

# The Application and Use of 2 Joint Inverse Kinematics for General Robotics.

## Background

Readers should have a basic understanding or background in the following concepts:

- Basic trigonometry
- How to work with angles
- Basic programming with angles

## Motivation

I have been building a bipedal robot over the summer of 2022. I have needed easy, simple, straight to the point resources on control concepts that are necessary to create walking gaits for the bipedal robot. One of the concepts necessary for these gaits, is inverse kinematics. Inverse Kinematics is in the valley of being easy enough for people who understand it easily to often not create easy resources, and being hard enough for the average person not being able to grasp the concept quickly.

My goal with this resource is to create a easy way for people to understand the math behind Inverse Kinematics, and a way to easily implement it on their robotics project.

# Terminology

Inverse Kinematics – The mathematical equations to finding the angles needed for a arm to reach a certain position at the end of the arm.

Arm – Defined as a controllable double pendulum, generally controlled by a series of motors and a micro controller.

Joint – Defined as a single pendulum on a arm. Both must be the same length.

# Mathematical Definitions

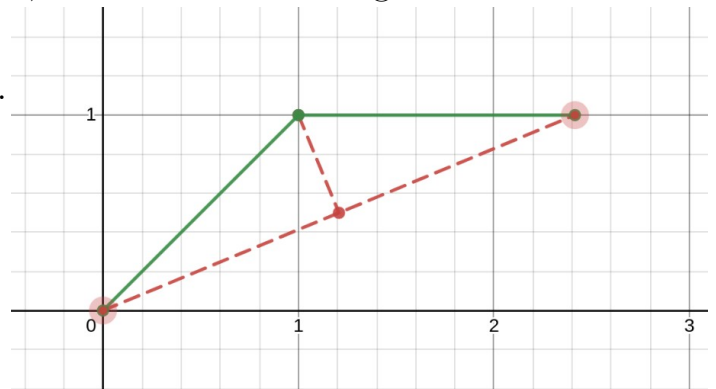
$J$  is defined as the length of both joints.

$T$  is the target coordinate of the endpoint of the arm.

# Inverse Kinematics Equations

You may notice how arms, at any position, will create two triangles back to back. We can use this to our advantage, because trigonometry is so easy to grasp.

First, we must find the  $\Phi$ , which is the angle between the X axis, and the imaginary line that connects the endpoint of the arm and the origin.



The equation for  $\Phi$  uses the inverse tangent function to find the angle from both the opposite and adjacent. The equation to find  $\Phi$  is the following:  $\Phi = \tan^{-1} \frac{T_y}{T_x}$ .

Now that we have  $\Phi$ , we need to find the length of the imaginary line connecting the origin and  $T$ , we'll call this variable  $B$ , for base. We can find  $B$  by using the equation  $B = \frac{T_x}{\cos \Phi}$ .

We only need one more variable to get the final angles of our arm. We need to find  $\Theta$ , which is the angle from the imaginary base, from which we calculated length last step, to the first joint of the arm. We can do this by halving the base, because we're trying to find the angle of single triangle, and using the joint length to find our angle between the base and the joint. The equation looks like this:

$$\Theta = \cos^{-1} \frac{.5B}{J}$$

Now that we have our variables, we can finally get the angles to set each arms. The math here is pretty simple: to find the angle from the X axis to the first joint, we add  $\Phi$  and  $\Theta$ :  $A_1 = \Phi + \Theta$ . To find the angle from the first joint's angle to the second joints angle we simply subtract:  $A_2 = \Phi - \Theta$ .

A complete calculation looks like this:

$$\Phi = \tan^{-1} \frac{T_y}{T_x}$$

$$B = \frac{T_x}{\cos \Phi}$$

$$\Theta = \cos^{-1} \frac{.5B}{J}$$

$$A_1 = \Phi + \Theta$$

$$A_2 = \Phi - \Theta$$

## Implementation

The following is a step by step way to implement inverse kinematics in a systematic environment, like computers and microprocessors.

1. Get the Target X, and Y Position,  $T_x$  and  $T_y$ , respectively
2. Get the Joint Length,  $J$
3. Calculate  $\Phi$  via the equation  $\Phi = \tan^{-1} \frac{T_y}{T_x}$

4. Calculate the base length,  $B$ , via the equation  $B = \frac{T_x}{\cos \Phi}$
5. Calculate  $\Theta$  via the equation  $\Theta = \cos^{-1} \frac{.5B}{J}$
6. Calculate the first angle,  $A_1$ , via the equation  $A_1 = \Phi + \Theta$
7. Calculate the second angle,  $A_2$ , via the equation  $A_2 = \Phi - \Theta$
8. If necessary, adjust the first and second angles before setting the motors/servos to them. (IE: flipping angles, or adding  $90^\circ$  to second angle.) This is dependent on the motor/servo itself, and the controller used.

## Resources

The following resources have helped me immensely in creating this document. A massive thanks to all of these people and their works!

### *Information:*

- <https://youtu.be/Q-UeYEpwXXU> – RoTechnic, “Easy inverse kinematics for robot arms”, January 23<sup>rd</sup>, 2022
- <https://youtu.be/IN8tjTk8ExI> – James Bruton, “How Robots Use Maths to Move”, March 2<sup>nd</sup>, 2021

### *Tools Used:*

- <https://www.desmos.com/> - Desmos Studio
- <https://libreoffice.org/> - The Document Foundation