**Abstract**

In this report we discuss the optimization technique that is applied to optimize a 2DOF robot's trajectory from point A to point B. The use of inverse kinematics to find the angle of the joints given a target coordinate for the end-effector is also discussed at length in this report.

**Introduction**

There are many applications that require the use of a robot arm in modern technology. Modern tools that make use of a robot arm include something as simple as a construction car to something as sophisticated as the Canadarm2. In order for such systems to work as efficiently as possible, it is crucial that the arm can find the optimal trajectory to go from one position to another. Thus robot motion planning has been one of the research of interest over the years [1].

The method we use to optimize...

**Inverse Kinematics**

Inverse kinematics is the process of calculating the joint angles given a desired coordinate of the end-effector. The robot sees an object and wants to grab it. At what angles should each individual joint go to in order for it to grab that object? Such is the calculation of inverse kinematics. The relationship is given as:

 (1)

Where = [x1 x2 x3...xn] and = [Ө1 Ө2 Ө3... Өn]

X is the set of each joints position while Ө is a set of each joints angle.

As one would imagine, there can be infinite many solutions to this problem. As shown in figure 1 there can be multiple angles at which the joints can be at and still have the end-effector reach the same location. The possibility of having infinite solutions means there must be one that is most optimal based on the current position of the arm. There is also a possibility that there may be no solution to the problem. The target location of the end-effector may be out of range of the workspace or it may require the joints to bend at constrained angles. These constraints will be defined as part of the objective function.

**FIGURE 1**

**Problem Model**

The objective function is to minimize the distance between the end-effector and the target location. The model is given by this formula:

min |Pe - Pc|

Subjected to:

self.l12 < (self.l[1]+self.l[2]) (2)

j1 != j2 != j3 (3)

The joints must not touch each other and the system must only work within the workspace. The objective is to rotate each join

We want to find the new angles of the joints

**Robot Arm Implementation in Python**

**Conclusion**

**References**

[1] P Savsani, R. L. Jhala, and V.J. Savsani, "Optimized Trajectory Planning of Robotic Arm Using teaching learning based optimization (TLBO) and artificial bee colony (ABV) optimization techniques", IEEE, 2013.

[2] W. Lalo, T. Brandt, D. Schramm, and M. Hiller "A Linear Optimization Approach to Inverse Kinematics of Redundant Robots with Respect to Manipulability", ISARC, 2008.