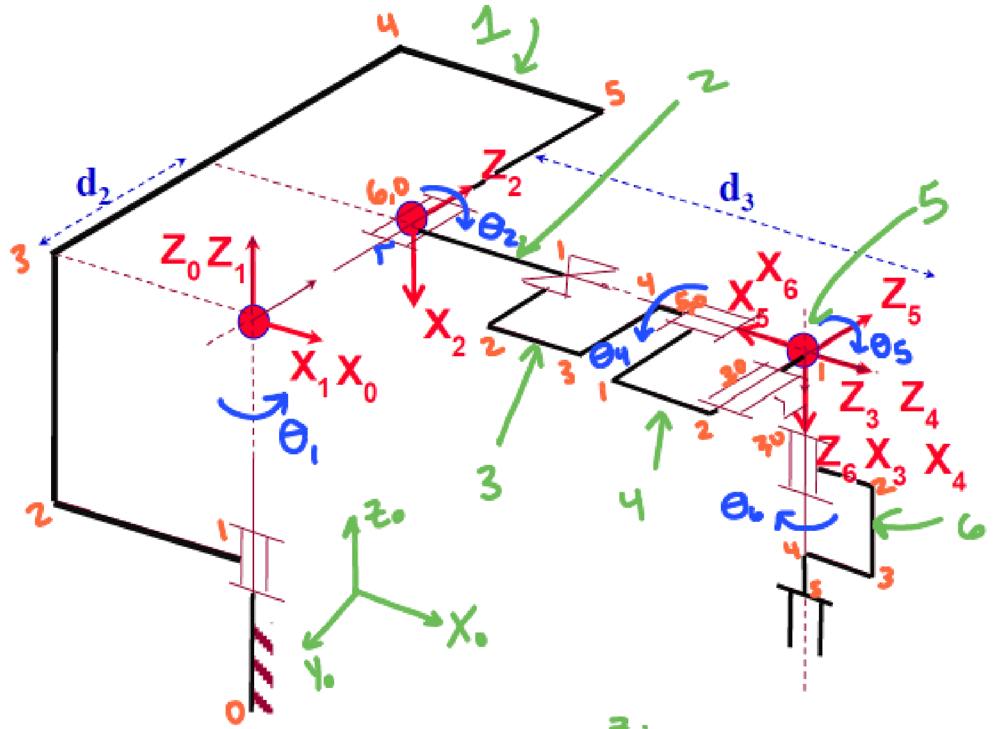
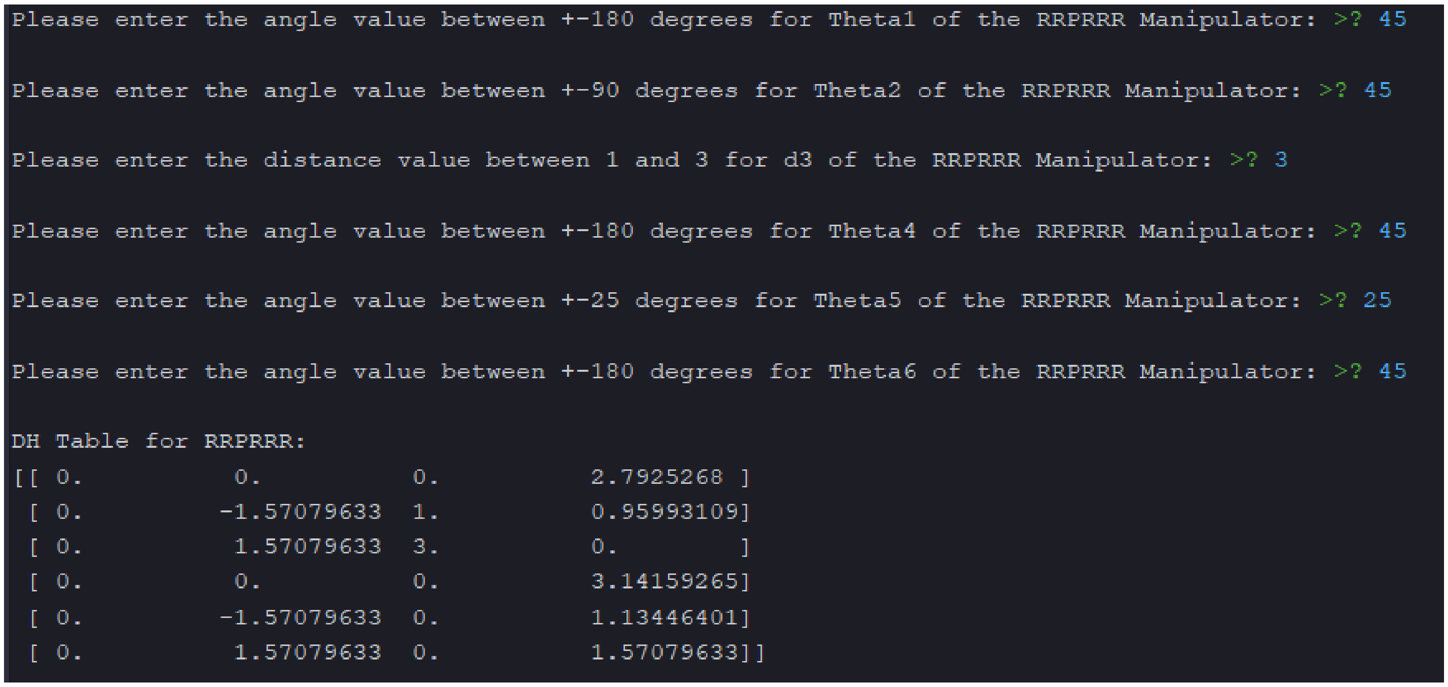
R^2PR^3 Manipulator



i	a_{i-1}	α_{i-1}	d_i	θ_i
1	0	0°	0	θ_1^*
2	0	-90°	d_2	$90° + \theta_2^*$
3	0	+90°	d_3	0°
4	0	0°	0	θ_4^*
5	0	-90°	0	$90° + \theta_5^*$
6	0	+90°	0	θ_6^*

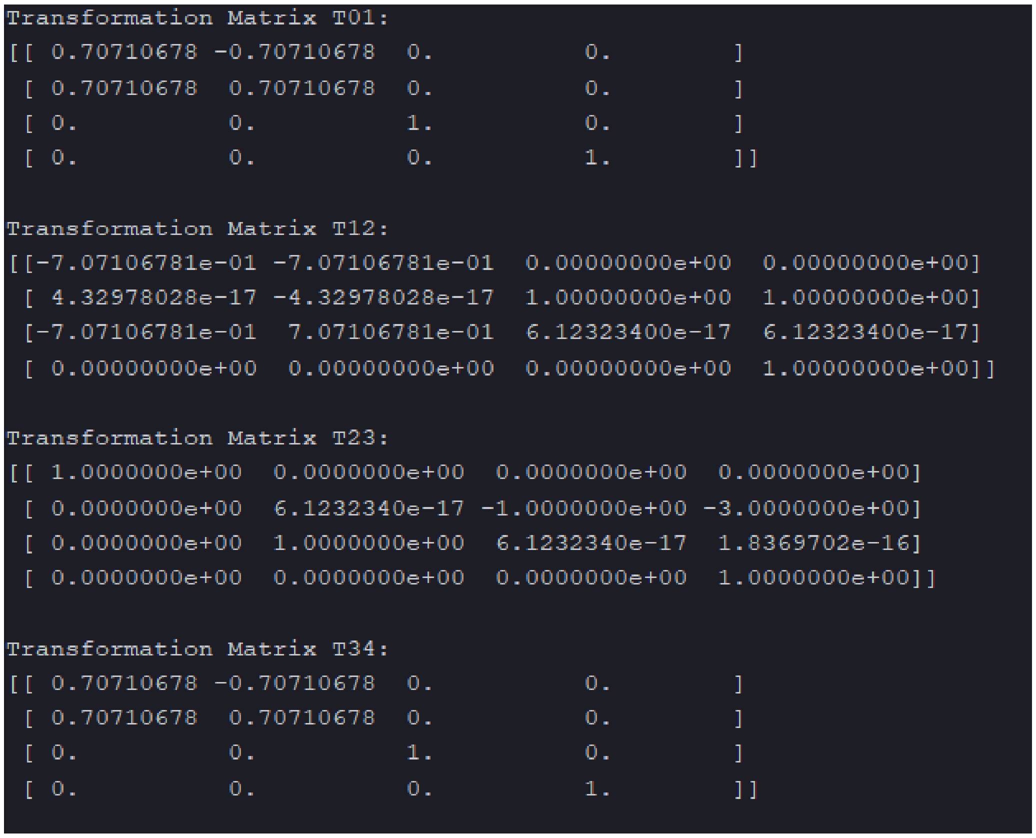
First, the user is asked to enter the joint parameters within specific limits. For simplicities sake, simple and consistent angles were chosen for the joint angles. The user selected parameters and calculated modified DH table can be seen below.



```
Please enter the angle value between +-180 degrees for Theta1 of the RRPRRR Manipulator: >? 45

Please enter the angle value between +-90 degrees for Theta2 of the RRPRRR Manipulator: >? 45

Please enter the distance value between 1 and 3 for d3 of the RRPRRR Manipulator: >? 3

Please enter the angle value between +-180 degrees for Theta4 of the RRPRRR Manipulator: >? 45

Please enter the angle value between +-25 degrees for Theta5 of the RRPRRR Manipulator: >? 25

Please enter the angle value between +-180 degrees for Theta6 of the RRPRRR Manipulator: >? 45

DH Table for RRPRRR:
[[ 0.          0.          0.          2.7925268 ]
 [ 0.         -1.57079633  1.          0.95993109]
 [ 0.          1.57079633  3.          0.        ]
 [ 0.          0.          0.          3.14159265]
 [ 0.         -1.57079633  0.          1.13446401]
 [ 0.          1.57079633  0.          1.57079633]]
```

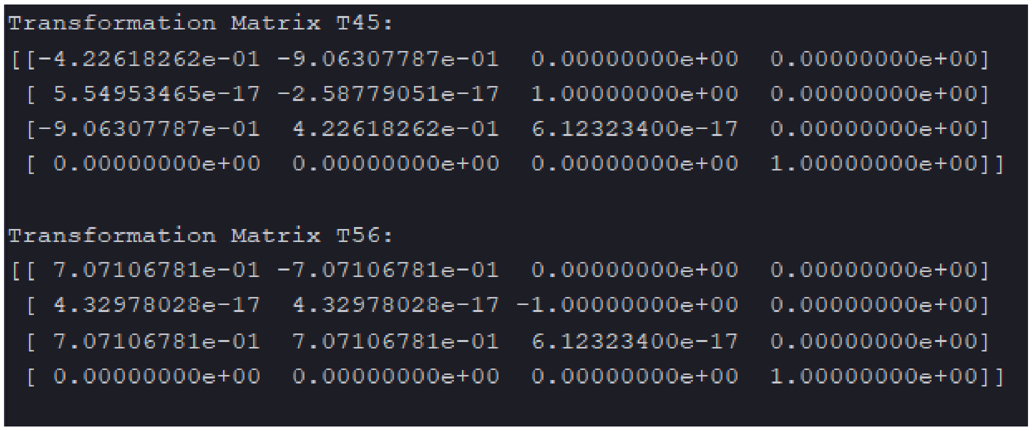
Next, all of the transformation matrices from the base to EE ($T_{i-1,i}$) are calculated and printed. Unfortunately, unlike MATLAB, the precision of floats cannot be reduce or truncated to view the matrices easier.

Transformations T01 through T45 can be seen below.



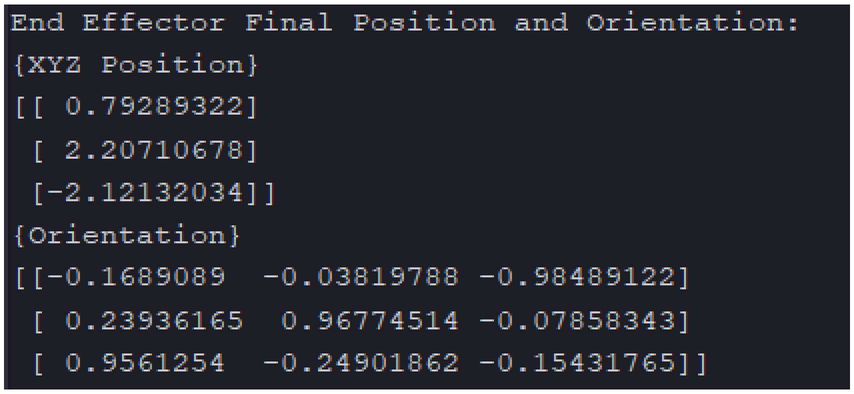
```
Transformation Matrix T01:
[[ 0.70710678 -0.70710678  0.          0.        ]
 [ 0.70710678  0.70710678  0.          0.        ]
 [ 0.          0.          1.          0.        ]
 [ 0.          0.          0.          1.        ]]

Transformation Matrix T12:
[[-7.07106781e-01 -7.07106781e-01  0.00000000e+00  0.00000000e+00]
 [ 4.32978028e-17 -4.32978028e-17  1.00000000e+00  1.00000000e+00]
 [-7.07106781e-01  7.07106781e-01  6.12323400e-17  6.12323400e-17]
 [ 0.00000000e+00  0.00000000e+00  0.00000000e+00  1.00000000e+00]]

Transformation Matrix T23:
[[ 1.0000000e+00  0.0000000e+00  0.0000000e+00  0.0000000e+00]
 [ 0.0000000e+00  6.1232340e-17 -1.0000000e+00 -3.0000000e+00]
 [ 0.0000000e+00  1.0000000e+00  6.1232340e-17  1.8369702e-16]
 [ 0.0000000e+00  0.0000000e+00  0.0000000e+00  1.0000000e+00]]

Transformation Matrix T34:
[[ 0.70710678 -0.70710678  0.          0.        ]
 [ 0.70710678  0.70710678  0.          0.        ]
 [ 0.          0.          1.          0.        ]
 [ 0.          0.          0.          1.        ]]
```

Transformations T45 and T56 can be seen below.



```
Transformation Matrix T45:
[[-4.22618262e-01 -9.06307787e-01  0.00000000e+00  0.00000000e+00]
 [ 5.54953465e-17 -2.58779051e-17  1.00000000e+00  0.00000000e+00]
 [-9.06307787e-01  4.22618262e-01  6.12323400e-17  0.00000000e+00]
 [ 0.00000000e+00  0.00000000e+00  0.00000000e+00  1.00000000e+00]]

Transformation Matrix T56:
[[ 7.07106781e-01 -7.07106781e-01  0.00000000e+00  0.00000000e+00]
 [ 4.32978028e-17  4.32978028e-17 -1.00000000e+00  0.00000000e+00]
 [ 7.07106781e-01  7.07106781e-01  6.12323400e-17  0.00000000e+00]
 [ 0.00000000e+00  0.00000000e+00  0.00000000e+00  1.00000000e+00]]
```

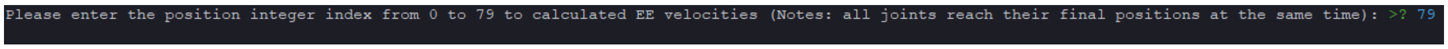
Next, the final position and orientation of the EE frame {6} was calculated and printed.



```
End Effector Final Position and Orientation:
{XYZ Position}
[[ 0.79289322]
 [ 2.20710678]
 [-2.12132034]]
{Orientation}
[[-0.1689089  -0.03819788 -0.98489122]
 [ 0.23936165  0.96774514 -0.07858343]
 [ 0.9561254  -0.24901862 -0.15431765]]
```

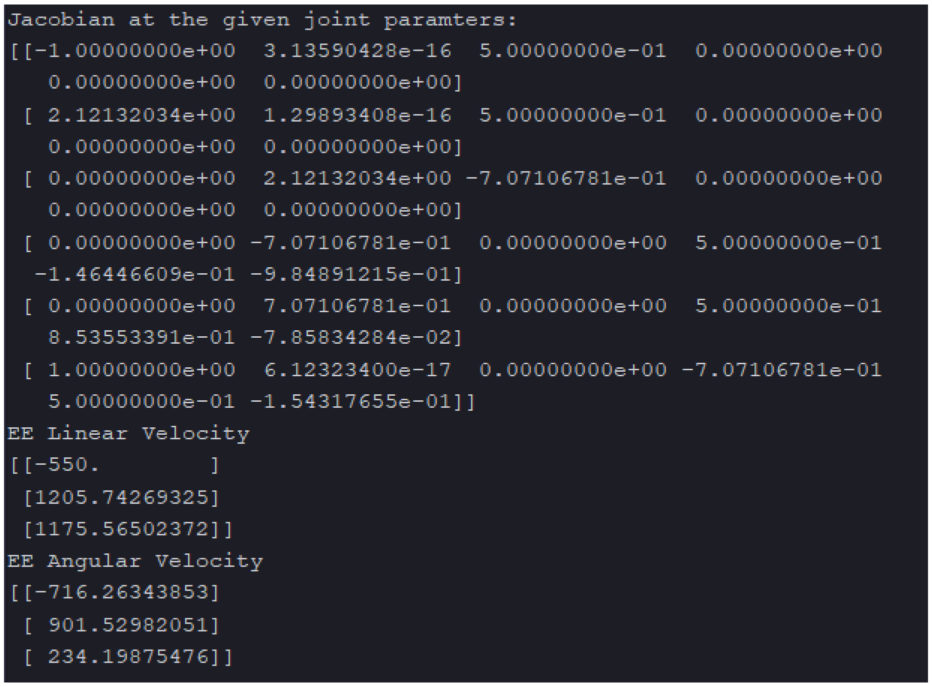
Next the velocity kinematics for the manipulator was calculated. This was based on the fact that all joints were commanded to reach their final positions at the same time, making the constant velocity calculations of each joint parameter doable. The user is prompted to decide where along the path of motion they would like to calculate the Jacobian and joint parameters. The position index they can choose is dependent on the number of frames of the simulation. Also, the number of frames and animation interval (in milliseconds) was used to estimate the time span of the simulated motion which was used in the constant velocity equation.

For this prompt, the user selected 79, which selects the final position of the joint parameters (for simplicity). The calculation for these values assumes that manipulator is still maintaining the constant velocity even though it stops at this position which is not physically possible (infinite deceleration) but it’s besides the point.



```
Please enter the position integer index from 0 to 79 to calculated EE velocities (Notes: all joints reach their final positions at the same time): >? 79
```

The Jacobian is shown below along with the EE linear and angular velocities. (Note: due to float precision, values are approximated and may not be what they actually are)



```
Jacobian at the given joint paramters:
[[-1.00000000e+00  3.13590428e-16  5.00000000e-01  0.00000000e+00
   0.00000000e+00  0.00000000e+00]
 [ 2.12132034e+00  1.29893408e-16  5.00000000e-01  0.00000000e+00
   0.00000000e+00  0.00000000e+00]
 [ 0.00000000e+00  2.12132034e+00 -7.07106781e-01  0.00000000e+00
   0.00000000e+00  0.00000000e+00]
 [ 0.00000000e+00 -7.07106781e-01  0.00000000e+00  5.00000000e-01
  -1.46446609e-01 -9.84891215e-01]
 [ 0.00000000e+00  7.07106781e-01  0.00000000e+00  5.00000000e-01
   8.53553391e-01 -7.85834284e-02]
 [ 1.00000000e+00  6.12323400e-17  0.00000000e+00 -7.07106781e-01
   5.00000000e-01 -1.54317655e-01]]
EE Linear Velocity
[[-550.        ]
 [1205.74269325]
 [1175.56502372]]
EE Angular Velocity
[[-716.26343853]
 [ 901.52982051]
 [ 234.19875476]]
```

For clarity (thanks to the fact that numpy arrays are not printed nicely), the Jacobian at the given joint parameters.

-1	0	0.5	0	0	0
2.12	0	0.5	0	0	0
0	2.12	-0.707	0	0	0
0	-0.707	0	0.5	-0.146	-0.984
0	0.707	0	0.5	0.854	-0.0785
1	0	0	-.707	0.5	-0.154