Project Report

Foundations of Robotics

1. Project Overview

- This project is inspired by the 'Plant Machete' by David Bowen[1]. We will simulate the kinematics of a 6 DOF robotic arm, that has a sword as an end-effector. I'll base my robot model on a substantively similar robot, the Universal Robots UR5, adjusting as necessary, such as adding the sword.

- This project includes three main functionalities:

1. **Forward kinematics:** function that will take the joint angles and will compute the position of each joint and end-effector.
2. **Inverse kinematics:** function that takes the desired end effector position and orientation as input and computes the joint angles as an output, including error messages when appropriate.
3. **Basic visualization:** using the Robotics Toolbox for MATLAB by Peter Corke, a simple visualization will show the motion of the robot.

- A range of references and resources were used to produce the model, including MATLAB documentation, materials offered by Universal Robots (such as the DH table of the UR5). Particularly for inverse kinematics, the solution is inspired by the theoretical derivation of Skovgaard (2018). More on this below.

2. Code package description, installation and documentation

- The code package includes three scripts:
    - forward_kinematics_ur5.m: implements forward kinematics
    - inverse_kinematics_ur5.m: implements inverse kinematics
    - ur5_project_v2.m: main script to run. From here the forward and inverse kinematics scripts will be called, and it will prompt the visualization.
- In order to run the visualization, MATLAB with the Robotics Toolbox by Peter Corke is required. The installation is referenced in footnote 2[2].
- Functions documentation:
    - **forward_kinematics_ur5(theta, d, a, alpha)**: takes four 6-element vectors theta, d, a, and alpha, corresponding to the DH parameters of the UR5. Vector theta corresponds to the desired joint angles. The function returns a transformation matrix ${}^{0}T_{6}$, that specifies the final orientation and position of the end effector, with respect to the base frame.

---

[1] https://www.dwbowen.com/plant-machete

[2] https://petercorke.com/toolboxes/robotics-toolbox/

- **inverse_kinematics_ur5(d, a, alpha, desiredPos, eulAngles, previous Joints):** takes 6-element vectors a, d, and alpha from the DH parameters of the UR5. desiredPos is a 3-element vector specifying the desired position of the end effector. eulAngles is a 3-element vector that specifies the desired orientation of the end effector, expressed as Euler angles. The function will then convert these angles to the proper rotation matrix. previousJoints is a 6-element vector that contains the joint angles that the robot is currently holding, prior to function call. These angles will be used to decide among multiple IK solutions (if they exist), choosing the one requiring the least movement from the joints.

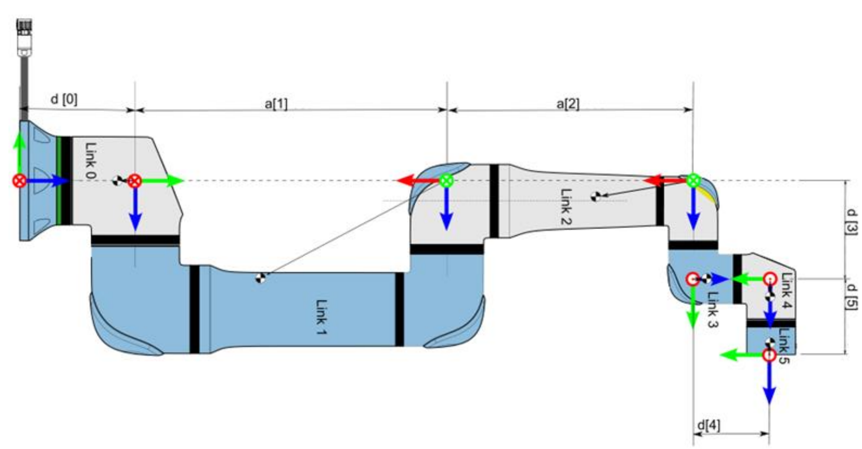
3. UR5: brief overview, DH parameters, and images

- The UR5 is a well-known robot that I'm somewhat familiar with, partially because it's available at Tandon. It has 6 DOF with a similar wrist to the plant machete. While some 6 DOF have a wrist center with three intersecting z_axis, that slightly simplify the inverse kinematics, in this case, I decided to go with the UR5 because it has the same shoulders as the 'Plant Machete', and its complexity is very similar.
- The DH parameters are provided in the table below:

DH Parameters and frames of the UR5 robot

UR5							
Kinematics	theta [rad]	a [m]	d [m]	alpha [rad]	**Dynamics**	Mass [kg]	Center of Mass [m]
Joint 1	0	0	0.089159	π/2	Link 1	3.7	[0, -0.02561, 0.00193]
Joint 2	0	-0.425	0	0	Link 2	8.393	[0.2125, 0, 0.11336]
Joint 3	0	-0.39225	0	0	Link 3	2.33	[0.15, 0.0, 0.0265]
Joint 4	0	0	0.10915	π/2	Link 4	1.219	[0, -0.0018, 0.01634]
Joint 5	0	0	0.09465	-π/2	Link 5	1.219	[0, 0.0018,0.01634]
Joint 6	0	0	0.0823	0	Link 6	0.1879	[0, 0, -0.001159]

Source: Universal Robots, link.



4. Forward Kinematics: brief overview

- The forward kinematics are straightforward. Based on the DH table provided above, we build the transformation matrices 0T1, 1T2,…, 5T6, and then we multiply them to obtain 0T1, 0T2, 0T3,…, 0T6.
- Then we plug the joint angles and 0T6 will be the final specification of position and orientation of the end effector with respect to the base frame.

5. Inverse Kinematics: brief overview

- Inverse kinematics are significantly more complex. Initially I tried a solver-based approach given a desired position and orientation (aka, given a 0T6), build a system of equations from the equivalencies between the given 0T6 and the expression of 0T6 in symbolic terms. However, this didn't yield reliable results.
- Therefore, I decided to move on with a geometrical solution. For this, I relied heavily in Skovgaard's (2018) work. They provide a theoretical approach to deriving the joint angles for a UR5 type robot. I also consulted references such as the Robotics Toolbox UR5 tools, to get a gist on how to solve the problem, as well as some MATLAB implementations of Skovgaard's work.
- The geometrical approach first solves the joints angles in a specific order: $\theta_1, \theta_5, \theta_6, \theta_3, \theta_2, \theta_4$.

6. References

- Skovgaard Andersen, Rasmus (2018). 'Kinematics of a UR5', Aalborg University. Available in https://www.studocu.com/co/document/universidad-tecnologica-de-pereira/mecanica-i/ur5-kinematics-cinematica-ur/51080984
- Robotics Toolbox by Peter Corke documentation. Available in https://petercorke.com/download/27/rtb/1050/rtb-manual.pdf
- Bazan, Vinicius (2014). 'Creating a Robot (SCARA) Animation in MATLAB', available in https://www.youtube.com/watch?v=StCAu04asOc. Code implementation available in https://github.com/VBazan/Robotics-Toolbox/blob/master/animation_script3.m
- Keating, Ryan (2014). 'UR5 Inverse Kinematics'. Johns Hopkins University. Available in https://tianyusong.com/wp-content/uploads/2017/12/ur5_inverse_kinematics.pdf
- JensOHI in Github, 'Inverse Kinematics Solver', available in https://github.com/JensOHI/IK_Solver_UR5/tree/main.
- Hawkins, Kelsey (2013), 'Analytic Inverse Kinematics for the Universal Robots UR-5/UR-10 Arms'. Available in https://repository.gatech.edu/server/api/core/bitstreams/e56759bc-92c8-43df-aa62-0dc47581459d/content
- Pradeep-roboticist on Github, 'UR5 Kinematics and Display'. https://github.com/pradeepr-roboticist/UR5Robot-MEX-Matlab