Robotic Applications Homework 1

EEL*5661

Professor: Dr. Zvi Roth

Jeffrey Young and Neelima Rajawat

Question 1:

1. The world coordinate frame {X_W, Y_W, Z_W} will be located in center of the southwest corner square A1(***see Figure 1***). Each square of the maze is 30’ in distance. For ease of calculations 1 square unit = 30’.

   The robot must be located north of the north maze wall with the X_R axis parallel to the north wall. The robot must take the quickest route possible to the south entrance while following the given constraints. Therefore, the robot will be located at the XY coordinate is (5,6) at its initial position (see ***Figure 1***). No rotation is required because the X_R axis is initially parallel to the north wall.

$$Starting\ Position = transl2(5{,}6) = \begin{matrix} 1 & 0 & 5 \\ 0 & 1 & 6 \\ 0 & 0 & 1 \end{matrix}$$

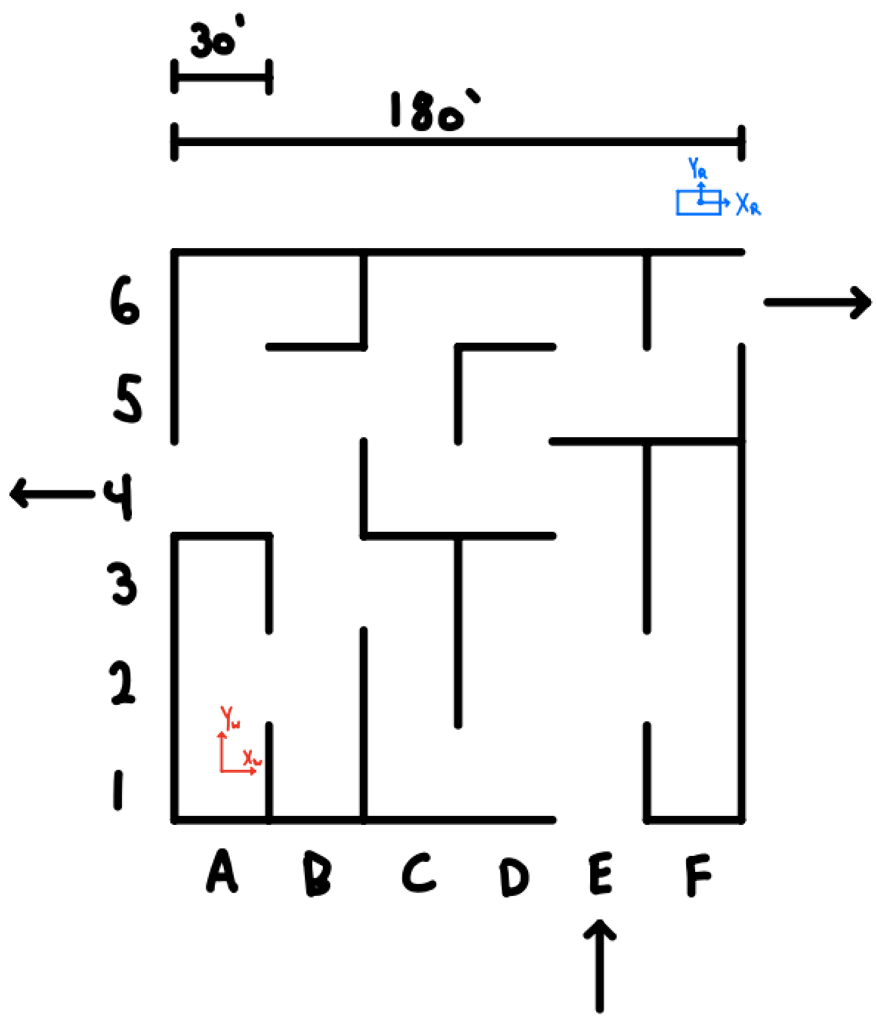

Figure 1- Starting Position

When the robot is exiting the east outlet the XY coordinate is (6,5) (see ***Figure 2***). No rotation will be required because the X_R axis is horizontal when exiting east outlet.

$$\mathit{Final\ Position\ East} = \mathit{transl2}(6,5) = \begin{matrix} 1 & 0 & 6 \\ 0 & 1 & 5 \\ 0 & 0 & 1 \end{matrix}$$

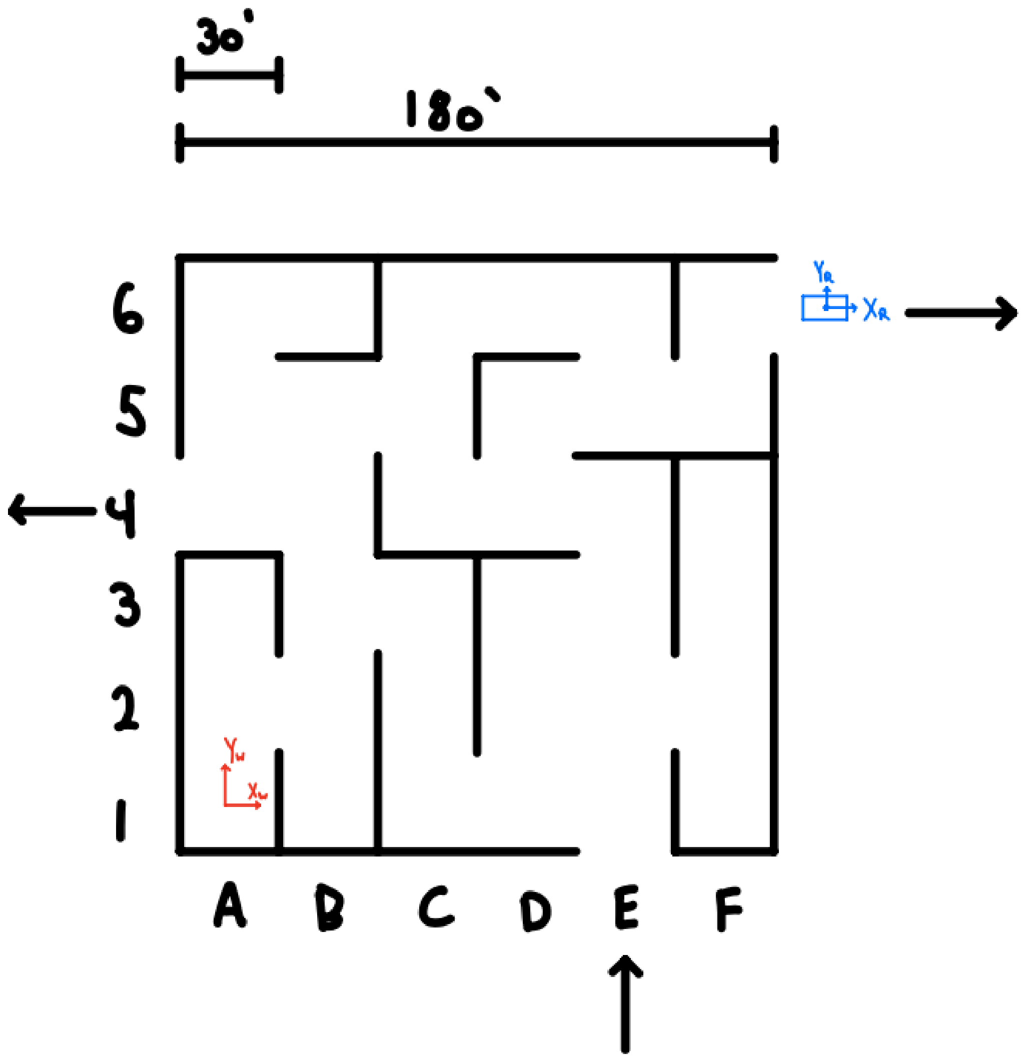

Figure 2- Final Position East

2. It makes sense to utilize the post multiplication of 2D homogeneous transformations to model the robot's movements because the robot must be rotated every single time that a new translation occurs and there are over 20 transformations in this problem. Using post-multiplication, we can complete transformations of the local frame with respect to the previous local frame. Whereas, with pre-multiplication we must complete these transformations with respect to the base coordinate frame. Completing a problem of this nature using pre-multiplication would be significantly more difficult and take much longer

to complete because we would have to complete every single transformation with respect to the base coordinate frame. Since the robot must be rotated every single time that a new translation occurs and there is such a large quantity of translations and rotations, we know that the use of post-multiplication will be significantly simpler for this problem.

3. ***Figure 3*** depicts the different rotations and translations that will occur:

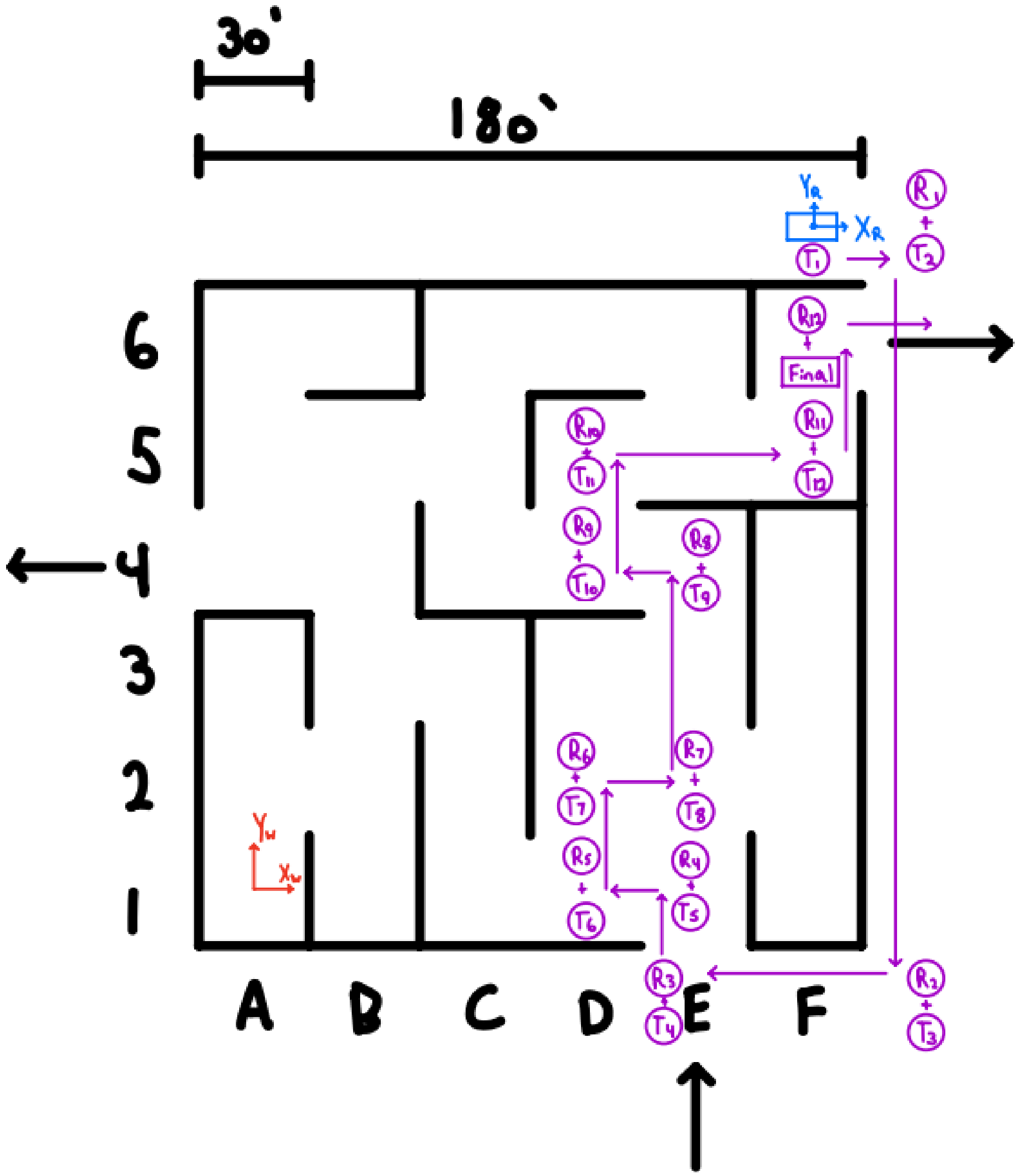

Figure 3 - Transformation Diagram

The sequence of the robot's steps is listed below:

Starting Position at XY coordinate (5,6): Translation (5,6) units
Translation1) Move 1 unit forward (+X_R): Translation (1,0)
Rotation1) Rotate 90 deg (pi/2)
Translation2) Move 7 units forward (+X_R): Translation (7,0)
Rotation2) Rotate 90 deg (pi/2)
Translation3) Move 2 units forward: Translation (2,0)
Rotation3) Rotate 90 deg (pi/2)
Translation4) Move 1 unit forward: Translation (1,0)
Rotation 4) Rotate -90 deg (-pi/2)
Translation5) Move 1 unit forward: Translation (1,0)
Rotation5) Rotate 90 deg (pi/2)
Translation6) Move 1 unit forward: Translation (1,0)
Pick Up package in D2
Rotation6) Rotate 90 deg (pi/2)
Translation7) Move 1 unit forward: Translation (1,0)
Rotation7) Rotate -90 deg (-pi/2)
Translation8) Move 2 units forward: Translation (2,0)
Rotation 8) Rotate -90 deg (-pi/2)
Translation9) Move 1 unit forward: Translation (1,0)
Rotation9) Rotate 90 deg (pi/2)
Translation10) Move 1 unit forward: Translation (1,0)
Rotation10) Rotate 90 deg (pi/2)
Translation11) Move 2 units forward: Translation (2,0)
Rotation11) Rotate -90 deg (-pi/2)
Translation12) Move 1 unit forward: Translation (1,0)
Rotation12) Rotate 90 deg (pi/2)
Translation13) Move 1 unit forward: Translation (1,0)

These 2D homogeneous transformations of the robot were analyzed using MATLAB code:

```
Forward1 = transl2(1,0);
Forward2 = transl2(2,0);
Forward7 = transl2(7,0);
Right = trot2(-90,'deg');
Left = trot2(90,'deg');
start = transl2(5,6), title Start;

T1 = start*Forward1, title T1;
R1 = T1*Right, title R1;
T2 = R1*Forward7, title T2;
R2 = T2*Right, title R2;
T3 = R2*Forward2, title T3;
R3 = T3*Right, title R3;
T4 = R3*Forward1, title T4;
R4 = T4*Left, title R4;
T5 = R4*Forward1, title T5;
R5 = T5*Right, title R5;
T6 =R5*Forward1, title T6;
R6 = T6*Right, title R6;
T7 = R6*Forward1, title T7;
R7 = T7*Left, title R7;
T8 =R7*Forward2, title T8;
R8 =T8*Left, title R8;
T9 =R8*Forward1, title T9;
R9 = T9*Right, title R9;
T10 = R9*Forward1, title T10;
R10 = T10*Right, title R10;
T11 = R10*Forward2, title T11;
R11 = T11*Left, title R11;
T12 = R11*Forward1, title T12;
R12 = T12*Right, title R12;
Final = R12*Forward1, title Final;
```

Figure 4- MATLAB Code for Homogeneous Transformations

The corresponding matrices for the homogenous transformations were calculated in MATLAB:

```
start =

     1     0     5
     0     1     6
     0     0     1


T1 =

     1     0     6
     0     1     6
     0     0     1


R1 =

    0.0000    1.0000    6.0000
   -1.0000    0.0000    6.0000
         0         0    1.0000


T2 =

    0.0000    1.0000    6.0000
   -1.0000    0.0000   -1.0000
         0         0    1.0000


R2 =

   -1.0000    0.0000    6.0000
   -0.0000   -1.0000   -1.0000
         0         0    1.0000


T3 =

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


R3 =

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


T4 =

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


R4 =

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


T5 =

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


R5 =

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


T6 =

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

```
R6 =

    1.0000   -0.0000    3.0000
    0.0000    1.0000    1.0000
         0         0    1.0000

R8 =

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

R10 =

    1.0000   -0.0000    3.0000
    0.0000    1.0000    4.0000
         0         0    1.0000

T7 =

    1.0000   -0.0000    4.0000
    0.0000    1.0000    1.0000
         0         0    1.0000

T9 =

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

T11 =

    1.0000   -0.0000    5.0000
    0.0000    1.0000    4.0000
         0         0    1.0000

R7 =

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

R9 =

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

R11 =

   -0.0000   -1.0000    5.0000
    1.0000   -0.0000    4.0000
         0         0    1.0000

T8 =

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

T10 =

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

T12 =

   -0.0000   -1.0000    5.0000
    1.0000   -0.0000    5.0000
         0         0    1.0000

R12 =

    1.0000   -0.0000    5.0000
    0.0000    1.0000    5.0000
         0         0    1.0000

Final =

    1.0000   -0.0000    6.0000
    0.0000    1.0000    5.0000
         0         0    1.0000
```

Figure 5 - Transformation Matrices

The final matrix in ***Figure 5*** displays the same values as the final matrix that was determined in 1.1. The final matrix when the robot exits through the east will be:

$$\begin{matrix} 1 & 0 & 6 \\ 0 & 1 & 5 \\ 0 & 0 & 1 \end{matrix}$$

Since the final matrix calculated in MATLAB is the same as the final matrix from question 1.1 we can confirm that we have properly determined the steps of the 2D homogeneous transformations of the robot when entering from the south, securing the package and exiting from the east.

4. ***Figure 6*** depicts the different rotations and translations that will occur. We are assuming that the alarm is sounded after the robot secures the package in D2. The robot then proceeds

to square F1 because it is the closest safe corner cell. The robot remains in square F1 until the alarm turns off and then proceeds to the east exit. For the simplicity and clarity of the figure, the translation and rotation notations were not included in the diagram:

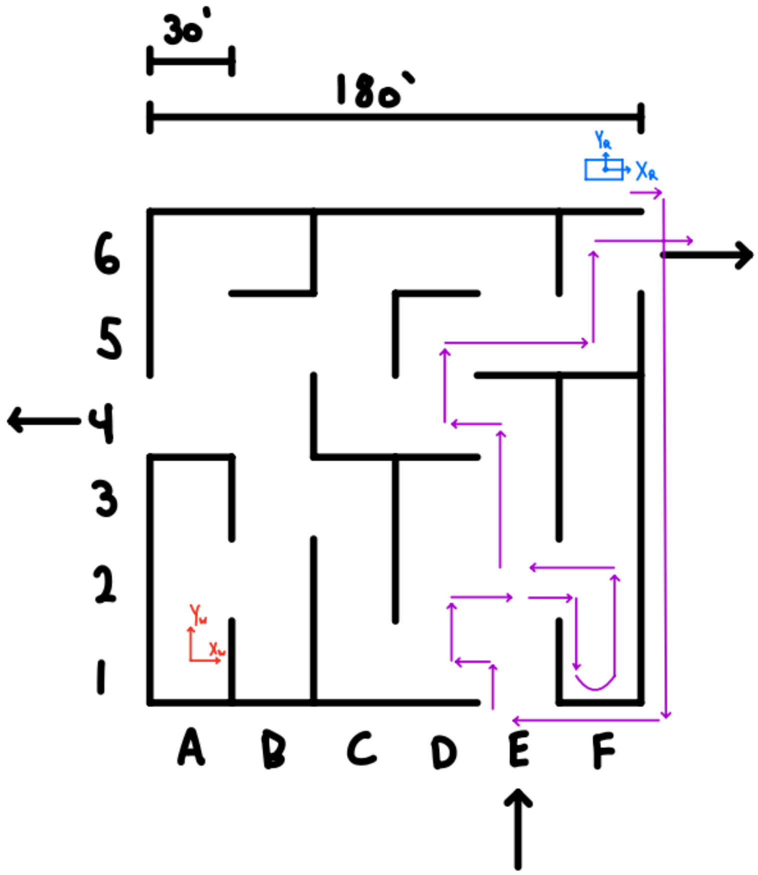

Figure 6- Transformations When Alarm is Activated

The sequence of the robot's steps is listed below. We are assuming that the alarm is sounded after the robot secures the package in D2:

Starting Position at XY coordinate (5,6): Translation (5,6) units
Translation1) Move 1 unit forward (+X_R): Translation (1,0)
Rotation1) Rotate 90 deg (pi/2)
Translation2) Move 7 units forward (+X_R): Translation (7,0)
Rotation2) Rotate 90 deg (pi/2)
Translation3) Move 2 units forward: Translation (2,0)
Rotation3) Rotate 90 deg (pi/2)
Translation4) Move 1 unit forward: Translation (1,0)
Rotation 4) Rotate -90 deg (-pi/2)
Translation5) Move 1 unit forward: Translation (1,0)
Rotation5) Rotate 90 deg (pi/2)
Translation6) Move 1 unit forward: Translation (1,0)
Pick Up package in D2

Alarm is sounded so robot moves to F1
Rotation6) Rotate 90 deg (pi/2)
Translation7) Move 2 units forward. Translation (2,0)
Rotation7) Rotate 90 deg (pi/2)
Translation8) Move 1 unit forward. Translation (1,0)
Robot is in square F1. When alarm stops robot proceeds to east exit
Rotation8) Rotate -90 deg (-pi/2)
Rotation9) Rotate -90 deg (-pi/2)
Translation9) Move 1 unit forward. Translation (1,0)
Rotation10) Rotate -90 deg (-pi/2)
Translation10) Move 1 unit forward. Translation (1,0)
Rotation11) Rotate 90 deg (pi/2)
Translation11) Move 2 units forward: Translation (2,0)
Rotation 12) Rotate -90 deg (-pi/2)
Translation12) Move 1 unit forward: Translation (1,0)
Rotation13) Rotate 90 deg (pi/2)
Translation13) Move 1 unit forward: Translation (1,0)
Rotation14) Rotate 90 deg (pi/2)
Translation14) Move 2 units forward: Translation (2,0)
Rotation15) Rotate -90 deg (-pi/2)
Translation15) Move 1 unit forward: Translation (1,0)
Rotation16) Rotate 90 deg (pi/2)
Translation16) Move 1 unit forward: Translation (1,0)

These 2D homogeneous transformations of the robot were analyzed using MATLAB code:

```
Forward1 = transl2(1,0);
Forward2 = transl2(2,0);
Forward7 = transl2(7,0);
Right = trot2(-90,'deg');
Left = trot2(90,'deg');
start = transl2(5,6), title Start;

T1 = start*Forward1, title T1;
R1 = T1*Right, title R1;
T2 = R1*Forward7, title T2;
R2 = T2*Right, title R2;
T3 = R2*Forward2, title T3;
R3 = T3*Right, title R3;
T4 = R3*Forward1, title T4;
R4 = T4*Left, title R4;
T5 = R4*Forward1, title T5;
R5 = T5*Right, title R5;
T6 =R5*Forward1, title T6;
R6 = T6*Right, title R6;
T7 = R6*Forward2, title T7;
R7 = T7*Right, title R7;
T8 =R7*Forward1, title T8;
R8 =T8*Left, title R8;
R9 =R8*Left, title R9;
T9 =R9*Forward1, title T9;
R10 = T9*Left, title R10;
T10 = R10*Forward1, title T10;
R11 = T10*Right, title R11;
T11 = R11*Forward2, title T11;
R12 = T11*Left, title R12;
T12 = R12*Forward1, title T12;
R13 = T12*Right, title R13;
T13 = R13*Forward1, title T13;
R14 = T13*Right, title R14;
T14 = R14*Forward2, title T14;
R15 = T14*Left, title R15;
T15 = R15*Forward1, title T15;
R16 = T15*Right, title R16;
Final = R16*Forward1, title Final;
```

Figure 7- MATLAB When Alarm Sounds in D2

The corresponding matrices for the homogenous transformations were calculated in MATLAB:

```
start =

     1     0     5
     0     1     6
     0     0     1


T1 =

     1     0     6
     0     1     6
     0     0     1


R1 =

    0.0000    1.0000    6.0000
   -1.0000    0.0000    6.0000
         0         0    1.0000


T2 =

    0.0000    1.0000    6.0000
   -1.0000    0.0000   -1.0000
         0         0    1.0000


R2 =

   -1.0000    0.0000    6.0000
   -0.0000   -1.0000   -1.0000
         0         0    1.0000
```

```
T3 =

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


R3 =

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


T4 =

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


R4 =

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


T5 =

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

```
R5 =

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


T6 =

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


R6 =

    1.0000   -0.0000    3.0000
    0.0000    1.0000    1.0000
         0         0    1.0000


T7 =

    1.0000   -0.0000    5.0000
    0.0000    1.0000    1.0000
         0         0    1.0000


R7 =

    0.0000    1.0000    5.0000
   -1.0000    0.0000    1.0000
         0         0    1.0000
```

```
T8 =

    0.0000    1.0000    5.0000
   -1.0000    0.0000    0.0000
         0         0    1.0000


R8 =

    1.0000   -0.0000    5.0000
    0.0000    1.0000    0.0000
         0         0    1.0000


R9 =

   -0.0000   -1.0000    5.0000
    1.0000   -0.0000    0.0000
         0         0    1.0000


T9 =

   -0.0000   -1.0000    5.0000
    1.0000   -0.0000    1.0000
         0         0    1.0000


R10 =

   -1.0000    0.0000    5.0000
   -0.0000   -1.0000    1.0000
         0         0    1.0000
```

```
T10 =

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


R11 =

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


T11 =

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


R12 =

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


T12 =

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

```
R13 =

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


T13 =

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


R14 =

    1.0000   -0.0000    3.0000
    0.0000    1.0000    4.0000
         0         0    1.0000


T14 =

    1.0000   -0.0000    5.0000
    0.0000    1.0000    4.0000
         0         0    1.0000


R15 =

   -0.0000   -1.0000    5.0000
    1.0000   -0.0000    4.0000
         0         0    1.0000
```

```
T15 =

   -0.0000   -1.0000    5.0000
    1.0000   -0.0000    5.0000
         0         0    1.0000


R16 =

    1.0000   -0.0000    5.0000
    0.0000    1.0000    5.0000
         0         0    1.0000


Final =

    1.0000   -0.0000    6.0000
    0.0000    1.0000    5.0000
         0         0    1.0000
```

Figure 8 - Transformation Matrices

The final matrix in ***Figure 8*** displays the same values as the final matrix that was determined in 1.1 and 1.3. The final matrix when the robot exits through the east will be:

$$\begin{matrix} 1 & 0 & 6 \\ 0 & 1 & 5 \\ 0 & 0 & 1 \end{matrix}$$

Since the final matrix calculated in MATLAB is the same as the final matrix from question 1.1 and 1.3 we can confirm that we have properly determined the steps of the 2D homogeneous transformations of the robot when entering from the south, securing the package, proceeding to F1 when the alarm sounds and exiting from the east after the alarm stops.

5. The robot will be starting north of the north wall in the same position as in question 1.1 through 1.4. It is not feasible for us to create an autonomous strategy to obtain the package in cell D2, thus we will hardcode the robots movements to this cell to ensure that the package is collected. Once the package is obtained from cell D2, an autonomous strategy will be utilized to navigate the robot to the mazes nearest exit. Furthermore, it is assumed that the object sensor is pointing in the positive X_R direction. The steps outlined below will be the autonomous logic of this robot to exit the maze from cell D2:

Step 1:
Robot rotates to the left (90 degrees)

Step 2:
Robot senses 1 unit forward (30')

Step 3:
If an obstacle has been sensed:
1) Robot rotates to the right (90 degrees)
2) Robot senses 1 unit forward (30’)
If there is still an obstacle being sensed:
1) Return to start of step 3
Else move onto step 4

Step 4:
Robot moves forward 1 unit ($+X_R$)
If no obstacle is sensed:
Return to start for step 4
Else move to step 5

Step 5:
Return to step 1

Figure 9 represents the starting point of the robot and the path that the robot will take using the autonomous logic of the robot navigating the maze:

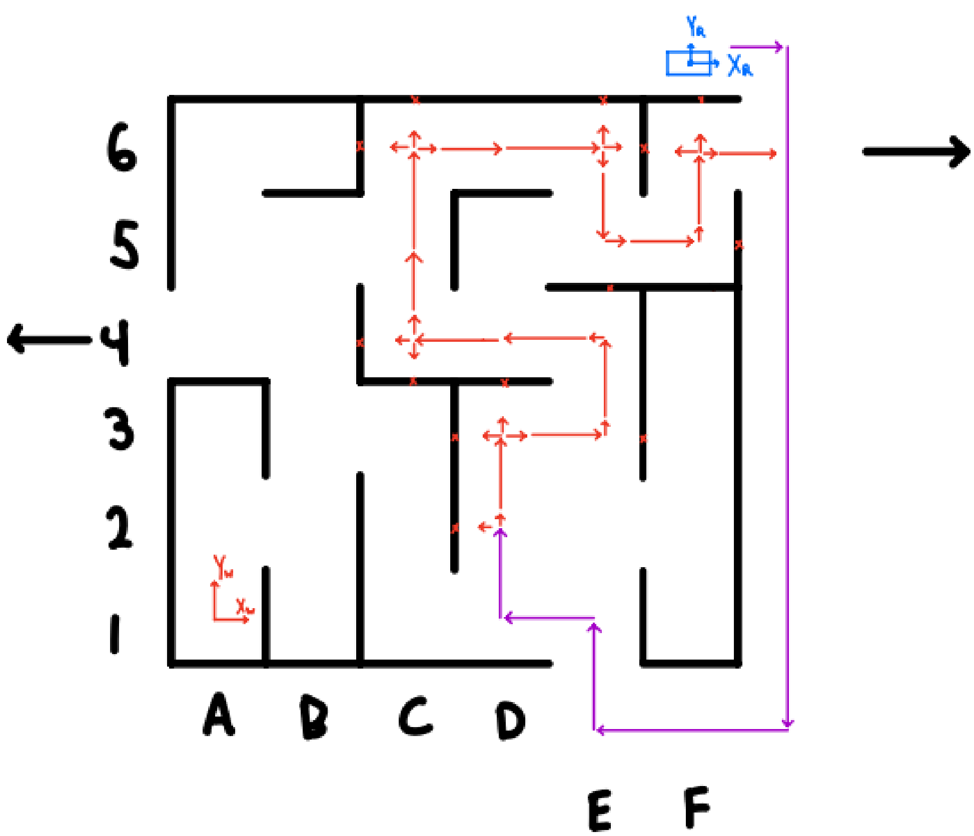

Figure 9 - Autonomous Maze Navigation

Figure 10 represents the MATLAB code required for this problem:

```
F1 = transl2(1,0);
F2 = transl2(2,0);
F7 = transl2(7,0);
R = trot2(-90,'deg');
L= trot2(90,'deg');
start = transl2(5,6), title Start;

%Hardcoding to get robot to secure package in D2
T1 = start*F1, title T1;
R1 = T1*R, title R1;
T2 = R1*F7, title T2;
R2 = T2*R, title R2;
T3 = R2*F2, title T3;
R3 = T3*R, title R3;
T4 = R3*F1, title T4;
R4 = T4*L, title R4;
T5 = R4*F1, title T5;
R5 = T5*R, title R5;
T6 =R5*F1, title T6;


 %Autonomous Code to get robot from D2 to exit
 R6 = T6*L, title R6
 R7 = R6*R, title R7
 T7 = R7*F1, title T7
 R8 = T7*L, title R8
 R9 = R8*R, title R9
 R10 = R9*R, title R10
 T8 = R10*F1, title T8
 R11 = T8*L, title R11
 T9 = R11*F1, title T9
 R12 = T9*L, title R12
 T10 = R12*F1, title T10
 T11 = T10*F1, title T11
 R13 = T11*L, title R13
 R14 = R13*R, title R14
 R15 = R14*R, title R15
 T12 = R15*F1, title T12
 T13 = T12*F1, title T13
 R16 = T13*L, title R16
 R17 = R16*R, title R17
 R18 = R17*R, title R18
 T14 = R18*F1, title T14
 T15 = T14*F1, title T15
 R19 = T15*L, title R19
 R20 = R19*R, title R20
 R21 = R20*R, title R21
 T16 = R21*F1, title T16
 R22 = T16*L, title R22
 T17 = R22*F1, title T17
 R23 = T17*L, title R23
 T18 = R23*F1, title T18
 R24 = T18*L, title R24
 R25 = R24*R, title R25
 R26 = R25*R, title R26
 Final = R26*F1, title Final
```

Figure 10 - Autonomous MATLAB Code

Figure 11 represents the resulting matrices produced by the MATLAB code:

```
start =

     1     0     5
     0     1     6
     0     0     1

T1 =

     1     0     6
     0     1     6
     0     0     1

R1 =

    0.0000    1.0000    6.0000
   -1.0000    0.0000    6.0000
         0         0    1.0000

T2 =

    0.0000    1.0000    6.0000
   -1.0000    0.0000   -1.0000
         0         0    1.0000

R2 =

   -1.0000    0.0000    6.0000
   -0.0000   -1.0000   -1.0000
         0         0    1.0000

T3 =

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

R3 =

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

T4 =

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

R4 =

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

T5 =

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

R5 =

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

T6 =

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

R6 =

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

R7 =

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

T7 =

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

R8 =

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

R9 =

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

R10 =

    1.0000   -0.0000    3.0000
    0.0000    1.0000    2.0000
         0         0    1.0000

T8 =

    1.0000   -0.0000    4.0000
    0.0000    1.0000    2.0000
         0         0    1.0000

R11 =

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

T9 =

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

R12 =

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

T10 =

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

T11 =

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

```
R13 =

    0.0000    1.0000    2.0000
   -1.0000    0.0000    3.0000
         0         0    1.0000

T13 =

   -0.0000   -1.0000    2.0000
    1.0000   -0.0000    5.0000
         0         0    1.0000

T14 =

    1.0000   -0.0000    3.0000
    0.0000    1.0000    5.0000
         0         0    1.0000

R14 =

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

R16 =

   -1.0000    0.0000    2.0000
   -0.0000   -1.0000    5.0000
         0         0    1.0000

T15 =

    1.0000   -0.0000    4.0000
    0.0000    1.0000    5.0000
         0         0    1.0000

R15 =

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

R17 =

   -0.0000   -1.0000    2.0000
    1.0000   -0.0000    5.0000
         0         0    1.0000

R19 =

   -0.0000   -1.0000    4.0000
    1.0000   -0.0000    5.0000
         0         0    1.0000

T12 =

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

R18 =

    1.0000   -0.0000    2.0000
    0.0000    1.0000    5.0000
         0         0    1.0000

R20 =

    1.0000   -0.0000    4.0000
    0.0000    1.0000    5.0000
         0         0    1.0000

R21 =

    0.0000    1.0000    4.0000
   -1.0000    0.0000    5.0000
         0         0    1.0000

R23 =

   -0.0000   -1.0000    5.0000
    1.0000   -0.0000    4.0000
         0         0    1.0000

T16 =

    0.0000    1.0000    4.0000
   -1.0000    0.0000    4.0000
         0         0    1.0000

T18 =

   -0.0000   -1.0000    5.0000
    1.0000   -0.0000    5.0000
         0         0    1.0000

R22 =

    1.0000   -0.0000    4.0000
    0.0000    1.0000    4.0000
         0         0    1.0000

R24 =

   -1.0000    0.0000    5.0000
   -0.0000   -1.0000    5.0000
         0         0    1.0000

R26 =

    1.0000   -0.0000    5.0000
    0.0000    1.0000    5.0000
         0         0    1.0000

T17 =

    1.0000   -0.0000    5.0000
    0.0000    1.0000    4.0000
         0         0    1.0000

R25 =

   -0.0000   -1.0000    5.0000
    1.0000   -0.0000    5.0000
         0         0    1.0000

Final =

    1.0000   -0.0000    6.0000
    0.0000    1.0000    5.0000
         0         0    1.0000
```

Figure 11 - Resultant Matrices of Autonomous Code

Since the final matrix displayed in ***Figure 11*** is the same as the final matrix from 1.1 and 1.3, thus we can confirm that our autonomous code is functioning properly.

Question 2:

1. ***Figure 12*** displays the world coordinate frame, the boxes coordinate frame, the balls coordinate frame and the initial and final position of the ball.

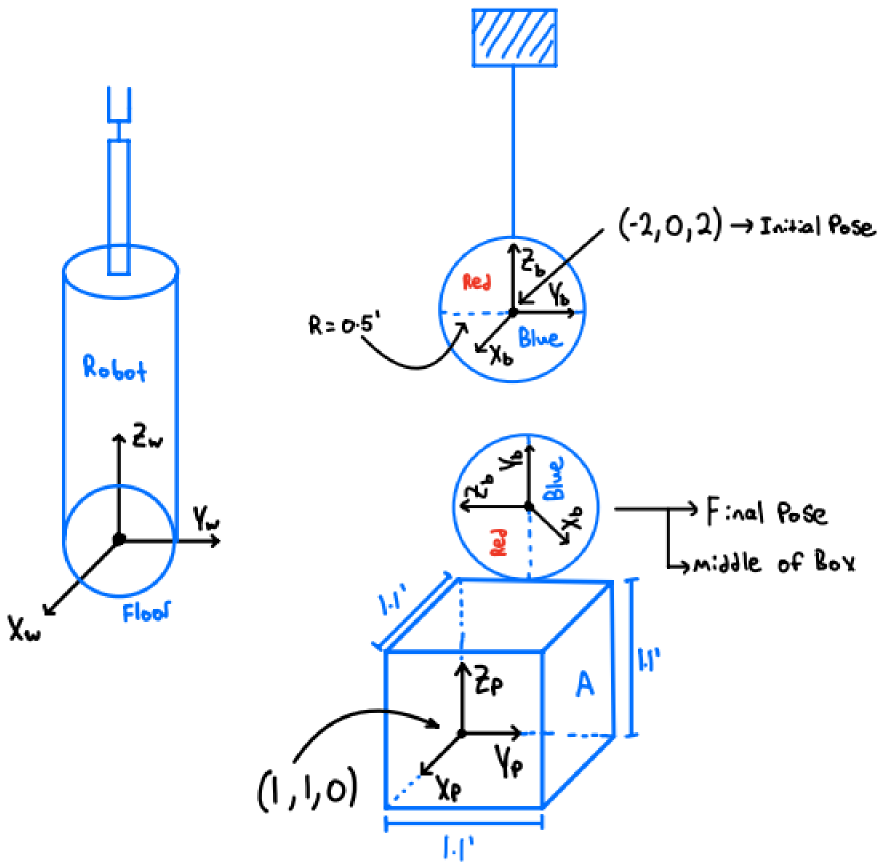

Figure 12- Initial and Final Ball Position

The 4X4 homogeneous transformation of the first pose of the ball is at coordinate location (-2,0,2). No rotations are required for the initial pose. This homogenous transformation is depicted below:

$$Initial\ Pose = transl(-2{,}0{,}2) = \begin{matrix} 1 & 0 & 0 & -2 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 2 \\ 0 & 0 & 0 & 1 \end{matrix}$$

The 4X4 homogeneous transformation of the final pose of the ball is when the ball is placed directly on top of the box. We know the height of the box is 1.1’ and the radius of the ball is 0.5’, thus the z coordinate will be 1.6. We know that the coordinate frame of the box is located at (1,1,0) relative to the world coordinate frame and the ball is placed on the center of this box, thus the X and Y coordinates will both be 1.5’. Furthermore, the ball must be rotated 90 degrees relative to the X axis to ensure that the blue half of the balls face is perpendicular to the boxes wall denoted as A. The homogeneous transformation is depicted below:

$$Final\ Pose = transl(1.5, 1.5, 1.6) * trotx(90) = \begin{matrix} 1 & 0 & 0 & 1.5 \\ 0 & 0 & -1 & 1.5 \\ 0 & 1 & 0 & 1.6 \\ 0 & 0 & 0 & 1 \end{matrix}$$

It is assumed that the transformation of the initial pose to the final pose was completed with respect to the initial pose. We know that the X, Y and Z coordinates for this transformation would be determined by subtracting the final pose coordinates from the initial pose coordinates. Thus, the X coordinate is 3.5, the Y coordinate is 1.5 and the Z coordinate is -0.4. Furthermore, the ball must be rotated 90 degrees relative to the X axis to ensure that the blue half of the balls face is perpendicular to the boxes wall denoted as A. The transformation of the final pose relative to the initial pose is depicted below:

$$Initial\ to\ Final = transl(3.5,1.5,-0.4) * trotx(90) = \begin{matrix} 1 & 0 & 0 & 3.5 \\ 0 & 0 & -1 & 1.5 \\ 0 & 1 & 0 & -0.4 \\ 0 & 0 & 0 & 1 \end{matrix}$$

2. The steps listed below will qualitatively explain how a human operator would guide this robot to complete the task:

<u>Step 1:</u>
To begin, the gripper will be fully open, and the robot gripper will move towards the hanging ball (negative X direction) until the gripping mechanism is in line with the ball.

<u>Step2:</u>
When the gripping mechanism and the ball are aligned the gripper will fully close (ball is gripped).

<u>Step 3:</u>
The gripping mechanism is fully closed and the gripper must pull the ball downwards slightly for the thin wire to snap (negative Z direction).

<u>Step 4:</u>
The robot's gripper is fully closed and now must move towards the pedestal where the ball will be placed.

Step 5:

The gripper is fully closed on the ball and the ball will be rotated 90 degrees relative to the X axis. This will ensure that the blue half of the balls face is perpendicular to the pedestal's wall denoted as A.

Step 6:

Once the ball is aligned with the pedestal in the correct orientation, the gripper will release the ball. At this point the process is complete.

3. Since the height of the robot and gripping mechanism is not specified within the question, we will make an assumption. For ease of calculations, it is assumed the height of the robot and gripping mechanism is approximately 2' which is the same height as the center of the ball (Z axis). This will allow for the robot gripper to move directly to the ball and grip it easily. The robotic grippers coordinate frame can be seen in ***Figure 13***.

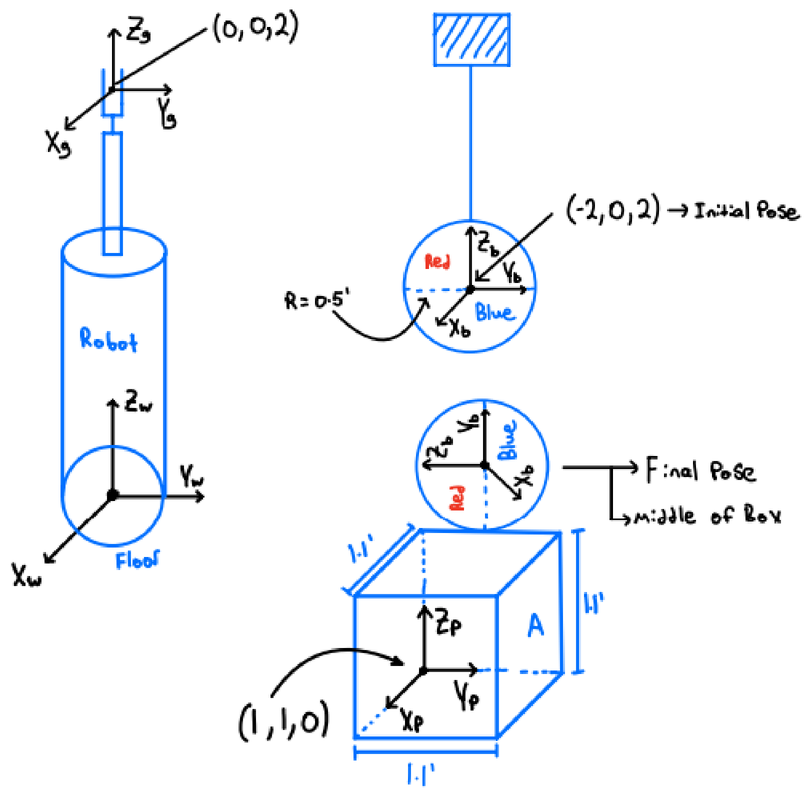

Figure 13 - Robot Height Assumption

Using post multiplication, we are capable of completing transformations of the local frame with respect to the previous local frame. The following steps explain the simple displacements that will occur when the gripper moves to the ball, grips it, rotates it, moves the ball to the surface of the pedestal and places it on top of the pedestal:

Starting position of gripper XYZ coordinate (0,0,2): Translate (0,0,2) units.

Translation1) Move -2 in X direction: Translation (-2,0,0)

Gripper position: (-2, 0, 2)

Gripper now aligned with ball. Gripper closes around Ball

Now must complete a slight downward translation to snap thin wire

Translation2) Move -0.2 in Z direction: Translation (0,0,-0.2)

Gripper Position: (-2, 0, 1.8)

Since the final position of the ball is at coordinates (1.5, 1.5, 1.6) and the coordinate frame of the ball of the and the gripper are aligned we must translate the gripper from (-2, 0, 1.8) to the coordinate (1.5, 1.5, 1.6)

Translation3) Move 3.5 in X direction, move 1.5 in Y direction and move -0.2 in Z direction: Translate (3.5, 1.5, -0.2)

Now must rotate for the gripper to ensure proper ball alignment once released

Rotation1) Rotate 90 degrees about X axis. trotx(90)

Final Gripper Position: (1.5, 1.5, 1.6) and rotated 90 degrees about X axis

Gripper opens fully and releases the ball. Process is complete

The final position of the gripper is the same as the final position of the ball from question 2.1, thus we can confirm our solution. The following MATLAB code and resulting matrices in ***Figure 14*** and ***Figure 15*** confirm this solution:

```
Initial = transl(0,0,2)
Final = Initial*transl(-2,0,0)*transl(0,0,-0.2)*transl(3.5,1.5,-0.2)*trotx(90)
```

Figure 14 - Post-multiplication MATLAB

```
Initial =

     1     0     0     0
     0     1     0     0
     0     0     1     2
     0     0     0     1


Final =

    1.0000         0         0    1.5000
         0         0   -1.0000    1.5000
         0    1.0000         0    1.6000
         0         0         0    1.0000
```

Figure 15 - Post-multiplication Resultant Matrices

4. Pre multiplications are completed with respect to the world coordinate frame. So, we will pre multiplying from right to the left:

Step 1:

We know the initial position of the gripper will be the matrix that is the farthest right:

$$Initial\ Pose = transl(0,0,2) = \begin{matrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 2 \\ 0 & 0 & 0 & 1 \end{matrix}$$

First, we will rotate the gripper 90 degrees about world frames X-axis. When the gripper is rotated the coordinates will change from (0,0,2) to (0,-2,0) and the gripper orientation has been rotated 90 degrees. This can be seen in ***Figure 16***:

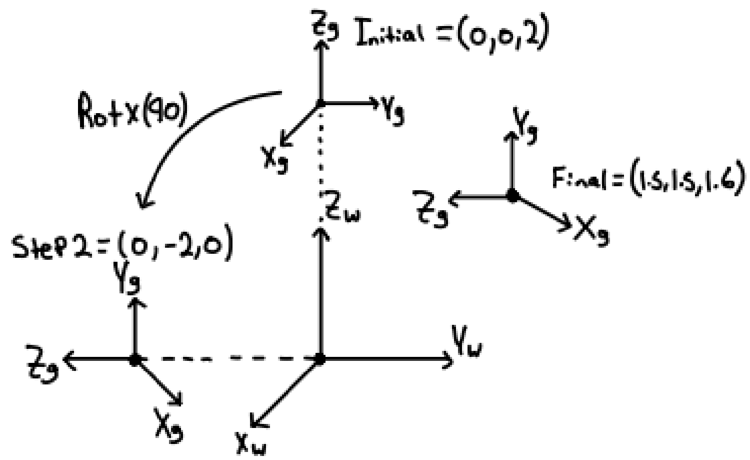

Figure 16 - Gripper Rotation

Step 2:

Now the gripper is translated from (0,-2,0) to the final position, (1.5,1.5,1.6). The translation will be:

X = 1.5-0= 1.5

Y = 1.5-(-2) = 3.5

Z = 1.6 – 0 = 1.6

Therefore, the total pre-multiplication expression is:

Final = transl(1.5, 3.5, 1.6)*trotx(90)*Initial

Figure 17 and ***Figure 18*** display the MATLAB and the resulting matrices of this pre-multiplication. Since the final matrix is the same as the final matrix from 2.3 we know that our solution is correct.

```
Initial = transl(0,0,2)
Final = transl(1.5,3.5,1.6)*trotx(90)*Initial
```

Figure 17 - Pre-multiplication MATLAB

```
Initial =

     1     0     0     0
     0     1     0     0
     0     0     1     2
     0     0     0     1


Final =

    1.0000         0         0    1.5000
         0         0   -1.0000    1.5000
         0    1.0000         0    1.6000
         0         0         0    1.0000
```

Figure 18 - Pre-multiplication Matrices

5. The MATLAB function tr2eul was utilized to determine the ZYZ Euler angles of the initial and final pose. ***Figure 19*** displays the MATLAB code for the initial pose and ***Figure 20*** displays the Euler angles for the initial pose:

```
Initial = transl(-2,0,2)
EulersInitial = tr2eul(Initial)
```

Figure 19- Initial Euler MATLAB

```
Initial =

     1     0     0    -2
     0     1     0     0
     0     0     1     2
     0     0     0     1


EulersInitial =

     0     0     0
```

Figure 20 - Euler Angles Initial

Therefore, the ZYZ Euler angles for the initial pose are (0,0,0). We know this to be true because no rotations were completed at the initial pose:

Figure 21 displays the MATLAB code for the initial pose and ***Figure 22*** displays the ZYZ Euler angles for the initial pose:

```
Final= transl(1.5, 1.5, 1.6)*trotx(90)
EulersFinal = tr2eul(Final)
```

Figure 21- Final Euler MATLAB

```
Final =

    1.0000         0         0    1.5000
         0         0   -1.0000    1.5000
         0    1.0000         0    1.6000
         0         0         0    1.0000


EulersFinal =

   -1.5708    1.5708    1.5708
```

Figure 22 - Final Euler Angles

Therefore, the ZYZ Euler angles for the final pose are (-1.5708, 1.5708, 1.5708). In degrees this equivalent to (-90, 90, 90).

6. We will apply the tr2angvec MATLAB function to the Final matrix (determined in 1.1) to find the Chasle's theorem screw axis vector (v) and the single screw rotation of the ball from its initial position to final position. ***Figure 23*** displays the MATLAB code for this problem and ***Figure 24*** displays the screw axis vector (v) and the single screw rotation (rotation):

```
Final= transl(1.5, 1.5,1.6)*trotx(90)
[rotation, v] = tr2angvec(Final)
```

Figure 23 – Chasle's MATLAB Code

```
Final =

    1.0000         0         0    1.5000
         0         0   -1.0000    1.5000
         0    1.0000         0    1.6000
         0         0         0    1.0000


rotation =

    1.5708


v =

     1     0     0
```

Figure 24 - Screw Axis Vector and Rotation

Therefore, the screw axis vector is (1,0,0) and the single screw rotation will be 90 degrees (1.5708 rad).