

Jancarlos Gomez

EEL5661

Homework 1

16 September 2021

Dr. Roth

**Problem 1: 2D Displacements**

**A mobile robotic platform can move only straight ahead or straight backwards, and it can perform only increment rotations of θ = π/4 or θ = -π/4 about a center axis. The robotic vehicle can translate only along its local Y axis, and it can rotate only about its own local Z axis. The room coordinate frame is located on the maze entrance corner of the left wall.**

A picture containing diagram

Description automatically generated

**1.1 Find a sequence of 3x3 homogeneous transformations that take the platform along a planar maze (shown below) from the shown “start” location to the shown “end” location**

Diagram, engineering drawing

Description automatically generated

Diagram, engineering drawing

Description automatically generated

First, we will define all the movements used to complete the maze.

R = Rotation T = Translation

Now we will multiply the movements shown above to create a sequence of transformations.

Sequence of 3x3 Homogeneous Transformations

\* =

**1.2 Are all the above transformations pre-multiplications? Or all post-multiplications? Or a mix? Why so?**

All of the above transformations are post-multiplications because the leftmost matrix transformation occurs first. All transformations are with respect to the local coordinate frame of the rectangle. The rectangle must rotate first before moving in the y-direction. For example, the rotation matrix is post multiplied with the translation matrix to ensure that the rectangle is facing the proper direction before it moves forward.

**1.3 Find the vehicle’s initial and final transformations with respect to the room coordinate frame. What is the overall homogeneous transformation that takes the robotic vehicle from “Start” to “End”?**

We will set the room coordinate frame as the new origin and find the initial and final positions of the rectangle. We can observe our results below in MATLAB:

Graphical user interface, text, application

Description automatically generated

**Figure 1 Finding Initial and Final Positions w.r.t Room Coordinate Frame**

Thus,

Referring to the results obtained earlier when using the local coordinate frame of the rectangle the overall 3x3 homogeneous transformation was:

We can use this same 3z3 homogeneous transformation with our new room coordinate frame to find the final position using the following equation:

Thus, the 3x3 transformation matrix is the same for both the local coordinate frame and the room coordinate frame.

**1.4 Plot (using MATLAB) a sequence of vehicle’s coordinate frames that describes the complete trip taken.**

Chart, scatter chart

Description automatically generated

**Figure 2 Sequence of 2D Displacements on Rectangle**

**1.5 Using MATLAB, animate the vehicle’s trip. Need to just submit the code of the program.**

Graphical user interface, text, application

Description automatically generated

**Figure 3 MATLAB Code for Problem 1**

Video of Animation (Double Click to Open)



**Problem 2: 3D Displacements**

**An old heavy sofa needs to be discarded. The top view map shows part of the geometry information. Feel free to add more information as needed.**

**Diagram

Description automatically generated**

**The sofa dimensions are 7’ x 3’ x 3’ (legs included). The exact starting location of the sofa, inside the house but not too far from the entrance door) is up to you to select. The same for the exact discarding location of the sofa. The bushes are 4’ tall (so don’t attempt to lift the sofa over the bushes). You have to mathematically model (by means of a sequence of 4x4 homogeneous transformation matrices) two possible sofa displacement motion options.**

**2.1 Option A: Two people lift the sofa and carry it parallel to the ground. Suggest (numerically) and explain each transformation that is needed. As you explain the selected figures and type of displacement, be sure to discuss if the transformations are pre- or post-multiplications. Use MATLAB to multiply the transformations to provide you with the overall displacement transformation, and to describe graphically the sofa’s motion.**

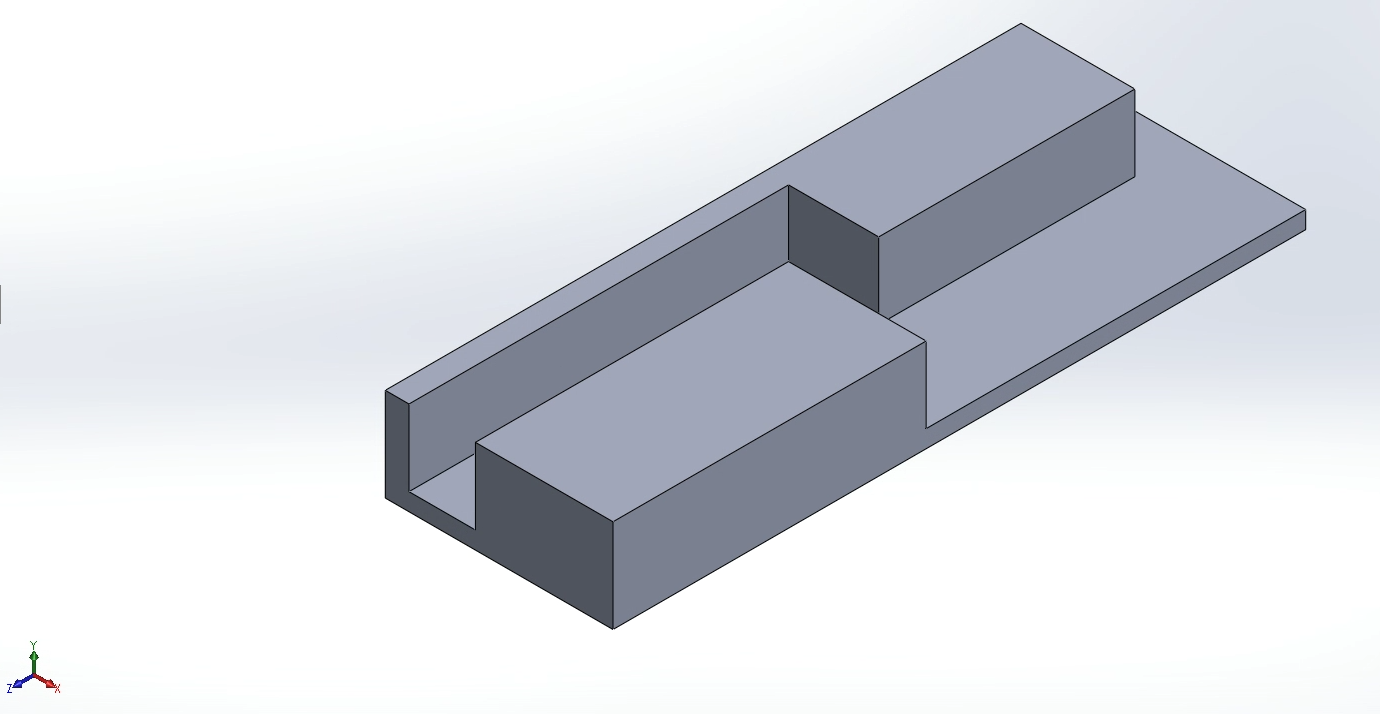
I created 3D models of the sofa and the path in Solidworks to get a better idea of how the sofa will move. Below are the models of the sofa and path:

A picture containing text, businesscard

Description automatically generatedShape

Description automatically generated with medium confidence

**Figure 4 3D Models of Sofa Dimensions**



**Figure 5 3D Model of Path**

The starting position of the sofa is right behind the door of the house. I picked my origin for this problem to be at the center of the meeting point between the sofa and the path. This is a local origin thus when a rotation occurs the axes on the sofa change direction. To make the problem easier I made the sofa be in a position to come out the door easily as shown below:

Graphical user interface, diagram, engineering drawing

Description automatically generated with medium confidence

**Figure 6 Starting Point of Sofa**

Graphical user interface, application

Description automatically generated with medium confidence

**Figure 7 Demonstration of Origin**

The sofa will require one vertical flip because there is not enough room to carry the sofa out in parallel all the way as shown below:

Engineering drawing

Description automatically generated

**Figure 8 Demonstration of Space Issue at the Turning Point**

For this problem, I will be translating the axes used in Solidworks to the ones used in MATLAB because different software use different axis labeling. The first movement the sofa must make is -16.75 in the x-direction:

A picture containing text, businesscard

Description automatically generated

**Figure 9 Sofa Moves -16.75 in x-Direction**

The sofa rotates -90 degrees in the y-direction, so it stands up on its smaller face allowing room to make the turn:

A picture containing text

Description automatically generated

**Figure 10 Sofa Rotates -90 Degrees in y-Direction**

The sofa must move 1.75 in the y-direction in order to have enough room to fall on its long face and keep moving towards the discarding point:

A picture containing text, sky, case

Description automatically generated

**Figure 11 Sofa Moves 1.75 in y-Direction**

The sofa rotates -90 degrees in the z-direction, so it lays on its longer face allowing room to go straight down the path:

Engineering drawing

Description automatically generated

**Figure 12 Sofa Rotates -90 Degrees in z-Direction**

The sofa now must move 16.75 in the z-direction in order to line up with the discarding point:

A picture containing engineering drawing

Description automatically generated

**Figure 13 Sofa Moves 16.75 in z Direction**

The sofa must move 7 in the x-direction to end up at the discarding point:

Engineering drawing

Description automatically generated

**Figure 14 Sofa at Discarding Point**

We will now define all the movements used to complete the path.

R = Rotation T = Translation

Now we will multiply the movements shown above to create a sequence of transformations.

Sequence of 4x4 Homogeneous Transformations

All the above transformations are post-multiplications because the leftmost matrix transformation occurs first. All transformations are with respect to the local coordinate frame of the sofa. Below is the MATLAB code for the movement of the Sofa carried by two people lifting it parallel to the ground except for the turning point due to the issue aforementioned:

Text

Description automatically generated

**Figure 15 MATLAB Code for 2.1 Option A**

Video of Animation (Double Click to Open)



Chart

Description automatically generated

**Figure 16 Sequence of 3D Displacements on Sofa (Blue = Translation, Red = Rotation)**

**2.2 Option B: One person tumbles the sofa all the way. A sketch of the tumbling motion is shown below. Repeat the requirements of 2.1.**

A picture containing text

Description automatically generated

**Hint: You may want to consider placing two coordinate frames on the sofa – one on its left side and one on its right side. The two frames are related by a constant transformation matrix. Say that the first tumbling is a rotation about the right-side frame. The next sofa rotation is about the leftside frame, but we can relate the left-side frame to the right-side frame.**

Diagram, engineering drawing

Description automatically generated

Using the same concept as problem 2.1; however, this time we will start 4.25 feet behind the door because each complete rotation on the long face of the sofa covers 7 feet and on the short face 3 feet. Since the sofa is 3 feet wide and the path at the turn is both 3.5 feet wide in the x and y directions, the closest face on the sofa must be 0.25 feet away, when the sofa is in the vertical position as shown in the image below:

A picture containing text

Description automatically generated

**Figure 17 Sofa Vertical Placement for Turn**

The previous starting point cannot be used because the number of rotations will not allow us to get in the proper position for the turning point, hence the offset of putting the sofa further inside the house at 4.25 feet in the x-direction. Below we can find the Sequence of 4x4 Homogeneous Transformations for the right and left sides of the sofa.

Sequence of 4x4 Homogeneous Transformations using the **Right Side**

Sequence of 4x4 Homogeneous Transformations using the **Left Side**

All the above transformations are post-multiplications because the leftmost matrix transformation occurs first. All transformations are with respect to the two local coordinate frames on each end of the sofa. Below is the MATLAB code for the movement of the Sofa by tumbling motion:

Text

Description automatically generated

Text

Description automatically generated

**Figure 18 MATLAB Code for 2.2 Option B**

Video of Animation (Double Click to Open)



The right end of the sofa is shown in blue, and the left end of the sofa is shown in red:

Chart, scatter chart

Description automatically generated

**Figure 19 Top View of Sequence of 3D Displacements**

Chart, scatter chart

Description automatically generated

**Figure 20 Sequence of 3D Displacements on Sofa**

**Problem 3: Task Description**

**The base frame of a 6R robot (that is not shown), equipped with a suction tool (that can be turned on and off), is identical with the room coordinate frame. Two identical 5x3x3 boxes that one of their sides is sticky (shown by R) are to be glued (R side to R side, to form a box of dimensions 5’x6’x3’) on top of a cylindrical pedestal (of height 7’, whose base lies on the XsYs plane and the base center is at (0,6’,0)). The top glued box comes from the top of another pedestal (of height 4’, whose base center is at (-7’,0,0)) at the shown orientation. The combined big box needs to be carried away and be placed on a conveyor located at a place of your choice.**

**Make additional assumptions as needed (for example, about the exact base location of the box in the initial and final position). Assume that the glue is fast acting and strong, and that the suction force is strong enough to lift the combined big box.**

**List all the robotic task steps. For each step show, and explain briefly, the kinematic equation (similarly to what was done in the lectures). What is the overall homogeneous transformation of the displacement of the box from its initial position to the final position?**

Diagram

Description automatically generated

Diagram, engineering drawing

Description automatically generated

On the previous page, we can observe the 3D steps the robot must take to unite the two boxes and drop off the final big box onto a conveyor. We can also observe a 2D aerial view of the problem and depict what obstacles are present and the location of the boxes. This is very useful for determining how long each joint should be. I have placed a conveyor belt on (7.5, 0, 4), this will serve as the final location for the big box. We are now ready to determine what steps the robot must take to complete its task.

**Task Steps**

**Step 1:** Robot moves the suction tool to the top of the vertical box.

**Step 2:** Robot turns on its suction tool.

**Step 3**: Robot changes the orientation of the box to horizontal to line up with the other box.

**Step 4**: Robot moves that box to a position over the other box.

**Step 5**: Robot turns off the suction tool to make the box fall and stick to the other box.

**Step 6:** Robot turns on the suction tool and picks up the combined big box.

**Step 7:** Robot moves the box to the drop-off point.

**Step 8**: Robot turns off the suction tool to release the box.

For this problem, we are only tasked with writing the robotic task steps and explaining the kinematic equation theoretically. The displacements of the boxes are to be explained mathematically. Since we are provided sufficient information in the problem statement to create a 6R Robot capable of completing the task, I decided to go a step further and develop the actual 6R Robot in MATLAB. On the previous page, we can observe that through trial and error I was able to determine the length of each joint for the 6R Robot. I then placed my findings into a MATLAB code and sought to find the twist angle through trial and error, Once I had these two parameters, I was able to set up a D-H parameter table for the 6R Robot with suction tool.

My approach was to start by sketching the general movement the robot would take first, then animating it in MATLAB, in order to obtain the kinematic equations. I moved the world frame/base frame to a different location because the instructions and drawing provided had some contradictions, so I adjusted the problem slightly. The start and finish locations on the drawing given in the problem are contradictory to the start and end locations given in the task description. Below is my MATLAB code with detailed comments to explain the process:

**MATLAB Code**

clc; clear; close all;

% create links using D-H parameters

% L = Link([theta, d, a, alpha]

L(1) = Link([0 0 0 -pi/2]); %base of closest cylinders is (0,1,0) hence the first rotating arm cannot be longer than 1

L(2) = Link([0 0 9.5 -pi/2]);

L(3) = Link([0 0 4 pi]);

L(4) = Link([0 0 3 pi]);

L(5) = Link([0 0 0.5 pi]);

L(6) = Link([0 0 0 0]);

% Joint limits

L(1).qlim = pi/180\*[0 360];

L(2).qlim = pi/180\*[0 360];

L(3).qlim = pi/180\*[0 360];

L(4).qlim = pi/180\*[0 360];

L(5).qlim = pi/180\*[0 360];

L(6).qlim = pi/180\*[0 360];

% create the robot model

Robot = SerialLink(L);

Robot.name = 'Robot'

set(gcf, 'Position', get(0, 'Screensize'));

plotvol([-15, 15, -15, 15,0,15])

% starting position

qz = [pi/2 -pi/2 -pi/2 0 -pi/2 0];

% next position

qr = [pi/2 -pi/2 -pi/2 0 -pi/2 0];

% generate a time vector

t = [0:0.056:1];

% computes the joint coordinate trajectory

b = jtraj(qz, qr, t);

% direct kinematics for each joint co-ordinate

step1 = Robot.fkine(b)

title('Suction tool on')

%Suction tool turns on

%Assume sticky face is the one facing the origin

Robot.plot(b)

qz = qr;

% next position

qr = [pi/2 -pi/2 -pi/2 0 0 0];

% computes the joint coordinate trajectory

b = jtraj(qz, qr, t);

% direct kinematics for each joint co-ordinate

step2 = Robot.fkine(b)

Robot.plot(b)

qz = qr;

% next position

qr = [pi/2 -pi/2 -pi/2 -pi/2 -pi/2 0];

% computes the joint coordinate trajectory

b = jtraj(qz, qr, t);

% direct kinematics for each joint co-ordinate

step3 = Robot.fkine(b)

Robot.plot(b)

qz = qr;

% next position

qr = [0 -pi/2 -pi/2 -pi/2 -pi/2 0];

% computes the joint coordinate trajectory

b = jtraj(qz, qr, t);

% direct kinematics for each joint co-ordinate

step4 = Robot.fkine(b)

Robot.plot(b)

title('Suction tool off')

pause(2)

%We grabbed the first box on the center of the top face so there is 1.5 in both directions

%the bottom most portion of this box is at height 11; the box in the arm is 1

%above the other box.

%Suction tool turns off box falls straight down and sticks to the other box

%making one big box

%the height of the arm is 12.5 which is shorter than the top of the big box at 13

%so we are safe to grab the big box from this point. Assume the suction tool is strong enough to grip the box from

%any point

qz = qr;

% next position

qr = [0 -pi/2 -pi/2 -pi/2 -pi/2 0];

% computes the joint coordinate trajectory

b = jtraj(qz, qr, t);

% direct kinematics for each joint co-ordinate

step5 = Robot.fkine(b)

title('Suction tool on')

Robot.plot(b)

%Suction tool turns on; grabs big box and takes it to the conveyor located

%at (7.5,0,4)

qz = qr;

% next position

qr = [-pi/2 -pi/2 -pi/2 -pi/2 -pi/2 0];

% computes the joint coordinate trajectory

b = jtraj(qz, qr, t);

% direct kinematics for each joint co-ordinate

step6 = Robot.fkine(b)

Robot.plot(b)

qz = qr;

% next position

qr = [-pi/2 -pi/2 -pi/2 0 0 0];

% computes the joint coordinate trajectory

b = jtraj(qz, qr, t);

% direct kinematics for each joint co-ordinate

step7 = Robot.fkine(b)

Robot.plot(b)

title('Suction tool off')

%suction tool off drop big box on conveyor belt

Video of Animation (Double Click to Open)

****

We can observe in the animation above that the 6R Robot functions properly. It follows the same steps outlined in the handwritten page above.

From the code above I was able to determine the following D-H parameters:

**D-H parameters of Robot**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Links |  | d | a |  |
| 0-1 | 0 | 0 | 0 |  |
| 1-2 | 0 |  | 9.5 |  |
| 2-3 | 0 | 0 | 4 | 180 |
| 3-4 | 0 |  | 3 |  |
| 4-5 | 0 | 0 | 0.5 |  |
| 5-6 | 0 | 0 | 0 | 0 |

With the D-H parameters we can use the following formula to find the A matrices:

Where ,

We will now determine the A matrices for each link:

We will now use the following equation to calculate the T matrix:

We can put this into MATLAB as shown below to generate the T matrix easily:

A picture containing calendar

Description automatically generated

**Figure 21 Calculating the T Matrix**

With the T matrix, we are now able to find the kinematic equations.

**Kinematic Equations**

A kinematic equation is defined as:

A kinematic equation consists of the following:

• Z - the transformation of the robot base frame with respect to the world frame.

• T - the transformation of the end of the manipulator with respect to its base.

• S – the representation of a tool with respect to the robot’s end. (Suction tool)

• Pn - the transformation of each step with respect to the world frame.

• PAn - the representation of the desired robot suction tool pose with respect to the object coordinate frame.

Since our base frame is equal to the world frame our Z is equal to the following:

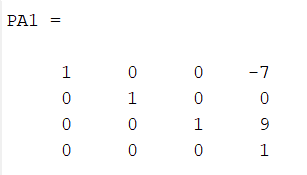
We will use the following equation to find Pn:

We can obtain PA by translation and rotation of the suction tool with respect to our base frame. I will be using the following MATLAB skeleton code to obtain results; I will be changing values for x and makehgtform():

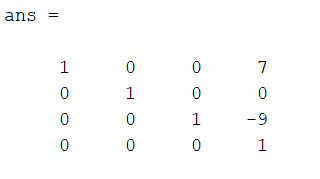
Calendar

Description automatically generated

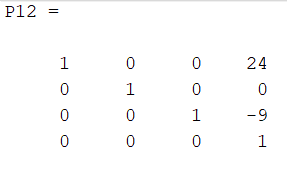
For step 1 we need the robot to move the suction tool to (-7,0,9), I have assumed that the suction tool is already facing downward. For step 2 we need to turn the suction tool on. From the MATLAB code we obtained for PA1:



Take the inverse to obtain



Now the P1,2 transformation for steps 1 and 2 can be found:



Step 3 is for the robot to change its orientation by 90 degrees to make the vertical box become horizontal. At this point, the suction tool goes through a translation and a rotation.

\*

Scatter chart

Description automatically generated with low confidence

A picture containing scatter chart

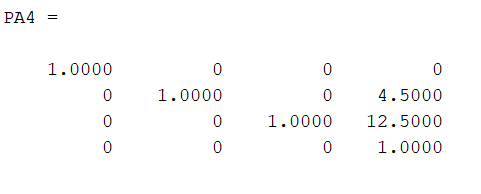
Description automatically generated

Now the P3 transformation for step 3 can be found:

A picture containing text

Description automatically generated

For steps 4 and 5 the robot moves the suction tool to a position on top of the other box (0,4.5,12.5) and releases the box.



Chart, scatter chart

Description automatically generated

Now the P4,5 transformation for steps 4 and 5 can be found:

Scatter chart

Description automatically generated with medium confidence

For steps 6, 7, and 8 – The robot turns on the suction tool picks up the big box and moves it to the conveyor located at (7.5,0,9.5), and turns off the suction tool.

Scatter chart

Description automatically generated

Chart, scatter chart

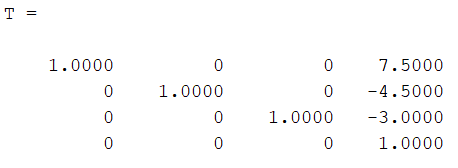
Description automatically generated

Now the P6,7,8 transformation for steps 6, 7, and 8 can be found:

The overall homogenous transformation for the big box can be found by taking the coordinates of the suction tool when suctioning the big box at its starting position and the coordinates of the tool at the drop-off position. We will use the following MATLAB code to obtain the overall homogeneous transformation:

Text

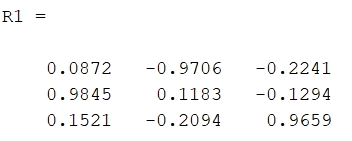
Description automatically generated



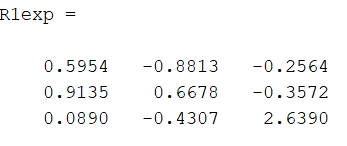
**Problem 4: Rotation Representations**

**Solve this problem using MATLAB RTB commands. No need to solve manually.**

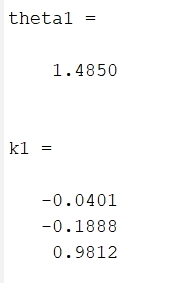
**4.1 Find the rotation matrix R1 obtained by the ZYZ Euler Angles (30** **°, -15** **° , 54** **°).**



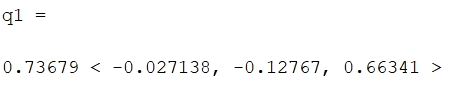
**4.2 What is the matrix exponential representation of R1?**



**4.3 What are the axis of rotation unit vector and the angle of rotation for R1?**



**4.4 What is the unit quaternion representation of R1?**



**4.5 Find the rotation matrix R2 = R1∙R1, and then repeat 4.2-4.4 for R2.**

**Rotation Matrix R2**

Text

Description automatically generated with low confidence

**Exponential Representation of R2**

Text

Description automatically generated

**Axis of Rotation Unit Vector and the Angle of Rotation for R2**

Text

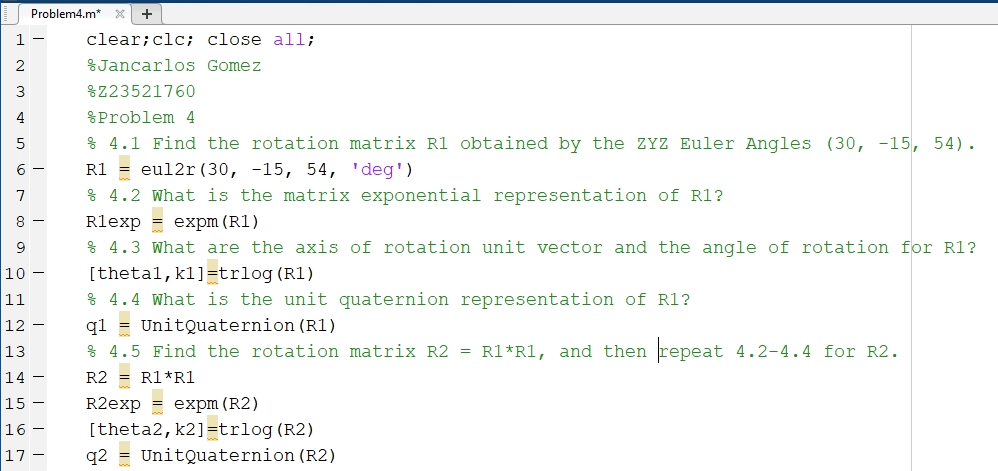
Description automatically generated with low confidence

**Unit Quaternion Representation of R2**

A picture containing chart

Description automatically generated

**MATLAB Code**



**Figure 22 MATLAB Code for Problem 4**

**References**

****

[**http://www.me.unm.edu/~starr/teaching/me582/postmultiply.pdf**](http://www.me.unm.edu/~starr/teaching/me582/postmultiply.pdf)

Robotics Toolbox for MATLAB® Release 10