Lab 2: Inverse Kinematics for the Lynx

MEAM 520, University of Pennsylvania

September 23, 2020

This lab consists of two portions, with a pre-lab due on **Wednesday, September 30, by midnight (11:59 p.m.)** and a lab (code+report) due on **Wednesday, October 7, by midnight (11:59 p.m.)**. Late submissions will be accepted until midnight on Saturday following the deadline, but they will be penalized by 25% for each partial or full day late. After the late deadline, no further assignments may be submitted; post a private message on Piazza to request an extension if you need one due to a special situation. This assignment is worth 50 points.

You may talk with other students about this assignment, ask the teaching team questions, use a calculator and other tools, and consult outside sources such as the Internet. To help you actually learn the material, what you submit must be your own work, not copied from any other individual or team. Any submissions suspected of violating Penn's Code of Academic Integrity will be reported to the Office of Student Conduct. When you get stuck, post a question on Piazza or go to office hours!

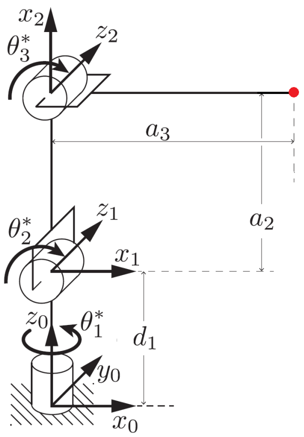
Individual vs. Pair Programming

Work closely with your partner throughout the lab, following these guidelines, which were adapted from "All I really needed to know about pair programming I learned in kindergarten," by Williams and Kessler, *Communications of the ACM*, May 2000. This article is available on Canvas under Files / Resources.

- Start with a good attitude, setting aside any skepticism, and expect to jell with your partner.
- Don't start alone. Arrange a meeting with your partner as soon as you can.
- Use just one setup, and sit side by side. For a programming component, a desktop computer with a large monitor is better than a laptop. Make sure both partners can see the screen.
- At each instant, one partner should be driving (writing, using the mouse/keyboard, moving the robot) while the other is continuously reviewing the work (thinking and making suggestions).
- Change driving/reviewing roles at least every 30 minutes, *even if one partner is much more experienced than the other.* You may want to set a timer to help you remember to switch.
- If you notice an error in the equation or code that your partner is writing, wait until they finish the line to correct them.
- Stay focused and on-task the whole time you are working together.
- Take a break periodically to refresh your perspective.
- Share responsibility for your project; avoid blaming either partner for challenges you run into.
- Recognize that working in pairs usually takes more time than working alone, but it produces better work, deeper learning, and a more positive experience for the participants.

1 Pre-lab Tasks (due September 30, 5 pts)

Directions: For the pre-lab component of this lab, you turn in your own **individual** work. Show your work to receive full credit. These calculations should be typed or written legibly. Submit a **pdf** on Gradescope containing your work.



This lab is about determining the IK equations for the Lynx. As you found in the previous labs, the Lynx is a 6-DOF robot. It has 3 degrees of freedom in the arm, 2 degrees of freedom in the wrist, and 1 degree of freedom in the gripper. The state of the gripper does not affect the location of our end effector frame.

Let's start by analyzing just the arm. The diagram on the right shows the first three links of the robots in the zero position. Positive joint directions are marked with arrows.

1. Write an equation for the position of the red dot at the end of the arm (i.e., the position of the center of the wrist) in frame 0 as a function of the joint variables.

2. Assume this robot has no joint limits and will not collide with itself in any configuration. As a step toward solving this robot's IK, draw a **diagram that shows how many inverse position kinematics solutions exist** in different regions around this robot; the number of solutions might be 0, 1, 2, 3, 4, . . . , ∞. Count the number of unique physical configurations; adding 2π to a joint angle is a trivial modification of a solution, so it does not increase the number of solutions.

3. Given a desired position of the red dot $[x \;\; y \;\; z]^\intercal$ for which at least one solution exists, find **all possible solutions** to this arm's inverse position kinematics. Derive closed-form equations for the joint variables in terms of x, y, and z and any needed robot parameters. Explain your steps - did you take a geometric or algebraic approach?

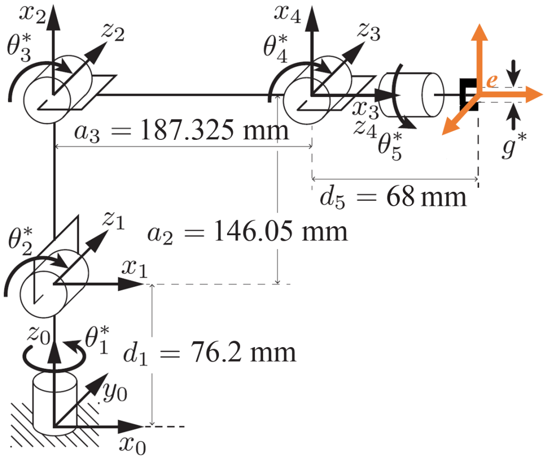
2 Lab (due October 7, 45 pts)

The remainder of the lab should be done with a partner. You may work with anyone you choose, but you must work with them for all parts of this assignment. You will both turn in the same report and code (see Submission Instructions), for which you are jointly responsible and you will both receive the same grade.

2.1 Methods

Include in your report the answers to the pre-lab questions, including any corrections you would like to make since the original submission.

Now, finish computing the inverse kinematics for the robot by adding the wrist back in. Again assume that the wrist has no joint limits and will not collide with itself in any configuration.



1. Say we want the origin of the coordinate frame e attached to the center of the gripper to be in the position $[x \;\; y \;\; z]^\intercal$. Draw a diagram of the set of positions that the center of the wrist can take in order for frame e to have this origin.

2. Now consider if we also specify the orientation of frame e, so that the frame is in the pose represented by the homogeneous transformation

$$T_e^0 = \begin{bmatrix} r_{11} & r_{12} & r_{13} & t_1 \\ r_{21} & r_{22} & r_{23} & t_2 \\ r_{31} & r_{32} & r_{33} & t_3 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

   What position should the center of the wrist take in order for the end effector frame e to be at the provided T_e^0? Explain your steps - did you take a geometric or algebraic approach? What should the joint angles of the wrist be?

3. Remember that the Lynx arm+wrist setup has 5 DOF while a rigid body in 3D space has 6 DOF. There are therefore some orientations that the Lynx end effector is simply not able to reach. Describe what orientations these are and write down a mathematical check that you can execute on the homogeneous transformation matrix T_e^0 to see if it is feasible.

4. Now, consider the case where the orientation of the end effector is infeasible but the target is reachable. We would still like to move the end effector into the correct location. Find the closest feasible transformation $\hat{T}_e^0$ by:

   - Projecting the desired z and y axes into the space of feasible orientations.
   - Computing an x axis to complete the rotation matrix.

   ** Note: This technique does not strictly give the “closest” feasible rotation. That solution would require the use of SVD and is beyond the scope of this class.

2.2 Coding

1. Write a function `[q isPos]=calculateIK(T)` with the following inputs and outputs.

   - Inputs: a homogeneous transformation matrix T_e^0, representing the end-effector frame expressed in the base frame.

- Outputs: 1) an $n \times 5$ matrix `q`, where each row is a set of 5 joint variable values that bring the gripper to this pose, and 2) a flag `isPos` to indicate if a solution is possible.
  (a) If the target transformation is located outside the reachable workspace, `isPos = 0` and `q` should be **empty**.
  (b) If the target transformation is inside the reachable workspace but infeasible, `isPos = 0` and q should contain the solutions for $\hat{T}_e^0$. Recall that you computed this in the third conceptual question for IK.
  (c) If the target transformation is feasible, `q` should be $n \times 5$, where each row is a **unique** possible solution in the range $(-\pi, \pi]$. If one of the joint angles is unconstrained (i.e., it can take any value), place `NaN` as the joint variable value for that joint.
- Add a **joint limit** check to narrow down your possible solutions. You do not need to consider self-collision.

2. The starter code in `Lab2.zip` contains a clean version of the code needed to connect to the simulator. Correct solution code for `calculateFK` will be posted after the Lab 1 late deadline (Sep. 26).

2.3 Evaluation

1. Include any evaluation tests you used to test whether your IK code is working properly.
2. Run the ROS simulator to check that the joint angles outputted by your code produce the expected position and orientation of the gripper end. Use the same procedure you used in Lab 0, except this time use the Lab 2 launch file, i.e.:

   ```
   $roslaunch al5d_gazebo lab2.launch
   ```

3. In addition to whatever tests you want to perform, a few targets are set up around the Lynx robots in the simulation. `TestIK_Sim` contains the homogeneous transformations for the targets. Run your code with Gazebo to have the gripper hit the targets.

   To run the code, follow the steps listed below.

   (a) Uncomment one of the T0e of the targets.
   (b) Run `TestIK_Sim` in a new terminal

   Report your results. Include in your writeup the commands that you executed to perform your tests, and the resulting qs output from `calculateIK`.

2.4 Analysis

1. Comment on any similarities or differences between the results of your IK function and the simulation experiments.
2. In some cases there may be multiple solutions to the IK problem. Consider how you would choose one out of the set for practical implementation. Write down at least 2 issues you would take into account to rule out some IK solutions or rank them.
3. As many of you noted in your Lab 0, the real Lynx robot will have some trouble reaching some positions due to gravity, torque limits in the joints, friction, etc. Explain what specific modifications to your FK/IK equations you might want to make to improve the accuracy of your results.

   Consider the following questions:

   - What are the major factors that affect your accuracy?

- What is the impact of these factors on the joint variable values q_i needed to reach a particular position and orientation?
- How much data or what additional information do you need to implement your proposed approach?
- How would you use this information to implement your proposed approach?
- Why do you expect this approach to address the major factors identified above?

3 Submission Instructions

Submit the assignment. One person from each pair should submit code and a pdf copy of the report to the Gradescope assignment for Lab 2. After selecting the files and uploading them, the website will take you to the next page, where in the top right corner you should add your group members. If you do not add your group members they will not get credit for the assignment.

3.1 Report

The format of the report is up to you, but you should make sure that it is clear, organized, and readable. The report should include:

1. Your answers to the conceptual questions in the Methods section, typed or legibly hand-written.
2. A short 1-pg description of how the concepts are incorporated into your code. Include pointers to important line numbers of subfunctions in your code. This will help the graders understand and provide feedback on your work. (This description can be bulleted. No need to use full sentences.)
3. Your experimental results, including a description of your experimental setup (i.e., what were your inputs) and collected data.
4. Your analysis comparing expectations against reality and extrapolations to general conclusions you would make from this lab.

3.2 Code Submission

Your code should be cleaned up so that it is easy to follow. Remove any commented-out commands that you are not using, and add comments to explain the tricky steps. Clearly indicate which parts of the code correspond to which parts of the lab. Your code submission should include **only**:

1. Your `calculateIK`.
2. Any additional functions needed to run your code not included in the original code.

Each file should be attached separately to Gradescope. Do **not** zip them into a single file attachment.