**2. Kinematic Calibration:**

Step 1:

Assume that there is only one configuration, that is .

Place the manipulator in this configuration and measure the actual position of the end effector, we have:

*0.39685078431539156*

*2.064380977101835*

where

Step 2:

Compute the end effector postion based on the assumed link lengths using forward kinematics equation as follows:

*0.4177376677004121*

*2.1730326074756157*

where

Step 3:

Define an error function to qualify the difference the measured and comuted positions as the following:

Step 4: Optimize link lengths

* Start with an initial guess for the link lengths, e.g., as step 2.
* Use an optimization method to minimize the error function by adjusting the link lengths . Here, I will use the library *scipy.optimize* in Python, and the optimal values are:

0.9620442720798735

0.9365663577745206

0.9598341076374409

Implementation in Python:

A screen shot of a computer program

Description automatically generated

**1. Kinematic Redundancy:**

According to Yoshikawa’s approach, we focus on modifying the inverse kinematics solution to avoid singularities.

Step 1: Calculate Jacobian matrix

For a simple 3 DOF robot, the equation as follows:

where

link lengths

joint angles

Step 2: Detect singularities

For each configuration, compute the determinant of the Jacobian, det(), and identify configurations where det() = 0. These configurations are the singularities to avoid!

Step 3: Define the objective function for joint angles optimization

1. Singularity penalty:

where is a small constant to prevent division by zero

1. Objective function:

where and are weights that balance the trade-off between minimizing position error and avoiding singularities.

Example:

Input:

* Link lengths:
* Target position: x = 0.6, y = 2.4
* Start with initial joint angles:

Output:

Optimized joint angles : 27.722967815649003

Optimized joint angles : 58.582329821010475

Optimized joint angles : 24.160442979883808

Implementation in Python:

A computer screen shot of text

Description automatically generated