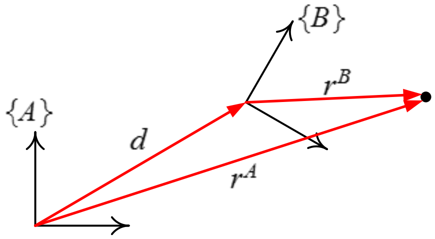
$$v^A = \begin{bmatrix} -2 & 4 & 2 \end{bmatrix}^T$$

- ► Evaluate the velocity of the point expressed in frame {B}, v^B
- ► Calculate the magnitude of v^A and v^B. Are they equal and why?

Figure 3: problem 4 description

PART 1:

Let d be the position vector of the origin of frame $\{B\}$ relative to frame $\{A\}$. Let r^A be the position vector of a point relative to frame $\{A\}$, and r^B be the position vector of the point relative to frame $\{B\}$ as shown in the following diagram



From the above diagram r^A is found as

$$r^A = R_B^A r^B + d \tag{1}$$

From the problem statement, $R_B^A = \begin{pmatrix} \frac{-1}{2} & 0 & \frac{-\sqrt{3}}{2} \\ 0 & 1 & 0 \\ \frac{\sqrt{3}}{2} & 0 & \frac{-1}{2} \end{pmatrix}$ and $d = \begin{pmatrix} -2 \\ 4 \\ 2 \end{pmatrix}$

Taking time derivative of (1) and using the chain rule results in

$$v^A = \frac{dR_B^A}{dt} r^B + R_B^A v^B + \frac{d}{dt} d \tag{2}$$

R_B^A does not depend on time, therefore $\frac{dR_B^A}{dt} = 0$. Since frame $\{B\}$ does not move relative to frame $\{A\}$, therefore $\frac{d}{dt} d = 0$. Using these results (2) simplifies to

$$v^A = R_B^A v^B$$

Solving for v^B from the above, and noting that $\left(R_B^A\right)^{-1} = \left(R_B^A\right)^T$ since it is a rotation matrix gives

$$v^B = \left(R_B^A\right)^T v^A$$

Substituting the values given in the problem in the above results in

$$\begin{aligned} v^B &= \begin{pmatrix} \frac{-1}{2} & 0 & \frac{-\sqrt{3}}{2} \\ 0 & 1 & 0 \\ \frac{\sqrt{3}}{2} & 0 & \frac{-1}{2} \end{pmatrix}^T \begin{pmatrix} -2 \\ 4 \\ 2 \end{pmatrix} \\ &= \begin{pmatrix} -\frac{1}{2} & 0 & \frac{1}{2}\sqrt{3} \\ 0 & 1 & 0 \\ -\frac{1}{2}\sqrt{3} & 0 & -\frac{1}{2} \end{pmatrix} \begin{pmatrix} -2 \\ 4 \\ 2 \end{pmatrix} \end{aligned}$$

Therefore

$$v^B = \begin{pmatrix} \sqrt{3} + 1 \\ 4 \\ \sqrt{3} - 1 \end{pmatrix}$$

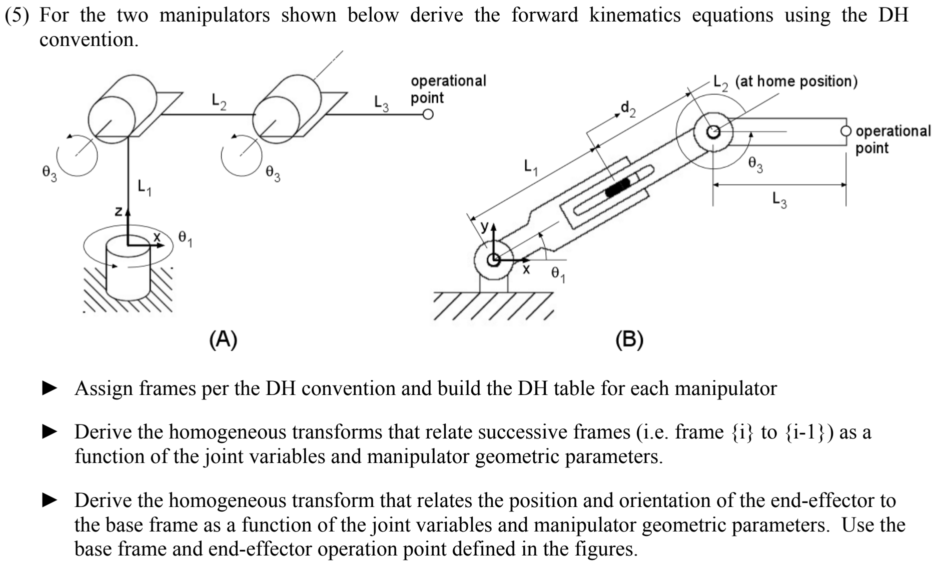
PART 2:

The norm of the velocity vectors are

$$\begin{aligned} \left\| v^A \right\| &= \left\| \begin{pmatrix} -2 \\ 4 \\ 2 \end{pmatrix} \right\| = 2\sqrt{6} \\ \left\| v^B \right\| &= \begin{pmatrix} \sqrt{3} + 1 \\ 4 \\ \sqrt{3} - 1 \end{pmatrix} = 2\sqrt{6} \end{aligned}$$

They have the same magnitude. The reason is that frame $\{B\}$ itself does not move nor rotate relative to $\{A\}$. Therefore frame B is fixed relative from frame $\{A\}$. Hence the velocity of a point relative to frame $\{A\}$ will have the same magnitude relative to $\{B\}$. The velocity vector has different representation depending on the frame of reference, but has the same magnitude.

0.5 Problem 5



(5) For the two manipulators shown below derive the forward kinematics equations using the DH convention.

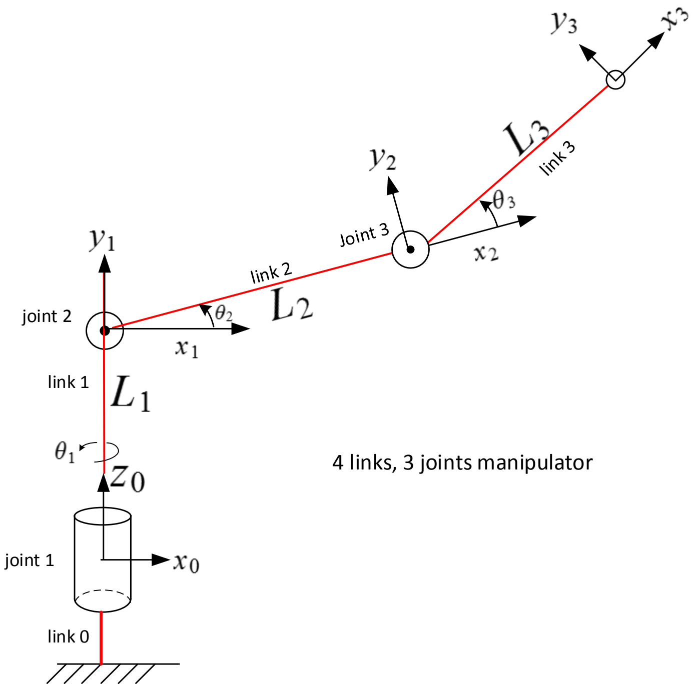
- Assign frames per the DH convention and build the DH table for each manipulator
- Derive the homogeneous transforms that relate successive frames (i.e. frame {i} to {i-1}) as a function of the joint variables and manipulator geometric parameters.
- Derive the homogeneous transform that relates the position and orientation of the end-effector to the base frame as a function of the joint variables and manipulator geometric parameters. Use the base frame and end-effector operation point defined in the figures.

Figure 4: problem 5 description

0.5.1 solution

PART (A)

The first step is to assign the z_i axes for each link as follows



The four Denavit-Hartenberg parameters are defined as follows[1]

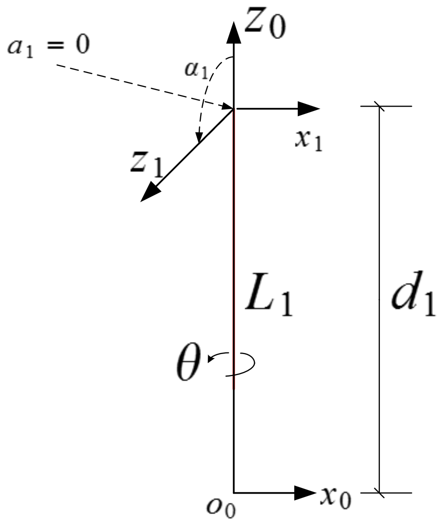
1. a_i (link length). The distance between axis z_{i-1} and z_i measured along x_i.
2. α_i (link twist angle). The angle between z_{i-1} and z_i measured in a plane normal to x_i using the right hand rule, around x_i (not x_{i-1}) to determine the positive sense of this angle.
3. d_i (link offset). The distance from origin o_{i-1} to the intersection of the x_i axis with z_{i-1} measured along z_{i-1} axis.
4. θ_i (Joint angle). The angle from x_{i-1} to x_i measured in plane normal to z_{i-1}.

When assigning the frames using the above rules, we need to insure[2] that $x_{i+1} \perp z_i$ and x_{i+1} intersects z_i. Using the above rules the DH table is written down. There is one row in the table for each link. Hence there will be three rows. Link 0 is the base link and attached to the ground and does not show in the table.

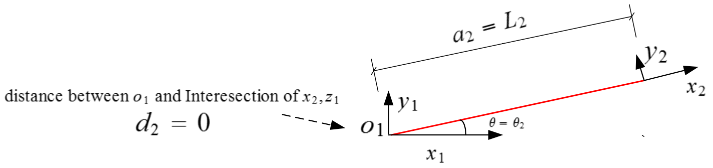
	a (link length)	α (link twist angle)	d (link offset)	θ (Joint angle)
link 1	0	90^0	L_1	θ_1
link 2	L_2	0	0	θ_2
link 3	L_3	0	0	θ_3

The following diagram shows the parameters for the first link

[1] Textbook, page 80.
[2] Text book, page 78.

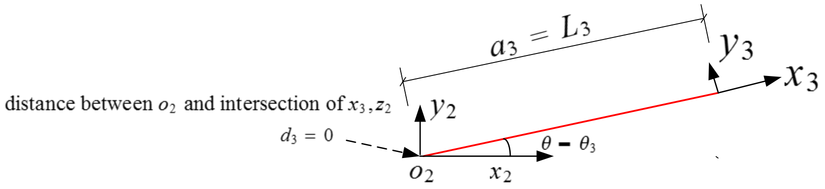


The following diagram shows the parameters for the second link



since $z_1 \| z_1$ then $\alpha_2 = 0$

The following diagram shows the parameters for the third link



since $z_3 \| z_2$ then $\alpha_3 = 0^0$

Now the forward transformations using equation (3.10) on page 77 of the textbook is found

$$A = \begin{pmatrix} C_\theta & -S_\theta C_\alpha & S_\theta S_\alpha & aC_\theta \\ S_\theta & C_\theta C_\alpha & -C_\theta S_\alpha & aS_\theta \\ 0 & S_\alpha & C_\alpha & d \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

Hence

$$A_1 = \begin{pmatrix} \cos\theta_1 & -\sin\theta_1\cos\frac{\pi}{2} & \sin\theta_1\sin\frac{\pi}{2} & 0 \\ \sin\theta_1 & \cos\theta_1\cos\frac{\pi}{2} & -\cos\theta_1\sin\frac{\pi}{2} & 0 \\ 0 & \sin\frac{\pi}{2} & \cos\frac{\pi}{2} & L_1 \\ 0 & 0 & 0 & 1 \end{pmatrix} = \begin{pmatrix} \cos\theta_1 & 0 & \sin\theta_1 & 0 \\ \sin\theta_1 & 0 & -\cos\theta_1 & 0 \\ 0 & 1 & 0 & L_1 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

$$A_2 = \begin{pmatrix} \cos\theta_2 & -\sin\theta_2\cos 0 & \sin\theta_2\sin 0 & L_2\cos\theta_2 \\ \sin\theta_2 & \cos\theta_2\cos 0 & -\cos\theta_2\sin 0 & L_2\sin\theta_2 \\ 0 & \sin 0 & \cos 0 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} = \begin{pmatrix} \cos\theta_2 & -\sin\theta_2 & 0 & L_2\cos\theta_2 \\ \sin\theta_2 & \cos\theta_2 & 0 & L_2\sin\theta_2 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

$$A_3 = \begin{pmatrix} \cos\theta_3 & -\sin\theta_3\cos 0 & \sin\theta_3\sin 0 & L_3\cos\theta_3 \\ \sin\theta_3 & \cos\theta_3\cos 0 & -\cos\theta_3\sin 0 & L_3\sin\theta_3 \\ 0 & \sin 0 & \cos 0 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} = \begin{pmatrix} \cos\theta_3 & -\sin\theta_3 & 0 & L_3\cos\theta_3 \\ \sin\theta_3 & \cos\theta_3 & 0 & L_3\sin\theta_3 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

Using the above, T_3^0 is found

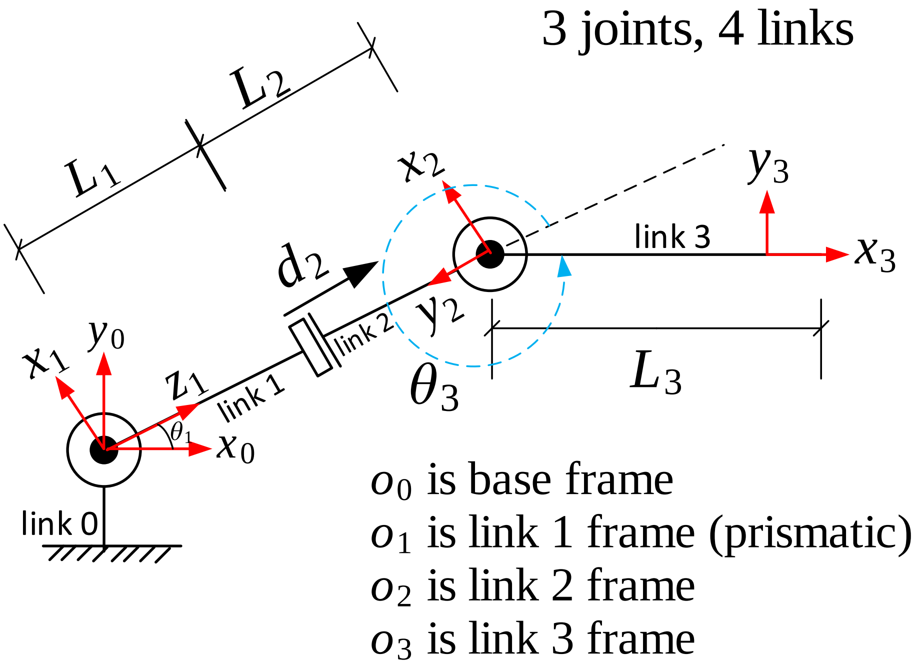
$$\begin{aligned} T_3^0 &= A_1 A_2 A_3 \\ &= \begin{pmatrix} \cos\theta_1 & 0 & \sin\theta_1 & 0 \\ \sin\theta_1 & 0 & -\cos\theta_1 & 0 \\ 0 & 1 & 0 & L_1 \\ 0 & 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} \cos\theta_2 & -\sin\theta_2 & 0 & L_2\cos\theta_2 \\ \sin\theta_2 & \cos\theta_2 & 0 & L_2\sin\theta_2 \\ 0 & 0 & 1 & L_1 \\ 0 & 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} \cos\theta_3 & -\sin\theta_3 & 0 & L_3\cos\theta_3 \\ \sin\theta_3 & \cos\theta_3 & 0 & L_3\sin\theta_3 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \end{aligned}$$

For verification of the above, let $\theta_1 = 0, \theta_2 = 0, \theta_3 = 0$ then the fourth column of T_3^0 gives the position vector of the end effector relative to the base when the manipulator is in the position in the problem. Substituting these values for the angles gives

$$T_3^0 = \begin{pmatrix} 1 & 0 & 0 & L_2 + L_3 \\ 0 & 0 & -1 & 0 \\ 0 & 1 & 0 & L_1 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

The above says that the end effector is at position vector $p^0 = (L_2 + L_3, 0, L_1)$ which means $x_3 = L_2 + L_3, y_3 = 0, z_3 = L_1$. From the diagram this result is correct.

PART (B) For the second manipulator, the same steps were repeated. The first step is to assign the axes for each link as follows



The DH table is written down

	a (link length)	α (link twist angle)	d (link offset)	θ (Joint angle)
link 1	0	90^0	0	$\theta_1 + 90^0$
link 2	0	-90^0	$L_1 + L_2 + d_2$	0
link 3	L_3	0	0	$\theta_3 - 90^0$

The forward transformations using equation (3.10) on page 77 of the textbook gives

$$A = \begin{pmatrix} C_\theta & -S_\theta C_\alpha & S_\theta S_\alpha & aC_\theta \\ S_\theta & C_\theta C_\alpha & -C_\theta S_\alpha & aS_\theta \\ 0 & S_\alpha & C_\alpha & d \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

Using the above, and noting that $\cos\left(x + 90^0\right) = -\sin x$,$\sin\left(x + 90^0\right) = \cos x$,$\cos\left(x - 90^0\right) = \sin x$ and $\sin\left(x - 90^0\right) = -\cos\left(x\right)$ results in

$$A_1 = \begin{pmatrix} \cos(\theta_1+90) & -\sin(\theta_1+90)\cos\frac{\pi}{2} & \sin(\theta_1+90)\sin\frac{\pi}{2} & 0 \\ \sin(\theta_1+90) & \cos(\theta_1+90)\cos\frac{\pi}{2} & -\cos(\theta_1+90)\sin\frac{\pi}{2} & 0 \\ 0 & \sin\frac{\pi}{2} & \cos\frac{\pi}{2} & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} = \begin{pmatrix} -\sin\theta_1 & 0 & \cos\theta_1 & 0 \\ \cos\theta_1 & 0 & \sin\theta_1 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

$$A_2 = \begin{pmatrix} \cos 0 & -\sin 0\cos(-90^0) & \sin 0\sin(-90^0) & 0\cos 0 \\ \sin 0 & \cos 0\cos(-90^0) & -\cos 0\sin(-90^0) & 0\sin 0 \\ 0 & \sin(-90^0) & \cos(-90^0) & L_1+L_2+d_2 \\ 0 & 0 & 0 & 1 \end{pmatrix} = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & -1 & 0 & L_1+L_2+d_2 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

$$A_2 = \begin{pmatrix} \cos\left(\theta_3-90^0\right) & -\sin\left(\theta_3-90^0\right)\cos 0 & \sin\left(\theta_3-90^0\right)\sin 0 & L_3\cos\left(\theta_3-90^0\right) \\ \sin\left(\theta_3-90^0\right) & \cos\left(\theta_3-90^0\right)\cos 0 & -\cos\left(\theta_3-90^0\right)\sin 0 & L_3\sin\left(\theta_3-90^0\right) \\ 0 & \sin 0 & \cos 0 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} = \begin{pmatrix} \sin\theta_3 & \cos\theta_3 & 0 & L_3\sin\theta_3 \\ -\cos\theta_3 & \sin\theta_3 & 0 & -L_3\cos\theta_3 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

From the above

$$\begin{aligned} T_3^0 &= A_1A_2A_3 \\ &= \begin{pmatrix} -\sin\theta_1 & 0 & \cos\theta_1 & 0 \\ \cos\theta_1 & 0 & \sin\theta_1 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & -1 & 0 & L_1+L_2+d_2 \\ 0 & 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} \sin\theta_3 & \cos\theta_3 & 0 & L_3\sin\theta_3 \\ -\cos\theta_3 & \sin\theta_3 & 0 & -L_3\cos\theta_3 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \\ &= \begin{pmatrix} \cos\theta_1\cos\theta_3-\sin\theta_1\sin\theta_3 & -\cos\theta_1\sin\theta_3-\cos\theta_3\sin\theta_1 & 0 & \cos\theta_1\left(L_1+L_2+d_2\right)+L_3\cos\theta_1\cos\theta_3-L_3\sin\theta_1\sin\theta_3 \\ \cos\theta_1\sin\theta_3+\cos\theta_3\sin\theta_1 & \cos\theta_1\cos\theta_3-\sin\theta_1\sin\theta_3 & 0 & \sin\theta_1\left(L_1+L_2+d_2\right)+L_3\cos\theta_1\sin\theta_3+L_3\cos\theta_3\sin\theta_1 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \end{aligned}$$

To verify the above, let $\theta_1 = 0, \theta_3 = 360^0, d_2 = 0$ then the fourth column of T_3^0 gives the position vector of the end effector relative to the base when the manipulator is in a straight horizontal position

$$T_3^0 = \begin{pmatrix} 1 & 0 & 0 & L_1+L_2+L_3 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

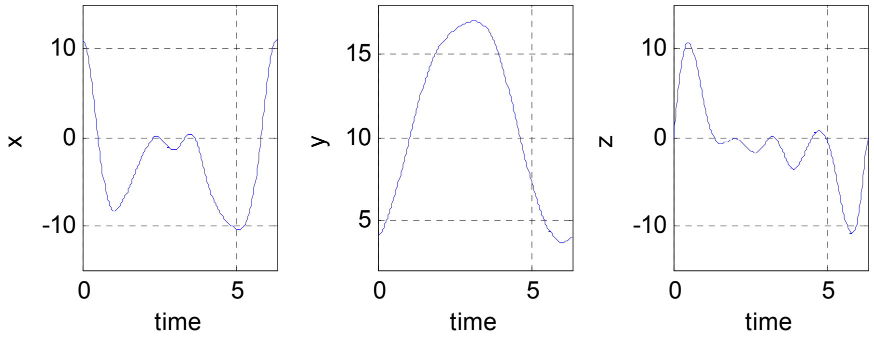
The above results show that $x_3 = L_1+L_2+L_3$ which is the expected result.

0.6 Problem 6

0.6.1 problem description

- Write Matlab code to plot the position (x, y, and z coordinates) of the end-effector (point E) as a function of time. Your plots should match the plots shown below. The joint motion, as a function of time, is given below. Make sure to include all supporting functions, including any custom plotting routines, with your homework submission. Your homework submission must provide clear, easy instructions to run you Matlab code

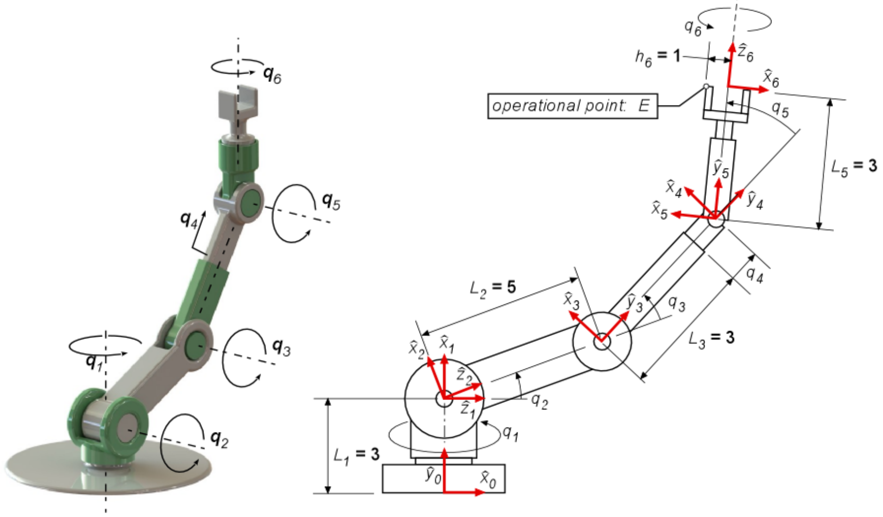
  Joint variable inputs:
  $q_1 = -\pi \sin(t),\ q_2 = \frac{\pi}{4}(1-\cos(t)),\ q_3 = \frac{\pi}{4}\sin(t),\ q_4 = \frac{1}{2}L_3(1-\cos(t)),\ q_5 = -\frac{\pi}{4}\sin(t),\ q_6 = \frac{\pi}{4}\sin(t)$
  Please animate the system over the time interval $t = 0:2\pi$.



- Write Matlab code to animate the manipulator using the Matlab scripts provided on the Learn@UW course page. The joint motion, as a function of time, is given above. Make sure to include all supporting functions, including any custom plotting routines, with your homework submission. Your homework submission must provide clear, easy instructions to run you Matlab code. To maintain consistency, please use the following rendering window view parameters.

```
%----set rendering window view parameters

% figure handle
f_handle = 1;

% axis limits
axis_limits = [-10 10 0 10 -10 10];

% camera position
render_view = [-1 1 -1];

% vertical orientation
view_up = [0 1 0];

% initialize rendering view
SetRenderingViewParameters(axis_limits,render_view,view_up,f_handle);
```

(6) For the manipulator shown in the adjacent figure



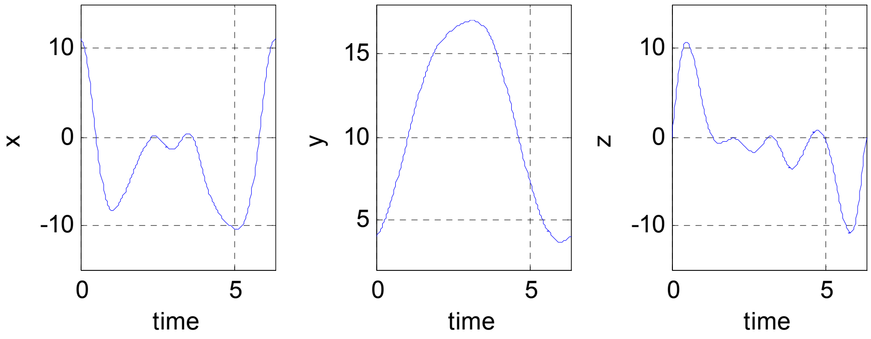
► Derive the homogeneous transforms that relate the successive frames as a function of the joint variable and manipulator geometric parameters (i.e. find: $T_1^0, T_2^1, T_3^2, T_4^3, T_5^4, T_6^5$). Use the coordinate frames as defined in the figure. Note that the frames are fixed to the links at various locations (e.g. proximal end, distal end). Assume that q_1 equals zero when the manipulator lies in the plane of the page. Attendee

► Write Matlab code to calculate the homogeneous transforms derived above as a function of the joint variables q_1, q_2, q_3, q_4, and q_5.

► Write Matlab code to calculate the homogeneous transformation matrices that describe the frame displacements relative to the ground frame {0} (i.e. numerically evaluate $T_1^0, T_2^0, T_3^0, T_4^0, T_5^0, T_6^0$).

- Write Matlab code to plot the position (x, y, and z coordinates) of the end-effector (point E) as a function of time. Your plots should match the plots shown below. The joint motion, as a function of time, is given below. Make sure to include all supporting functions, including any custom plotting routines, with your homework submission. Your homework submission must provide clear, easy instructions to run you Matlab code

  Joint variable inputs:
  $q_1 = -\pi\sin(t),\; q_2 = \frac{\pi}{4}(1-\cos(t)),\; q_3 = \frac{\pi}{4}\sin(t),\; q_4 = \frac{1}{2}L_3(1-\cos(t)),\; q_5 = -\frac{\pi}{4}\sin(t),\; q_6 = \frac{\pi}{4}\sin(t)$
  Please animate the system over the time interval $t = 0:2\pi$.



- Write Matlab code to animate the manipulator using the Matlab scripts provided on the Learn@UW course page. The joint motion, as a function of time, is given above. Make sure to include all supporting functions, including any custom plotting routines, with your homework submission. Your homework submission must provide clear, easy instructions to run you Matlab code. To maintain consistency, please use the following rendering window view parameters.

```
%----set rendering window view parameters

% figure handle
f_handle = 1;

% axis limits
axis_limits = [-10 10 0 10 -10 10];

% camera position
render_view = [-1 1 -1];

% vertical orientation
view_up = [0 1 0];

% initialize rendering view
SetRenderingViewParameters(axis_limits,render_view,view_up,f_handle);
```

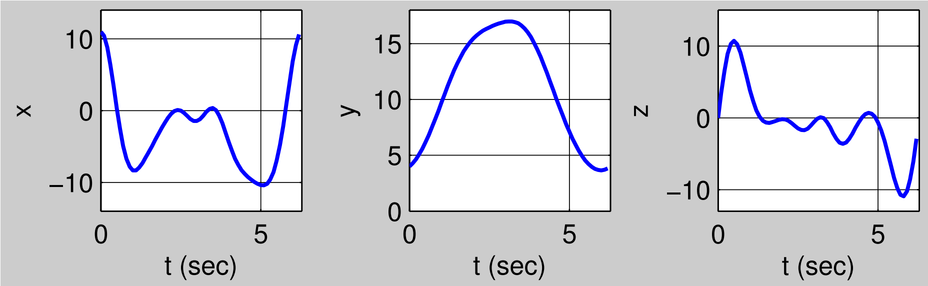
0.6.2 solution

The homogeneous transformation T_i^{i-1} was derived by inspection giving the following results

$$T_1^0=\begin{pmatrix}0&\sin q_1&\cos q_1&0\\1&0&0&L_1\\0&\cos q_1&-\sin q_1&0\\0&0&0&1\end{pmatrix},T_2^1=\begin{pmatrix}\cos q_2&0&\sin q_2&0\\0&1&0&0\\-\sin q_2&0&\cos q_2&0\\0&0&0&1\end{pmatrix},T_3^2=\begin{pmatrix}\cos q_3&\sin q_3&0&0\\0&0&-1&0\\-\sin q_3&\cos q_3&0&L_2\\0&0&0&1\end{pmatrix}$$

$$T_4^3=\begin{pmatrix}1&0&0&0\\0&1&0&L_3+q_4\\0&0&1&0\\0&0&0&1\end{pmatrix},T_5^4=\begin{pmatrix}\cos q_5&\sin q_5&0&0\\-\sin q_5&\cos q_5&0&0\\0&0&1&0\\0&0&0&1\end{pmatrix},T_6^5=\begin{pmatrix}-\cos q_6&\sin q_6&0&0\\0&0&1&L_5\\\sin q_6&\cos q_6&0&0\\0&0&0&1\end{pmatrix}$$

The following is plot of the x, y, z coordinates of the end effector E



The following Matlab script **problem_6_part_1.m** calculates the homogeneous transformation T_6^0 and plots the above figures

```
%This scripts plots the x,y,z coordinates of the end effector E
%for problem 5, HW1 , ME 739.
%to run, type this script name on the Matlab console
%    problem_6_part_1
%The matlab path must include the ME 739 rendering software
%Nasser M. Abbasi 2/16/15

clear all; close all;
%define syms to use to build the T matrices
syms q1 q2 q3 q4 q5 q6 L1 L2 L3 L5 t
L1 = 3;
L2 = 5;
L3 = 3;
L5 = 3;
h6 = 1;
q1 = -pi*sin(t);
q2 = pi/4*(1-cos(t));
q3 = pi/4*sin(t);
q4 = 1/2*L3*(1-cos(t));
q5 = -pi/4*sin(t);
q6 = pi/4*sin(t);

%define the 6 transformation matrices T01 to T56 in syms
T01 = [0        sin(q1) cos(q1)   0;
       1        0       0         L1;
       0        cos(q1) -sin(q1)  0;
       0        0       0         1];

T12 = [cos(q2)  0       sin(q2)  0;
       0        1       0        0;
       -sin(q2) 0       cos(q2)  0;
       0        0       0        1];

T23 = [cos(q3)  sin(q3) 0        0;
       0        0       -1       0;
       -sin(q3) cos(q3) 0        L2;
       0        0       0        1];

T34 = [1        0       0        0;
       0        1       0        L3+q4;
       0        0       1        0;
       0        0       0        1];

T45 = [cos(q5)  sin(q5) 0        0;
       -sin(q5) cos(q5) 0        0;
       0        0       1        0;
       0        0       0        1];

T56 = [-cos(q6) sin(q6) 0        0;
       0        0       1        L5;
       sin(q6)  cos(q6) 0        0;
       0        0       0        1];

%Now obtain T06 to allow finding the end effector coordinates
T06 = T01*T12*T23*T34*T45*T56;

%handle to function to evaluate T06 at each instance of time
endPos = @(t0) subs(T06,t,t0)

%set up time scale, and evaluate the end effector coordinates
%saving result in a matrix for plotting later.
timeScale = 0:.1:2*pi;
coords    = zeros(length(timeScale),3);

%generate the coordinates of the end effector
for i = 1:length(timeScale)
    p = endPos(timeScale(i))*[-h6 0 0 1]’;
    coords(i,:) = p(1:3);
end

%now plot the result. First col is the x-coordinates,
%second col is y-coord, and third col is z-coordinate.
subplot(1,3,1);
plot(timeScale,coords(:,1), ’LineWidth’,1.5);
xlabel(’t (sec)’); ylabel(’x’);
grid; axis square; xlim([0 2*pi]);ylim([-14 14]);
set(gca, ’GridLineStyle’, ’-’);

subplot(1,3,2);
```

```
plot(timeScale,coords(:,2), 'LineWidth',1.5);
xlabel('t (sec)'); ylabel('y');
grid; axis square; xlim([0 2*pi]);ylim([0 18]);
%axis square; axis tight
set(gca, 'GridLineStyle', '-');

subplot(1,3,3);
plot(timeScale,coords(:,3),'LineWidth',1.5);
xlabel('t (sec)'); ylabel('z');
grid; axis square; xlim([0 2*pi]);ylim([-13 15]);
set(gca, 'GridLineStyle', '-');
%export_fig(gcf,'problem_6_part_1.pdf');
```

The following matlab script calculates $T_1^0, T_2^0, T_3^0, T_4^0, T_5^0, T_6^0$, where $T_2^0 = T_1^0 T_2^1, T_3^0 = T_2^0 T_3^2, T_4^0 = T_3^0 T_4^3, T_5^0 = T_4^0 T_5^4, T_6^0 = T_5^0 T_6^5$

For example, for T_2^0 the result is

$$T_1^0 = \begin{pmatrix} 0 & \sin q_1 & \cos q_1 & 0 \\ 1 & 0 & 0 & L_1 \\ 0 & \cos q_1 & -\sin q_1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

$$T_2^0 = T_1^0 T_2^1 = \begin{pmatrix} 0 & \sin q_1 & \cos q_1 & 0 \\ 1 & 0 & 0 & L_1 \\ 0 & \cos q_1 & -\sin q_1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} \cos q_2 & 0 & \sin q_2 & 0 \\ 0 & 1 & 0 & 0 \\ -\sin q_2 & 0 & \cos q_2 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} = \begin{pmatrix} -\cos q_1 \sin q_2 & \sin q_1 & \cos q_1 \cos q_2 & 0 \\ \cos q_2 & 0 & \sin q_2 & L_1 \\ \sin q_1 \sin q_2 & \cos q_1 & -\cos q_2 \sin q_1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

The complete calculation was done in the following Matlab script `problem_6_part_2.m`. The script is run by typing its name on the Matlab console.

```
%This calculates T01,T02,T03,T04,T05,T06 numerically
%for problem 5, HW1 , ME 739.
%to run, just type this script name on the Matlab console
%    problem_6_part_2
%
%The matlab path must include the ME 739 rendering software
%Nasser M. Abbasi 2/16/15

clear all; close all;
%define syms to use to build the T matrices
syms q1 q2 q3 q4 q5 q6 L1 L2 L3 L5 t
L1 = 3;
L2 = 5;
L3 = 3;
L5 = 3;
h6 = 1;
q1 = -pi*sin(t);
q2 = pi/4*(1-cos(t));
q3 = pi/4*sin(t);
```

```
q4 = 1/2*L3*(1-cos(t));
q5 = -pi/4*sin(t);
q6 = pi/4*sin(t);

%define the 6 transformation matrices T01 to T56 in syms
%define the 6 transformation matrices T01 to T56 in syms
T01 = [0        sin(q1) cos(q1)   0;
       1        0       0         L1;
       0        cos(q1) -sin(q1)  0;
       0        0       0         1];

T12 = [cos(q2)  0       sin(q2)  0;
       0        1       0        0;
       -sin(q2) 0       cos(q2)  0;
       0        0       0        1];

T23 = [cos(q3)  sin(q3) 0        0;
       0        0       -1       0;
       -sin(q3) cos(q3) 0        L2;
       0        0       0        1];

T34 = [1        0       0        0;
       0        1       0        L3+q4;
       0        0       1        0;
       0        0       0        1];

T45 = [cos(q5)  sin(q5) 0        0;
       -sin(q5) cos(q5) 0        0;
       0        0       1        0;
       0        0       0        1];

T56 = [-cos(q6) sin(q6) 0        0;
       0        0       1        L5;
       sin(q6)  cos(q6) 0        0;
       0        0       0        1];

%Now calculate T02,T03,T04,T05,T06
T02 = T01*T12;
T03 = T02*T23;
T04 = T03*T34;
T05 = T04*T45;
T06 = T05*T56;

%handle to function to evaluate each Tij at each instance of time
calcT = @(T,t0) double(subs(T,t,t0));

%now calculate all the T’s at some specific time. The problem
%does not says what time instance to use, so we use t=0 for
%illustration

timeToShow = 1; %change this to different time as needed

fprintf(’T01 at t=1 is \n’); calcT(T01,timeToShow)
fprintf(’T02 at t=1 is \n’); calcT(T02,timeToShow)
fprintf('T03 at t=1 is \n'); calcT(T03,timeToShow)
fprintf('T04 at t=1 is \n'); calcT(T04,timeToShow)
fprintf('T05 at t=1 is \n'); calcT(T05,timeToShow)
fprintf('T06 at t=1 is \n'); calcT(T06,timeToShow)
```

The above script calculates numerically all the transformation matrices using the joint variable inputs given in the problem. At the end it prints each matrix. The problem did not indicate for which value of t to use to calculate the matrices, hence for illustration these are displayed for $t = 0$ and $t = 1$ second. A variable inside the script can be used to change the time instance. The following is the output from running the above script for illustration

At $t = 0$ the output is

```
T01 at t=0 is
     0     0     1     0
     1     0     0     3
     0     1     0     0
     0     0     0     1

T02 at t=0 is
     0     0     1     0
     1     0     0     3
     0     1     0     0
     0     0     0     1

T03 at t=0 is
     0     1     0     5
     1     0     0     3
     0     0    -1     0
     0     0     0     1

T04 at t=0 is
     0     1     0     8
     1     0     0     3
     0     0    -1     0
     0     0     0     1

T05 at t=0 is
     0     1     0     8
     1     0     0     3
     0     0    -1     0
     0     0     0     1

T06 at t=0 is
     0     0     1    11
    -1     0     0     3
     0    -1     0     0
     0     0     0     1
```

At $t = 1$ the output is

```
T01 at t=1 is
            0      -0.4777     -0.87852            0
            1            0            0            3
            0     -0.87852       0.4777            0
            0            0            0            1

T02 at t=1 is
      0.31034      -0.4777     -0.82188            0
      0.93553            0      0.35325            3
     -0.16875     -0.87852       0.4469            0
            0            0            0            1

T03 at t=1 is
      0.74949     -0.45834       0.4777      -4.1094
      0.52172      0.85312            0       4.7663
     -0.40753      0.24922      0.87852       2.2345
            0            0            0            1

T04 at t=1 is
      0.74949     -0.45834       0.4777      -5.8005
      0.52172      0.85312            0       7.9139
     -0.40753      0.24922      0.87852        3.154
            0            0            0            1

T05 at t=1 is
      0.31034     -0.82188       0.4777      -5.8005
      0.93553      0.35325            0       7.9139
     -0.16875       0.4469      0.87852        3.154
            0            0            0            1

T06 at t=1 is
     0.048223      0.56761     -0.82188      -8.2661
     -0.73855      0.57425      0.35325       8.9736
      0.67247      0.58997       0.4469       4.4947
            0            0            0            1
```

The manipulator was animated using the UW software. The following script `problem_6_part_3.m` written for this purpose. Typing the name of the script in Matlab starts the animation.

This script `problem_6_part_3.m` assumes the Matlab path is set to include the UW rendering software.

```
%This calculates T01,T02,T03,T04,T05,T06 numerically
%for problem 5, HW1 , ME 739.
%to run, just type this script name on the Matlab console
%
%    problem_6_part_3
%The matlab path must include the ME 739 rendering software
%
%Nasser M. Abbasi 2/17/15

clear all; close all; clc;

%----set rendering window view parameters
```

```
f_handle = 1; % figure handle
axis_limits = [-10 10 0 13 -10 10]; %needed little bit more space
render_view = [-1 1 -1]; % camera position
view_up = [0 1 0]; % vertical orientation
% initialize rendering view
SetRenderingViewParameters(axis_limits,render_view,view_up,f_handle);

ADD_BASE = false; %set to TRUE to see base rendered, does not move.
DO_MOVIE = false; %set true to make frames for movie
ANIMATION_TIME = 10; %10 seconds to animation
DEL_T          = 0.05; %time betweeb each animation loop. smaller time
                       %make it run slower but more accurate

%define syms to use to build the T matrices
syms q1 q2 q3 q4 q5 q6 L1 L2 L3 L5 t
L1 = 3;
L2 = 5;
L3 = 3;
L5 = 3;
h6 = 1;
q1 = -pi*sin(t);
q2 = pi/4*(1-cos(t));
q3 = pi/4*sin(t);
q4 = 1/2*L3*(1-cos(t));
q5 = -pi/4*sin(t);
q6 = pi/4*sin(t);

%define the 6 transformation matrices T01 to T56 in syms
%define the 6 transformation matrices T01 to T56 in syms
T01 = [0        sin(q1) cos(q1)   0;
       1        0       0         L1;
       0        cos(q1) -sin(q1)  0;
       0        0       0         1];

T12 = [cos(q2)  0       sin(q2)  0;
       0        1       0        0;
       -sin(q2) 0       cos(q2)  0;
       0        0       0        1];

T23 = [cos(q3)  sin(q3) 0        0;
       0        0       -1       0;
       -sin(q3) cos(q3) 0        L2;
       0        0       0        1];

T34 = [1        0       0        0;
       0        1       0        L3+q4;
       0        0       1        0;
       0        0       0        1];

T45 = [cos(q5)  sin(q5) 0        0;
       -sin(q5) cos(q5) 0        0;
       0        0       1        0;
       0        0       0        1];
```

```
T56 = [-cos(q6) sin(q6) 0       0;
       0        0       1       L5;
       sin(q6)  cos(q6) 0       0;
       0        0       0       1];

   %Now calculate T02,T03,T04,T05,T06
T02 = T01*T12;
T03 = T02*T23;
T04 = T03*T34;
T05 = T04*T45;
T06 = T05*T56;

%handle to function to evaluate each Tij at each instance of time
%this is called during running the animation
calcT = @(T,t0) double(subs(T,t,t0));

%base, does not move
if ADD_BASE
    linkColor = [0 0 0]; plotFrame=0; normalized_location=-1;
    nSides = 4; radius = 4; r = L1;  axis_aligned = 2;
    d0 = CreateLinkRendering(r ,radius, nSides, axis_aligned ,normalized_location, ...
        linkColor,plotFrame, f_handle);
end

%now create all the links
linkColor = [1 .1 0]; plotFrame=0; normalized_location=-1;
nSides=20; radius=2; r=L1;  axis_aligned = 1;
d1 = CreateLinkRendering(r ,radius, nSides, axis_aligned ,normalized_location, ...
    linkColor,plotFrame, f_handle);

linkColor = [.5 .2 1]; plotFrame=0; normalized_location=-1;
nSides=4; r=L2;  radius=1.2; axis_aligned=3;
d2 = CreateLinkRendering(r ,radius, nSides,axis_aligned ,normalized_location, ...
    linkColor,plotFrame,  f_handle);

linkColor = [1 .1 1]; plotFrame=0; normalized_location=-1;
nSides=4; r=L3;  radius=1.15; axis_aligned=2;
d3 = CreateLinkRendering(r ,radius, nSides,axis_aligned ,normalized_location,...
    linkColor,plotFrame,f_handle);

linkColor = [0 0 1]; plotFrame=0; normalized_location=1;
nSides=4;   r=L5;  radius=.9; axis_aligned=2;
d4 = CreateLinkRendering(r ,radius, nSides,axis_aligned,normalized_location,...
    linkColor,plotFrame,  f_handle);

linkColor = [.3 .5 .3]; plotFrame=0; normalized_location=-1;
nSides=4;   r=L5;  radius=.7; axis_aligned=2;
d5 = CreateLinkRendering(r ,radius, nSides,axis_aligned ,normalized_location,...
    linkColor,plotFrame,  f_handle);

linkColor = [.5 1 .5]; plotFrame=0; normalized_location=-1;
nSides=4;   r=L5;  radius=0.5; axis_aligned=3;
d6 = CreateLinkRendering(r ,radius, nSides,axis_aligned,normalized_location,...
    linkColor,plotFrame,  f_handle);
```

```
%now start the animation
timeScale = 0: DEL_T :ANIMATION_TIME;
k         = 0; %for screen shots counting, to make movie
for i = 1:length(timeScale)
    T = calcT(T01,timeScale(i));
    UpdateLink(d1,T);

    T = calcT(T02,timeScale(i));
    UpdateLink(d2,T);

    T = calcT(T03,timeScale(i));
    UpdateLink(d3,T);

    T = calcT(T04,timeScale(i));
    UpdateLink(d4,T);

    T = calcT(T05,timeScale(i));
    UpdateLink(d5,T);

    T = calcT(T06,timeScale(i));
    UpdateLink(d6,T);
    title(sprintf(’time = %3.3f (sec)’,timeScale(i)));

    if DO_MOVIE
       k = k+1;
       I = getframe(gcf);
       imwrite(I.cdata, sprintf(’frame%d.png’,k));
    end

    %p=T*[-h6 0 0 1]’;  to show the end effector path if needed
    %plot3(p(1),p(2),p(3),’o’);
    drawnow
end
```

The following is a movie of the first few seconds of the Matlab run. (posted as animation gif in my site).