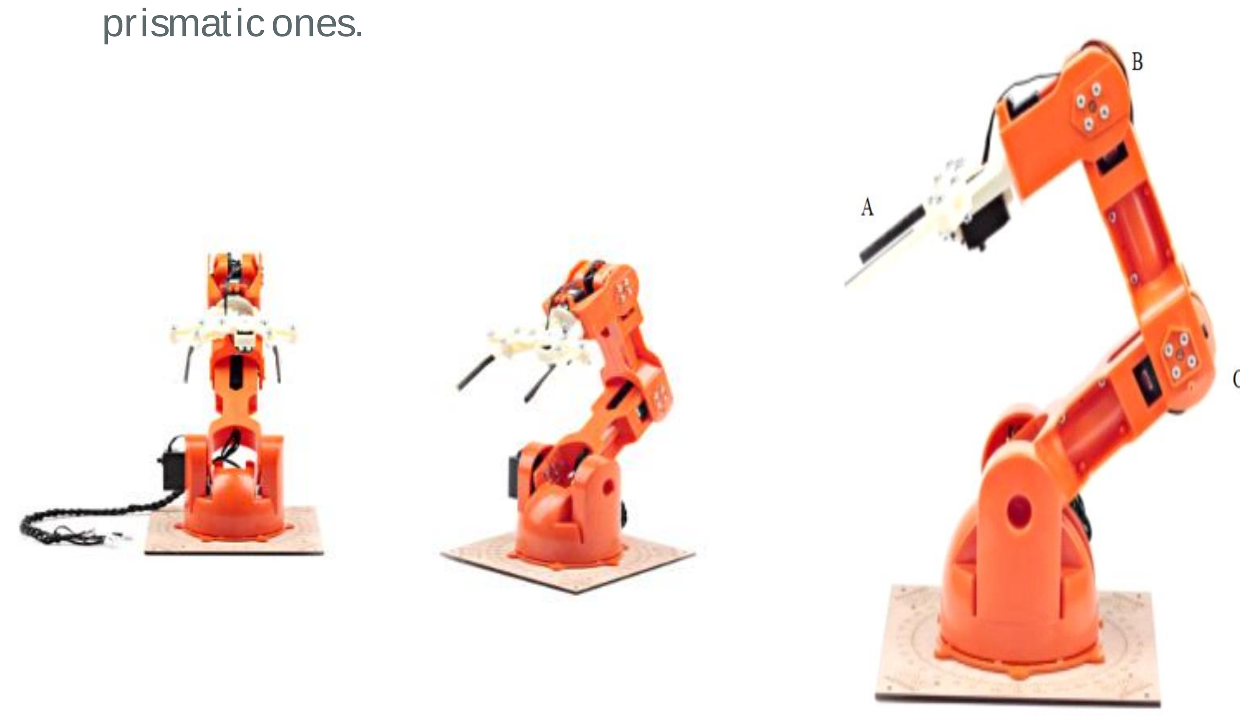
Robot Kinematics

Supervisor:
Julius Ross

Project Members:
Ativ Aggarwal,
Leticia Garcia,
Sara Sayeed

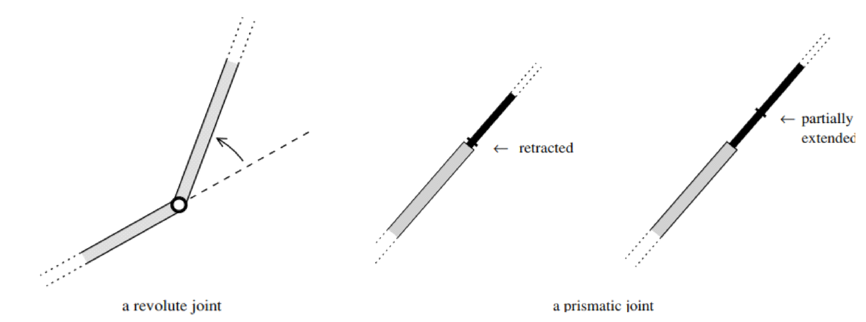
Basics

- 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.



Basics

- A revolute joint allows a rotation of one segment relative to another. We will assume that the axis of rotation is perpendicular to the place. Thus, each segment has their own axis.
- A prismatic joint allows one segment of a robot to move by sliding. We will assume for simplicity that all joints are in the same plane, and the axes of rotation of all revolute joints are perpendicular to that and that the translation axis all lie in the same plane.



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], where m is the minimum length and M is the maximum length .

Configuration Space

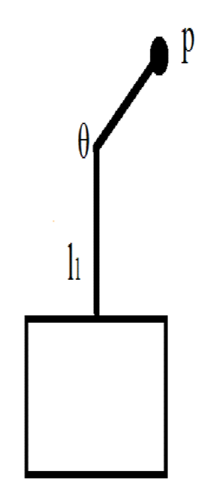
- The Joint Space (J) is the set of configurations of the joints. For robots with revolute and prismatic joint J will be a cartesian product of the circle S and interval I.
- The configuration space (C) is the position (P) of the hand
- There is a map $J \to C$ that takes the actual configuration of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

$$J : S \times S \times S \to C : \mathbb{R}^2 or \mathbb{R}^3$$

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).

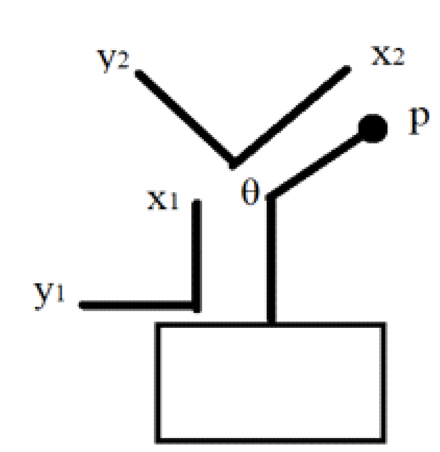


Forward Problem Continued

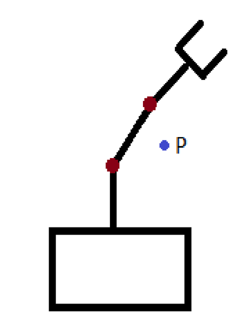
- Consider coordinate (xi, yi) whose origin is at ith joint with xi parallel to the ith arm and yi perpendicular counterclockwise.
- Given the example for the previous slide we're going to use it to solve the forward problem for two arms.
  1. In the (x_2, y_2) coordinate p has coordinate (l,0).
  2. In the (x_1, y_1) coordinate p has coordinate of the following equation.

$$\begin{pmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{pmatrix} + \begin{pmatrix} 1 \\ 0 \end{pmatrix} + \begin{pmatrix} l \\ 0 \end{pmatrix}$$

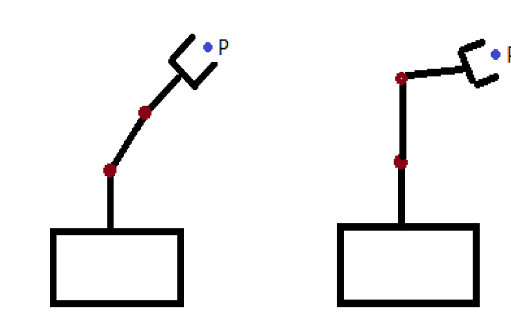
  3. This method could be repeated multiple times and used depending on the number of arms of the robot.



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 -

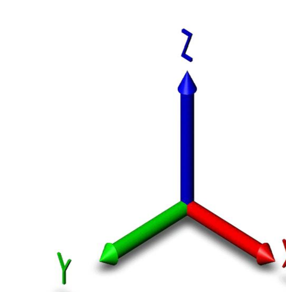
$$S \times S \times S \xrightarrow{f} J = \mathbb{R}^3$$

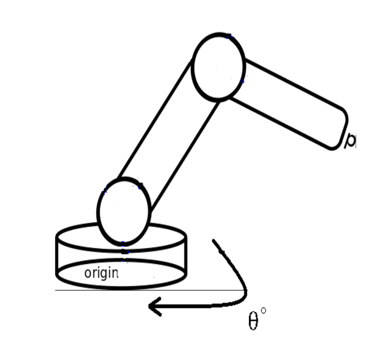
- Calculate $\Theta = \tan^{-1}(y/x)$
- Rotate base revolute joint to angle of Θ

$$\tilde{x} = x / \cos\Theta$$

$$y \Rightarrow 0$$

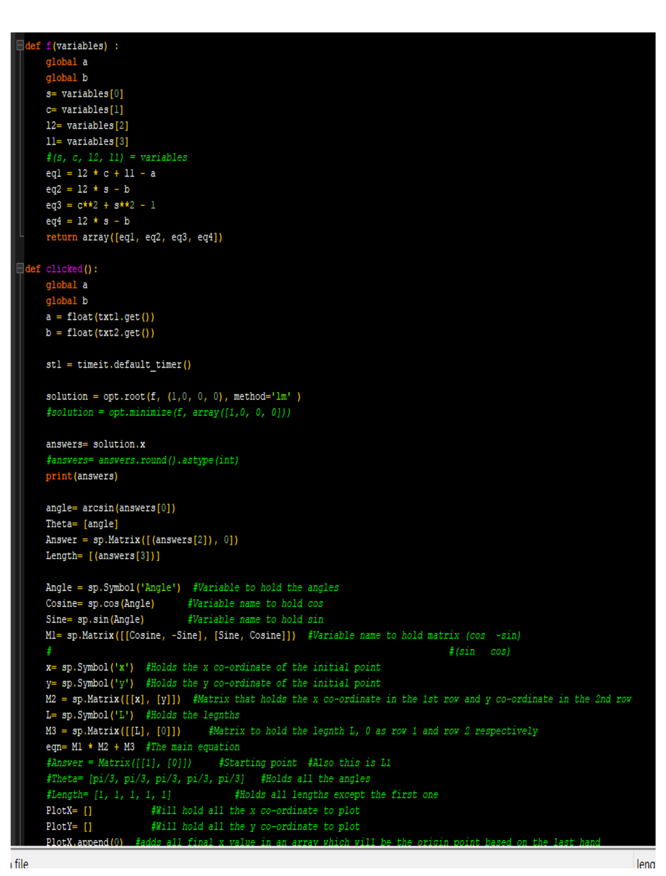
- Approach a 2- Dimensional and calculate for $(\overline{x}, z)$





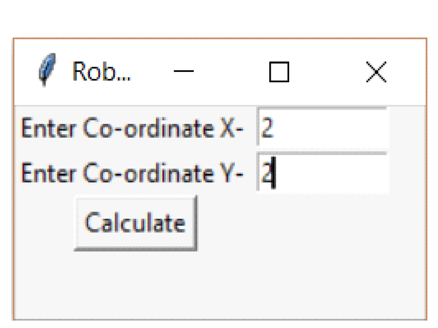
Code

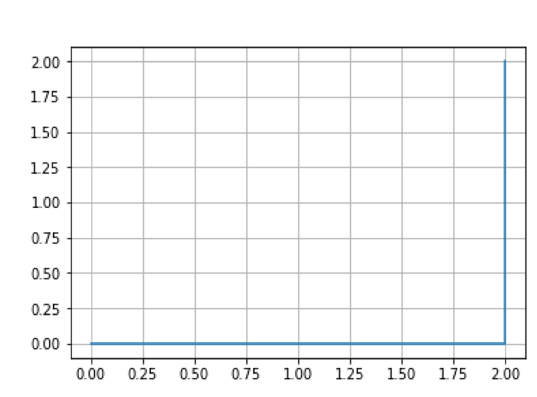
- Enter x, y coordinates in GUI
- Scipy Library's optimize method is used to calculate angles and lengths
- A plot is also shown to help visualize in 2-D



Code

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

Possible Extensions

- Consideration of a real robot with prismatic joints, to get a better understanding of all the movements of the robot. This would also let us access more points 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.