Robot Manipulation and Mobility Project 2022

Muslim Adedamola, Nurlan Kabdyshev

November 2022

Abstract

Analytical solution has been widely used for solving the inverse kinematic problem. In this final project, we are solving a direct and inverse kinematics for the serial-parallel manipulator which has three universal joints. We are solving this problem using an algebraic approach by using the screw displacement method. The simulation of the main leg of the manipulator was implemented. Overall, the results showed that it is possible to use such a method to solve the inverse kinematic of the serial-parallel manipulator.

Keywords—direct kinematics, inverse kinematics, matrices

1 Introduction

This Project requires us to analyze the forward and inverse kinematics of a manipulator using screw displacement and parallel manipulator theory. The manipulator has 3 universal joints. A universal joint can be considered as two revolute joints whose axes are orthogonal.
One of the most popular and effective methods to solve the direct and inverse kinematics is screw displacement. The specific axes are placed on the manipulator according to the initial, final, and all intermediate frames. After that, the directions and origins of these axes are found. Matrices from matrix element equation were derived.
First, in this problem, we find the direct kinematics of the manipulator. This is done by using the screw displacement method. We identify the screw axes (their directions and origins) and then build the table according to the found values. And finally, we simulate the manipulator in MATLAB.

1.1 Goals of the Project

- To Formalize Direct Kinematics of a Manipulator Using Successive Screw Displacement
- To Formalize Inverse Kinematics of a Manipulator
- To simulate the manipulator using ball-and-stick model

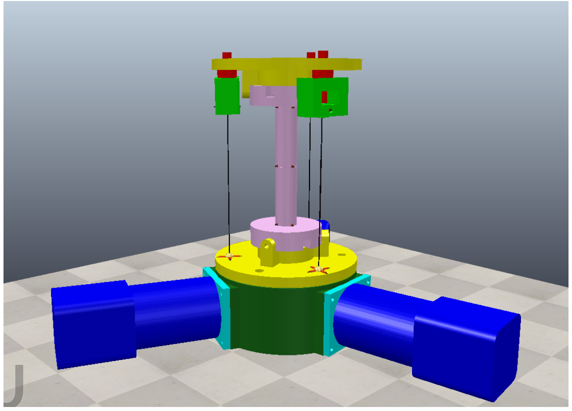


Figure 1: Model of the manipulator

2 Forward Kinematics

Formalizing Forward Kinematics only considering central leg using screw theory. We placed 3 screws on every universal joint of the central leg, which axis are aligned. After successive screw displacement and considering we don't have prismatic joints, we have the following table:

Joint i	S_i	S_oi
1	$\theta_1 = (0,1,0); \phi_1 = (1,0,0)$	(0,0,0); (0,0,0)
2	$\theta_1 = (0,1,0); \phi_1 = (1,0,0)$	(0,0,L_1); $(0,0,L_1)$
3	$\theta_1 = (0,1,0); \phi_1 = (1,0,0)$	(0,0,$L_1 + L_2$); $(0,0,L_1+L_2)$

For all joint t=0.
To obtain matrix equation, we need the matrix element equation:

```
a11 = (Sx^2 - 1)*(1 - cos(theta)) + 1;
a12 = Sx*Sy*(1 - cos(theta)) - Sz*sin(theta);
a13 = Sx*Sz*(1 - cos(theta)) + Sy*sin(theta);
a21 = Sy*Sx*(1 - cos(theta)) + Sz*sin(theta);
a22 = (Sy^2 - 1)*(1 - cos(theta)) + 1;
a23 = Sy*Sz*(1 - cos(theta)) - Sx*sin(theta);
a31 = Sz*Sx*(1 - cos(theta)) - Sy*sin(theta);
a32 = Sz*Sy*(1 - cos(theta)) + Sx*sin(theta);
a33 = (Sz^2 - 1)*(1 - cos(theta)) + 1;
a14 = t*Sx - Sox*(a11 - 1) - Soy*a12 - Soz*a13;
a24 = t*Sy - Sox*a21 - Soy*(a22 - 1) - Soz*a23;
a34 = t*Sz - Sox*a31 - Soy*a32 - Soz*(a33 - 1);
```

Now, we can obtain matrices of screw displacement due to $\theta \, and \, \phi$

$$A_{1\theta_1} = \begin{bmatrix} cos(\theta_1) & 0 & sin(\theta_1) & 0 \\ 0 & 1 & 0 & 0 \\ -sin(\theta_1) & 0 & cos(\theta_1) & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad A_{1\phi_1} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & cos(\phi_1) & -sin(\phi_1) & 0 \\ 0 & sin(\phi_1) & cos(\phi_1) & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$A_{2\theta_2} = \begin{bmatrix} cos(\theta_2) & 0 & sin(\theta_2) & -L_1 sin(\theta_2) \\ 0 & 1 & 0 & 0 \\ -sin(\theta_2) & 0 & cos(\theta_2) & -L_1 cos(\theta_2) - 1 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$A_{2\phi_2} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & cos(\phi_2) & -sin(\phi_2) & L_1 sin(\phi_2) \\ 0 & sin(\phi_2) & cos(\phi_2) & -L_1 cos(\phi_2) - 1 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$A_{3\theta_1} = \begin{bmatrix} cos(\theta_3) & 0 & sin(\theta_3) & -(L_1 + L_2)(sin(\phi_3)) \\ 0 & 1 & 0 & 0 \\ -sin(\theta_3) & 0 & cos(\theta_3) & -(L_1 + L_2)(cos(\phi_3)) \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$A_{3\phi_3} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & cos(\phi_3) & -sin(\phi_3) & (L_1 + L_2)(sin(\phi_2)) \\ 0 & sin(\phi_2) & cos(\phi_2) & -(L_1 + L_2)(cos(\phi_2) - 1) \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Using the matrices, we can obtain forward kinematics:
$A_{1\theta_1} * A_{1\phi_1} * A_{2\theta_2} * A_{2\phi_2} * A_{3\theta_3} * A_{3\phi_3} =$

```
[3*cos(theta)^4 - 4*cos(theta)^2 + cos(theta)^5 + 1, cos(2*theta)/32 + cos(3*theta)/4 - cos(5*theta)/4 -
(7*cos(2*theta)^2)/8 - cos(3*theta)^2/16 + 29/32, sin(4*theta)/2 - (3*sin(2*theta))/32 + sin(5*theta)/4 +
sin(6*theta)/32 - sin(theta)/4, 2*t - 3*t*cos(theta)^2 - t*cos(theta)^3 + 2*t*cos(theta)^4 + t*cos(theta)^5 +
2*t*cos(theta) + 3*Soz*cos(theta)*sin(theta) + 3*Soz*cos(theta)^2*sin(theta) - 3*Soz*cos(theta)^3*sin(theta) -
4*Soz*cos(theta)^4*sin(theta) - Soz*cos(theta)^5*sin(theta)]

[cos(theta)*(cos(theta) - 2*cos(theta)^2 - cos(theta)^3 + 2),    3*cos(theta)^4 - 4*cos(theta)^2 + cos(theta)^5 +
1,               -cos(theta)*sin(theta)*(cos(theta) + 3*cos(theta)^2 + cos(theta)^3 - 2),          (17*t)/8 -
(t*cos(3*theta))/4 - (t*cos(4*theta))/8 - (5*Soz*sin(theta))/8 + (5*t*cos(theta))/4 - (Soz*sin(2*theta))/4 +
(7*Soz*sin(3*theta))/16 + (3*Soz*sin(4*theta))/8 + (Soz*sin(5*theta))/16]

[-cos(theta)*sin(theta)*(cos(theta) + 3*cos(theta)^2 + cos(theta)^3 - 2),              sin(4*theta)/2 -
(3*sin(2*theta))/32 + sin(5*theta)/4 + sin(6*theta)/32 - sin(theta)/4,   3*cos(theta)^4 - 4*cos(theta)^3 -
4*cos(theta)^2 + 4*cos(theta)^5 + cos(theta)^6 + 1, t*sin(2*theta) - Soz - 2*t*sin(theta) + 5*Soz*cos(theta)^2 +
4*Soz*cos(theta)^3 - 3*Soz*cos(theta)^4 - 4*Soz*cos(theta)^5 - Soz*cos(theta)^6 - 2*t*cos(theta)^3*sin(theta) -
t*cos(theta)^4*sin(theta)]

[0,0,0,1]
```

3 Inverse Kinematics

The inverse kinematics problem of the overall parallel manipulator is to find the length of the three tendons that let to achieve a reachable pose.
Let the lengths of the tendons be ***K1, K2, and K3*** respectively. For the purpose of analysis, as seen in Figure 1a and 1b, two Cartesian systems *A(x,y,z)* and *B(u,v,w)* are attached to the fixed base and moving platform

respectively. The following points are used:

- As seen in Fig1a, points A1, A2, and A3 lie on the x-y plane while point B1, B2, and B3 lie on the u-v plane.
- As seen in Fig1b, the origin O of the fixed coordinate system is located at the centroid of triangle A1-A2-A3 and the X-axis points in the direction $\overline{OA_1}$.
- Similarly, the origin P of the moving coordinate system is located at the centroid of triangle B1-B2-B3 and the U-axis points in the direction of $\overline{PB_1}$
- Both triangles **A1-A2-A3 and B1-B2-B3** are equilateral triangles with $\overline{OA_1} = \overline{OA_2} = \overline{OA_3} = \overline{PB_1} = \overline{PB_2} = \overline{PB_3} = \mathbf{d}$.

Let the transformation from the moving platform to the fixed base be described by a position vector $\mathbf{p} = \overline{OP}$ (shown in Fig1a) and the 3 by 3 rotation matrix ${}^{A}R_{B}$ previously obtained from the forward kinematics problem. Also, let **u, v,** and **w** be the three unit vectors defined along the *u, v, and w* axes of the moving coordinate system **B**. Therefore, the rotation matrix can be expressed in terms of direction cosines as **u, v, and w** as

$$ {}^{A}R_{B} = \begin{bmatrix} u_x & v_x & w_x \\ u_y & v_y & w_y \\ u_z & v_y & w_z \end{bmatrix} \tag{1} $$

which is equal to the rotation part of the complex transformation matrix previously obtained from the direct kinematics.
Let a_i and ${}^{B}b_i$ be the position vectors of points A_i and B_i in the coordinate system **A** and **B** respectively. As seen in the Figure 2, since the distance of A_1, A_2, A_3, B_1, B_2, and B_3 are **d** from the centre of the triangle, and the three tendons are displaced on a circle at 120 degrees each, using cosine rule and Pythagoras theorem, the length of the sides, vertical, and horizontal lines can be easily obtained. As seen in Figure 2, the coordinates A_i and B_i are given by

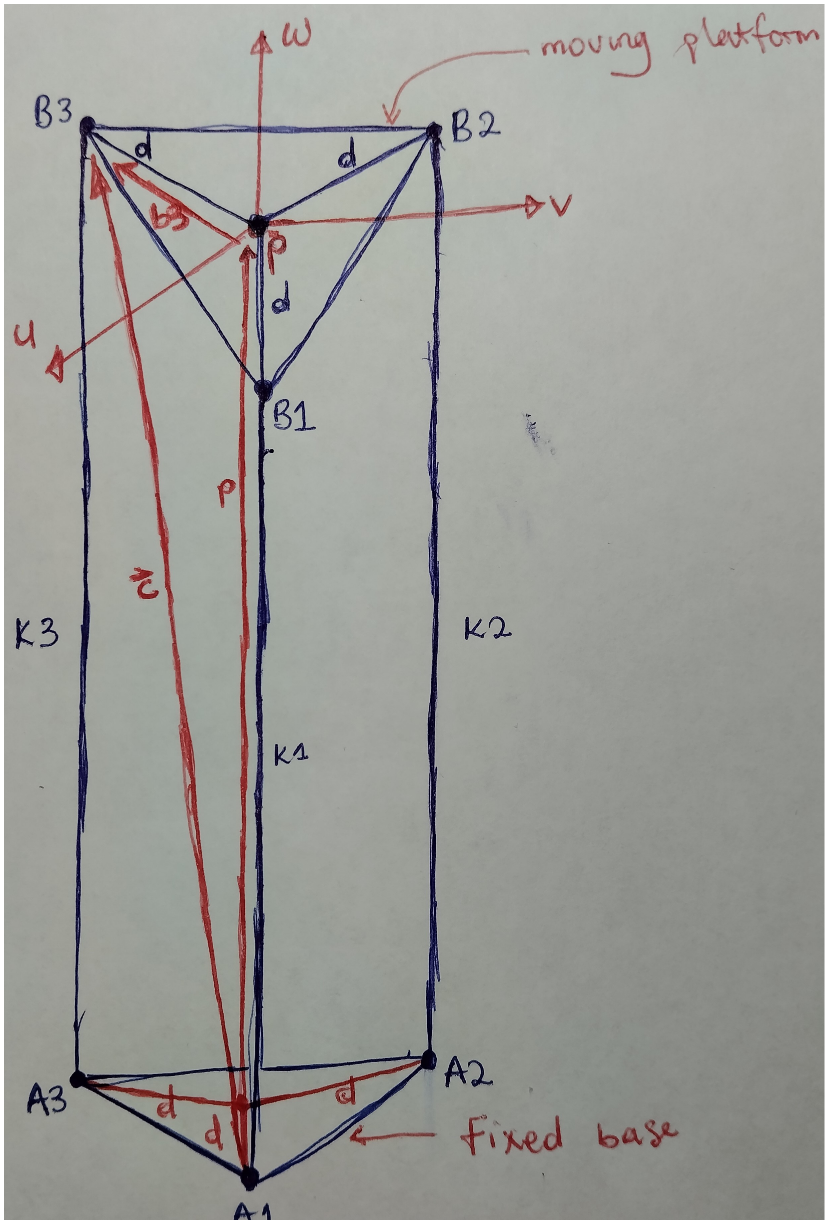
$$ {}^{A}a1 = \begin{bmatrix} d, & 0, & 0 \end{bmatrix}^{\top} \tag{2} $$

$$ {}^{A}a2 = \begin{bmatrix} -d/2, & (d\sqrt{3})/2, & 0 \end{bmatrix}^{\top} \tag{3} $$

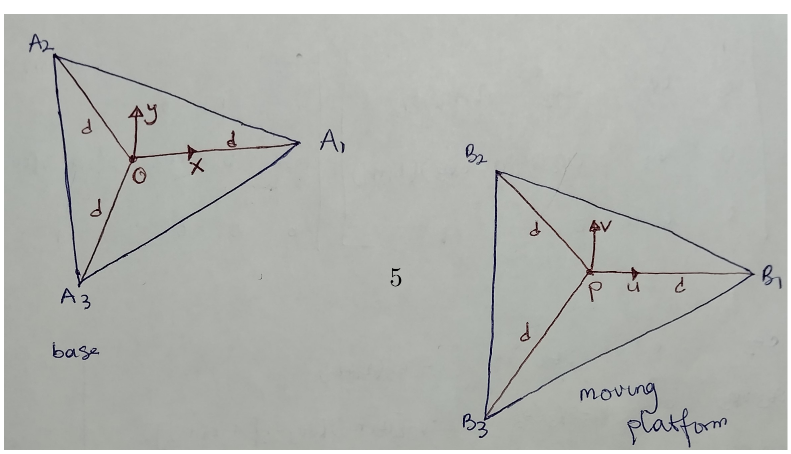
$$ {}^{A}a3 = \begin{bmatrix} -d/2, & -(d\sqrt{3})/2, & 0 \end{bmatrix}^{\top} \tag{4} $$

$$ {}^{B}b1 = \begin{bmatrix} d, & 0, & 0 \end{bmatrix}^{\top} \tag{5} $$

$$ {}^{B}b2 = \begin{bmatrix} -d/2, & (d\sqrt{3})/2, & 0 \end{bmatrix}^{\top} \tag{6} $$



(a) Fig1a



(b) Fig1b

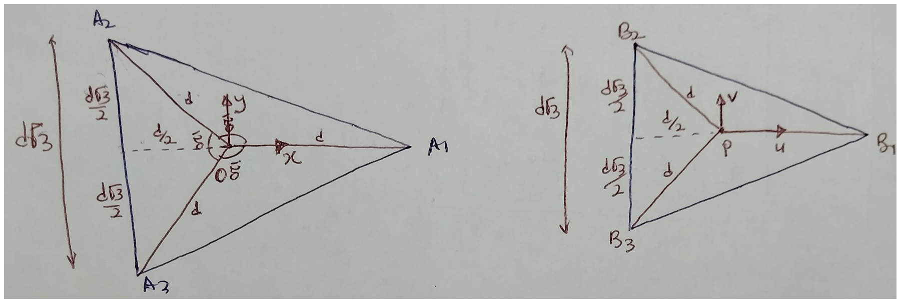


Figure 3: Coordinates Ai and Bi

$$^{B}b3 = \begin{bmatrix} -d/2, & -(d\sqrt{3})/2, & 0 \end{bmatrix}^{\top} \tag{7}$$

The vector $\mathbf{p}$ has coordinates in x, y, z and can be written as:

$$\mathbf{P} = \begin{bmatrix} P_x, & P_y, & P_z \end{bmatrix}^{\top} \tag{8}$$

The position vector $\mathbf{C}\boldsymbol{i}$ of $\mathbf{B}\boldsymbol{i}$ with respect to the fixed coordinate system can be obtained using vector loop equation from the following transformation (see figure 1a):

$$\mathbf{C}\boldsymbol{i} = \mathbf{p} + {}^{B}R_{A}{}^{B}b_i \tag{9}$$

Putting the values of ${}^{B}R_A$, $\mathbf{p}$, and ${}^{B}b_i$ in Equation 8, that is, substituting Equations 1, 5, 6, 7, 8 into Equation 9 gives:

$$C_1 = \begin{bmatrix} P_x + dU_x \\ P_y + dU_y \\ P_z + dU_z \end{bmatrix} \tag{10}$$

$$C_2 = \begin{bmatrix} P_x - \frac{dU_x}{2} + \frac{d\sqrt{3}}{2}V_x \\ P_y - \frac{dU_y}{2} + \frac{d\sqrt{3}}{2}V_y \\ P_z - \frac{dU_z}{2} + \frac{d\sqrt{3}}{2}V_z \end{bmatrix} \tag{11}$$

$$C_3 = \begin{bmatrix} P_x - \frac{dU_x}{2} - \frac{d\sqrt{3}}{2}V_x \\ P_y - \frac{dU_y}{2} - \frac{d\sqrt{3}}{2}V_y \\ P_z - \frac{dU_z}{2} - \frac{d\sqrt{3}}{2}V_z \end{bmatrix} \tag{12}$$

To calculate the length of each tendons **K1, K2, and K3**, writing vector-loop equation, we can say as shown in Figure 3,

$$\mathbf{a} + \mathbf{k} = \mathbf{c}, then, \mathbf{k} = \mathbf{c} - \mathbf{a} \tag{13}$$

Therefore, the length of each tendon is given by:

$$K_i^2 = [C_i - a_i]^{\top}[C_i - a_i] \tag{14}$$

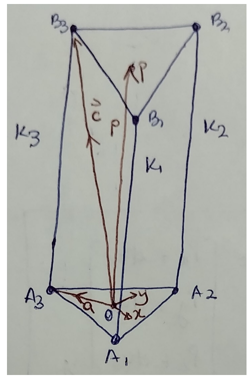


Figure 4: Vector Loop Equation to find K, length of tendons

where i = 1, 2, and 3 C_1, C_2, C_3 have been calculated respectively as seen in equations 10, 11, and 12. Values of a_1, a_2, and a_3 have also been calculated as seen in equations 2 to 4. Substituting the respective values of C_1, C_2, C_3 and a_1, a_2, and a_3 into equation 14, we obtain $k_1, k_2, and k_3$ which are the lengths of the tendons.

$$K_1^2 = [C_1 - a_1]^\top [C_1 - a_1] \tag{15}$$

and

$$C_1 = \begin{bmatrix} P_x + dU_x \\ P_y + dU_y \\ P_z + dU_z \end{bmatrix}$$

and

$$a1 = \begin{bmatrix} d \\ 0 \\ 0 \end{bmatrix}$$

then

$$K_1^2 = \begin{bmatrix} P_x + dU_x - d \\ P_y + dU_y \\ P_z + dU_z \end{bmatrix}^\top \begin{bmatrix} P_x + dU_x - d \\ P_y + dU_y \\ P_z + dU_z \end{bmatrix}$$

then

$$K_1^2 = \begin{bmatrix} P_x + dU_x - d & P_y + dU_y & P_z + dU_z \end{bmatrix}^\top \begin{bmatrix} P_x + dU_x - d \\ P_y + dU_y \\ P_z + dU_z \end{bmatrix}$$

then we obtain
$K_1^2 = P_x^2 + P_y^2 + P_z^2 + 2d(P_xU_x + P_yU_y + P_zU_z - dU_x) - 2dP_x + d^2(U_x^2 + U_y^2 + U_z^2) + d^2$
from which we can obtain K_1 as

$K_1 =$
$\sqrt{P_x^2 + P_y^2 + P_z^2 + 2d(P_xU_x + P_yU_y + P_zU_z - dU_x) - 2dP_x + d^2(U_x^2 + U_y^2 + U_z^2) + d^2}$
In the same way,

$$K_2^2 = [C_2 - a_2]^\top [C_2 - a_2] \tag{16}$$

and

$$C_2 = \begin{bmatrix} P_x - \frac{dU_x}{2} + \frac{d\sqrt{3}}{2}V_x \\ P_y - \frac{dU_y}{2} + \frac{d\sqrt{3}}{2}V_y \\ P_z - \frac{dU_z}{2} + \frac{d\sqrt{3}}{2}V_z \end{bmatrix}$$

and

$$a2 = \begin{bmatrix} -d/2 \\ (d\sqrt{3})/2 \\ 0 \end{bmatrix}$$

then

$$K_2^2 = \begin{bmatrix} P_x - 1/2(dU_x) + \frac{d\sqrt{3}}{2}V_x + d/2 \\ P_y - 1/2(dU_y) + \frac{d\sqrt{3}}{2}V_y - d\sqrt{3}/2 \\ P_z - 1/2(dU_z) + \frac{d\sqrt{3}}{2}V_z \end{bmatrix}^\top \begin{bmatrix} P_x + dU_x - d \\ P_y + dU_y \\ P_z + dU_z \end{bmatrix}$$

then we obtain

$$\begin{aligned} K_2^2 = P_x^2 + P_y^2 + P_z^2 - d(P_xU_x + P_yU_y + \\ P_zU_z) + d\sqrt{3}(P_xV_x + P_yV_y + P_zV_z) + d(P_x - \\ P_y\sqrt{3}) + \frac{d^2(V_x\sqrt{3} - 3V_y)}{2} - \frac{d^2(U_x - \sqrt{3}U_y)}{2} + 2d^2 \end{aligned} \tag{17}$$

and K_2 is the square root of equation 17.
Applying the same to K_3, we have

$$\begin{aligned} K_3^2 = P_x^2 + P_y^2 + P_z^2 - d(P_xU_x + P_yU_y + \\ P_zU_z) - d\sqrt{3}(P_xV_x + P_yV_y + P_zV_z) + d(P_x + \\ P_y\sqrt{3}) - \frac{d^2(V_x\sqrt{3} + 3V_y)}{2} - \frac{d^2(U_x + \sqrt{3}U_y)}{2} + 2d^2 \end{aligned} \tag{18}$$

where K_3 is the square root of Equation 18.

4 Simulation

After that, we have implemented both visual and interactive model to the MATLAB:

```
robot = rigidBodyTree("DataFormat","column");
base = robot.Base;
rotatingBase = rigidBody("rotating_base");
arm1 = rigidBody("arm1");
arm2 = rigidBody("arm2");
gripper = rigidBody("gripper");
collBase = collisionCylinder(0.1,0.01); % cylinder: radius, length
collBase.Pose = trvec2tform([0 0 0.01/2]);
coll1 = collisionCylinder(0.01,0.25);
coll1.Pose = trvec2tform([0 0 0.25/2]);
coll2 = collisionCylinder(0.01,0.25);
coll2.Pose = trvec2tform([0 0 0.25/2]);
collGripper = collisionCylinder(0.1,0.01);
collGripper.Pose = trvec2tform([0 0 0.01/2]);
addCollision(rotatingBase, collBase)
addCollision(arm1, coll1)
addCollision(arm2, coll2)
addCollision(gripper, collGripper)
jntBase = rigidBodyJoint("base_joint");
jnt1 = rigidBodyJoint("jnt1","revolute");
jntGripper = rigidBodyJoint("gripper_joint","revolute");
jnt2 = rigidBodyJoint("jnt2","revolute");
jntGripper = rigidBodyJoint("gripper_joint","revolute");
jnt1.JointAxis = [1 1 0];
jnt2.JointAxis = [1 1 0];
jntGripper.JointAxis = [1 1 0];
setFixedTransform(jnt1, trvec2tform([0 0 0]))
setFixedTransform(jnt2, trvec2tform([0 0 0.25]))
setFixedTransform(jntGripper, trvec2tform([0 0 0.25]))
bodies = {base, rotatingBase, arm1, arm2, gripper};
joints = {[], jntBase, jnt1, jnt2, jntGripper};

figure("Name","Assemble_Robot","Visible","on")
for i = 2:length(bodies) % Skip base. Iterate through adding bodies and joints.
        bodies{i}.Joint = joints{i};
        addBody(robot, bodies{i}, bodies{i-1}.Name)
        show(robot,"Collisions","on","Frames","on");
        drawnow;
end
showdetails(robot)
gui = interactiveRigidBodyTree(robot,"MarkerScaleFactor",0.25);
```

Here you can see the results of the simulation. The model behaves as it has 3 universal joints and end effector can be interacted with and other joints will be simulated accordingly.

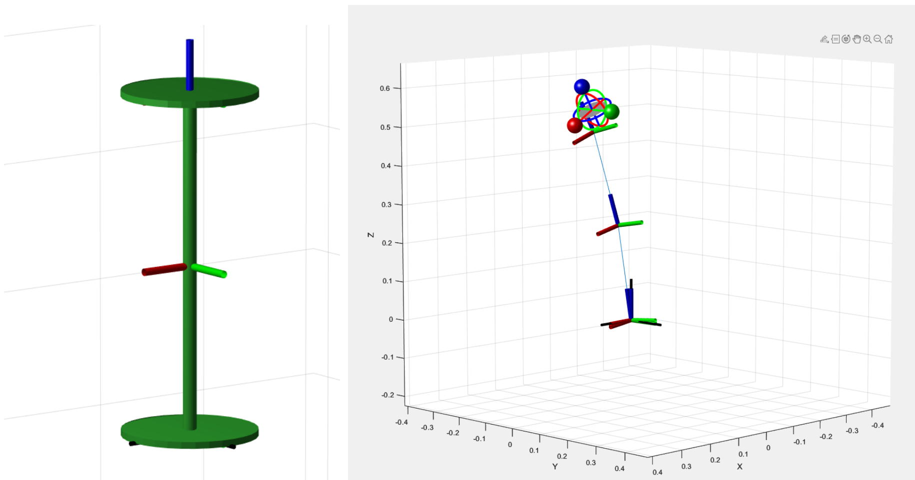


Figure 5: The visual and interactive MATLAB model

5 Conclusion

In this project we have found Direct and Inverse Kinematics for a serial/parallel manipulator using Screw Theory. We found the length of the three tendons that let to achieve a reachable pose. We simulated the robot in MATLAB and plotted 3d model of robot and simulated interactive simplified model.