Homework 1: Rotations and Reference Frames

EECS/ME/BioE C106A/206A Introduction to Robotics

Fall 2023

Note: This problem set includes a programming component. Your deliverables for this assignment are:

1. A PDF file submitted to the `HW1 (pdf)` Gradescope assignment with all your work and solutions to the written problems.
2. The provided `hw1.py` file submitted to the `HW1 (code)` Gradescope assignment with your implementation to the programming components.

Problem 1: Properties of Rotations

State whether each transformation matrix below is a valid rotation. Justify your answer.

(a) $\begin{bmatrix} 1 & 1 \\ 0 & 1 \end{bmatrix}$

(b) $\begin{bmatrix} -1 & 0 \\ 0 & -1 \end{bmatrix}$

(c) $\begin{bmatrix} 0 & 1 \\ -1 & 0 \end{bmatrix}$

(d) $\begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix}$

Problem 2: Euler Angles

Consider two initially coincident reference frames, A and B. Frame B is then rotated about the Z axis by $\pi/4$ radians.

a) Sketch the coordinate frames A and B after the rotation.

b) Write the rotation matrix R_{AB} that will take a point from the B frame and represent it in the A frame.

c) Write the rotation matrix R_{BA}.

d) What are the coordinates in frame A of a point with coordinates $p_B = [0, 0, 1]^T$ given with respect to frame B?

e) What are the coordinates in frame B of a point with coordinates $p_A = [1, 1, 0]^T$ given with respect to frame A?

Problem 3: Multiple Euler Angles

(a) A frame is rotated first about the Z axis by angle $\frac{\pi}{2}$, then about the mobile Y axis by an angle of $\frac{\pi}{2}$, then about the mobile X axis an angle of $\frac{\pi}{2}$. *Hint: This is an intrinsic rotation.*

  (i) Draw the frame before and after the rotation. Label all axes.

  (ii) Write the net rotation matrix.

(b) A frame is rotated first about the Z axis by angle $\frac{\pi}{2}$, then about the original Y axis by an angle of $\frac{\pi}{2}$, then about the original X axis an angle of $\frac{\pi}{2}$. *Hint: This is an extrinsic rotation.*

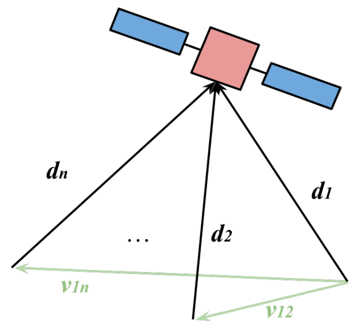
  (i) Draw the frame before and after the rotation. Label all axes.

  (ii) Write the net rotation matrix.

Problem 4: Save the Satellite!

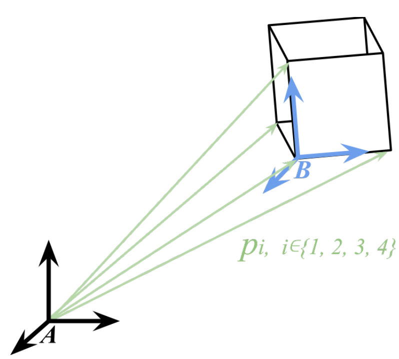
Your autonomous space startup, SpaceY, has just launched its first satellite! Everything is going great when all of a sudden, it loses orientation data and begins spinning wildly out of control. Let's see what we can do to recover our satellite's orientation data and put our mission back on track.

(a) Although our onboard sensors have failed, we still have a set of n sensors on the ground that we can use to help locate our satellite. The only problem is - each sensor i only reports a scalar distance d_i to the satellite rather than an XYZ position.



Find an expression for the XYZ position of the satellite in terms of $d_1, ..., d_n$ and $v_{12}, ..., v_{1n} \in \mathbb{R}^3$, the vectors pointing from Sensor 1 to the other n-1 sensors. Your solution should assume that each sensor points in a different direction and that no three sensors are collinear. It should be specified in the frame of Sensor 1. *Hint: How can we use least-squares to help solve this problem?*

(b) Using our solution above, our tracking station, which has an orthonormal frame A, has picked up the XYZ positions of four points $p_1,\ p_2,\ p_3,\ p_4 \in \mathbb{R}^3$ on the satellite's body.



Choosing p_1 as the origin of the body frame B, find an expression for three vectors $v_{b1}, v_{b2}, v_{b3} \in \mathbb{R}^3$ with respect to frame A that form an orthonormal basis on the body of the satellite. You may leave your answer in terms of p_1, p_2, p_3, p_4. *Hint: What procedure from linear algebra can we use to make a set orthonormal?*

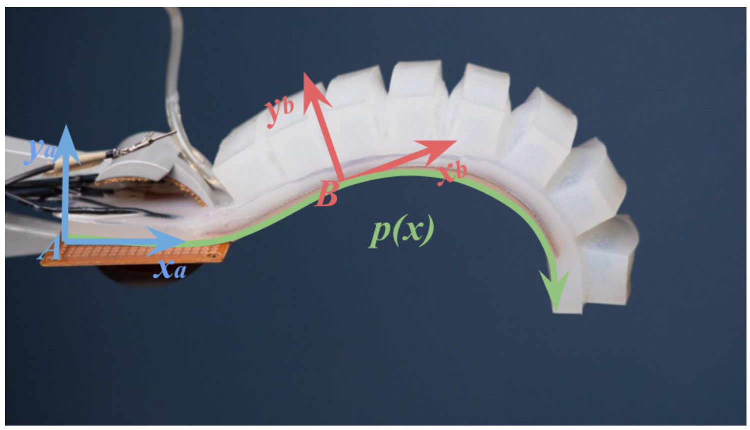
(c) Find the rotation matrix R_{ab} from the body frame B of the satellite to the base frame A in terms of v_{b1}, v_{b2}, v_{b3}. Congratulations, you've successfully recovered the orientation data of your satellite!

Problem 5: Squishy Rotations

Soft robotics is concerned with the design, control, and modelling of squishy, rather than rigid, robots! Using soft robotic grippers, we can easily pick up objects, move around, and more without the danger of having a rigid robot arm swinging around! Let's see how we can apply the concepts of rotations to study the motion of soft robots.

1. As opposed to rigid body robots, soft robots *deform*, and bend as if they have an infinite number of degrees of freedom. Let's model the xy profile of a soft robot using an n^{th} order polynomial:

$$y = f(x) = a_n x^n + a_{n-1} x^{n-1} + ... + a_1 x + a_0 = \sum_{k=0}^{n} a_k x^k \tag{1}$$



Let's define a coordinate frame A with $\boldsymbol{x_A}$ pointing right, $\boldsymbol{y_A}$ pointing up, and $\boldsymbol{z_A}$ pointing out of the page (shown in blue in the image above). Using $f(x)$, we can write the XYZ coordinates of the set of positions along the soft robot with respect to frame A as:

$$\boldsymbol{p(x)} = \begin{bmatrix} x \\ \sum_{k=0}^{n} a_k x^k \\ 0 \end{bmatrix} \in \mathbb{R}^3 \tag{2}$$

Calculate $\boldsymbol{t(x)} \in \mathbb{R}^3$, the unit tangent vector to $\boldsymbol{p(x)}$ at position x in terms of a_i and x. Your answer should be specified with respect to frame A.

2. Let's define a frame B in terms of the A frame along the length of the soft robot. If we define $\boldsymbol{x_B} = \boldsymbol{t(x)}$ and $\boldsymbol{z_B}$ to be the vector pointing out of the page, calculate the rotation matrix R_{AB} from frame B to frame A. This rotation matrix allows us to specify the orientation of our soft gripper at all points along its length! *Hint: All coordinate frames must be right-handed.*

3. Say we have found $f(x) = 3x^3 + 4x$. Calculate the rotation matrix at $x = 1$.

Problem 6: Rotation Matrices in Action

Tarun wants to plot the motion of a car moving in 2D space for his next research paper, but there's one problem! His plotting code doesn't take the car's orientation into consideration when plotting, so nothing looks right (figure 1).

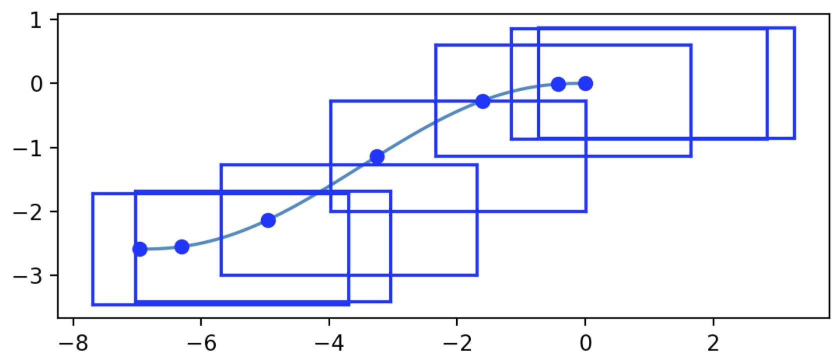


Figure 1: Cars can't move like this!

Edit the `hw1.py` to take orientation into account when plotting the car, and create a visualization by running `car_vis.py`. Once you are satisfied with your visualization, submit only `hw1.py` to the `HW1 (code)` assignment on Gradescope.