Assignment 3: Inverse Differential Kinematics

Robot Kinematics and Dynamics

Prof. Howie Choset

Contents

1 Overview

This assignment reinforces the following topics:

- Jacobians
- Psuedoinverses

2 Background

2.1 The Inverse Problem

So far we have studied the functions that perform forward kinematics, which means going from a manipulator's configuration space, Q to its workspace, W:

$$\mathbf{W} = f(\mathbf{Q})$$

In other words, we have determined the function for a robot arm that uses its joint angles to calculate the position and orientation of its end effector (as well as other frames on the robot).

You can probably think of a lot of cases where we want to do the inverse: given a specified effector position (and potentially orientation), what are the joint angles that will achieve this? This inverse process is problematic because the forward kinematics function, f, is (usually) non-linear. That means that we (usually) cannot just say:

$$\mathbf{Q} = f^{-1}(\mathbf{W})$$

This means that finding the inverse function for the kinematics of an arm is difficult (but by no means impossible – in the next assignment we will tackle several methods to do this).

2.2 Differential IK

While the general problem of going from workspace positions to joint angles is difficult, we can begin with the *differential* problem – going from workspace velocities to configuration space velocities.

$$\dot{\mathbf{Q}} = f^{-1}(\dot{\mathbf{W}})$$

From the last assignment, you know that the function relating these two sets of velocities is the Jacobian, $\mathbf{J}$. A key property of the Jacobian is that it is linear (given the configuration $\mathbf{Q}$ of the arm). In practical terms, this means that if we know an arm's joint angles and desired instantaneous end-effector velocities, we can calculate the required joint velocities by inverting the Jacobian:

$$\dot{\mathbf{Q}} = J^{-1}(\mathbf{Q})\dot{\mathbf{W}}$$

Because (for a given configuration) the Jacobian can be represented as a matrix, this is a much easier problem than the general IK problem we began with. It doesn't directly give us the joint angles for a given end effector position, but it can be used to help solve for these, as we'll see in the assignment.

2.3 Pseudoinverse

But wait – although inverting matrices is easy for computers, not all matrices (and not all Jacobians) are invertible! In order for a matrix to be invertible, it has to be square, and have a non-zero determinant. In terms of a robot arm this means that it has to be:

- Fully constrained (the same number of workspace and configuration space variables)
- Non-singular (the arm's instantaneous motion can span the dimension of the workspace, meaning J is full rank)

Fortunately, mathematicians have developed a technique called the *pseudoinverse* to help get around the problem of not having a square or full rank matrix. The pseudoinverse has some, but not all, of the properties of a true matrix inverse. The general method to compute this quantity involves a technique called *singular value decomposition*, but when the matrix is *full rank*, there are analytical formulas that can be used to find the inverse.

In this case, we basically make the matrix square by multiplying it by its transpose. We then invert the new matrix, and multiply the result by the transpose of the original matrix to get the dimensions to match up. The exact order of multiplication depends on the dimension of the matrix:

2.3.1 Underconstrained

To determine the joint velocities that result in a given a workspace velocity, we need to solve a system of linear equations (given a joint configuration). If there are n joint variables and m workspace variables, then the Jacobian has dimensions m x n. Consider the case where $m < n$ i.e. there are more joints than the number of workspace variables. This results in a system of m equations (i.e. constraints) in n unknowns (i.e. joint velocities) where $m < n$, hence the name underconstrained.

Assuming the Jacobian is full rank, such a system always has a solution. This means we can always find joint velocities that result in a given end effector velocity. However, there is a whole family of solutions that satisfy these constraints. Therefore, one might be interested in searching the best solution in this family, measured with respect to some criterion. For the purpose of this discussion, we are interested in solutions (joint velocities) which result in the smallest effort in moving the arm. We can quantify this effort using the squared norm of the joint velocities.

Posing this as an optimization problem, we want to find $\dot{\mathbf{Q}}$ such that we minimize $\dot{\mathbf{Q}}^T\dot{\mathbf{Q}}$ satisfying $\dot{\mathbf{W}} = J(\mathbf{Q})\dot{\mathbf{Q}}$ where $\dot{\mathbf{W}}$ and $J(\mathbf{Q})$ are known.

This optimization problem is equivalent to a least-norm problem for linear systems of equations and has an exact analytical solution given by:

$$\dot{\mathbf{Q}} = J^+\dot{\mathbf{W}} \tag{1}$$

where J^+ is known as the right pseudoinverse of the Jacobian and is given by:

$$J^+ = J^T(JJ^T)^{-1}$$

2.3.2 Overconstrained

This case is when the number of rows the Jacobian has is greater than (or equal to) the number of columns. In other words, there are more equations than unknowns.

In this case, the arm cannot move the end effector with the exact desired velocity, but this solution will try to solve for the "best" joint velocities in terms of the sum-squared error of end effector velocities. This is called the left puesdoinverse.

$$J^\# = (J^TJ)^{-1}J^T$$

As a side note, the pseudoinverse in this overconstrained case is really useful even outside of kinematics, and is used to quickly and reliably do things like solve for best-fit lines to data.

2.3.3 Perfectly Constrained

Note that in the case of a square full rank matrix, the psuedoinverse will result in $\mathbf{J}^{-1}$.

2.3.4 Practical Usage

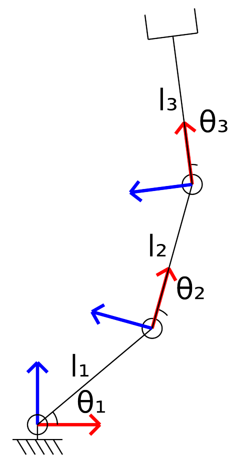
When using the psuedoinverse for real-world applications, languages such as MATLAB have a function to compute the general psuedoinverse of a matrix (i.e., one that works regardless of matrix rank or dimension). In MATLAB, this single command is `pinv`. Though, it is still useful to understand what the pseudoinverse conceptually gives you in each of these cases. – e.g., a best fit or lowest effort solution. It is also important to understand the limitations that apply when the system is in a singular configuration.

3 Written Questions

For all of the following questions, please fully compute all answers and show your work for full credit, unless otherwise specified. Submit your answers in a PDF entitled `writeup.pdf` in your handin directory. All answers must be typed, but diagrams may be hand-drawn and scanned in. However, they must be tidy and fully legible! Consider drawing them in a black or blue pen. All units are in radians and meters, where appropriate.

1) RRR Robot

Consider the following robot



Assume $\begin{bmatrix} \theta_1 \\ \theta_2 \\ \theta_3 \end{bmatrix} = \begin{bmatrix} \frac{\pi}{2} \\ \frac{\pi}{3} \\ \frac{\pi}{3} \end{bmatrix}$.

(1) [1 point] What is the dimension of the end-effector Jacobian? Consider end effector positions $\left[\begin{smallmatrix} x \\ y \end{smallmatrix}\right]$ in $\mathbb{R}^2$.

(2) [1 point] Is this system underconstrained, overconstrained, or neither? Why?

(3) [6 points] On separate plots, draw the velocity vector for the end effector caused by
- Joint 1 moving with unit velocity (joints 2 and 3 have zero velocity).
- Joint 2 moving with unit velocity (joints 1 and 3 have zero velocity).
- Joint 3 moving with unit velocity (joints 1 and 2 have zero velocity).

The direction of these vectors should be correct, and the relative magnitudes should be approximately correct, but do not worry about exact magnitudes. You can complete this problem by inspection or by using properties of the Jacobian.

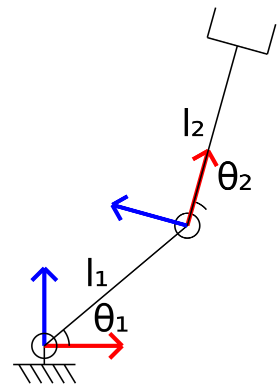
(4) [2 points] What is the dimension of the space spanned by these vectors? Are the 1st and 2nd vectors linearly independent? The 1st and 3rd? The 2nd and 3rd? All three?

(5) [1 point] What is the rank of J for this configuration?

(6) [1 point] When identifying the inverse differential kinematics for this problem, would you use the inverse, the left psuedoinverse, or the right psuedoinverse?

2) RR Robot

Consider the following robot



Assume $\begin{bmatrix} \theta_1 \\ \theta_2 \end{bmatrix} = \begin{bmatrix} \frac{\pi}{4} \\ \frac{\pi}{4} \end{bmatrix}$.

(1) [1 point] What is the dimension of the end-effector Jacobian? Consider end effector positions $\begin{bmatrix} x \\ y \end{bmatrix}$ in $\mathbb{R}^2$.

(2) [1 point] Is this system underconstrained, overconstrained, or neither? Why?

(3) [4 points] On separate plots, draw the velocity vector for the end effector caused by
- Joint 1 moving with unit velocity (joints 2 has zero velocity).
- Joint 2 moving with unit velocity (joints 1 has zero velocity).

The direction of these vectors should be correct, and the relative magnitudes should be approximately correct, but do not worry about exact magnitudes. You can complete this problem by inspection or by using properties of the Jacobian.

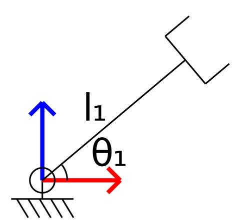
(4) [2 points] What is the dimension of the space spanned by these vectors? Are the 1st and 2nd vectors linearly independent?

(5) [1 point] What is the rank of J for this configuration?

(6) [1 point] When identifying the inverse differential kinematics for this problem, would you use the inverse, the left psuedoinverse, or the right psuedoinverse?

3) R Robot

Consider the following robot



Assume $\theta_1 = \frac{\pi}{4}$.

(1) [1 point] What is the dimension of the end-effector Jacobian? Consider end effector positions$\begin{bmatrix} x \\ y \end{bmatrix}$ in $\mathbb{R}^2$.

(2) [1 point] Is this system underconstrained, overconstrained, or neither? Why?

(3) [2 points] Draw the velocity vector for the end effector caused by joint 1 moving with unit velocity. The direction of these vectors should be correct, but do not worry about the magnitude. You can complete this problem by inspection or by using properties of the Jacobian.

(4) [2 points] What is the dimension of the space spanned by this vector?

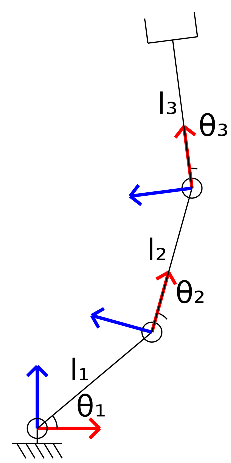
(5) [1 point] What is the rank of J for this configuration?

(6) [1 point] When identifying the inverse differential kinematics for this problem, would you use the inverse, the left psuedoinverse, or the right psuedoinverse?

(7) [3 points] What are the benefits of a left pseudoinverse versus a right pseudoinverse?

4) RRR Robot, revisited with rotation

Consider the following robot



(1) [1 point] What is the dimension of the end-effector Jacobian, now considering $\begin{bmatrix} x \\ y \\ \theta \end{bmatrix}$ end effector positions in SE(2).

(2) [1 point] Is this system underconstrained, overconstrained, or neither? Why?

(3) [1 point] Write out the end-effector Jacobian.

(4) [3 points] Assume $\theta_1 = 0.6$, $\theta_2 = 0.6$, and $\theta_3 = 0.4$ (all values in radians). Write out the end effector velocity $\begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{\theta} \end{bmatrix}$ for each of:

- Joint 1 moving with unit velocity (joints 2 and 3 have zero velocity).
- Joint 2 moving with unit velocity (joints 1 and 3 have zero velocity).
- Joint 3 moving with unit velocity (joints 1 and 2 have zero velocity).

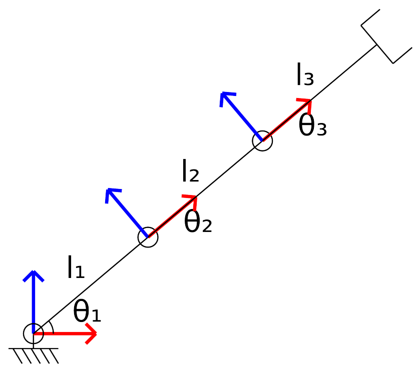
(5) [2 points] What is the dimension of the space spanned by these vectors? Are the 1st and 2nd vectors linearly independent? The 1st and 3rd? The 2nd and 3rd? All three?

(6) [1 point] What is the rank of J for this configuration?

(7) [2 points] When identifying the inverse differential kinematics for this problem, would you use the inverse, the left psuedoinverse, or the right psuedoinverse?

5) RRR Singular Configuration

Consider the following robot



(1) [1 point] What is the dimension and rank of the end-effector Jacobian? Consider end effector positions $\left[\begin{smallmatrix} x \\ y \end{smallmatrix}\right]$ in $\mathbb{R}^2$. Do the dimension and rank differ? why?

(2) [6 points] On separate plots, draw the velocity vector for the end effector caused by
- Joint 1 moving with unit velocity (joints 2 and 3 have zero velocity).
- Joint 2 moving with unit velocity (joints 1 and 3 have zero velocity).
- Joint 3 moving with unit velocity (joints 1 and 2 have zero velocity).

The direction of these vectors should be correct, and the relative magnitudes should be approximately correct, but do not worry about exact magnitudes. You can complete this problem by inspection or by using properties of the Jacobian.

(3) [2 points] What is the dimension of the space spanned by these vectors? Are the 1st and 2nd vectors linearly independent? The 1st and 3rd? The 2nd and 3rd? All three?

(4) [1 point] What is the rank of J for this configuration?

6) Local Inverse Kinematics

Consider a general arm with revolute joints. Assume you already have the kinematic map to the end effector, f, and the Jacobian of the end effector, J.

The robot arm begins with joint angles Θ_0, resulting in an end effector position of $\mathbf{p_0} = f(\Theta_0)$. The end effector needs to pick up an object at point $\mathbf{p_1}$.

Note: this is a conceptual problem: do not consider joint limits, and assume that all relevant positions are within the workspace of the robot. Furthermore, all answers are in terms of the given variables and known constants.

(1) [2 points] What are the initial joint angle velocities $\dot{\Theta}_0$ that start to move the end effector directly towards $\mathbf{p_1}$,?

(2) [3 points] Assume the end effector has moved at $\dot{\Theta}_0$ for a small amount of time dt. What joint angles velocities $\dot{\Theta}_{dt}$ will move the end effector from this new starting point directly towards $\mathbf{p_1}$?

(3) [2 points] Describe (in a couple of sentences) how you would command the robot to move from $\mathbf{p_0}$ to $\mathbf{p_1}$ in approximately 1 second.

Note that this process describes how differential IK can be used to solve IK *numerically*, given a nearby starting point. We will explore this concept in more depth in the next assignment.

4 Feedback

1) Feedback Form *5 points*

We are always looking to improve the class! To that end, we're looking for your feedback on the assignments. When you've completed the assignment, please fill out the feedback form.

5 Code Questions

Copy the Code Handout folder to some location of your choice. Open Matlab and navigate to that location. Whenever you work on the assignment, go into this directory and run `setup.m`.

1) General Robot Class

Open directory `ex_01`.

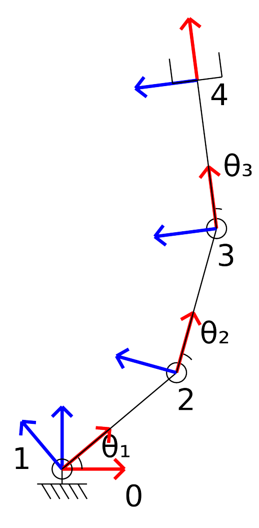
Surely, you don't want to keep writing forward kinematics and jacobians for every single assignment in this class!

For this exercise, we will walk you through the creation of a general MATLAB class that we will continue to use and extend for the remainder of this class. We will begin with writing forward kinematics for any RR arm, and then compute the Jacobian for any RR arm.

First, open the file Robot.m, and study this file. This is a basic MATLAB class, but don't worry, we aren't going to force too much CS on you! The idea here is that you can pass in parameters to the *constructor* of the robot class, and the returned variable then has functions that can be called, and can operate on the values you passed in. For example, the following code

```
r = Robot([l1; l2], [linkmass1; linkmass2], [jointmass1; jointmass2], 0);
frames = r.fk([0; pi/2]);
```

can be used to find the forward kinematics of a robot with link lengths of l1 and l2 for $\theta = \left[\begin{smallmatrix} 0 \\ \frac{\pi}{2} \end{smallmatrix} \right]$



Forward Kinematics Your job is to write forward kinematics that can work for any number of revolute joints in a chain. You will fill in the `forward_kinematics` function in the `Robot.m` file. This computes $\mathbf{H}_\mathrm{i}^0$ for each frame (i > 0) in the figure above, including the end effector frame.

The code describes the conventions used for the resulting variable.

One suggestion that should help is to break this problem into three components:

- First consider frame 1.
- Use a for loop to iterate from frame 2 to $n-1$; use $\mathbf{H^0_{i-1}}$ to help compute frame $\mathbf{H^0_i}$.
- Compute the end effector frame, using $\mathbf{H^0_{n-1}}$ to help.

Verification To verify the results, run the `sample_arm` and `sample_path` functions, and verify that the plots match (as on assignment 1).

6 Submission Checklist

- ☐ Create a PDF of your answers to the written questions with the name `writeup.pdf`.
- ☐ Make sure you have `writeup.pdf` in the same folder as the `create_submission.m` script.
- ☐ Run `create_submission.m` in Matlab.
- ☐ Upload `handin-3.tar.gz` to Canvas.
- ☐ After completing the entire assignment, fill out the feedback form[1].

[1] https://canvas.cmu.edu/courses/6365/quizzes/14004