Nishant Kumar Bharali
906641308

HW - 3 (Robotics and Automation)

[1] Here is the MATLAB Implementation for the question with the results attached.

```
function T = forwardkinem(M, S, theta)
T = eye(4) % initialization of transformation matrix
[row, col] = size(theta)
for i = 1:row  % length iteration
  T = T * expm(bracket(S(:,i)) * theta(i))
end
T = T * M % pre-multiply to M
    function S_matrix = bracket(S1)
        S_matrix = [ 0 -S1(3) S1(2) S1(4);
            S1(3) 0 -S1(1) S1(5);
            -S1(2) S1(1) 0 S1(6);
            0 0 0 0];
    end
end
```

```
>>
>>
>>
>>
>> forwardkinem(M, S, theta)
T =

Diagonal Matrix

   1   0   0   0
   0   1   0   0
   0   0   1   0
   0   0   0   1

row = 3
col = 1
T =

  -0.0000  -1.0000        0   1.0000
   1.0000   0.0000        0   1.0000
        0        0   1.0000        0
        0        0        0   1.0000

T =

  -0.7043  -0.0897   0.7043  -0.1594
   0.5449  -0.7043   0.4551   1.4206
   0.4551   0.7043   0.5449   0.6266
        0        0        0   1.0000

T =

  -0.5614   0.4345   0.7043  -0.1305
  -0.1127  -0.8833   0.4551   1.9571
   0.8198   0.1761   0.5449   1.5826
        0        0        0   1.0000

T =

  -0.5614   0.4345   0.7043  -0.1305
  -0.1127  -0.8833   0.4551   1.9571
   0.8198   0.1761   0.5449   1.5826
        0        0        0   1.0000

ans =

  -0.5614   0.4345   0.7043  -0.1305
  -0.1127  -0.8833   0.4551   1.9571
   0.8198   0.1761   0.5449   1.5826
        0        0        0   1.0000

>>
```

[2] [2.1]

$M = T_{0b}(0)$, when the robot is in home position.

Using the diagram;

$$M = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 3L \\ 0 & 0 & -1 & -2L \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

z_b aligned with $-z_0$

x_b aligned with y_0

S_i is the screw for the i-th joint when the robot is in home position.

$$S = \begin{bmatrix} \omega_s \\ -\omega_s \times q \end{bmatrix}$$

. ω_s is unit vector in direction of positive axis rotation

screw for translation / prismatic joint:

$$S = \begin{bmatrix} 0 \\ v_s \end{bmatrix}$$

unit vector same as above but for translation

- **Joint 1 :-**

$$\omega_{s_1} = \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} \qquad q_1 = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix}$$

$$\rightarrow S_1 = \begin{bmatrix} 0 \\ 0 \\ 1 \\ 0 \\ 0 \\ 0 \end{bmatrix}$$

- **Joint 2 :**

$$\omega_{s_2} = \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix} \qquad q_2 = \begin{bmatrix} 0 \\ 0 \\ -2L \end{bmatrix}$$

$$\rightarrow S_2 = \begin{bmatrix} 1 \\ 0 \\ 0 \\ 0 \\ -2L \\ 0 \end{bmatrix} \qquad (-\omega_{s_2} \times q) = \begin{bmatrix} 0 \\ -2L \\ 0 \end{bmatrix}$$

- **Joint 3 : (Prismatic)** $\rightarrow S = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 1 \\ 0 \end{bmatrix} = [0\,0\,0\,0\,1\,0]^T$

$$v_{s_3} = \begin{bmatrix} 0 \\ 1 \\ 0 \end{bmatrix}$$

q is not required here as it's a translation

- for joint 4 :

Again a prismatic joint :-

$$S_4 = \begin{bmatrix} 0 \\ v_s \end{bmatrix} ; \quad v_{s_4} = \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix}$$

translation along $+z_0$ axis

$$\mapsto S_4 = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 1 \end{bmatrix}$$

- for joint 5 :-

Revolute joint :

$$S_5 = \begin{bmatrix} \omega_s \\ -\omega_s \times q \end{bmatrix} ; \quad \omega_{s_5} = \begin{bmatrix} 0 \\ 1 \\ 0 \end{bmatrix}$$

$$q_{s_5} = \begin{bmatrix} 0 \\ 2L \\ -L \end{bmatrix} \qquad (-\omega_s \times q) \xrightarrow{=} \begin{bmatrix} L \\ 0 \\ 0 \end{bmatrix}$$

apply

$$\mapsto S_5 = \begin{bmatrix} 0 \\ 1 \\ 0 \\ +L \\ 0 \\ 0 \end{bmatrix} = \begin{bmatrix} 0 \\ 1 \\ 0 \\ L \\ 0 \\ 0 \end{bmatrix}$$

• for joint 6 :-

$$\omega_{s_6} = \begin{bmatrix} 0 \\ 0 \\ -1 \end{bmatrix} \qquad q_6 = \begin{bmatrix} 0 \\ 3L \\ -L \end{bmatrix}$$

$$\rightarrow \quad S_6 = \begin{bmatrix} 0 \\ 0 \\ -1 \\ -3L \\ 0 \\ 0 \end{bmatrix} \qquad \left(\because \omega_s \times q = -\hat{i}(-1 \times 3L) \right)$$

$$\Rightarrow -\omega_s \times q = \begin{bmatrix} -3L \\ 0 \\ 0 \end{bmatrix}$$

Using the formula to get $T(\theta)$:

$$T(\theta) = e^{[S_1]\theta_1} e^{[S_2]\theta_2} e^{[S_3]\theta_3} e^{[S_4]\theta_4} e^{[S_5]\theta_5} e^{[S_6]\theta_6}.$$

$$M$$

Bookmark | Section | Run to End

FILE | NAVIGATE | CODE | ANALYZE | SECTION | RUN

/ › MATLAB Drive

untitled2.m | fkinematicsE1.m | untitled5 *

/MATLAB Drive/fkinematicsE1.m

```
clc; close all;
syms theta1 theta2 theta3 theta4 theta5 theta6 L real

M = [0 1 0 0; 1 0 0 3*L; 0 0 -1 -2*L; 0 0 0 1];
S1 = [0; 0; 1; 0; 0; 0];
S2 = [1; 0; 0; 0; -2*L; 0];
S3 = [0; 0; 0; 0; 1; 0];
S4 = [0; 0; 0; 0; 0; 1];
S5 = [0; 1; 0; L; 0; 0];
S6 = [0; 0; -1; -3*L; 0; 0];
T = expm(bracket(S1) * theta1) * expm(bracket(S2) * theta2) * expm(bracket(S3) * theta3) * expm(bracket(S4) * theta4) * ...
expm(bracket(S5) * theta5) * expm(bracket(S6) * theta6) * M;


function S_matrix = bracket(S)
    S_matrix = [ 0 -S(3) S(2) S(4);
            S(3) 0 -S(1) S(5);
            -S(2) S(1) 0 S(6);
            0 0 0 0];
end
```

Command Window

```
>> disp(T)
[sin(theta6)*(cos(theta1)*cos(theta5) - sin(theta1)*sin(theta2)*sin(theta5)) - cos(theta2)*cos(theta6)*sin(theta1), cos(theta6)*(cos(theta1)*cos(theta5) - sin(theta1)*sin(theta2)*sin(theta5)) + cos(the
[sin(theta6)*(cos(theta5)*sin(theta1) + cos(theta1)*sin(theta2)*sin(theta5)) + cos(theta1)*cos(theta2)*cos(theta6), cos(theta6)*(cos(theta5)*sin(theta1) + cos(theta1)*sin(theta2)*sin(theta5)) - cos(the
[                                   cos(theta6)*sin(theta2) - cos(theta2)*sin(theta5)*sin(theta6),                                    - sin(theta2)*sin(theta6) - cos(the
[                                                                                               0,

>>
```

MATLAB Drive

untitled2.m × fkinematicsE1.m × untitled5 * × +

/MATLAB Drive/fkinematicsE1.m

```
clc; close all;

theta1 = pi/2;  % Example values, replace with your actual values
theta2 = 0;
theta3 = 2;
theta4 = -0.5;
theta5 = 2*pi/3;
theta6 = -pi/4;
L = 1.0;

M = [0 1 0 0; 1 0 0 3*L; 0 0 -1 -2*L; 0 0 0 1];
S1 = [0; 0; 1; 0; 0; 0];
S2 = [1; 0; 0; 0; -2*L; 0];
S3 = [0; 0; 0; 0; 1; 0];
S4 = [0; 0; 0; 0; 0; 1];
S5 = [0; 1; 0; L; 0; 0];
S6 = [0; 0; -1; -3*L; 0; 0];
T = expm(bracket(S1) * theta1) * expm(bracket(S2) * theta2) * expm(bracket(S3) * theta3) * expm(bracket(S4) * theta4) * ...
expm(bracket(S5) * theta5) * expm(bracket(S6) * theta6) * M;


function S_matrix = bracket(S)
    S_matrix = [ 0 -S(3) S(2) S(4);
            S(3) 0 -S(1) S(5);
            -S(2) S(1) 0 S(6);
            0 0 0 0];
end
```

Command Window

```
>> disp(T)
   -0.7071   -0.7071   -0.0000   -5.0000
    0.3536   -0.3536   -0.8660   -0.8660
    0.6124   -0.6124    0.5000   -1.0000
         0         0         0    1.0000

>> 
```

NAVIGATE CODE ANALYZE SECTION RUN

MATLAB Drive

ed2.m × fkinematicsE1.m × +

AB Drive/fkinematicsE1.m

```
clc; close all;

theta1 = -pi/2;  % Example values, replace with your actual values
theta2 = -pi/4;
theta3 = 0;
theta4 = 1;
theta5 = pi/3;
theta6 = pi/2;
L = 2.0;

M = [0 1 0 0; 1 0 0 3*L; 0 0 -1 -2*L; 0 0 0 1];
S1 = [0; 0; 1; 0; 0; 0];
S2 = [1; 0; 0; 0; -2*L; 0];
S3 = [0; 0; 0; 0; 1; 0];
S4 = [0; 0; 0; 0; 0; 1];
S5 = [0; 1; 0; L; 0; 0];
S6 = [0; 0; -1; -3*L; 0; 0];
T = expm(bracket(S1) * theta1) * expm(bracket(S2) * theta2) * expm(bracket(S3) * theta3) * expm(bracket(S4) * theta4) * ...
expm(bracket(S5) * theta5) * expm(bracket(S6) * theta6) * M;


function S_matrix = bracket(S)
    S_matrix = [ 0 -S(3) S(2) S(4);
            S(3) 0 -S(1) S(5);
            -S(2) S(1) 0 S(6);
            0 0 0 0];
end
```

Command Window

```
>> disp(T)
   -0.6124   -0.7071   -0.3536    5.6569
   -0.5000   -0.0000    0.8660    1.7321
   -0.6124    0.7071   -0.3536   -6.8284
         0         0         0    1.0000

>>
```

$\boxed{2.7}$ <u>Case I</u> :-

$$L = 1, \quad \theta = \left[\pi/2, 0, 2, -0.5, 2\pi/3, -\pi/4\right]^T$$

$$T_{06} = \begin{bmatrix} -0.707 & -0.707 & 0 & -5 \\ 0.3536 & -0.3536 & -0.866 & -0.866 \\ 0.6124 & -0.6124 & 0.5 & -1 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

<u>Case II</u> : $L = 2, \quad \theta = \left[-\pi/2, -\pi/4, 0, 1, \pi/3, \pi/2\right]^T$

$$T_{06} = \begin{bmatrix} -0.6124 & -0.7071 & -0.3536 & 5.6569 \\ -0.5 & 0 & 0.866 & 1.7321 \\ -0.6124 & 0.7071 & -0.3536 & -6.8284 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

3 $M = T_{sb}(0)$, in this case $T_{0b}(0)$ when robot is in home position.

$$M = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & L_1+L_2+L_3+L_4 \\ 0 & 0 & 1 & h \\ & 0 & & 1 \end{bmatrix}$$

z_b aligns with $+z_0$
y_b aligns with $+y_0$

S_i is the screw for the i-th joint when the robot is in home position.

$$S = \begin{bmatrix} \omega_s \\ -\omega_s \times q \end{bmatrix} \quad \& \quad S = \begin{bmatrix} 0 \\ V_s \end{bmatrix}$$

revolute joint having ω_s ang velocity as unit vector in direction of positive axis rotation

linear velocity unit vector in direction of translation.

- for joint 1 (prismatic):

At the fixed frame, it's placed:

$$\omega_{s_1} = \begin{bmatrix} 0 \\ 1 \\ 0 \end{bmatrix}$$

$$\therefore \hookrightarrow S_1 = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 1 \\ 0 \end{bmatrix}$$

- for joint 2 (revolute):

$$\omega_{s_2} = \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} \qquad q = \begin{bmatrix} 0 \\ L_1 \\ 0 \end{bmatrix}$$

$$\therefore -\omega_{s_2} \times q = \begin{bmatrix} L_1 \\ 0 \\ 0 \end{bmatrix}$$

$$\hookrightarrow \therefore S_2 = \begin{bmatrix} 0 \\ 0 \\ 1 \\ L_1 \\ 0 \\ 0 \end{bmatrix}$$

- **for joint 3 :**

joint rotation about axis $-x_0$;

$$\therefore \quad \omega_{s_3} = \begin{bmatrix} -1 \\ 0 \\ 0 \end{bmatrix} \quad ; \quad q = \begin{bmatrix} 0 \\ L_1 \\ h \end{bmatrix}$$

$$\therefore \quad S_3 = \begin{bmatrix} -1 \\ 0 \\ 0 \\ 0 \\ -h \\ L_1 \end{bmatrix} \qquad (-\omega_{s_3} \times q) = \begin{bmatrix} 0 \\ -h \\ L_1 \end{bmatrix}$$

- **for joint 4 :**

joint rotation about axis $-x_0$;

$$\omega_{s_4} = \begin{bmatrix} -1 \\ 0 \\ 0 \end{bmatrix} \quad ; \quad q = \begin{bmatrix} 0 \\ L_1+L_2 \\ h \end{bmatrix}$$

$$\rightarrow \therefore \quad S_4 = \begin{bmatrix} -1 \\ 0 \\ 0 \\ 0 \\ -h \\ L_1+L_2 \end{bmatrix} \qquad (-\omega_{s_4} \times q) = \begin{bmatrix} 0 \\ -h \\ L_1+L_2 \end{bmatrix}$$

- for joint 5 :

joint rotation about axis $-x_0$;

$$\omega_{s5} = \begin{bmatrix} -1 \\ 0 \\ 0 \end{bmatrix} ; \quad q = \begin{bmatrix} 0 \\ L_1 + L_2 + L_3 \\ h \end{bmatrix}$$

$$\rightarrow \quad S_5 = \begin{bmatrix} -1 \\ 0 \\ 0 \\ 0 \\ -h \\ L_1 + L_2 + L_3 \end{bmatrix}$$

- for joint 6 :

joint rotation about $+y_0$ axis ;

$$\omega_{s6} = \begin{bmatrix} 0 \\ 1 \\ 0 \end{bmatrix} ; \quad q = \begin{bmatrix} 0 \\ L_1 + L_2 + L_3 + L_4 \\ h \end{bmatrix}$$

$$\text{or} \quad q = \begin{bmatrix} 0 \\ L_1 \\ h \end{bmatrix}$$

$$\rightarrow \therefore S_6 = \begin{bmatrix} 0 \\ 1 \\ 0 \\ -h \\ 0 \\ 0 \end{bmatrix}$$

$$-\omega_{s6} \times q = \begin{bmatrix} -h \\ 0 \\ 0 \end{bmatrix}$$

/ › MATLAB Drive

untitled2.m × fkinematicsE1.m × +

/MATLAB Drive/fkinematicsE1.m

```
clc; close all;
syms h L1 L2 L3 L4 theta1 theta2 theta3 theta4 theta5 theta6;

M = [1 0 0 0; 0 1 0 L1+L2+L3+L4; 0 0 -1 h; 0 0 0 1];
S1 = [0; 0; 0; 0; 1; 0];
S2 = [0; 0; 1; L1; 0; 0];
S3 = [-1; 0; 0; 0; -h; L1];
S4 = [-1; 0; 0; 0; -h; L1+L2];
S5 = [-1; 0; 0; 0; -h; L1+L2+L3];
S6 = [0; 1; 0; -h; 0; 0];
T = expm(bracket(S1) * theta1) * expm(bracket(S2) * theta2) * expm(bracket(S3) * theta3) * expm(bracket(S4) * theta4) * ...
expm(bracket(S5) * theta5) * expm(bracket(S6) * theta6) * M;


function S_matrix = bracket(S)
    S_matrix = [ 0 -S(3) S(2) S(4);
            S(3) 0 -S(1) S(5);
            -S(2) S(1) 0 S(6);
            0 0 0 0];
end
```

Command Window

```
>> disp(T)
[(((exp(-theta3*1i)/2 + exp(theta3*1i)/2)*((exp(-theta2*1i)*1i)/2 - (exp(theta2*1i)*1i)/2)*((exp(-theta4*1i)*1i)/2 - (exp(theta4*1i)*1i)/2) + (exp(-theta4*1i)/2 + exp(theta4*1i)/2)*((exp(-theta2*1i
[                        - (((exp(-theta2*1i)/2 + exp(theta2*1i)/2)*(exp(-theta3*1i)/2 + exp(theta3*1i)/2)*((exp(-theta4*1i)*1i)/2 - (exp(theta4*1i)*1i)/2) + (exp(-theta2*1i)/2 + exp(theta2*1i)
[
[

>>
```

MATLAB Drive

2.m × fkinematicsE1.m ×

Drive/fkinematicsE1.m

```matlab
clc; close all;

h = 2;
L1 = 1; L2 = 1; L3 = 1; L4 = 1;
theta1 = 1;
theta2 = pi/4;
theta3 = -pi/4;
theta4 = 0;
theta5 = pi/4;
theta6 = pi/2;

M = [1 0 0 0; 0 1 0 L1+L2+L3+L4; 0 0 -1 h; 0 0 0 1];
S1 = [0; 0; 0; 0; 1; 0];
S2 = [0; 0; 1; L1; 0; 0];
S3 = [-1; 0; 0; 0; -h; L1];
S4 = [-1; 0; 0; 0; -h; L1+L2];
S5 = [-1; 0; 0; 0; -h; L1+L2+L3];
S6 = [0; 1; 0; -h; 0; 0];
T = expm(bracket(S1) * theta1) * expm(bracket(S2) * theta2) * expm(bracket(S3) * theta3) * expm(bracket(S4) * theta4) * ...
expm(bracket(S5) * theta5) * expm(bracket(S6) * theta6) * M;


function S_matrix = bracket(S)
    S_matrix = [ 0 -S(3) S(2) S(4);
                S(3) 0 -S(1) S(5);
                -S(2) S(1) 0 S(6);
                0 0 0 0];
end
```

Command Window

```
>> disp(T)
   -0.0000   -0.7071   -0.7071   -1.7071
   -0.0000    0.7071   -0.7071    3.7071
   -1.0000         0    0.0000    3.4142
         0         0         0    1.0000

>> 
```

Using the formula to get $T(\theta)$:

$$\Rightarrow T(\theta) = e^{[S_1]\theta_1} e^{[S_2]\theta_2} e^{[S_3]\theta_3} e^{[S_4]\theta_4} e^{[S_5]\theta_5} e^{[S_6]\theta_6} \cdot M$$

$\boxed{3.2}$ $h=2$, $L_1 = L_2 = L_3 = L_4 = 1$, $\theta = [1, \pi/4, -\pi/4, 0, \pi/4, \pi/2]$

$$T_{06} = \begin{bmatrix} 0 & -0.707 & -0.707 & -1.707 \\ 0 & +0.707 & -0.707 & 3.707 \\ -1 & 0 & 0 & 3.4142 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$\boxed{4}$ $M = T_{0b}(0)$

$$M = \begin{bmatrix} 1 & 0 & 0 & (2+\sqrt{3})L \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & (\sqrt{3}-1)L \\ & 0 & & 1 \end{bmatrix}$$

for joint 1 :
(θ_1)

$$\omega_{s_1} = \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} \qquad q = \begin{bmatrix} L \\ 0 \\ 0 \end{bmatrix}$$

apply $(-\omega_{s_1} \times q)$ $\rightarrow \begin{bmatrix} 0 \\ -L \\ 0 \end{bmatrix}$

$$\therefore \quad S_1 = \begin{bmatrix} 0 \\ 0 \\ 1 \\ 0 \\ -L \\ 0 \end{bmatrix}$$

for joint 2 :

$$\omega_{s_2} = \begin{bmatrix} 0 \\ 1 \\ 0 \end{bmatrix} \qquad q = \begin{bmatrix} L \\ 0 \\ 0 \end{bmatrix}$$

$$S_2 = \begin{bmatrix} 0 \\ 1 \\ 0 \\ 0 \\ 0 \\ L \end{bmatrix}$$

for joint 3 :-

$$\omega_{s_3} = \begin{bmatrix} 0 \\ 1 \\ 0 \end{bmatrix} \qquad q = \begin{bmatrix} (\sqrt{3}+1)L \\ 0 \\ -L \end{bmatrix}$$

$$\hookrightarrow S_3 = \begin{bmatrix} 0 \\ 1 \\ 0 \\ L \\ 0 \\ (\sqrt{3}+1)L \end{bmatrix}$$

for joint 4 :-

$$\omega_{s_4} = \begin{bmatrix} 0 \\ 1 \\ 0 \end{bmatrix} ; \quad q = \begin{bmatrix} (2+\sqrt{3})L \\ 0 \\ (\sqrt{3}-1)L \end{bmatrix}$$

$$S_4 = \begin{bmatrix} 0 \\ 1 \\ 0 \\ (1-\sqrt{3})L \\ 0 \\ (2+\sqrt{3})L \end{bmatrix} = \begin{bmatrix} 0 & 1 & 0 & (1-\sqrt{3})L & 0 & (2+\sqrt{3})L \end{bmatrix}^T$$

4.1 Even if we change θ_5 & θ_6, T_{0b} will not change as;

$$T_{0b} = e^{[S_1]\theta_1} e^{[S_2]\theta_2} e^{[S_3]\theta_3} e^{[S_4]\theta_4} \cdot M$$

It's independent of the variables θ_5 & θ_6 and making any changes at this joint will not affect T_{0b} as our arbitrary point of interest is at the revolute joint at the angle of θ_4, and our body frame lies there, thus we find T_{0b} with fixed frame {0} to body frame {b}, and any alterations to further joints will not affect the kinematics of this formation.

4.4 Case I :-

$$L=1,\ \theta=[\pi/4, \pi/4, 0, \pi/2, 2, -\pi/4]^T$$

$$T_{0b} = \begin{bmatrix} -0.5 & -0.707 & 0.5 & 2.7321 \\ -0.5 & 0.707 & 0.5 & 1.7321 \\ -0.707 & 0 & -0.707 & -1.4142 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Case II :-

$$L=1,\ \theta=[-\pi/2, 0, \pi/6, -\pi/3]$$

$$T_{0b} = \begin{bmatrix} 0 & 1 & 0 & 1 \\ -0.866 & 0 & 0.5 & -3.4641 \\ 0.5 & 0 & 0.866 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Below provided is the MATLAB implementation.

FILE NAVIGATE CODE ANALYZE SECTION RUN

/ > MATLAB Drive

untitled2.m × fkinematicsE1.m ×

/MATLAB Drive/fkinematicsE1.m

```matlab
clc; close all;
syms L theta1 theta2 theta3 theta4;

M = [1 0 0 (2+sqrt(3))*L; 0 1 0 0; 0 0 1 (sqrt(3)-1)*L; 0 0 0 1];
S1 = [0; 0; 1; 0; -L; 0];
S2 = [0; 1; 0; 0; 0; L];
S3 = [0; 1; 0; L; 0; (1+sqrt(3))*L];
S4 = [0; 1; 0; (1-sqrt(3))*L; 0; (2+sqrt(3))*L];

T = expm(bracket(S1) * theta1) * expm(bracket(S2) * theta2) * expm(bracket(S3) * theta3) * expm(bracket(S4) * theta4) * M;


function S_matrix = bracket(S)
    S_matrix = [ 0 -S(3) S(2) S(4);
            S(3) 0 -S(1) S(5);
            -S(2) S(1) 0 S(6);
            0 0 0 0];
end
```

Command Window

```
>> disp(T)
[                  ((exp(-theta1*1i)/2 + exp(theta1*1i)/2)*(exp(-theta2*1i)/2 + exp(theta2*1i)/2)*(exp(-theta3*1i)/2 + exp(theta3*1i)/2) - (exp(-theta1*1i)/2 + exp(theta1*1i)/2)*((
[((exp(-theta2*1i)/2 + exp(theta2*1i)/2)*(exp(-theta3*1i)/2 + exp(theta3*1i)/2)*((exp(-theta1*1i)*1i)/2 - (exp(theta1*1i)*1i)/2) - ((exp(-theta1*1i)*1i)/2 - (exp(theta1*1i)*1i)/2)*((exp(-theta2*1i)*1i)/
[                                                                                                                                                                                  - ((exp
[

>>
```

MATLAB Drive

untitled2.m × fkinematicsE1.m * × +

/MATLAB Drive/fkinematicsE1.m

```matlab
clc; close all;

%Question 4.2 Case I:
L = 1;
theta1 = pi/4;
theta2 = pi/4;
theta3 = 0;
theta4 = pi/2;

M = [1 0 0 (2+sqrt(3))*L; 0 1 0 0; 0 0 1 (sqrt(3)-1)*L; 0 0 0 1];
S1 = [0; 0; 1; 0; -L; 0];
S2 = [0; 1; 0; 0; 0; L];
S3 = [0; 1; 0; L; 0; (1+sqrt(3))*L];
S4 = [0; 1; 0; (1-sqrt(3))*L; 0; (2+sqrt(3))*L];

T = expm(bracket(S1) * theta1) * expm(bracket(S2) * theta2) * expm(bracket(S3) * theta3) * expm(bracket(S4) * theta4) * M;


function S_matrix = bracket(S)
    S_matrix = [ 0 -S(3) S(2) S(4);
            S(3) 0 -S(1) S(5);
            -S(2) S(1) 0 S(6);
            0 0 0 0];
end
```

Command Window

```
>> disp(T)
   -0.5000   -0.7071    0.5000    2.7321
   -0.5000    0.7071    0.5000    1.7321
   -0.7071         0   -0.7071   -1.4142
         0         0         0    1.0000

>>
```

FILE NAVIGATE CODE ANALYZE SECTION RUN

/ › MATLAB Drive

untitled2.m × fkinematicsE1.m × +

/MATLAB Drive/fkinematicsE1.m

```
clc; close all;

%Question 4.2 Case II:
L = 1;
theta1 = -pi/2;
theta2 = 0;
theta3 = pi/6;
theta4 = -pi/3;

M = [1 0 0 (2+sqrt(3))*L; 0 1 0 0; 0 0 1 (sqrt(3)-1)*L; 0 0 0 1];
S1 = [0; 0; 1; 0; -L; 0];
S2 = [0; 1; 0; 0; 0; L];
S3 = [0; 1; 0; L; 0; (1+sqrt(3))*L];
S4 = [0; 1; 0; (1-sqrt(3))*L; 0; (2+sqrt(3))*L];

T = expm(bracket(S1) * theta1) * expm(bracket(S2) * theta2) * expm(bracket(S3) * theta3) * expm(bracket(S4) * theta4) * M;


function S_matrix = bracket(S)
    S_matrix = [ 0 -S(3) S(2) S(4);
            S(3) 0 -S(1) S(5);
            -S(2) S(1) 0 S(6);
            0 0 0 0];
end
```

Command Window

```
>> disp(T)
    0.0000    1.0000   -0.0000    1.0000
   -0.8660    0.0000    0.5000   -3.4641
    0.5000         0    0.8660    0.0000
         0         0         0    1.0000

>>
```