Robot Kinematics

Basics
- The robot is consisted of two arms and one hand.
- Point A refers to the hand and point B and C refers to the joints of the robot.
- The robot uses joints called revolute and prismatic.
- In this picture our robot only consist of revolute joints and not prismatic ones.

Configuration Space
1. Revolute joint is measured by an angle moving counterclockwise. Denoted as S and its parameter is from zero to 2π.
2. A Prismatic joint is parameterized by \( I = [m, M, l] \), where \( m \) is the minimum length and \( M \) is the maximum length.

Configuration Space
- The Joint Space \( J \) is the set of configurations of the joints. For robots with revolute and prismatic joint \( J \) will be a cartesian product of the circle \( S \) and interval \( I \).
- The configuration space \( C \) is the position \( (P) \) of the hand.
- There is a map \( J \rightarrow C \) that takes the actual configuration of the joints to the position of the hand.
- In the above case \( J : S \times S \times S \rightarrow C : R^2 \rightarrow R^3 \)

Forward Problem
In this example we have an angle and two given lengths.
- Starting with the forward problem we are given the angles and lengths of the robot and our goal is to find the endpoint \( (p) \).

Forward Problem Continued
- Consider coordinate \((x_i, y_i)\) whose origin is at \( i \)th joint with \( x_i \) parallel to the \( i \)th arm and \( y_i \) perpendicular counterclockwise.
- Given the example for the previous slide we’re going to use it to solve the forward problem for two arms.
  1. In the \((x_1, y_1)\) coordinate \( (p) \) has coordinate \((0,0)\).
  2. In the \((x_2, y_2)\) coordinate \( (p) \) has coordinate \((l,0)\).
- This method could be repeated multiple times and used depending on the number of arms of the robot.

Inverse Problem
- Given the endpoint \( (p) \), we are to find the lengths of each arm and the angles of each joint.
- May have more than one solution or sometimes none at all.
- Find the solution which is within the constraint of our joint space.

Three Dimensional
Coordinates given in form of \((x, y, z)\)
- Configuration Space - \( S \times S \times S \rightarrow J \rightarrow R^3 \)
- Calculate \( \theta = \arctan\left(\frac{y}{x}\right) \)
- Rotate base revolute joint to angle of \( \theta \)
  \( x = \frac{z}{\sin \theta} + \frac{x_i}{\cos \theta} \)
  \( y = \frac{z}{\sin \theta} \)
- Approach a 2-Dimensional

Code
- Enter \( x, y \) coordinates in GUI
- Scipy Library’s optimize method is used to calculate angles and lengths
- A plot is also shown to help visualize in 2-D

Example:

Summary
- We understood and used the forward problem and incorporated this into our robot. The same thing was done using the inverse problem.
- We developed a deeper understanding of the configuration space of the robot.
- Many solutions might exist theoretically, however, only a few or none are applicable.
- The Levenberg-Marquardt method was used to solve for least squares that led us to solve the root of our equations for the robot.

Possible Extensions
- Consideration of a real robot with prismatic joints, to get a better understanding of all movements of the robot. This would also let us access more joints because not all prismatic joints are fixed.
- We could have further explored the limitations of the configuration space.
- Efficiency of more concise answers of the movement of the robot would have been implemented.