

Forward Kinematics

ECE 470 Lab 3 Intro

Ben Walt

UIUC

Fall 2020



- You can find this document and any others that I share in my [Box Share Folder](#)
- You can also find it on my [Website](#)
- I recommend reviewing the **numpy** package if you have not used it before. I provide a few helpful hints on a Jupyter Notebook via my website above or [here](#).



- Our goal is to implement Forward Kinematics on the UR3
- The robot should accept joint angle input from the user and return T_{06} .
- Prep for Next Lab:
 - Continue to work on Lab 3 during your off week
 - Prepare for Lab 4 - Inverse Kinematics
 - Write your lab report



What are Forward Kinematics?

- Forward Kinematics translates from joint angles to an end effector position and orientation
- $(\theta_1, \theta_2, \theta_3, \theta_4, \theta_5, \theta_6) \rightarrow T_{06}$
- $T_{06} = \begin{bmatrix} R_{06} & p_{06} \\ 0 & 1 \end{bmatrix}$
- What we really care about in this lab is the translation vector, p_{06} , in T_{06} .
- Thus given the joint angles, we can find the end effector position.



How to obtain the Forward Kinematics?

- We will use the product of exponential method to obtain the forward kinematics.
- To do so, we need the following:
 - \mathbf{M} - the T_{06} in the zero configuration
 - S for all 6 joints
- For details see Chapter 4 of *Modern Robotics*



- You are required to create a symbolic solution to the forward kinematics in addition to the work on the robot.
- To do so, you can use MATLAB or Python:
 - MATLAB has the 'syms' command to create symbolic variables
 - Python has the SymPy package to do the same
- MATLAB is rather straight forward to use.
- SymPy is a bit more challenging to use, so it will be up to you to figure it out if you want to use it.



A demonstration of your working code is required for this lab.

- Show me (your TA) your working program
- Online students will demo their simulation over Zoom
- In-person students can demo in-person or via Zoom
- All demos will be live (no recordings), so please plan ahead to see me in office hours or during lab time - prior to the due date.
 - Group A - Week of 19 October
 - Group B - Week of 26 October
- **Please read the requirements carefully. Every semester people want to demo without completing the lab properly.**



- Lab reports will be due according to the schedule in [GradeScope](#).
 - Group A - Week of 26 October
 - Group B - Week of 2 November
- Please look closely at the lab report guidelines document - [How to Write a Lab Report](#)
- Read Section 3.6 carefully and ensure that you answer all the questions there.



Lab 3 - Hints and Help



- Read the code carefully. You should only need to edit `lab3_func.py`, but you are encouraged to read `lab3_exec.py` to get a better sense of how the program works.
- Please review the **entire** Lab 3 manual - there are many useful figures at the end of Chapter 3.
- The offsets of Section 3.5.1 are already taken care of so you do not need to do any additional correction.
- You need to test your code with a variety of configurations using the angle limits provided in Section 3.4.2 prior to doing a demo.
- The arm will always go to the requested position. Your code only changes the T_{06} output.



Lab 3 Report Details

- Ask yourself "What were the most important things we did in this lab?" This should be the focus of your report.
- **Figures, Figures, Figures** - Figures are a vital part of explaining what you did in this lab. I expect your own quality figures - i.e. not from the manual, nothing hand-drawn, clearly labeled, etc.
- Use code snippets as needed to help explain what you did
- Include your code (**lab3_func.py** only) as an appendix in your report.
- Include your symbolic derivation code in the appendix as well
- Follow the How to Write a Lab Report Guidelines!



- Look at section 3.10 of the Lab Manual to prepare for Lab 4 (Inverse Kinematics).
- Finish your Lab 3 solution before the demo due date.
- Start writing your lab report following the lab report guidelines.

