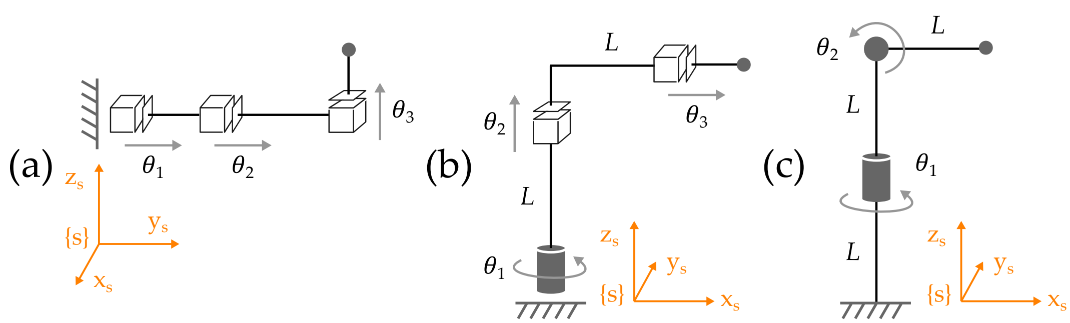
Problem Set 5

Robotics & Automation
Dylan Losey, Virginia Tech

Instructions. Please write legibly and do not attempt to fit your work into the smallest space possible. It is important to show all work, but basic arithmetic can be omitted. You are encouraged to use Matlab when possible to avoid hand calculations, but print and submit your commented code for non-trivial calculations. You can attach a pdf of your code to the homework, use live scripts or the publish feature in Matlab, or include a snapshot of your code. Do not submit .m files — we will not open or grade these files.

For this assignment we are asking you to also submit **videos** of your simulations. Follow the instructions to **label** these videos based on the problem number, and then submit them all within a **single zipped folder**.

1 Analytical Inverse Kinematics



You are considering purchasing an industrial robot arm. Before you buy, you want to find the inverse kinematics of the available options. For each of the robots shown above find equations for θ in terms of $p = [x, y, z]^T$, the position of the end-effector in the world frame.

1.1 (5 points)

Find the inverse kinematics of robot (a).

1.2 (10 points)

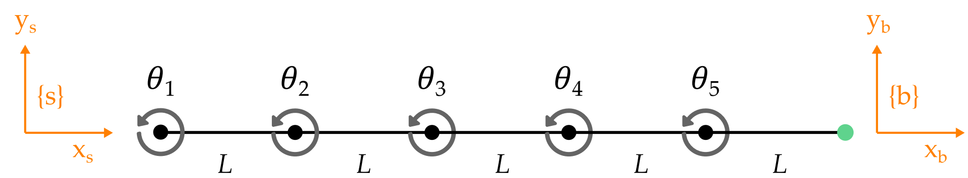
Find the inverse kinematics of robot (b).

1.3 (15 points)

Find the inverse kinematics of robot (c).

1.4 (5 points)

Imagine that the end-effector of robot (a) is at position $p = [0, 2L, 0]^T$. How many inverse kinematics solutions are there at this position?



2 Numerical Inverse Kinematics

You are building a snake robot. This snake robot moves in a plane and has 5 joints, making it a redundant robot. You are using this redundancy to mimic the motion of real snakes.

2.1 (25 points)

Implement the numerical inverse kinematics algorithm. Leave $b = 0$ within the Jacobian pseudoinverse. Using your code, find the inverse kinematics solutions when:

Case 1 $L = 1$ and the desired end-effector pose is:

$$T_{sb} = \begin{bmatrix} rotz(\pi/4) & \begin{bmatrix} 3 \\ 2 \\ 0 \end{bmatrix} \\ 0 & 1 \end{bmatrix} \tag{1}$$

Case 2 $L = 1$ and the desired end-effector pose is:

$$T_{sb} = \begin{bmatrix} rotz(\pi/2) & \begin{bmatrix} -2 \\ 4 \\ 0 \end{bmatrix} \\ 0 & 1 \end{bmatrix} \tag{2}$$

Case 3 $L = 1$ and the desired end-effector pose is:

$$T_{sb} = \begin{bmatrix} rotz(0) & \begin{bmatrix} 3 \\ -1 \\ 0 \end{bmatrix} \\ 0 & 1 \end{bmatrix} \tag{3}$$

For each of these cases let your initial joint position be $\theta_0 = [\pi/8, \pi/8, \pi/8, \pi/8, \pi/8]^T$.

3 Simulating the Robot

This problem uses the same redundant snake robot as in Problem 2. Here you will make videos of the robot to visualize your inverse kinematics solutions. Start by **downloading** the Matlab file `make_video.m` that was provided with this assignment.

3.1 (25 points)

Combine `make_video.m` with your numerical inverse kinematics code. You will need to get the (x, y) position of each link (in addition to the end-effector position) to plot the snake robot (see example at top of next page). Turn in the following MP4 videos:

- Video for **Case 1** in Problem 2. Title this video **Problem3_1.mp4**
- Video for **Case 2** in Problem 2. Title this video **Problem3_2.mp4**
- Video for **Case 3** in Problem 2. Title this video **Problem3_3.mp4**

Please adjust the video frame rate (`v.FrameRate = your_rate_here;`) so that each video is a few seconds long (more than 3 secs, less than 10 secs).

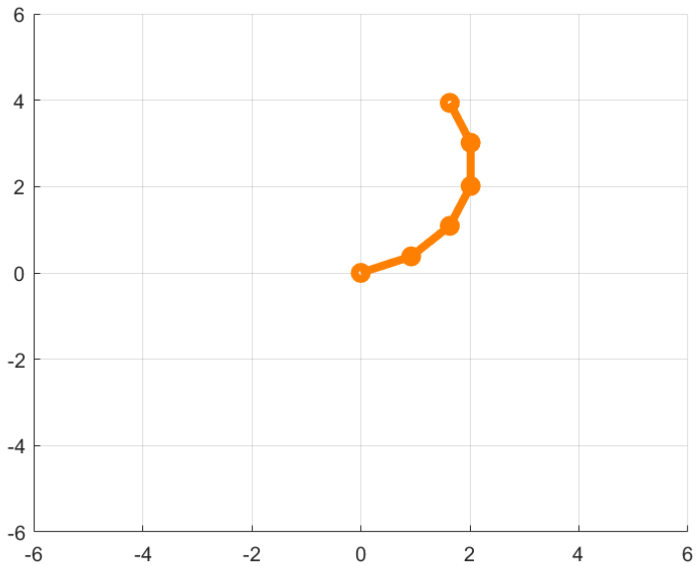


Figure 1: Plot of the snake robot in its initial position $\theta = [\pi/8, \pi/8, \pi/8, \pi/8, \pi/8]^T$

4 Jacobian Pseudoinverse and Redundancy

This problem continues exploring the redundant snake robot introduced in Problem 2. So far we have left $b = 0$ in our Jacobian pseudoinverse. More generally, choosing b allows us to set a **secondary objective** for the inverse kinematics of redundant robots. Recall that numerical inverse kinematics finds a solution for θ such that $T_{sb}(\theta)$ equals the desired end-effector pose. But when working with redundant robots, multiple solutions are often possible. Choosing b affects which of these solutions the algorithm selects.

4.1 (15 points)

Set b as the following vector (and update b as θ_1 changes):

$$b = \begin{bmatrix} -\theta_1 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix} \tag{4}$$

This choice of b **minimizes the first joint angle**. In other words, the robot will look for an inverse kinematics solution where $|\theta_1| \to 0$. Show that this actually works:

- Make a video for **Case 3** in Problem 2 with b chosen as Equation 4. Title this video **Problem4.mp4**
- Compare the final joint position θ for **Case 3** when $b = 0$ and when b is Equation 4. Show that $|\theta_1|$ is smaller with the secondary objective.
- State one reason why you might want to minimize the first joint angle (or any joint angle). What is a practical benefit?