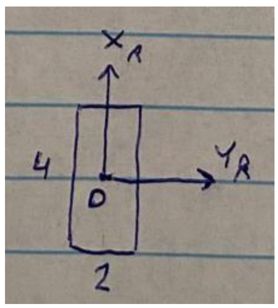
COT 4930 COT 5930 EEL 4930 EEL 5661 Robotic Applications (Fall 2022)

Shaun Pritchard 9/26/2022

Homework 1

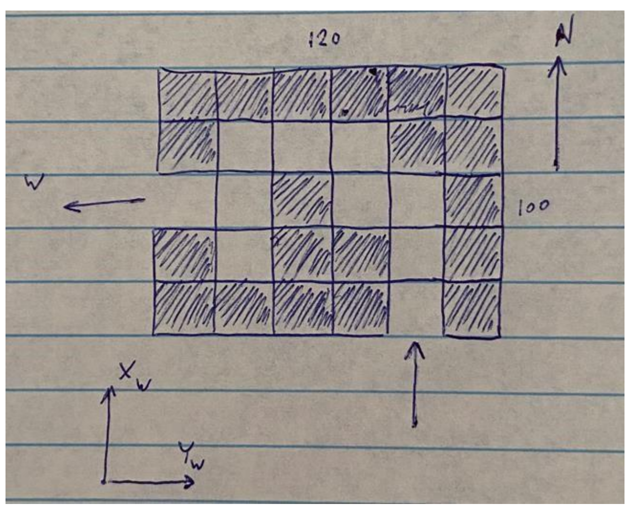
Problem 1: 2D Displacements

A mobile robotic platform can move only straight ahead or straight backwards, along its local X_R axis, and it can perform only incremental rotations of $\theta = \pi/2$ or $\theta = -\pi/2$ [radians] about a vertical center axis that passes through the local point O.



The robotic vehicle cannot translate along its local Y_R axis.

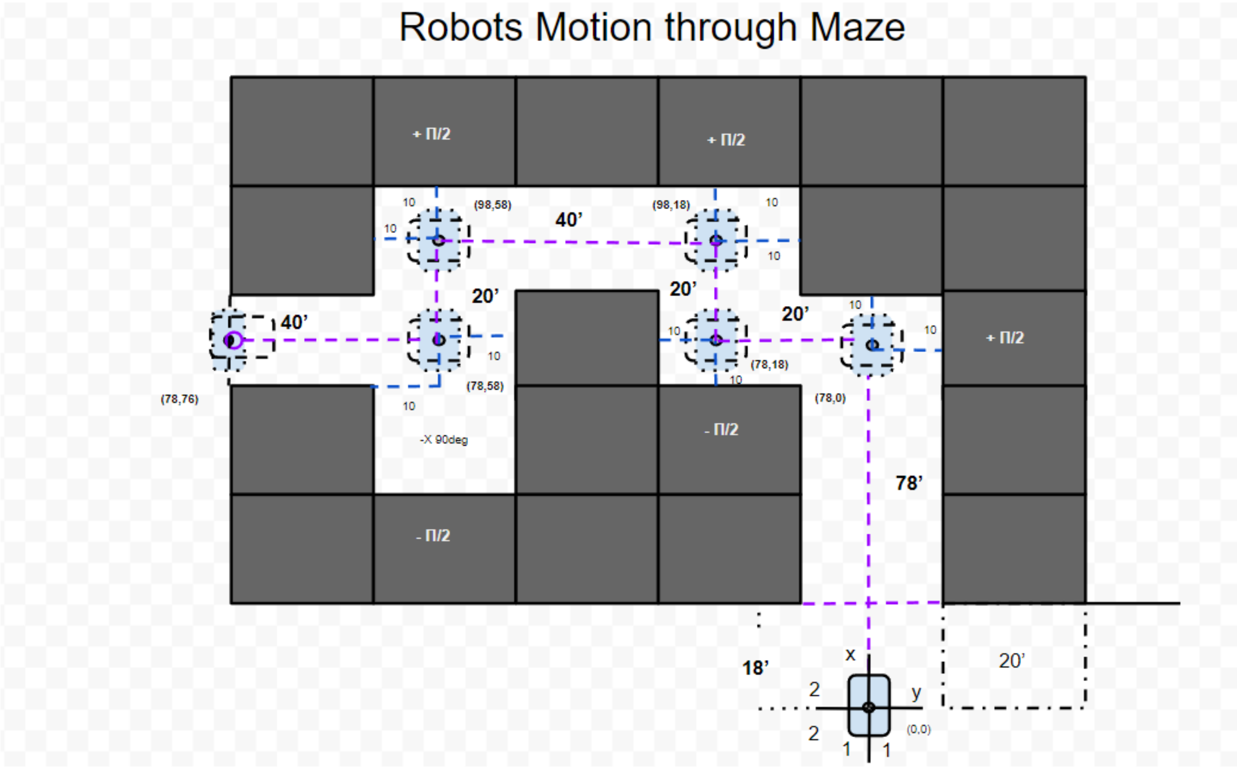
The robot is standing **initially** near the south entrance of a maze, at a distance of 20' (along X_R), facing the entrance along the north direction. The robot needs to traverse the maze and come out via the west outlet. The maze and world coordinate frame are shown below:



The maze is a 6 by 5 grid of squares of size 20'. Dark color means a wall and light color means an available space. Note that the robot's dimensions are much smaller than each grid square length, so maneuvering inside the maze is not difficult.

1.1 Place the world coordinate frame at some arbitrary location (say, the SW corner of the maze, or along the south wall 40' east of the SW corner, or any other choice that you decide to make).

The world coordinate is positioned at South 40′ position



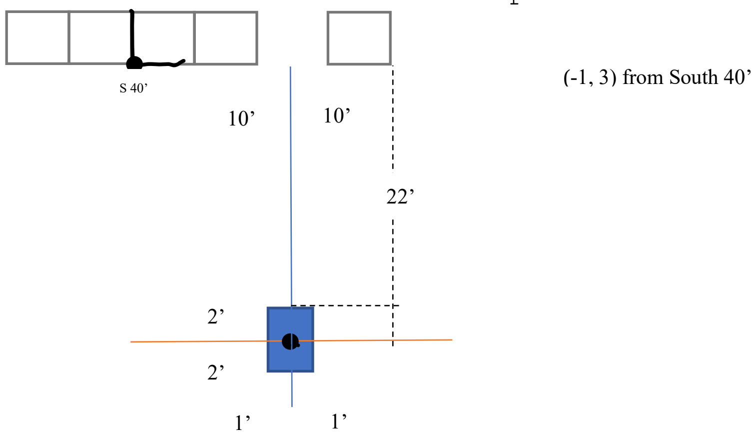
1) What is the 2D homogeneous transformation of the robot, at its initial position, with respect to the world frame?

Moves in negative X direction than positive y direction. The world coordinate is positioned at South 40′ position

$$\begin{vmatrix} -1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{vmatrix}$$

From the SW corner at 40’ the robot is at position (X: -22, Y:50) positions in the (x,y) direction from the south west corner of the world coordinate.

The initial position is centered that of 4′ x 2′ meaning the direct center is P = (2′x 1′) of the robot 20′ south of the initial entrance in the center of 20′ which C=10′ where the world coordinate frame is at the South position at 40′.

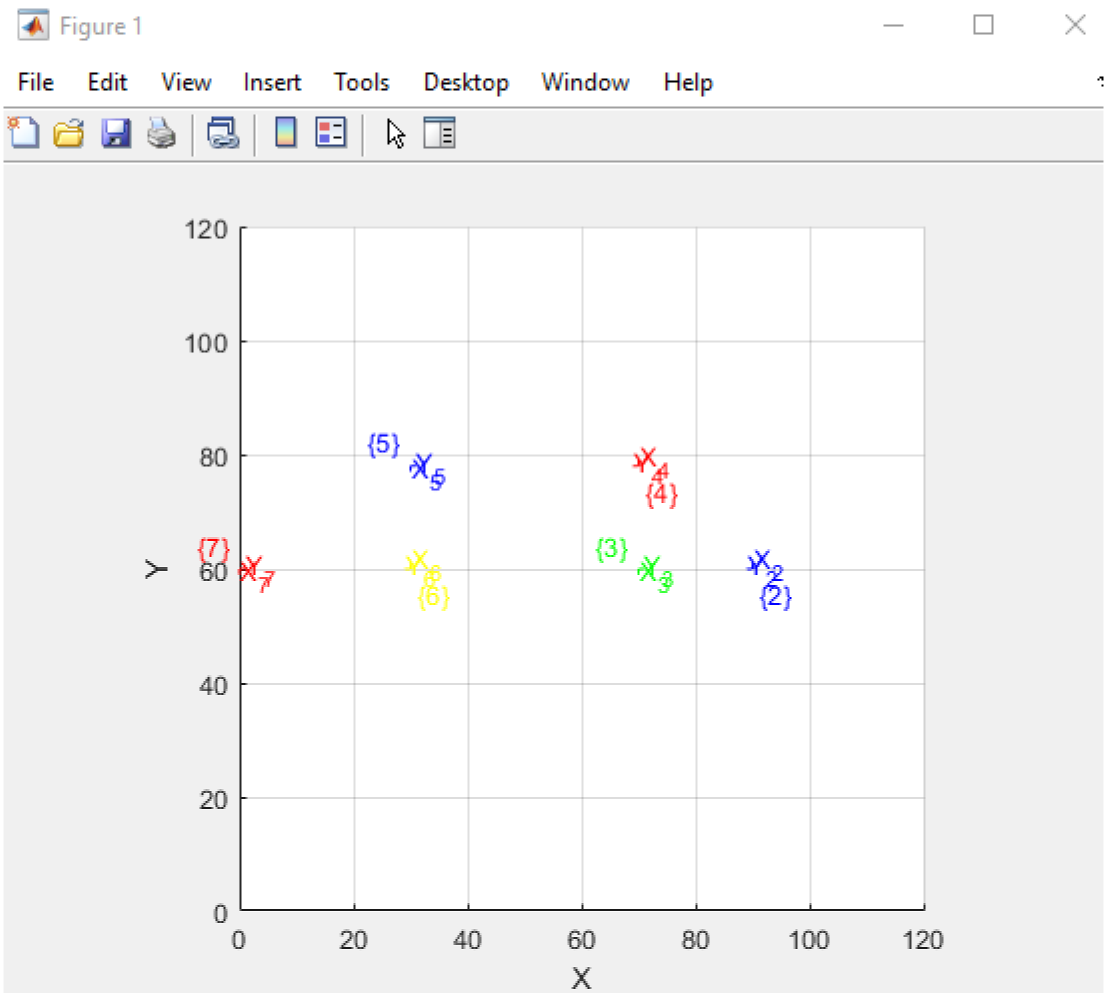


2) What is the 2D homogeneous transformation of the robot with respect to the world frame, just as it comes out of the west outlet? These homogeneous transformations are 3x3 matrices.

world coordinate is positioned at South 40′ position to last position up[on exit of maze

```
| 0   1  0 |
|-1   0  0 |
| 0   0  1 |
```

1.2 Because the robot’s motions are clearly defined with respect to its own local coordinate frame, it makes sense to model a sequence of robot motions (inside the maze) by post-multiplication of 2D homogeneous transformations. <u>Assume that a human operator drives the robot by a remote-control device, and that the human operator sees the whole maze including the robot inside</u>. Decide what should be the sequence steps of traversing the maze (translation by …, followed by rotation of …., followed again by translation, etc.), find each step’s 2D 3x3 homogeneous transformation matrices, and let MATLAB perform the multiplication of the matrices.



By the cube position:

```
R1 = trot2(-pi/2)

R2 = trot2( pi/2)

% Initila Position
F0 = transl2(5,-1)
plotvol([0 6 0 5])
trplot2(F0, 'frame', '1', 'color', 'r')


T1= transl2(5,3) * R2
trplot2(T1, 'frame', '2', 'color', 'b')

T2= transl2(4,3) * R1
trplot2(T2, 'frame', '3', 'color', 'g')

T3= transl2(4,4) * R2
trplot2(T3, 'frame', '4', 'color', 'r')

T4= transl2(2,4)* R1
```

```
trplot2(T4, 'frame', '5', 'color', 'b')

T5= transl2(2,3)* R2
trplot2(T5, 'frame', '6', 'color', 'y')

T6= transl2(0,3)* R1
trplot2(T6, 'frame', '7', 'color', 'r')
```

```
R1 =

    0.0000    1.0000         0
   -1.0000    0.0000         0
         0         0    1.0000


R2 =

    0.0000   -1.0000         0
    1.0000    0.0000         0
         0         0    1.0000


F0 =

     1     0     5
     0     1    -1
     0     0     1


T1 =

    0.0000   -1.0000    5.0000
    1.0000    0.0000    3.0000
         0         0    1.0000


T2 =

    0.0000    1.0000    4.0000
   -1.0000    0.0000    3.0000
         0         0    1.0000
```

T3 =

```
    0.0000   -1.0000    4.0000
    1.0000    0.0000    4.0000
         0         0    1.0000
```

T4 =

```
    0.0000    1.0000    2.0000
   -1.0000    0.0000    4.0000
         0         0    1.0000
```

T5 =

```
    0.0000   -1.0000    2.0000
    1.0000    0.0000    3.0000
         0         0    1.0000
```

T6 =

```
    0.0000    1.0000         0
   -1.0000    0.0000    3.0000
         0         0    1.0000
```

By 20' increments centered on robot

```
R1 = trot2(-pi/2)
R2 = trot2( pi/2)

% Initila Position
F0 = transl2(90,-18)
plotvol([0 120 0 120])
trplot2(F0, 'frame', '1', 'color', 'r')


T1= transl2(90,60) * R2
trplot2(T1, 'frame', '2', 'color', 'b')

T2= transl2(70, 60) * R1
trplot2(T2, 'frame', '3', 'color', 'g')

T3= transl2(70,78) * R2
```

```
trplot2(T3, 'frame', '4', 'color', 'r')

T4= transl2(30,78)* R1
trplot2(T4, 'frame', '5', 'color', 'b')

T5= transl2(30,60)* R2
trplot2(T5, 'frame', '6', 'color', 'y')

T6= transl2(0,60)* R1
trplot2(T6, 'frame', '7', 'color', 'r')

R1 =

    0.0000    1.0000         0
   -1.0000    0.0000         0
         0         0    1.0000


R2 =

    0.0000   -1.0000         0
    1.0000    0.0000         0
         0         0    1.0000


F0 =

     1     0    90
     0     1   -18
     0     0     1


T1 =

    0.0000   -1.0000   90.0000
    1.0000    0.0000   60.0000
         0         0    1.0000


T2 =

    0.0000    1.0000   70.0000
   -1.0000    0.0000   60.0000
         0         0    1.0000


T3 =

    0.0000   -1.0000   70.0000
    1.0000    0.0000   78.0000
         0         0    1.0000


T4 =
```

```
    0.0000    1.0000   30.0000
   -1.0000    0.0000   78.0000
         0         0    1.0000


T5 =

    0.0000   -1.0000   30.0000
    1.0000    0.0000   60.0000
         0         0    1.0000


T6 =

    0.0000    1.0000         0
   -1.0000    0.0000   60.0000
         0         0    1.0000
```

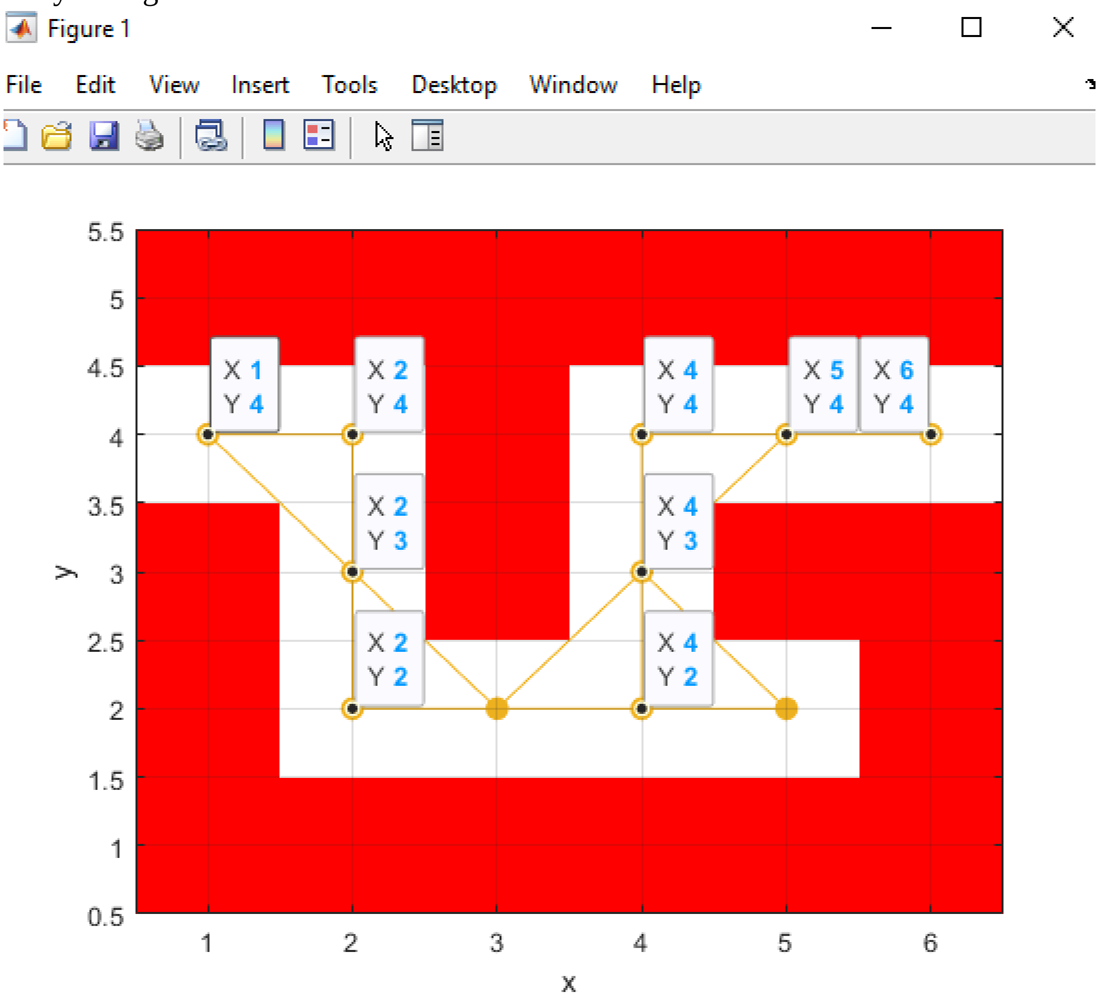
1.3 Now assume that the robot moves autonomously. It has a sensor that can detect if there is an obstacle ahead. The robot obviously needs to move straight ahead, as far as it can. Decide how the robot should interpret the readings coming from the obstacle-avoidance sensor (Turn left? Turn right? Take a reverse motion and then turn?). Again, write down the sequence of motions and their homogeneous transformations. Many will be identical to what you wrote in (1.2), but here and there you will have to deviate. Again, let MATLAB perform the multiplications.

Start at initial position facing north at F0 position

```
F0 = transl2(90,-18)

Move forward in north direction  78' to position (90,60)
Then turn -pi/2 90 degrees west
T1= transl2(90,60) * R2
trplot2(T1, 'frame', '1', 'color', 'b')

Move forward in west direction  18' to position (70,60)
Then turn pi/2 90 degrees north
T2= transl2(70, 60) * R1
trplot2(T2, 'frame', '2', 'color', 'g')

Move forward in north direction  18' to position (70,78)
Then turn -pi/2 90 degrees west
T3= transl2(70,78) * R2
trplot2(T3, 'frame', '4', 'color', 'r')


Move forward in west direction  18' to position (30,78)
Then turn -pi/2 90 degrees south
T4= transl2(30,78)* R1
trplot2(T4, 'frame', '5', 'color', 'b')
```

```
Move forward in south direction  78’ to position (90,60)
Then turn -pi/2 90 degrees west
T5= transl2(30,60)* R2
trplot2(T5, 'frame', '6', 'color', 'y')

Move forward in west direction  78’ to position (90,60)
T6= transl2(0,60)* R1
trplot2(T6, 'frame', '7', 'color', 'r')
```

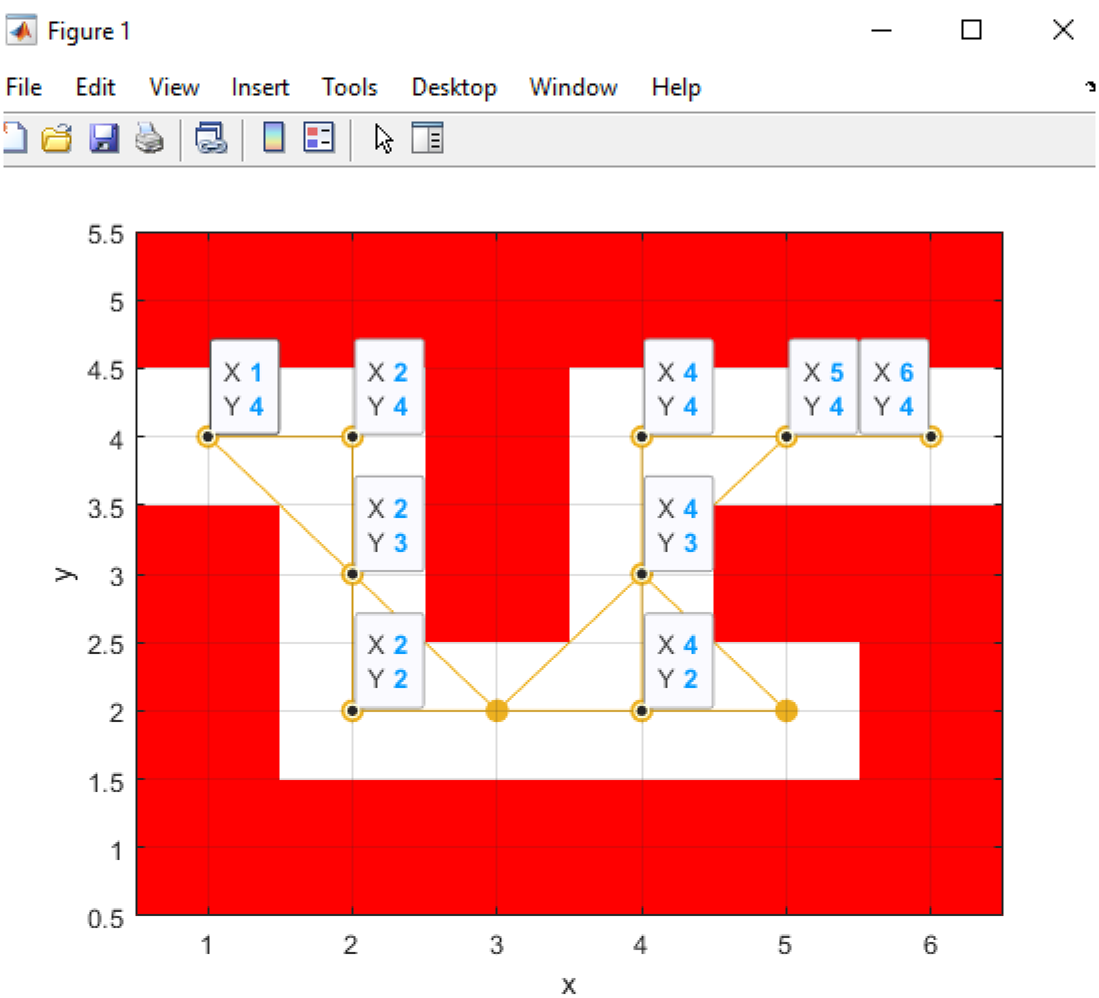
1.4 Your maze-traversing method, that you developed in (1.3), should work for most, if not all, other mazes. Create a new maze (maybe slightly larger than the previous one and show that your algorithm works.




```
track = [1 1 1 1 1 1;
         1 0 0 0 0 1;
         1 0 1 0 1 1;
         0 0 1 0 0 0;
         1 1 1 1 1 1];

obs = [0 0 0 0 0 0;
       0 0 0 0 0 0;
```

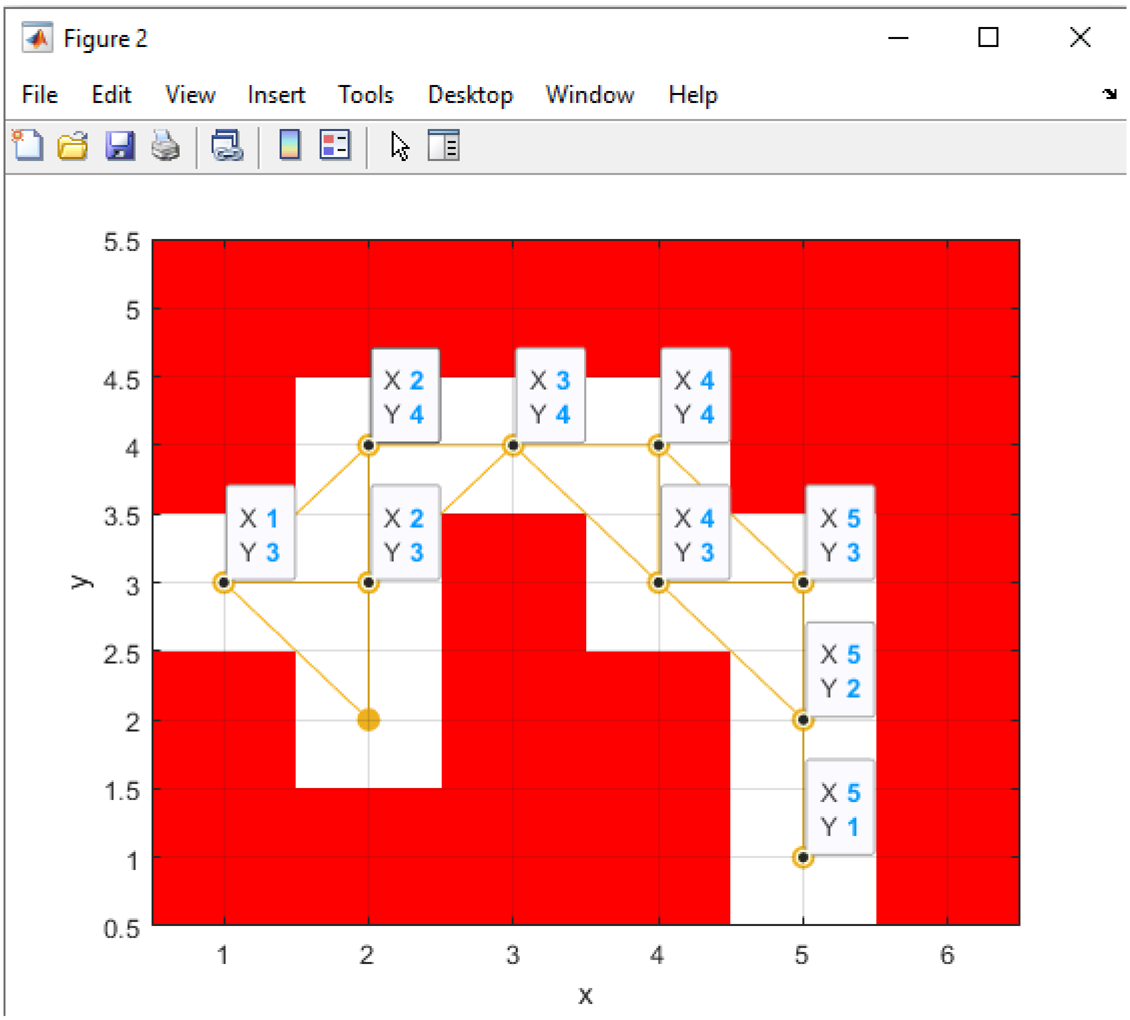
```
       0 0 0 0 0 0;
       0 0 0 0 0 0;
       0 0 0 0 0 0];


goal = [6,2]; % goal point
start = [1,2]; % starting
prm = PRM(track); %create nav object
prm.plan()
%prm.query(start, goal)
prm.plot
```

1.5 Bonus 0.5% coding activity: Write the MATLAB code that plots your 1.4 robot motion overlayed on top of the maze's map. MATLAB has a "map" command. RTB has a "makemap" command.



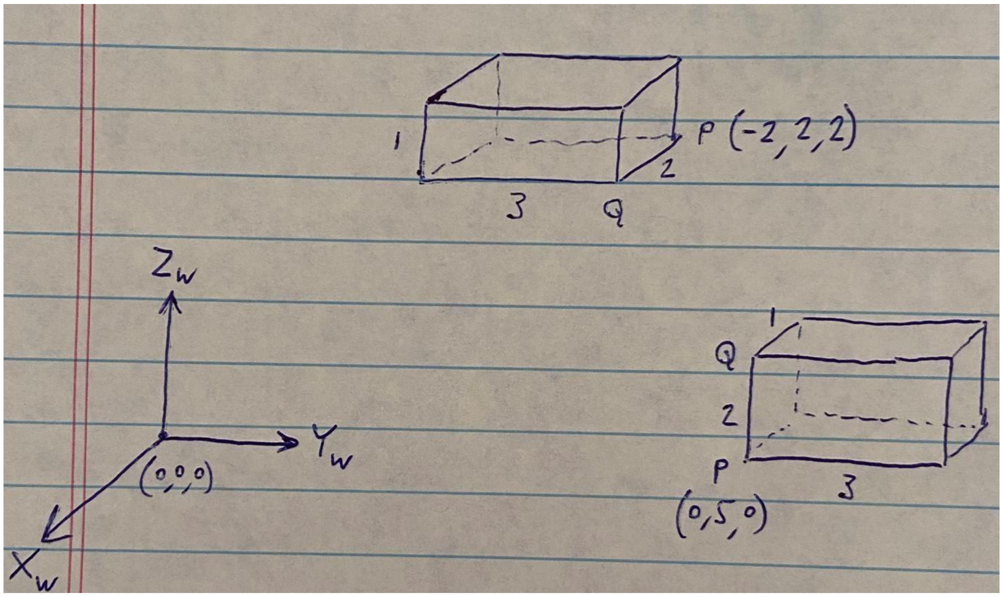

```
track = [1 1 1 1 0 1;
         1 0 1 1 0 1;
         0 0 1 0 0 1;
```

```
         1 0 0 0 1 1;
         1 1 1 1 1 1];

obs = [0 0 0 0 0 0;
       0 0 0 0 0 0;
       0 0 0 0 0 0;
       0 0 0 0 0 0;
       0 0 0 0 0 0];

goal = [5,1]; % goal point
start = [1,3]; % starting
prm = PRM(track); %create nav object
prm.plan(start, goal)
prm.plot
```



Problem 2: 3D Displacements

In the picture below you see a rectangular box of dimensions 1'x2'x3' in two poses with respect to the world coordinate frame {W}.



At the initial position of the box, the XYZ coordinates of point P are (0,5,0). At the final position of the box, point Q is at (-2,2,2) with respect to the world frame. Note a second point Q, at each of the positions. **Error Correction: In the final position, point Q is at (-2,2,2), not point P.**

2.1

1) What is the 4x4 homogeneous transformation of the first pose of the box with respect to the world frame?

T1_P =

1	0	0	0
0	1	0	5
0	0	1	0
0	0	0	1

T1_Q =

$$\begin{matrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 5 \\ 0 & 0 & 1 & 2 \\ 0 & 0 & 0 & 1 \end{matrix}$$

2) What is the transformation of the second position with respect to the world frame? What is the transformation from the first pose of the box to its second pose?

T2_P =

$$\begin{matrix} \mathbf{1} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{1} & \mathbf{0} & \mathbf{2} \\ \mathbf{0} & \mathbf{0} & \mathbf{1} & \mathbf{2} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{1} \end{matrix}$$

T2_Q =

$$\begin{matrix} \mathbf{1} & \mathbf{0} & \mathbf{0} & \mathbf{-2} \end{matrix}$$

0	**1**	**0**	**2**
0	**0**	**1**	**2**
0	**0**	**0**	**1**

2.2 Describe the sequence of "simple displacements" that take the box from its initial pose to its final pose. By "simple displacement step" we mean "translation, or a rotation by 90^0 or -90^0".

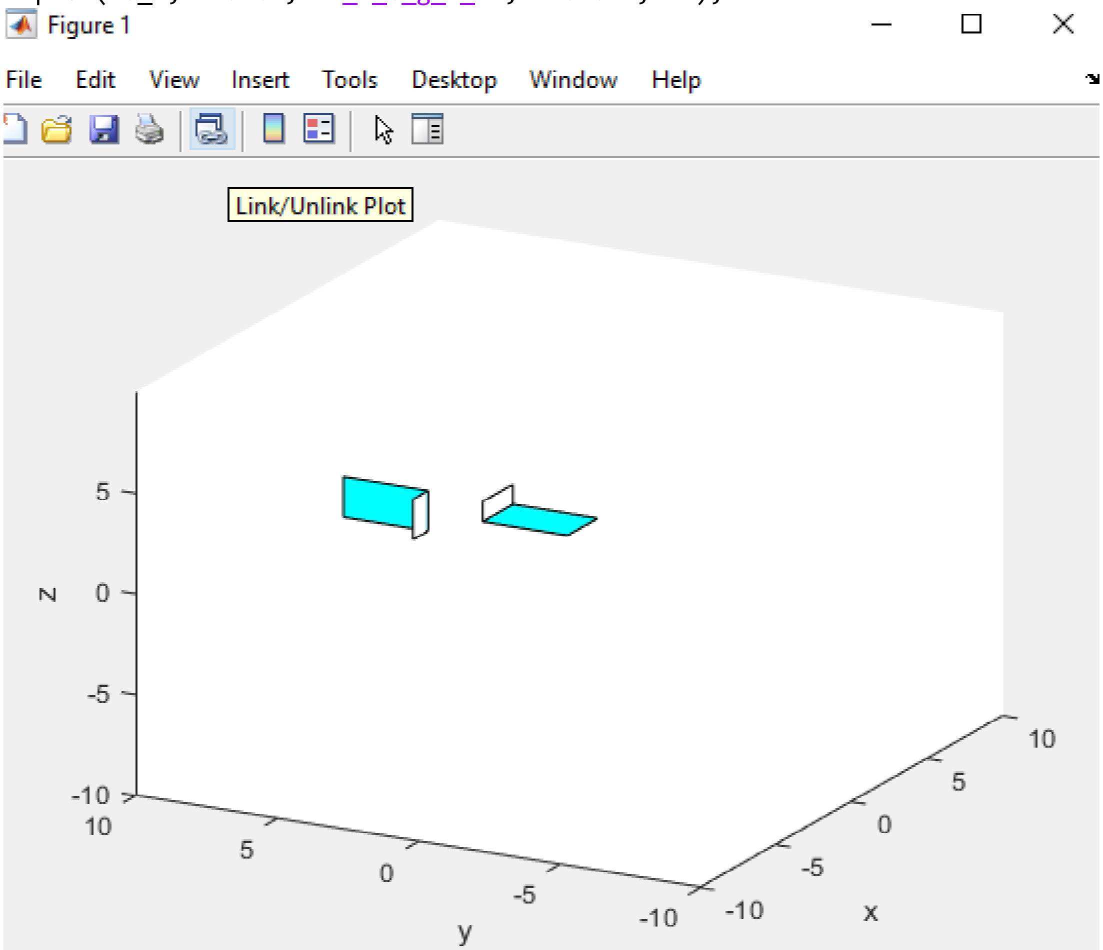
Assume that each step is done with respect to the box's local coordinate frame arrived at the end of each previous step. Are you creating a sequence of post-multiplied transformations, or premultiplied transformations? (Answer: Post multiplications).

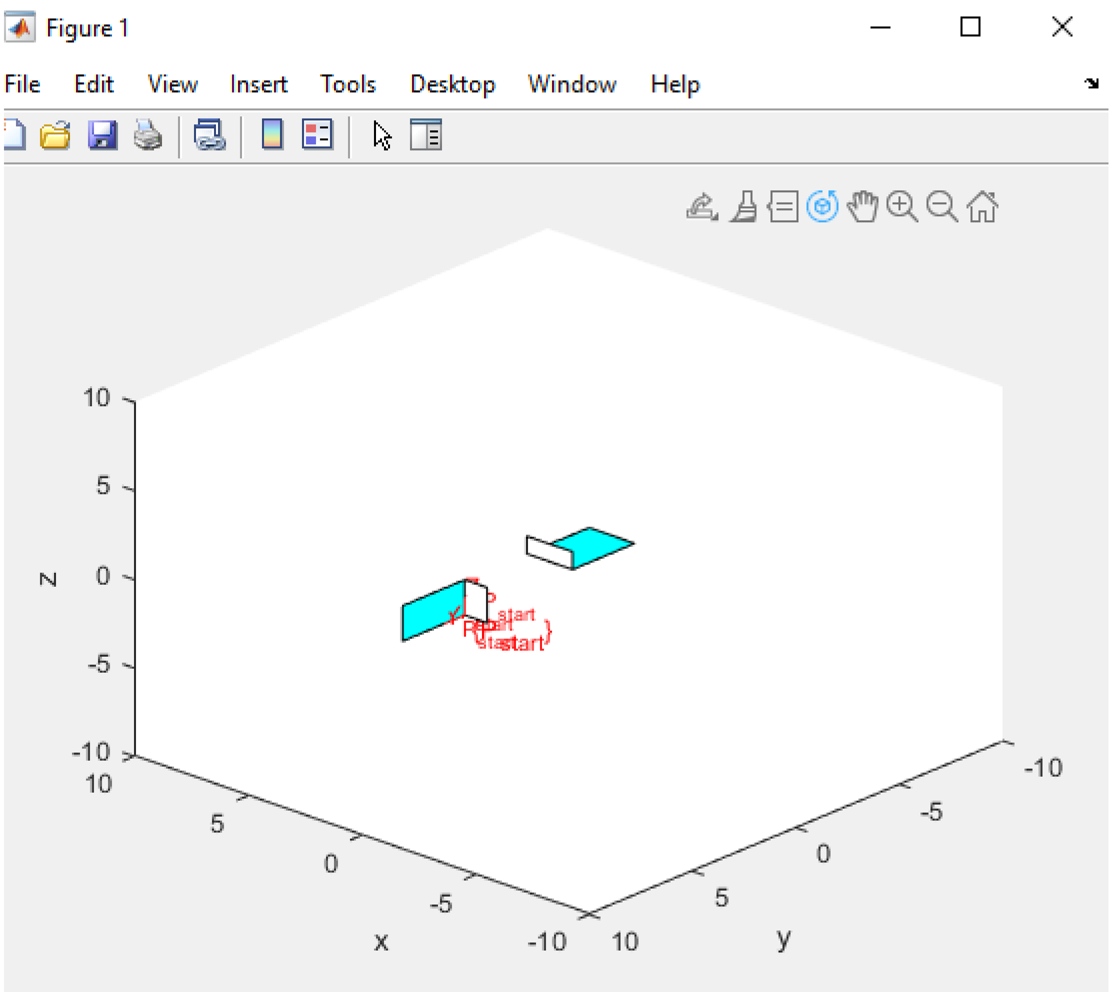
We are using post multiplication the initial position from W is (0,0,0) P1 moves only in the goes in the Y direction by 5 P1 = (0,5,0) Q1=(0,5,2) , than using two of the rotations and one rotation that is used twice to flip it 90deg twice to make the final position, than we see P2 = (-0,2,2) Q2 =(-2,2,2)

2.3 simple displacements' homogeneous transformations? Show a sketch of the box position and orientation at the end of each step. Do the sketching either manually or with the aid of MATLAB.

```
%    P Q
x1 = [0 0 0 0];
y1 = [5 5 8 8];
z1 = [0 2 2 0];
%     P Q
x2 = [0 0 -1 -1];
y2 = [5 5 5 5];
z2 = [0 2 2 0];

%    P2 Q2
x3 = [0 -2 -2 0];
y3 = [2 2 -1 -1];
z3 = [2 2 2 2];

%    P2 Q2
x4 = [0 -2 -2 0];
y4 = [2 2 2 2];
z4 = [2 2 3 3];

% Grid Limits
figure
xlim([-10 10])
ylim([-10 10])
zlim([-10 10])
```

```
% Axis Labels
xlabel('x')
ylabel('y')
zlabel('z')
hold on
patch(x1,y1,z1,'c')
patch(x2,y2,z2,'w')
patch(x3,y3,z3,'c')
patch(x4,y4,z4,'w')

T0_P = transl(0,5,0);
trplot(T0_P,'frame', 'P_b_e_g_i_n', 'color', b');
```





2.4 Use MATLAB to find the ZYZ Euler angles of the box homogeneous transformations (of 2.1) in its initial and final poses.

- Ra=rotz(z1)roty(y1)rotz(z2)
- Ra1=eul2tr(z1,y1,z2) Ra==Ra1 (edited)
- ZYZ = eul2tr(tr2eul(R1 * R2 * R3,'deg'),'deg');

- ZYX = rpy2tr(z,y,x,'deg');
- [theta_zyx,v_zyx] = tr2angvec(zyx);
- angvec2tr(theta_zyx,v_zyx)

EUL(:,:,2) =

1	0	0
0	1	0
0	0	1

EUL(:,:,3) =

1	0	0
0	1	0
0	0	1

EUL(:,:,4) =

1	0	0
0	1	0
0	0	1

2.5 Use MATLAB to find the screw axis vector (with respect to {W}) and the single screw rotation that takes the box from its initial position to its final position.

```
x1= 1.5708
y1 = -1.5708
z1 = 1.5708
x = 0;
y = 5;
z = 2;
zyx = rpy2tr(z,y,x,'deg');
[theta_zyx,v_zyx] = tr2angvec(zyx);
angvec2tr(theta_zyx,v_zyx)
```

ans =

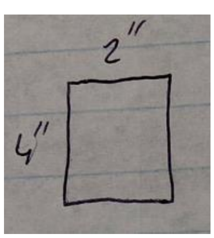
0.9962	0.0030	0.0871	0
0.0000	0.9994	-0.0349	0
-0.0872	0.0348	0.9956	0
0	0	0	1.0000

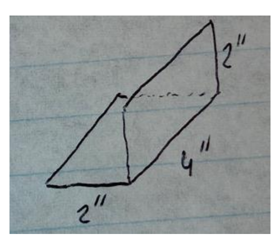
TSP =

-0.4481	-0.4959	-0.7439	0
0.4959	0.5544	-0.6683	0
0.7439	-0.6683	-0.0025	0
0	0	0	1.0000

Problem 3: Task Description

The task involves the welding of two 2”x4” planar rectangular pieces of metal (like the one shown below) to create a L-shaped bracket (also shown below):

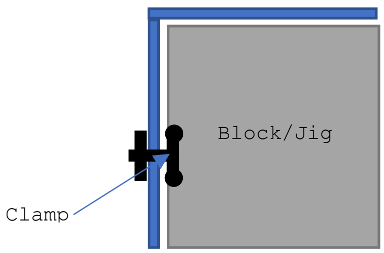




The thickness of each piece of metal is d = 2mm. There are hundreds of such pieces stacked inside a gravity feeder. One cycle of the task involves the making of one bracket.

3.1 Break one task cycle into a sequence of steps that a human operator must take to get the job done. Describe each step verbally (and very briefly).

- I would have a jig or a block that would hold one pleate on its side to the right.
- While placing the other plate on top of the block creating a seem in the L formation of the two plates.
- I would then have to manually pick up a welder and bead together the plates to create one bracket.



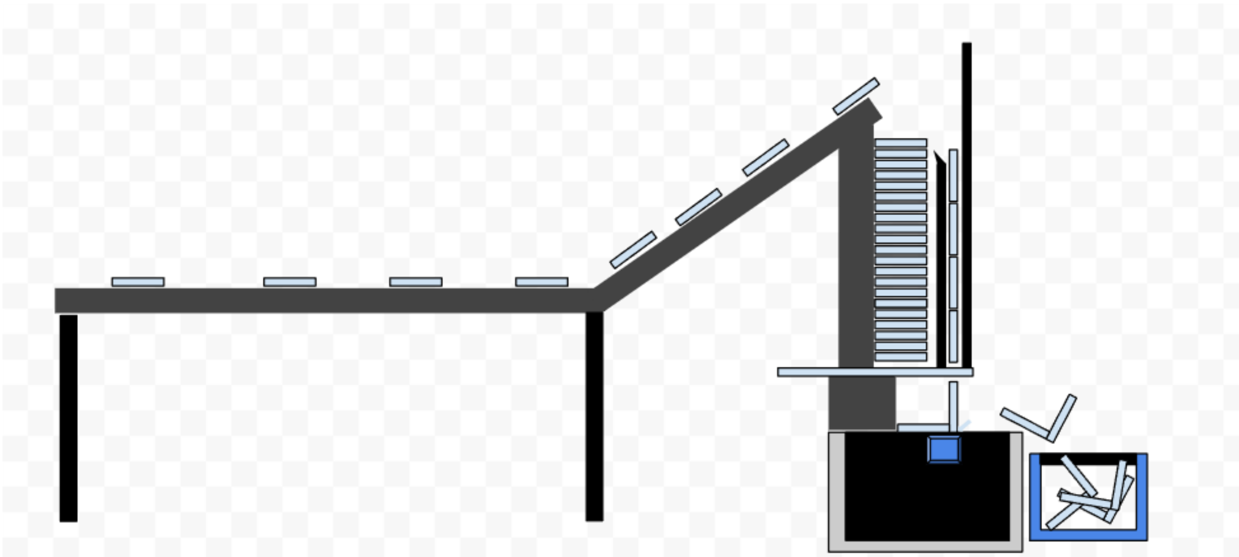
3.2 Need to replace the manual labor by automation. Revisit the steps of (3.1) and try to find a sequence of steps, that can be automated, to perform the job, starting from a part falling from the feeder into a conveyor belt, and ending with the outgoing bracket product placed on the same (or maybe another?) conveyor belt.

- I would have horizontal and vertical stack of plate being divided by two feeds, that were fed in from the west, split on a feeder before they were divided into horizontal and vertical feeders. N number of plates is infinite and continuous for this use-case.
- One feed that would rotate the plate -90 deg on the x axis dropping the plate down vertically in the Z direction through a feed on its side to a jig.
- The second on would remain in the original position being feed moving toward the Y direction in to catch the plate horizontally in a jig.

- Both plates would meet together at the exact L bracket formation together to create a seem.
- A pressure sensor would detect the plates are in correct position trigging the robot welder.
- As a robotic welder would travel in the 4” X direction to weld them together by the specified programed travel time and length.
- A mechanism and jig holding the bracket would release and Rotate 45 degrees in the positive X direction to release the bracket down a slide which would gather all the finished brackets.

Robot Motion

A) After plates where feed through a conveyer belt at a 60 degree angle to a separator where ½ plates separated in their respective vertical feeder bins and gravity moved plates in the respective positions.
B) An actuator would release the plates based on timing of the robots previous fish shed process
C) Move P1 plate in horizontal position
D) Move P2 plate in vertical position
E) Move P1 + P2 Welder on plates seem in Z direction
F) Eject the flange into storage.

G) 

3.3 What is the minimum number of robot manipulators that you think may be needed to perform the steps outlined in (3.2)?

I would only need 2 manipulators 3 if we are counting the feeding belt. My method would use gravity and the velocity of a forward feeder into a pre-made jig to position the plates together. With on robot to weld and one robot to eject it down a slide into a bin to collect the finished parts.

Try to get by with as few robots as possible. For each robot mention the geometry that the robot must have, and its end-effector's tool.

- This is based on human organization of standing North facing the machine from the side where the plate is feed in from the west and ejected to the east.
- Robot #1 the welder, would travel 4" in X direction , reset and then travel back in the negative X direction waiting the arrival of the new plate.

- Robot # 2 would simply rotate the jig in which the bracket was being help 45 degrees to dump the bracket down a slide into a collector.

Does each robot work at a different station of the automation line? Are there stations in which more than one robot may be needed? Briefly answer each question.

- The robots are centralized around the jig and plate feeders. There are no stations other than described needed. That is unless more robots are needed to attach the brackets to something else.

3.4 In addition to the robot (or robots) mentioned in (3.3), are there any other special-purpose devices or rigs that are needed?

- Yes, the main robot will need to be attached to a welding machine, which will have an automated welding feed of welding rod. It will have to have sensors and ability to alert and sequences stop of production when its resources need to be refilled.
- Robot #1 will also need a sensor to determine of the weld it makes is correct and have ability to catch errors in the weld
- In conjunction with robot # 2 sensors on robot #1 will trigger the release and ejection of the part and reset for the next bracket to be made.

Can each robot be programmed “by-doing”?

- specific actions based on the sensors input and timing sequences programmed to welled and eject each part.

Deadline for submission via Canvas Assignments: 9/23/2022 by 11:59 PM.