Robotic Science and Systems - Exercise 1

Kevin Robb

February 9, 2022

1

A frame $\{B\}$ is obtained from frame $\{A\}$ by first rotating about $\hat{x}_A$ by θ and then translating by $\begin{bmatrix}1 & 2 & 3\end{bmatrix}^T$ in the $\{A\}$ frame.

1.a Find the homogenous transformation matrix ${}^AT_B$.

This transformation between $\{A\}$ and $\{B\}$ can be written simply with the given rotation and translation.

$$ {}^AT_B = \left(\begin{array}{c|c} R & t \\ \hline 0 & 1 \end{array}\right) = \left(\begin{array}{ccc|c} 1 & 0 & 0 & 1 \\ 0 & \cos\theta & -\sin\theta & 2 \\ 0 & \sin\theta & \cos\theta & 3 \\ \hline 0 & 0 & 0 & 1 \end{array}\right). $$

We can check this by seeing that multiplying this by some vector at the origin of $\{B\}$ (e.g. ${}^Bp = \begin{bmatrix}0 & 0 & 0\end{bmatrix}^T$) will give the representation of this vector in frame $\{A\}$.

1.b Compute the homogenous transformation matrix ${}^BT_A$.

We know that ${}^BT_A = {}^AT_B{}^{-1}$. We can compute this inverse by using properties of the transformation. Since ${}^AT_B \in SE(3)$, its rotation submatrix $R \in SO(3)$ satisfies $R^{-1} = R^T$ (i.e., R is orthonormal). Since the translation is relative to frame $\{A\}$, we can invert it by first undoing the rotation and then applying the negative of the original translation. Applied, this looks like

$$ {}^BT_A = {}^AT_B{}^{-1} = \left(\begin{array}{c|c} R^T & R^T \cdot -t \\ \hline 0 & 1 \end{array}\right) = \left(\begin{array}{ccc|c} 1 & 0 & 0 & -1 \\ 0 & \cos\theta & \sin\theta & -2\cos\theta - 2\sin\theta \\ 0 & -\sin\theta & \cos\theta & 3\sin\theta - 3\cos\theta \\ \hline 0 & 0 & 0 & 1 \end{array}\right). $$

1.c Compute ${}^Bp$ given $\theta = \pi/4$.

A vector (in its affine representation) will be transformed in the following way from its representation in $\{A\}$ to that in $\{B\}$.

$$ {}^Bp = {}^B T_A \cdot {}^A p. $$

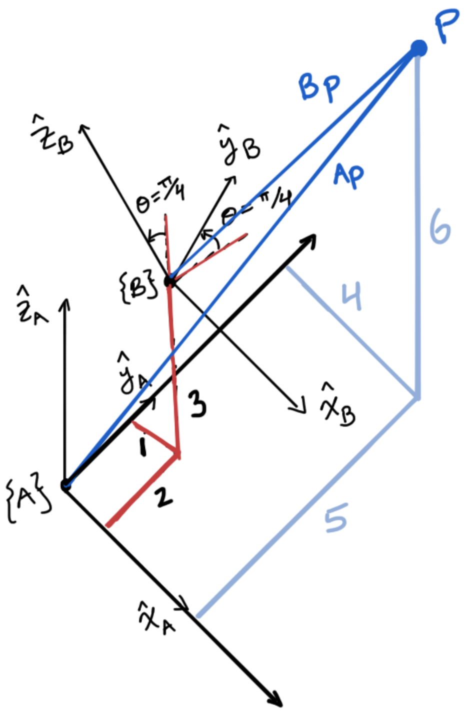
Using the example vector in $\{A\}$, ${}^{A}p = \begin{bmatrix}4 & 5 & 6\end{bmatrix}^T$, and letting $\theta = \pi/4$, we can thus find this vector's equivalent representation in $\{B\}$, ${}_{B}p$.

$$
\begin{aligned}
{}^{B}p = {}^{B}T_A \cdot {}^{A}p &= \left(\begin{array}{ccc|c} 1 & 0 & 0 & -1 \\ 0 & \cos\frac{\pi}{4} & \sin\frac{\pi}{4} & -2\cos\frac{\pi}{4} - 2\sin\frac{\pi}{4} \\ 0 & -\sin\frac{\pi}{4} & \cos\frac{\pi}{4} & 3\sin\frac{\pi}{4} - 3\cos\frac{\pi}{4} \\ \hline 0 & 0 & 0 & 1 \end{array}\right) \cdot \left[\begin{array}{c} 4 \\ 5 \\ 6 \\ \hline 1 \end{array}\right] \\
&= \left(\begin{array}{ccc|c} 1 & 0 & 0 & -1 \\ 0 & \frac{\sqrt{2}}{2} & \frac{\sqrt{2}}{2} & -2\sqrt{2} \\ 0 & -\frac{\sqrt{2}}{2} & \frac{\sqrt{2}}{2} & 0 \\ \hline 0 & 0 & 0 & 1 \end{array}\right) \cdot \left[\begin{array}{c} 4 \\ 5 \\ 6 \\ \hline 1 \end{array}\right] \\
&= \left[\begin{array}{c} 3 \\ \frac{7}{2}\sqrt{2} \\ \frac{1}{2}\sqrt{2} \\ \hline 1 \end{array}\right]
\end{aligned}
$$

Thus the vector ${}^{B}p = \begin{bmatrix}3 & 7/\sqrt{2} & 1/\sqrt{2}\end{bmatrix}^T$. ■

1.d Sketch the frames for the specifications in part (c).

The sketch in Figure 1.d shows both coordinate frames and the point relative to both of them. Our result in part (c) seems to align with this, as we can see from the figure that the x-coordinate should be 3, y should be around 3-6, and z should be around 1. We see this agrees with our analytical result.



2

A frame $\{A\}$ is rotated about $\hat{y}_A$ by θ to form frame $\{B\}$, which is then rotated about $\hat{z}_B$ by ϕ to form frame $\{C\}$.

Given a position vector ${}^C p$, we want to find the rotation matrix ${}^A R_C \in SO(3)$ s.t. ${}^A p ={}^A R_C \cdot {}^C p$.

We can first form ${}^B R_A$ and ${}^C R_B$ from the information given, then compose and invert them.

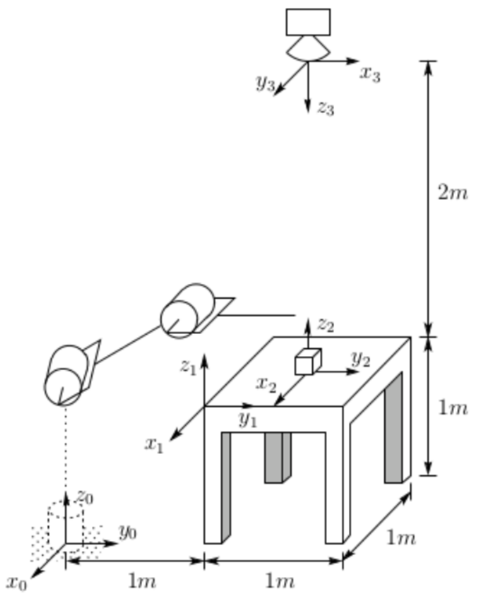
$$
{}^A R_B = \begin{pmatrix} \cos\theta & 0 & \sin\theta \\ 0 & 1 & 0 \\ -\sin\theta & 0 & \cos\theta \end{pmatrix}
$$

$$
{}^B R_C = \begin{pmatrix} \cos\phi & -\sin\phi & 0 \\ \sin\phi & \cos\phi & 0 \\ 0 & 0 & 1 \end{pmatrix}
$$

Then the combined rotation is

$$
{}^A R_C ={}^A R_B \cdot {}^B R_C = \begin{pmatrix} \cos\theta\cos\phi & -\cos\theta\sin\phi & \sin\theta \\ \sin\phi & \cos\phi & 0 \\ -\sin\theta\cos\phi & \sin\theta\sin\phi & \cos\theta \end{pmatrix}. \quad \blacksquare
$$

3 Spong, Problem 2.37



3.a

The homogenous transformations relating each frame in the figure to frame $\{0\}$ can be found easily by observing how each basis vector is transformed, and what the relative translation is between pairs of frames.

These are as follows:

$$
{}^0T_1 = \left(\begin{array}{ccc|c} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 1 \\ \hline 0 & 0 & 0 & 1 \end{array}\right)
$$

$$
{}^0T_2 = \left(\begin{array}{ccc|c} 1 & 0 & 0 & -0.5 \\ 0 & 1 & 0 & 1.5 \\ 0 & 0 & 1 & 1 \\ \hline 0 & 0 & 0 & 1 \end{array}\right)
$$

$$
{}^0T_3 = \left(\begin{array}{ccc|c} 0 & 1 & 0 & -0.5 \\ 1 & 0 & 0 & 1.5 \\ 0 & 0 & -1 & 3 \\ \hline 0 & 0 & 0 & 1 \end{array}\right) \qquad \blacksquare
$$

3.b

We can then find the transformation from the cube's frame {2} to the camera's frame {3} via the following sequence:

$$
\begin{aligned}
{}^3T_2 &= {}^3T_0 \cdot {}^0T_2 = {}^0T_3{}^{-1} \cdot {}^0T_2 \\
&= \left(\begin{array}{ccc|c} 0 & 1 & 0 & -1.5 \\ 1 & 0 & 0 & 0.5 \\ 0 & 0 & -1 & 3 \\ \hline 0 & 0 & 0 & 1 \end{array}\right) \cdot \left(\begin{array}{ccc|c} 1 & 0 & 0 & -0.5 \\ 0 & 1 & 0 & 1.5 \\ 0 & 0 & 1 & 1 \\ \hline 0 & 0 & 0 & 1 \end{array}\right) \\
&= \left(\begin{array}{ccc|c} 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 0 & -1 & 2 \\ \hline 0 & 0 & 0 & 1 \end{array}\right) \qquad \blacksquare
\end{aligned}
$$

We can then compare this to the figure as a sanity check, and it does indeed make sense.

3.c

If the robot then pushes the cube and the camera detects its origin at ${}^3p = \begin{bmatrix} 0.2 & -0.3 & 2 \end{bmatrix}^T$ in frame {3}, the position of the cube in the base frame becomes

$$
\begin{aligned}
{}^0p &= {}^0 T_3 \cdot {}^3 p \\
&= \left(\begin{array}{ccc|c} 0 & 1 & 0 & -0.5 \\ 1 & 0 & 0 & 1.5 \\ 0 & 0 & -1 & 3 \\ \hline 0 & 0 & 0 & 1 \end{array}\right) \cdot \left[\begin{array}{c} 0.2 \\ -0.3 \\ 2 \\ \hline 1 \end{array}\right] \\
&= \left[\begin{array}{c} -0.8 \\ 1.7 \\ 1 \\ \hline 1 \end{array}\right]. \qquad \blacksquare
\end{aligned}
$$

3.d

If we want the robot to push the cube to the corner of the table s.t. ${}^1p = \begin{bmatrix} -0.1 & 0.1 & 0 \end{bmatrix}^T$, we can write this relative pose as

$$ {}^1T_2 = \left(\begin{array}{ccc|c} 1 & 0 & 0 & -0.1 \\ 0 & 1 & 0 & 0.1 \\ 0 & 0 & 1 & 0 \\ \hline 0 & 0 & 0 & 1 \end{array}\right) $$

We can then expect the camera's view of the cube will be

$$ \begin{aligned} {}^3T_2 &= {}^0T_3{}^{-1} \cdot {}^0T_1 \cdot {}^1T_2 \\ &= \left(\begin{array}{ccc|c} 0 & 1 & 0 & -1.5 \\ 1 & 0 & 0 & 0.5 \\ 0 & 0 & -1 & 3 \\ \hline 0 & 0 & 0 & 1 \end{array}\right) \cdot \left(\begin{array}{ccc|c} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 1 \\ \hline 0 & 0 & 0 & 1 \end{array}\right) \cdot \left(\begin{array}{ccc|c} 1 & 0 & 0 & -0.1 \\ 0 & 1 & 0 & 0.1 \\ 0 & 0 & 1 & 0 \\ \hline 0 & 0 & 0 & 1 \end{array}\right) \\ &= \left(\begin{array}{ccc|c} 0 & 1 & 0 & -0.4 \\ 1 & 0 & 0 & 0.4 \\ 0 & 0 & -1 & 2 \\ \hline 0 & 0 & 0 & 1 \end{array}\right) \quad \blacksquare \end{aligned} $$

4 3-DOF manipulator forward kinematics

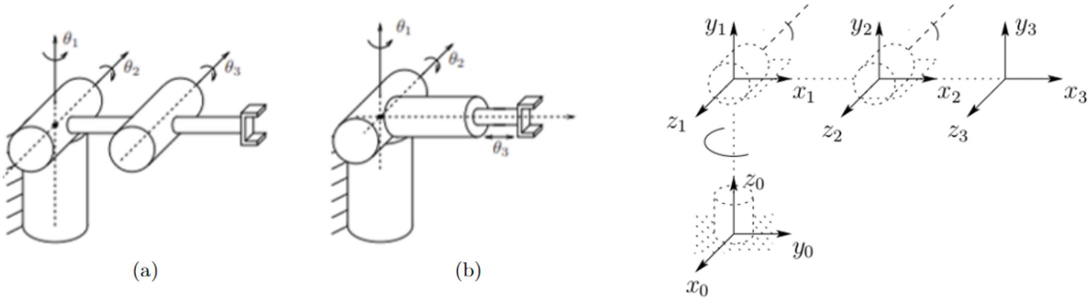


Figure 1: (a) L_1,L_2,L_3 are the lengths of the three segments.
(b) L_1,L_2 are the same as (a), and $L_3 = \theta_3$ is variable, depending on the translational joint.

Note: I have assumed all angles are positive counter-clockwise, obeying the right-hand-rule for whatever axis they are about; this means θ_2 increases opposite the direction the arrow is drawn in the figure. I have also assumed in the RRP case that $\hat{z}_3$ should point forward along the direction of motion and $\hat{x}_3$ should point up.

4.a RRR Manipulator

We will use the provided set of intermediate coordinate frames. I will separate transforms into their movement at $\theta_i = 0$ and the effect of movement at that joint alone, since it helps me keep things clear. The forward kinematics map here is a composition of all these transformations, so

$$
{}^{0}T_3 = \left(\begin{array}{ccc|c} 0 & 0 & 1 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & L_1 \\ \hline 0 & 0 & 0 & 1 \end{array}\right) \cdot \left(\begin{array}{ccc|c} \cos\theta_1 & 0 & \sin\theta_1 & 0 \\ 0 & 1 & 0 & 0 \\ -\sin\theta_1 & 0 & \cos\theta_1 & 0 \\ \hline 0 & 0 & 0 & 1 \end{array}\right) \cdot \left(\begin{array}{ccc|c} \cos\theta_2 & -\sin\theta_2 & 0 & 0 \\ \sin\theta_2 & \cos\theta_2 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ \hline 0 & 0 & 0 & 1 \end{array}\right)
$$

$$
\cdot \left(\begin{array}{ccc|c} 1 & 0 & 0 & L_2 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ \hline 0 & 0 & 0 & 1 \end{array}\right) \cdot \left(\begin{array}{ccc|c} \cos\theta_3 & -\sin\theta_3 & 0 & 0 \\ \sin\theta_3 & \cos\theta_3 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ \hline 0 & 0 & 0 & 1 \end{array}\right) \cdot \left(\begin{array}{ccc|c} 1 & 0 & 0 & L_3 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ \hline 0 & 0 & 0 & 1 \end{array}\right)
$$

$$
= \left(\begin{array}{ccc|c} -\cos(\theta_2+\theta_3)\cdot\sin(\theta_1) & \sin(\theta_2+\theta_3)\cdot\sin(\theta_1) & \cos(\theta_1) & -\sin(\theta_1)\cdot(L3\cdot\cos(\theta_2+\theta_3)+L2\cdot\cos(\theta_2)) \\ \cos(\theta_2+\theta_3)\cdot\cos(\theta_1) & -\sin(\theta_2+\theta_3)\cdot\cos(\theta_1) & \sin(\theta_1) & \cos(\theta_1)\cdot(L3\cdot\cos(\theta_2+\theta_3)+L2\cdot\cos(\theta_2)) \\ \sin(\theta_2+\theta_3) & \cos(\theta_2+\theta_3) & 0 & L1+L3\cdot\sin(\theta_2+\theta_3)+L2\cdot\sin(\theta_2) \\ \hline 0 & 0 & 0 & 1 \end{array}\right) \quad \blacksquare
$$

I carried out this solution using the Symbolic Math Toolbox in Matlab, with the following code.

```
syms th1 th2 th3 L1 L2 L3
t1=[0,0,1,0;1,0,0,0;0,1,0,L1;0,0,0,1];
t2=[cos(th1),0,sin(th1),0;0,1,0,0;-sin(th1),0,cos(th1),0;0,0,0,1];
t3=[cos(th2),-sin(th2),0,0;sin(th2),cos(th2),0,0;0,0,1,0;0,0,0,1];
t4=[1,0,0,L2;0,1,0,0;0,0,1,0;0,0,0,1];
t5=[cos(th3),-sin(th3),0,0;sin(th3),cos(th3),0,0;0,0,1,0;0,0,0,1];
t6=[1,0,0,L3;0,1,0,0;0,0,1,0;0,0,0,1];
simplify(t1*t2*t3*t4*t5*t6)
```

4.b RRP Manipulator

The transformation up to the joint 3 is the same as in part (a). The final joint is prismatic rather than revolute. Here L_3 varies from 0 to some maximum. We will assume the variable θ_3 fills the role of this varying L_3 in this case. We also want the final coordinate frame of this prismatic joint to have $\hat{z}_3$ pointing forwards, along the direction of motion, with $\hat{x}_3$ pointing “upwards”, using the example set of frames for (a) as comparison. Thus,

$$
{}^0T_3 = \left(\begin{array}{ccc|c} 0 & 0 & 1 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & L_1 \\ \hline 0 & 0 & 0 & 1 \end{array}\right) \cdot \left(\begin{array}{ccc|c} \cos\theta_1 & 0 & \sin\theta_1 & 0 \\ 0 & 1 & 0 & 0 \\ -\sin\theta_1 & 0 & \cos\theta_1 & 0 \\ \hline 0 & 0 & 0 & 1 \end{array}\right) \cdot \left(\begin{array}{ccc|c} \cos\theta_2 & -\sin\theta_2 & 0 & 0 \\ \sin\theta_2 & \cos\theta_2 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ \hline 0 & 0 & 0 & 1 \end{array}\right)
$$

$$
\cdot \left(\begin{array}{ccc|c} 1 & 0 & 0 & L_2 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ \hline 0 & 0 & 0 & 1 \end{array}\right) \cdot \left(\begin{array}{ccc|c} 0 & 0 & 1 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ \hline 0 & 0 & 0 & 1 \end{array}\right) \cdot \left(\begin{array}{ccc|c} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & \theta_3 \\ \hline 0 & 0 & 0 & 1 \end{array}\right)
$$

$$
= \left(\begin{array}{ccc|c} -\cos(\theta_2)\cdot\sin(\theta_1) & \sin(\theta_1)\cdot\sin(\theta_2) & \cos(\theta_1) & -\cos(\theta_2)\cdot\sin(\theta_1)\cdot(L_2+\theta_3) \\ \cos(\theta_1)\cdot\cos(\theta_2) & -\cos(\theta_1)\cdot\sin(\theta_2) & \sin(\theta_1) & \cos(\theta_1)\cdot\cos(\theta_2)\cdot(L_2+\theta_3) \\ \sin(\theta_2) & \cos(\theta_2) & 0 & L_1 + L_2\cdot\sin(\theta_2) + \theta_3\cdot\sin(\theta_2) \\ \hline 0 & 0 & 0 & 1 \end{array}\right). \quad \blacksquare
$$

I similarly computed this final solution using Matlab, with the following code.

```
syms th1 th2 th3 L1 L2
t1=[0,0,1,0;1,0,0,0;0,1,0,L1;0,0,0,1];
t2=[cos(th1),0,sin(th1),0;0,1,0,0;-sin(th1),0,cos(th1),0;0,0,0,1];
t3=[cos(th2),-sin(th2),0,0;sin(th2),cos(th2),0,0;0,0,1,0;0,0,0,1];
t4=[1,0,0,L2;0,1,0,0;0,0,1,0;0,0,0,1];
t5=[0,0,1,0;1,0,0,0;0,1,0,0;0,0,0,1];
t6=[1,0,0,0;0,1,0,0;0,0,1,th3;0,0,0,1];
simplify(t1*t2*t3*t4*t5*t6)
```

4.c Verify using Robotics Toolbox in Matlab

I wrote the following code to create an arm matching 4(a). I plotted it (Figure 3) to verify its zero position and its movements, and then found its forwards kinematics map, which exactly matches my result in part 4(a).

```
syms th1 th2 th3 L1 L2 L3
% create the arm in 4-a.
%L1 = 1; L2 = 1; L3 = 1; %needs actual values to plot
L(1) = Revolute('d', L1, 'a', 0, 'alpha', pi/2);
L(2) = Revolute('d', 0, 'a', L2, 'alpha', 0);
L(3) = Revolute('d', 0, 'a', L3, 'alpha', 0);
arm_a = SerialLink(L, 'name', 'arm_a');
%arm_a.plot([pi/4 0 pi/4])
fk = arm_a.fkine([th1 th2 th3]);
simplify(fk);
% for some reason this gives huge answers with factors of
% 8112963841460668663546605165575 /
% 16225927682921336339157801028128
% that it won't simplify on its own even though it equals 1/2.
% thus I have to do it manually.
fk_a = [0.5*cos(th1+th2+th3)+0.5*cos(th2-th1+th3),
        -0.5*sin(th1+th2+th3)+0.5*sin(th2-th1+th3), sin(th1),
        0.5*L3*cos(th2-th1+th3)+0.5*L2*cos(th1+th2)+0.5*L2*cos(th1-th2)+0.5*L3*cos(th1+th2+th3);
        0.5*sin(th1+th2+th3)-0.5*sin(th2-th1+th3),
        0.5*cos(th1+th2+th3)-0.5*cos(th2-th1+th3), -1*cos(th1),
        0.5*L2*sin(th1+th2)-0.5*L3*sin(th2-th1+th3)+0.5*L2*sin(th1-th2)+0.5*L3*sin(th1+th2+th3);
        sin(th2+th3), cos(th2+th3), 0, L1 + L3*sin(th2+th3) + L2*sin(th2);
        0,0,0,1];
simplify(fk_a)
```

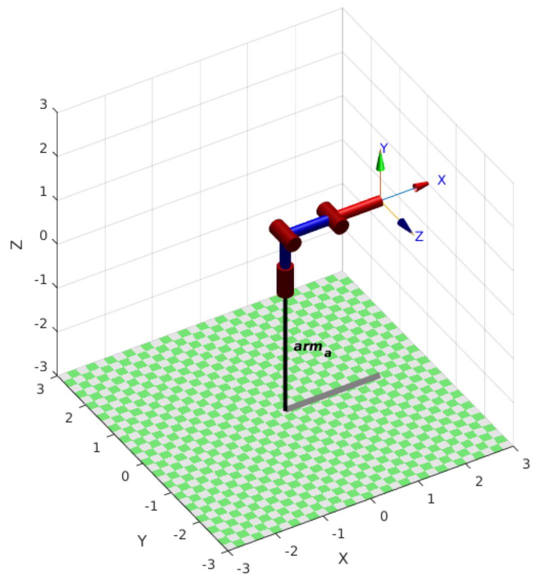


Figure 2: Arm for Q4(a) created as a SerialLink object, and shown at its ‘zero’ configuration.

To create a representation for the arm in part (b), I switched to the `ETS3` package, since it was not possible to specify the set of joints and frames I needed with the DH parameters only. This code is shown below, and the resulting arm is in Figure 3.

```
syms th1 th2 th3 L1 L2
import ETS3.*
%L1 = 1; L2 = 1; th3 = 1;
% first two joints
E3 = Rz(pi/2) * Tz(L1) * Rz('q1') * Ry(pi/2) * Rx(pi) * Rz(pi/2) * Rz('q2') * Tx(L2);
% prismatic joint
E3 = E3 * Ry(pi/2) * Rz(pi/2) * Tz(th3);
%E3.teach() %plots the arm. must uncomment line above to set variables to values.
simplify(E3.fkine([th1 th2])) %find fwd kin in general.
```

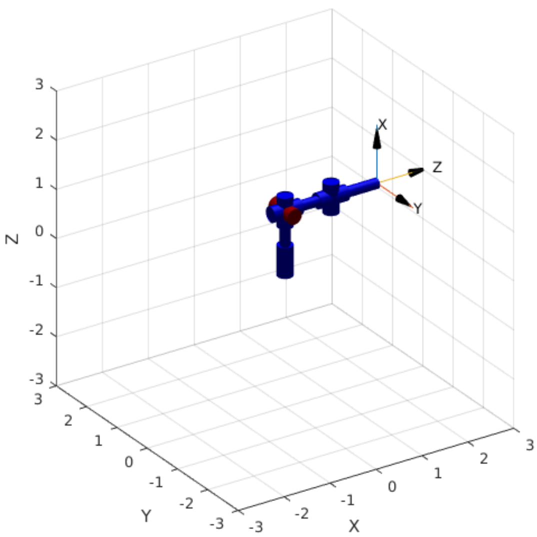
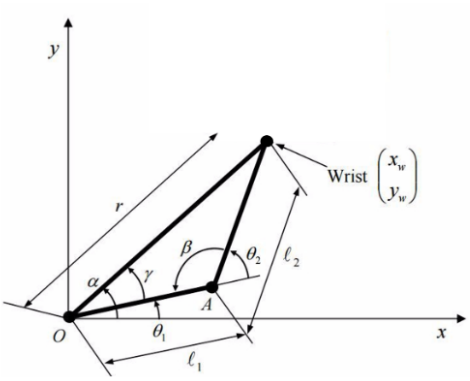
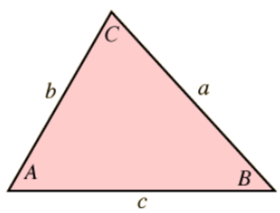


Figure 3: Arm for Q4(b) created as an ETS3 object, and shown when the first two joint angles are zero, and the prismatic joint is extended 1 meter.

5 2-DOF Planar Arm - Inverse Kinematics



$$c^2 = a^2 + b^2 - 2ab\cos C$$



We want to provide a desired wrist position $\begin{bmatrix} x_w & y_w \end{bmatrix}^T$ and find a joint configuration $\begin{bmatrix} \theta_1 & \theta_2 \end{bmatrix}^T$ to achieve this position. We will derive a geometric solution using α, β, and γ.

5.a

Using the law of cosines, we can see that

$$r^2 = l_1^2 + l_2^2 - 2l_1 l_2 \cos\beta$$
$$\Rightarrow \cos\beta = \frac{l_1^2 + l_2^2 - r^2}{2l_1 l_2}$$

5.b

Since $\beta + \theta_2 = \pi$,

$$\theta_2 = \pi - \arccos\left(\frac{l_1^2 + l_2^2 - r^2}{2l_1 l_2}\right)$$

5.c

We can imagine creating triangles using the shared wrist point and angle-hypotenuse pairs (γ, r) and (θ_2, l_2), with equal opposite side. This gives us the relationship

$$r \sin\gamma = l_2 \sin\theta_2$$
$$\Rightarrow \gamma = \arcsin\left(\frac{l_2}{r}\sin\theta_2\right)$$

5.d

We can imagine one triangle with hypotenuse r and angle α from the x-axis to write an expression for α.

$$\tan\alpha = \frac{y_w}{x_w}$$

We know $\alpha = \theta_1 + \gamma$, so

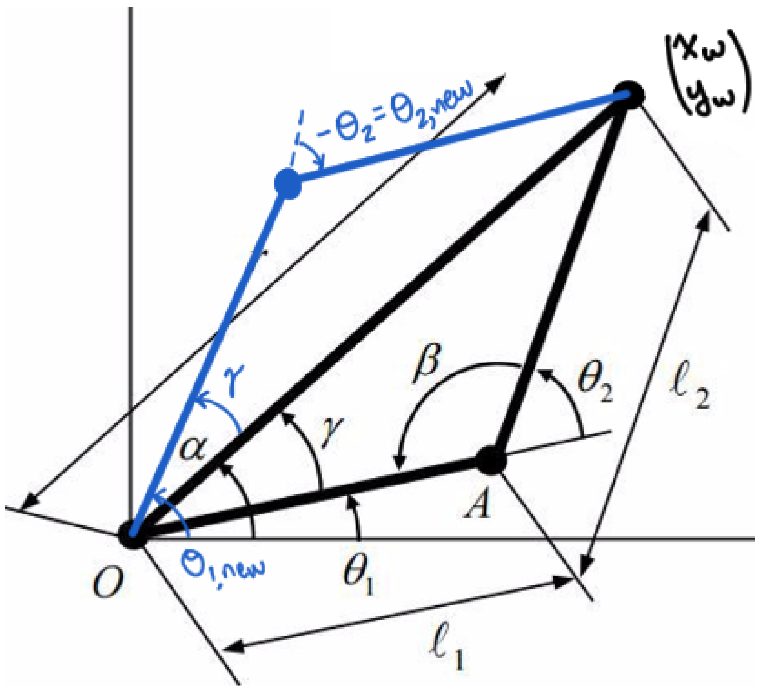
$$\theta_1 = \alpha - \gamma = \arctan\left(\frac{y_w}{x_w}\right) - \arcsin\left(\frac{l_2}{r}\sin\theta_2\right)$$

Thus we have our solution for finding both needed joint angles for a given wrist position.

5.e

There will be exactly 1 solution when the distance of the wrist from the origin is exactly $r = l_1 + l_2$, the maximum distance the arm can reach. This is because the arm will be straight. There will be 2 solutions otherwise (if the point is reachable), since the arm could be reflected across the line labeled "r" in the diagram, meaning two different sets of joint angles will reach the same position. There will be 0 solutions if the desired point is not in the manipulator's work area; this will be the case if the point is further than $(l_1 + l_2)$ away from the origin, or if it is closer than $|l_1 - l_2|$. Additionally, there could be further limitations to the work area if either of the joint angles has only a limited range of motion.

For this other solution, the joint angles take the form $\theta_1|new = \alpha + \gamma$ and $\theta_2|new = -\theta_2$. This is shown in Figure 5.e.



6 Programming Section

My code for the programming section is in the accompanying `.zip` file.