ASSIGNMENT REPORT

15MH301 – FUNDAMENTALS OF ROBOTICS

ASSIGNMENT 1 (AY 2019-2020, ODD)

SUBMITTED BY

NAME: CHINMAY VILAS SAMAK

REGISTER NUMBER: RA1711018010102

DATE OF SUBMISSION: 12th SEPTEMBER 2019

DUE DATE : 12TH SEPTEMBER, 2019

DEPARTMENT OF MECHATRONICS ENGINEERING

SRM INSTITUTE OF SCIENCE AND TECHNOLOGY

KATTANKULATHUR

FACULTY SIGNATURE

(Ranjith Pillai R)

EVALUATION /10

REMARKS:

INSTRUCTIONS

- Answer all questions
- The assignment report should be submitted in the format given.
- The assignment report should be submitted online on the day of submission given above and the submission will be blocked thereafter.
- All the reports will go through plagiarism check. High similarity index will be rejected.
- If MATLAB script to be added, publish the script and attach.
- Any hand written/ derivation should be added in appendix.
- After online submission, the hard copy needs to be submitted
- Date of Submission/presentation will not be changed
- It's the effort and learning that matters!!! Do your best!!!

NOTE: Format of naming the report file for submission

Format: Assign1_15MH301_B(NO.) _RA/IA/16/17_ (REG.NO.LAST 3 DIGIT)

E.g. For a student in batch 1 with register number RA17018010056

File should be named as **Assign1_15MH301_B1_RA17_056**

QUESTION 1:

Consider the two cases and write the homogenous transformation matrix for both the cases of frame pose

a) A frame {B} is translated by [3 2 1] units and then rotated by 60 degree along X axis with respect to frame {A}. (Use MATLAB)
b) A frame {B} is rotated by 60 degree along X axis with respect to frame {A} and then translated by [3 2 1] units. (use MATLAB)
c) Is both the homogenous transformation matrix same? Justify your answer.
d) Show the final pose of both the frames in the single plot using MATLAB

1. UNDERSTANDING OF PROBLEM STATEMENT:

This problem requires the knowledge of homogeneous transformation and its application. It is required to find the transformation of frame $\{B\}$ with respect to frame $\{A\}$.

2. SOLUTION APPROACH

1. Define pose of initial frame $\{A\}$ (at origin with no rotation).
2. Find the homogeneous transformation of frame $\{B\}$ with respect to frame $\{A\}$ for both the cases.
3. Compare both the homogeneous transformation matrices.
4. Visualize the frames for both cases.

3. INFERENCE/CONCLUSION

Since sequential motion mathematically means sequential multiplication of transformation matrices (current axis convention) and it is known that matrix multiplication is not commutative in nature (meaning $A * B \neq B * A$), it can be concluded that the sequence of motion matters while computing the final pose of a body.

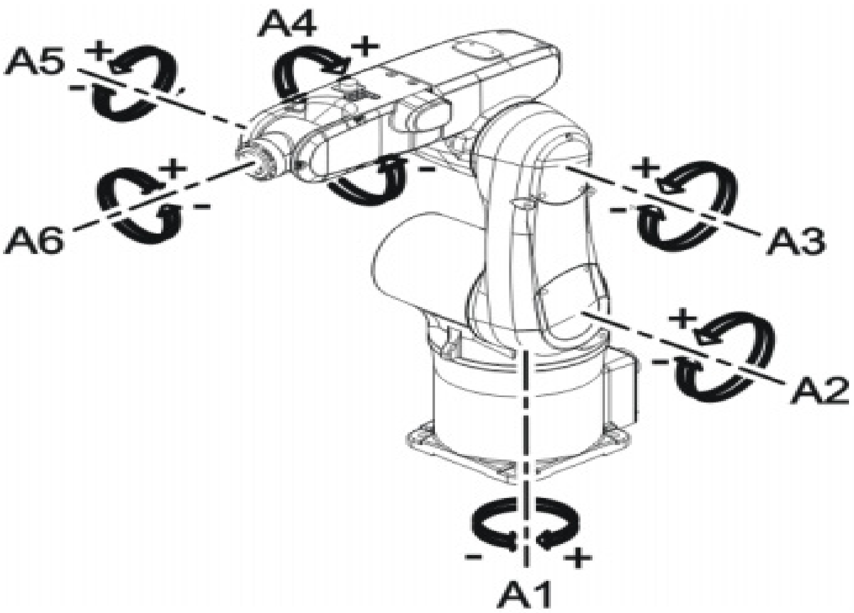
The two cases differ in the sequence of motion. Assuming the pose of frame $\{A\}$ to be {0,0,0,0,0,0} i.e. at origin with no rotation, in first case, frame $\{B\}$ is first translated and then rotated which means that its final position is $(3,2,1)$ and its orientation is changed by $60°$ with respect to the same x-axis. However, in the second case, frame $\{B\}$ is first rotated and then translated which means that its final orientation is changed by $60°$ with respect to the initial x-axis and the translation translates it along the current (rotated) x-axis which takes it to a different position altogether.

QUESTION 2:

Derive the forward kinematic model of the KUKA KR5 R650 manipulator given below. Use datasheet given below to interpret the various parameter values (d and a values in DH) when using DH.

(Use MATLAB to multiply all matrices to arrive at the final matrix defining the pose of end effector with respect to base).

(** The frames should be assigned neatly in the manipulator diagram itself. Also find the data sheet in the last appendix section).



1. UNDERSTANDING OF PROBLEM STATEMENT:

This problem requires finding the forward kinematic model for KUKA KR5 Sixx R650, a manipulator with 6 revolute joints (6 DOF) using standard DH convention.

2. SOLUTION APPROACH

1. Assign frames for each joint and end-effector.
2. Formulate the DH Parameter Table.
3. Compute the forward kinematics.
4. Visualize the result.

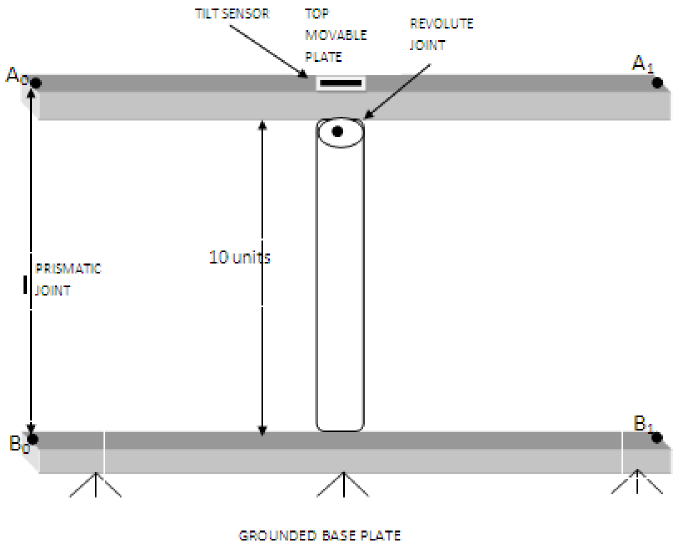
3. INFERENCE/CONCLUSION

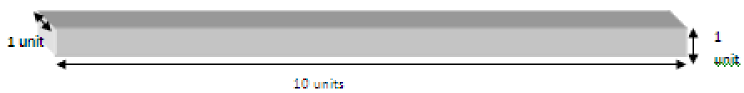
The forward kinematic model for KUKA KR5 Sixx R650 manipulator was computed using $fkine()$ function in *MATLAB.* The result was verified for the home position (as given in question figure).

This problem strengthens the concepts of DH convention and forward kinematics and their application to compute forward kinematics for actual industrial manipulators. It also provides hands-on simulation experience using *Robotics Toolbox* in *MATLAB.*

QUESTION 3:

Consider a 1 DOF beam as shown in figure hinged by a revolute joint at the center and other side of the joint is welded to the fixed base plate. The dimension of the beam and the base plate is given below. One pneumatic cylinder is connected along A_0, B_0. The beam has a tilt sensor to measure the inclination of beam along one axis. It is well known that the angle of inclination can only be achieved by appropriate length of actuation. Hence, the control loop requires a joint level actuator length feedback to ensure the minimum error in actuator length. Your task is to find the kinematic model which can find the length of actuator (piston length) for a desired angle of inclination using the tilt sensor. Follow the hints below to achieve the same.





If the top tilting beam is assigned frame {A} and bottom base plate is assigned frame {B}, answer the following

a) Assign the frames {A} and {B} to the top plate and base plate center using right hand convention.
b) Which is the axis of tilt as per the frame assigned by you? (x, y, z)
c) Find the vector points A_0, B_0, A_1 and B_1 as per the frame assigned. (x, y, z)
d) Find the length of the prismatic joint at home position (zero-degree tilt of beam)
e) If the top plate is tilted by an angle of 30 degree counter clockwise along the axis of tilt you have fixed, find the new location for the points A_0.
f) Find the length of actuator A_0 B_0 required to make the tilt of 30 degree of top platform. (Hint: Find Euclidian distance between points A_0 $B_{0)}$
(****Use MATLAB to solve)

1. UNDERSTANDING OF PROBLEM STATEMENT:

This problem requires finding the kinematic model for the given 1 DOF system (beam tilt mechanism) and hence establish a relation between the tilt angle θ and actuator length l.

It is to be assumed that the linear actuator is attached to the assembly using 2 passive joints at both ends. Hence this problem is to be solved in a generalized way, assigning frames and finding the relative transformations between them (and not using geometric approach).

2. SOLUTION APPROACH

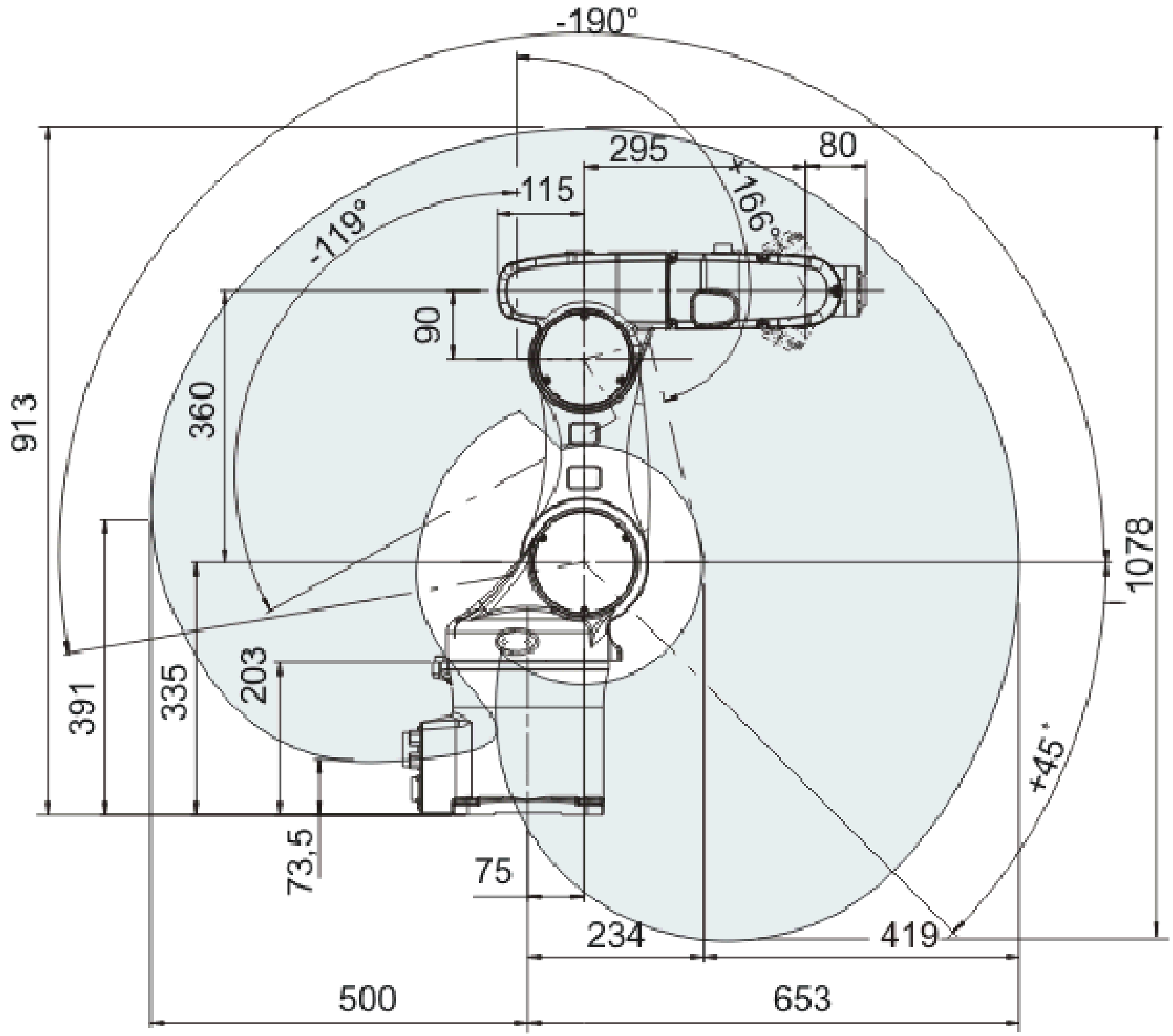
1. Assign frames at points A_0 and B_0 as well as for the pivot point P.
2. Define mechanical constraints for the system.
3. Find the homogeneous transformation of frame $\{P\}$ with respect to frame $\{B_0\}$.
4. Find the homogeneous transformation of frame $\{A_0\}$ with respect to frame $\{P\}$.
5. Compute the homogeneous transformation of frame $\{A_0\}$ with respect to frame $\{B_0\}$ by multiplying the above two matrices (step 3 and 4) in same sequence.
6. Find the co-ordinates for points A_0, B_0 A_1 and B_1.
7. Compute actuator length (Euclidian distance) between the points A_0 and B_0.
8. Verify the results for $0°$ and $30°$.

3. INFERENCE/CONCLUSION

The generalized kinematic model for the given 1 DOF system was found, thereby establishing a relation between the tilt angle θ and the actuator length l. The result was verified for tilt angles of $0°$, $30°$ and $-30°$.

This problem strengthens the concepts of kinematic modelling and transformations and their application to solve real-world problems (just like this one).

APPENDIX



-190°
295
80
115
+166°
-119°
90
360
913
1078
391
335
203
73,5
75
234
419
+45°
500
653

APPENDIX - 1

Define initial frame $\{A\}$

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

Case 1: Frame $\{B\}$ is translated by $[3 \quad 2 \quad 1]^T$ units and then rotated by 60° along x-axis with respect to frame $\{A\}$.

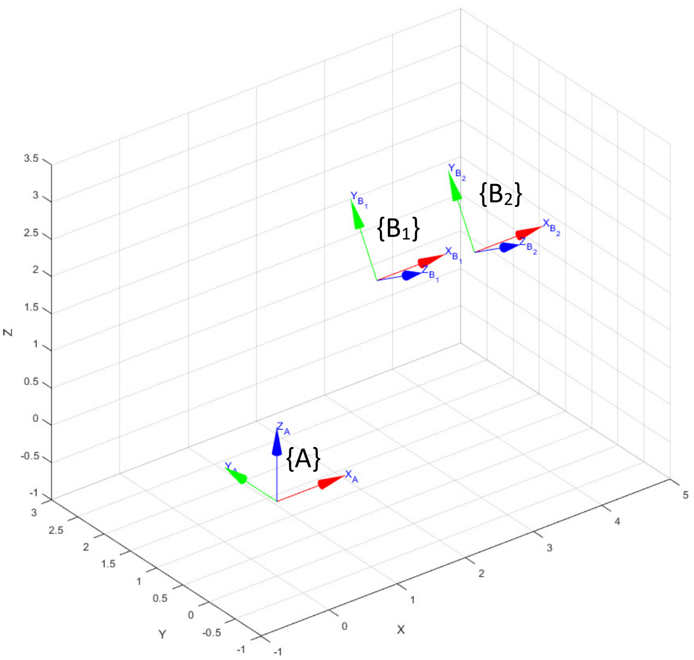
$$B_1 = \begin{bmatrix} 1 & 0 & 0 & 3 \\ 0 & 1 & 0 & 2 \\ 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 1 \end{bmatrix} * \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0.5 & -0.866 & 0 \\ 0 & 0.866 & 0.5 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 3 \\ 0 & 0.5 & -0.866 & 2 \\ 0 & 0.866 & 0.5 & 1 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Case 2: Frame $\{B\}$ is rotated by 60° along x-axis with respect to frame $\{A\}$ and then translated by $[3 \quad 2 \quad 1]^T$ units.

$$B_2 = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0.5 & -0.866 & 0 \\ 0 & 0.866 & 0.5 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} * \begin{bmatrix} 1 & 0 & 0 & 3 \\ 0 & 1 & 0 & 2 \\ 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 3 \\ 0 & 0.5 & -0.866 & 0.134 \\ 0 & 0.866 & 0.5 & 2.2321 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

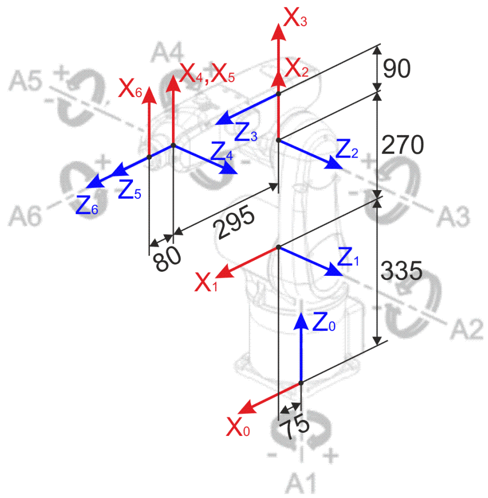
*Since $B_1 \neq B_2$ we can say that both homogeneous transformation matrices are **NOT** same.*

Frame Visualization:



APPENDIX - 2

Assign Frames:



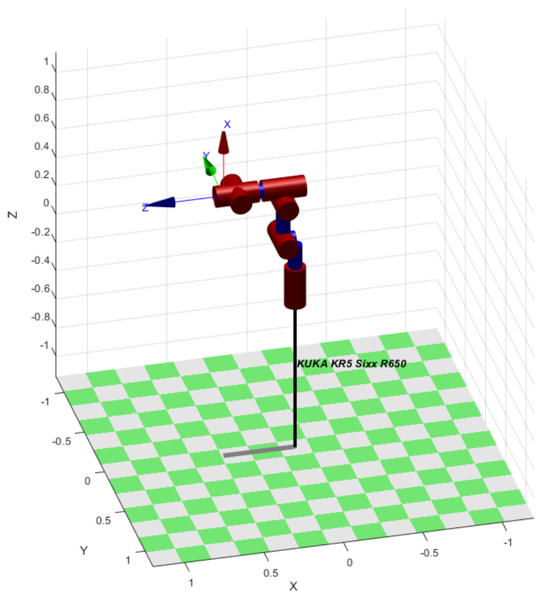
All dimensions are in mm

DH Parameter Table:

$\boldsymbol{\theta}$	$\boldsymbol{d}$	$\boldsymbol{a}$	$\boldsymbol{\alpha}$
$\theta_1(JV)$	$335mm$	$75mm$	$-90°$
$\theta_2(JV) - 90°$	$0mm$	$270mm$	$0°$
$\theta_3(JV)$	$0mm$	$90mm$	$-90°$
$\theta_4(JV)$	$295mm$	$0mm$	$90°$
$\theta_5(JV)$	$0mm$	$0mm$	$-90°$
$\theta_6(JV)$	$80mm$	$0mm$	$0°$

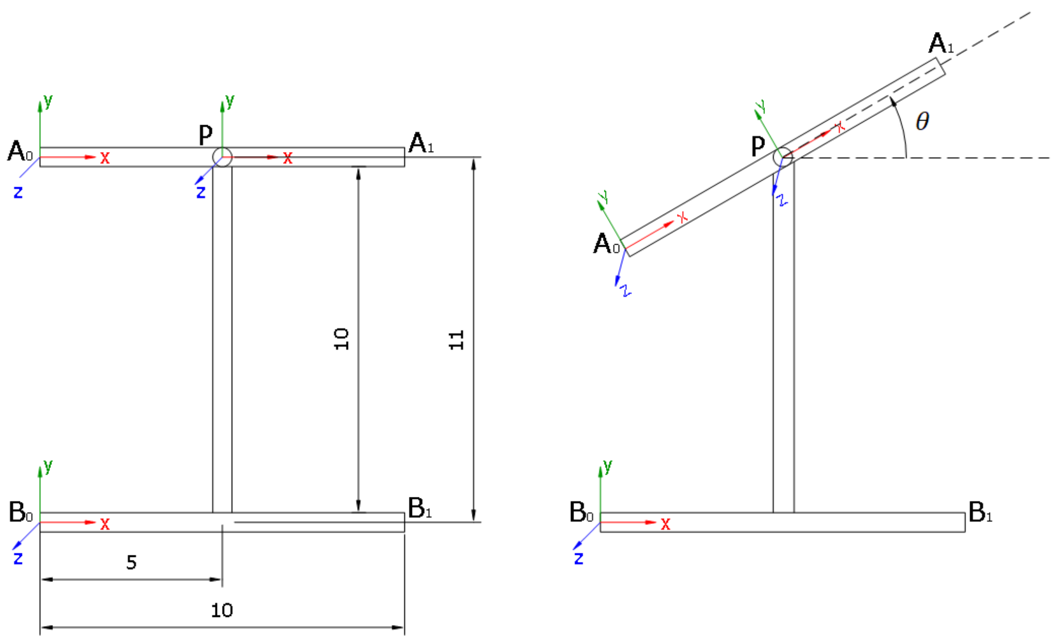
Forward Kinematics for Home Position:

$$^{0}T_6 = \begin{bmatrix} 0 & 0 & 1 & 0.45 \\ 0 & -1 & 0 & 0 \\ 1 & 0 & 0 & 0.695 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$



APPENDIX - 3

Diagram:



Transformations:

$$^{B0}T_P = \begin{bmatrix} cos\theta & -sin\theta & 0 & 5 \\ sin\theta & cos\theta & 0 & 11 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$^{P}T_{A0} = \begin{bmatrix} 1 & 0 & 0 & -5 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$^{B0}T_{A0} = {}^{B0}T_P * {}^{P}T_{A0} = \begin{bmatrix} cos\theta & -sin\theta & 0 & 5 - 5 * cos\theta \\ sin\theta & cos\theta & 0 & 11 - 5 * sin\theta \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Co-ordinates:

$B_0 = (0,0,0)$

$B_1 = (10,0,0)$

$A_0 = (5 - 5 * cos\theta, 11 - 5 * sin\theta, 0)$

$A_1 = (15 - 5 * cos\theta, 11 - 5 * sin\theta, 0)$

Actuator Length (Euclidian distance between A_0 and B_0):

$$l(\theta) = |\overline{B_0 A_0}| = \sqrt{(x_{A_0} - x_{B_0})^2 + (y_{A_0} - y_{B_0})^2 + (z_{A_0} - z_{B_0})^2}$$

$$l(\theta) = \sqrt{[(5 - 5 * cos\theta) - 0]^2 + [(11 - 5 * sin\theta) - 0]^2 + (0 - 0)^2}$$

$$\boxed{l(\theta) = \sqrt{(5 - 5 * cos\theta)^2 + (11 - 5 * sin\theta)^2}}$$

Verification:

Home Position: $l(0°) = \sqrt{(5 - 5)^2 + (11 - 0)^2} = 11\, units$

Actuator Retracted: $l(30°) = \sqrt{(5 - 4.33)^2 + (11 - 2.5)^2} = 8.5263\, units$

Actuator Extended: $l(-30°) = \sqrt{(5 - 4.33)^2 + (11 + 2.5)^2} = 13.5166\, units$

MATLAB Programming & Simulation Attachment

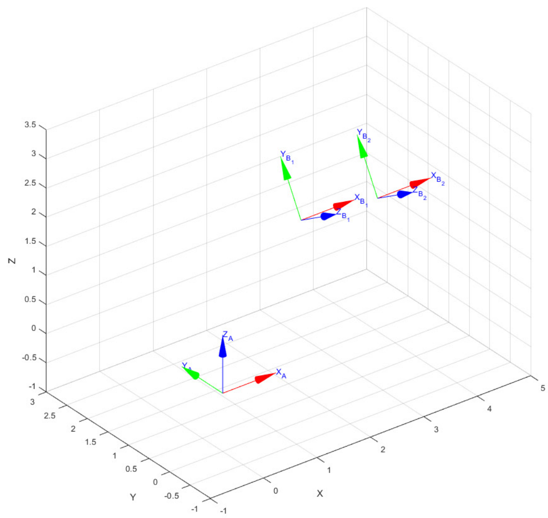
Author: Chinmay Vilas Samak (Student, B.Tech. Mechatronics Engineering, RA1711018010102)
Course: Fundamentals of Robotics (15MH301)

1. Homogenous Transformations

```
fprintf('-----------------------------------------------------------------\n');
fprintf('1. Homogenous Transformations\n');
fprintf('-----------------------------------------------------------------\n');

% Frame Definition & Transformations
A = trotx(0) % Definition of frame {A}

T1 = transl(3,2,1); % Translation along x,y,z axes by 3,2,1 units. Yields 4x4 homogeneous transformation matrix with zero rotation.
T2 = trotx(60); % Rotation along x-axis by 60 degrees. Yields 4x4 homogeneous transformation matrix with zero translation.

B1 = T1*T2 % Frame {B} case 1
B2 = T2*T1 % Frame {B} case 2

% Comparing the Transformations
if B1 == B2
    fprintf("Both Homogeneous Transformation Matrices Are Same.\n");
else
    fprintf("Both Homogeneous Transformation Matrices Are NOT Same.\n");
end

% Frame Visualization
Bounds = [-1 5 -1 3 -1 3.5]; % Axis bounds [xmin xmax ymin ymax zmin zmax]
figure('Name','Frame Visualization','NumberTitle','off','WindowState','Maximized');
trplot(A, 'frame', 'A', 'axis', Bounds, 'rgb', 'arrow'); % Plot frame {A}
hold on;
tranimate(A, B1, 'frame', 'B_1', 'axis', Bounds, 'rgb', 'arrow'); % Animate frame {B} case 1
hold on;
tranimate(A, B2, 'frame', 'B_2', 'axis', Bounds, 'rgb', 'arrow'); % Animate frame {B} case 2
hold off;
```

```
-----------------------------------------------------------------
1. Homogenous Transformations
-----------------------------------------------------------------

A =

     1     0     0     0
     0     1     0     0
     0     0     1     0
     0     0     0     1
```

```
B1 =

    1.0000         0         0    3.0000
         0    0.5000   -0.8660    2.0000
         0    0.8660    0.5000    1.0000
         0         0         0    1.0000


B2 =

    1.0000         0         0    3.0000
         0    0.5000   -0.8660    0.1340
         0    0.8660    0.5000    2.2321
         0         0         0    1.0000

Both Homogeneous Transformation Matrices Are NOT Same.
```

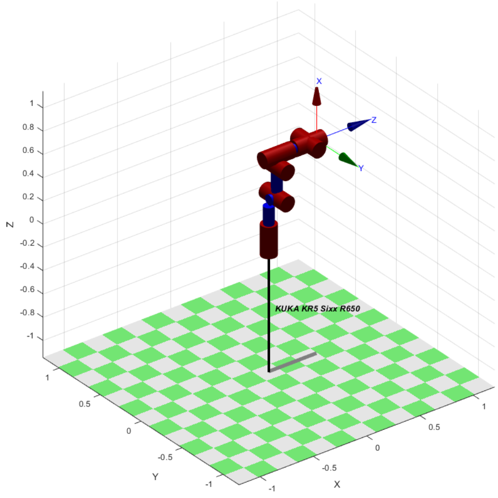


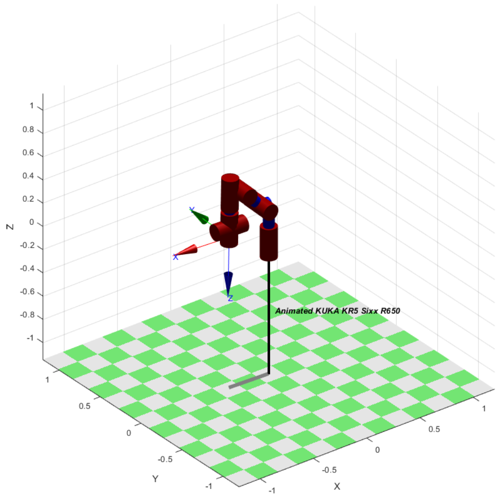
2. Forward Kinematic Model of KUKA KR5 Sixx R650 Manipulator

```
fprintf('-----------------------------------------------------------------\n');
fprintf('2. Forward Kinematic Model of KUKA KR5 Sixx R650 Manipulator\n');
fprintf('-----------------------------------------------------------------\n');

% Define DH Parameters
L(1) = Link([0 0.335 0.075 -pi/2]);
L(2) = Link([0 0 0.270 0]);
L(2).offset = -pi/2;
L(3) = Link([0 0 0.090 -pi/2]);
L(4) = Link([0 0.295 0 pi/2]);
L(5) = Link([0 0 0 -pi/2]);
L(6) = Link([0 0.080 0 0]);
```

```
% Define the Robot Objects
Robot = SerialLink(L, 'name', 'KUKA KR5 Sixx R650');
Animated_Robot = SerialLink(L, 'name', 'Animated KUKA KR5 Sixx R650'); % Define another robot object with different name because if the robot already exists then that same graphical model will be found and moved.

% Forward Kinematics
FKM = Robot.fkine([0 0 0 0 0 0]) % Forward Kinematic Model (FKM) for home position

% Robot Visualization
figure('NumberTitle', 'off', 'Name', 'Robot Visualization','WindowState','Maximized');
Robot.plot([0 0 0 0 0 0]); % Home position

% Define Joint Angle Variables
th1 = 0;
th2 = 0;
th3 = 0;
th4 = 0;
th5 = 0;
th6 = 0;

% Robot Animation
figure('NumberTitle', 'off', 'Name', 'Robot Animation','WindowState','Maximized');
for i = 0:0.05:pi/2
    th2 = i/2;
    th3 = i/2;
    Animated_Robot.plot([th1 th2 th3 th4 th5 th6]);
end
for i = pi/2:-0.05:0
    th2 = i/2;
    th3 = i/2;
    Animated_Robot.plot([th1 th2 th3 th4 th5 th6]);
end
for i = 0:0.05:pi
    th1 = i;
    th2 = i/4;
    th3 = i/4;
    th4 = i/2;
    th6 = -i/2;
    Animated_Robot.plot([th1 th2 th3 th4 th5 th6]);
end
```

```
-----------------------------------------------------------------
2. Forward Kinematic Model of KUKA KR5 Sixx R650 Manipulator
-----------------------------------------------------------------


FKM =
         0         0         1      0.45
         0        -1         0         0
         1         0         0     0.695
         0         0         0         1
```





3. Kinematic Model of 1 DOF Beam Tilt Mechanism

```
fprintf('-----------------------------------------------------------------\n');
fprintf('3. Kinematic Model of 1 DOF Beam Tilt Mechanism\n');
fprintf('-----------------------------------------------------------------\n');

% Tilt Angle
input = inputdlg('Enter Tilt Angle in Degrees: '); % User input for tilt angle in degrees
theta_degrees = str2double(input); % Convert string input to numerical value
theta = deg2rad(theta_degrees); % Convert degrees to radians
fprintf('Tilt Angle: %0.4f degrees\n', theta_degrees);

if (theta_degrees >= -45 && theta_degrees <= 45) % Mechanical constraint
```

```
    % Define Transformations
    B0_T_P = [cos(theta) -sin(theta) 0 5; sin(theta) cos(theta) 0 11; 0 0 1 0; 0 0 0 1];
    P_T_A0 = [1 0 0 -5; 0 1 0 0; 0 0 1 0; 0 0 0 1];
    B0_T_A0 = B0_T_P * P_T_A0;

    % Coordinates of Points A0 and B0
    x_A0 = B0_T_A0(1,4);
    y_A0 = B0_T_A0(2,4);
    z_A0 = B0_T_A0(3,4);

    x_B0 = 0;
    y_B0 = 0;
    z_B0 = 0;

    % Compute the Actuator Lenghth
    l = sqrt((x_A0 - x_B0)^2 + (y_A0 - y_B0)^2 + (z_A0 - z_B0)^2); % Euclidian distance
between points
    fprintf('Actuator Length: %0.4f units\n',l);

else
    fprintf('The mechanical constraint is not satisfied!\nPlease enter tilt angle in range of
[-45,45] degrees.\n');

end
```

```
-----------------------------------------------------------------
3. Kinematic Model of 1 DOF Beam Tilt Mechanism
-----------------------------------------------------------------
Tilt Angle: 30.0000 degrees
Actuator Length: 8.5264 units
```

Published with MATLAB® R2018a