

# Inverse Kinematics for a 12 DOF Quadraped

Christopher Schicho

October 22, 2020

## Contents

<b>1</b>	<b>Introduction</b>	<b>2</b>
1.1	Purpose . . . . .	2
<b>2</b>	<b>3 DOF Leg Inverse Kinematics</b>	<b>2</b>
2.1	Z-Axis . . . . .	3
2.2	X-Axis . . . . .	5
2.3	Y-Axis . . . . .	7
2.4	Final Leg Model . . . . .	10
<b>3</b>	<b>Body Inverse Kinematics</b>	<b>11</b>
3.1	Pitch . . . . .	11
3.2	Yaw . . . . .	11
3.3	Roll . . . . .	11
3.4	Final Body Model . . . . .	11
<b>4</b>	<b>Outroduction</b>	<b>11</b>
<b>5</b>	<b>References</b>	<b>11</b>

# 1 Introduction

This Paper is about the inverse kinematics for a 12 DOF quadruped robot. It is based on an open source project called "open-quadruped" . Nevertheless, this paper describes the kinematic in a general way. Therefore, it is useable for other 12 DOF quadruped projects as well.

## 1.1 Purpose

The purpose of this paper is to give you an oversight about what the inverse kinematics is about and to show you, how you use this for your own project.

# 2 3 DOF Leg Inverse Kinematics

Firstly, let us define what inverse kinematics is about. The key idea of inverse kinematics is to only care about the position of the end of a kinematic chain. We do not want to care about all the variable joint parameters needed to reach this position. Therefore, we need to create a mathematical process to calculate all of these variable joint parameters.

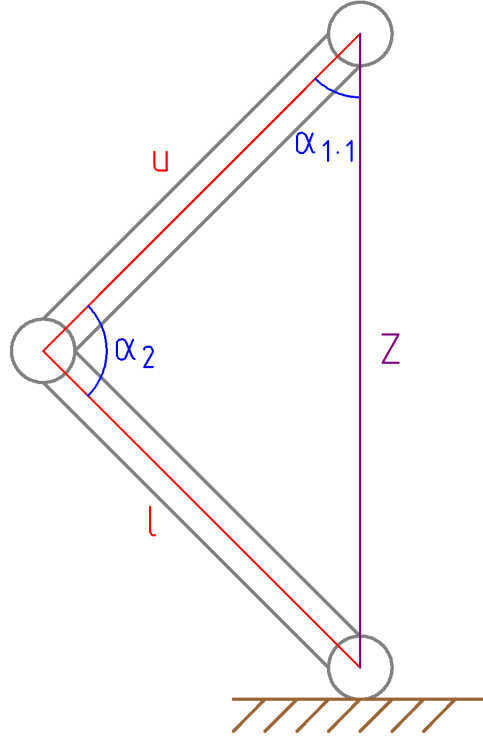
This is exactly what this section will discover. We will start with calculating every single axis and combine them to a complete leg inverse kinematics model once we are done with these single steps.

## 2.1 Z-Axis

The easiest way to start is by starting with the z-axis. The z-axis is responsible for moving up and down the quadruped's body. In this step we will calculate the upper leg's and the lower leg's joint angles.

In order to calculate the necessary joint parameters we need some details about the used quadruped. For this axis we need to know the length from pivot to pivot of the upper and lower leg. The used measurement unit is not important for this calculation, as long as you use the same for every length / variable. However, I will use millimeters in the following calculations.

In the picture below you can see the quadruped's leg and also the triangle we will need in order to calculate our joint parameters. Due to the Design of this model, the end of the lower leg will always be directly under the pivot of the upper leg, but note that this is only true if we only perform z-axis movements. This is a good configuration and makes calculating the angles much easier.



The first calculation uses cosine rule to calculate the angle  $\alpha_{1.1}$ , where  $Z$  is the input for the z-axis,  $u$  the length of the upper leg and  $l$  the length of the lower leg. This calculation is only directly applied to the quadruped if you want

to perform the z-axis movements without including the other axis, otherwise the value of  $\alpha_1$  will be calculated in the x-axis step.

$$\alpha_{1.1} = \arccos \frac{u^2 + Z^2 - l^2}{2 * u * Z} \quad (1)$$

The second calculation uses the same rule for calculating  $\alpha_2$ . This is the angle which will determine the wrist movement later.

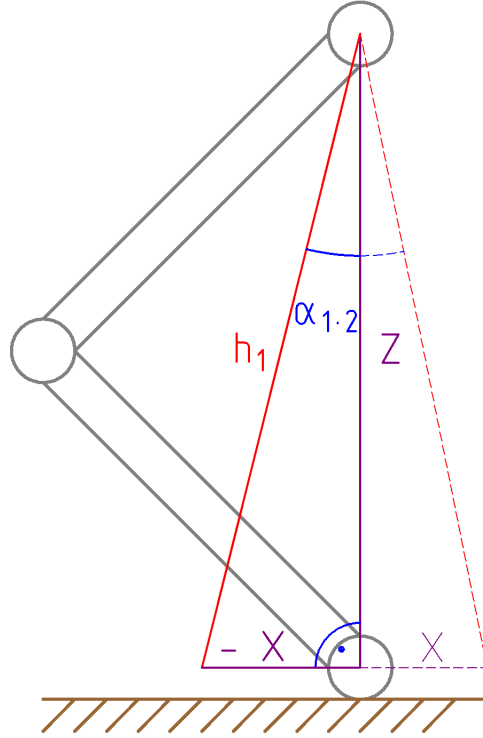
$$\alpha_2 = \arccos \frac{u^2 + l^2 - Z^2}{2 * u * l} \quad (2)$$

At this point we are done with the z-axis calculation. The quadruped is now able to perform movements along the z-axis.

## 2.2 X-Axis

The next step is to calculate the x-axis calculations. This axis is responsible for moving the body back and forth. In this step we do not need any further information about the quadruped.

The picture below shows again one leg of the quadruped, however, this time it looks a little bit different. Note that x-axis may have positive and negative values. This is slightly different compared to the z-axis which is only positive.



At first we calculate the hypotenuse from our new triangle by using the Pythagoras theorem. The variables  $X$  and  $Z$  represent the input for these axis. we will need this length to calculate the desired angle  $\alpha_1$ .

$$h_1 = \sqrt{X^2 + Z^2} \quad (3)$$

Next we calculate the angle  $\alpha_{1.2}$ . This angle represents the difference between our desired position and the quadruped's initial position.

$$\alpha_{1.2} = \arccos \frac{Z}{h_1} \quad (4)$$

In this step we need to make a case distinction due to the fact that  $X$  can be either positive or negative. If  $X$  is negative we want to subtract  $\alpha_{1.2}$  from the initial angle, otherwise we want to add it. We need this step because otherwise the quadruped would only move in one direction. This angle will determine the shoulder movement later.

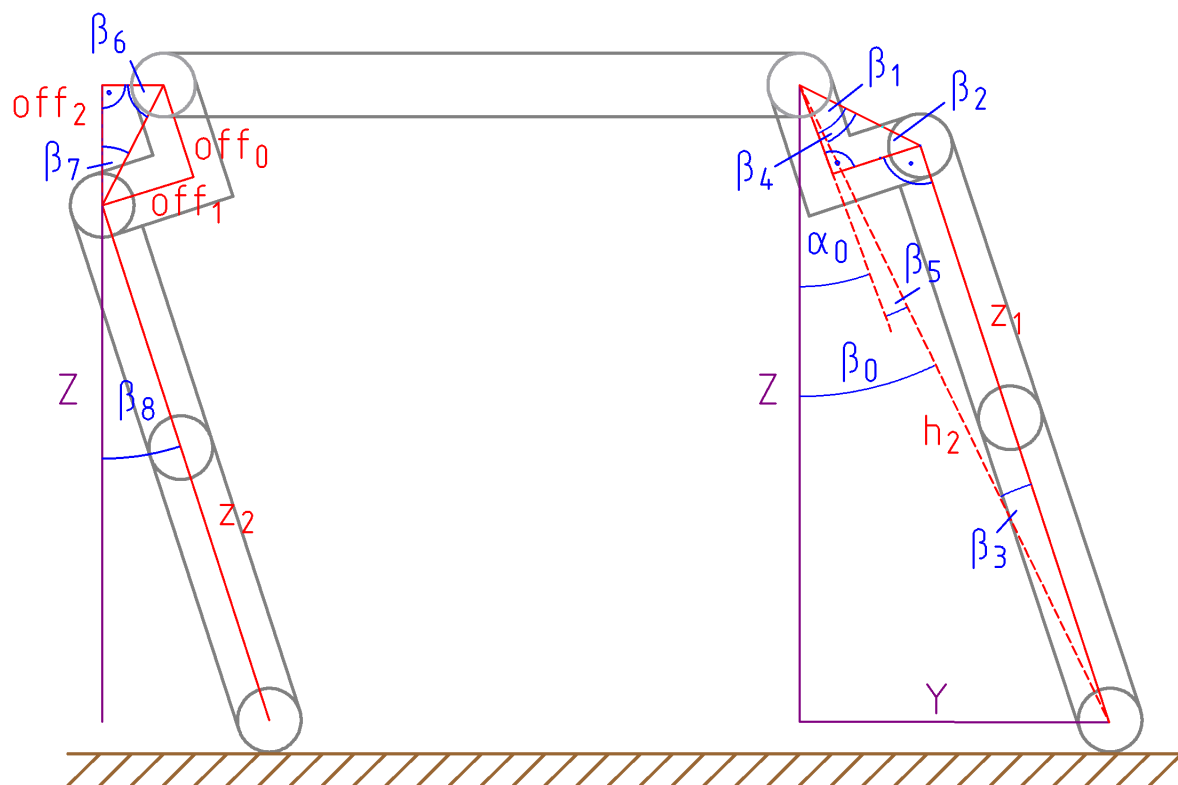
$$\alpha_1 = \begin{cases} \alpha_{1.1} + \alpha_{1.2} & \alpha_{1.2} \geq 0 \\ \alpha_{1.1} - \alpha_{1.2} & \alpha_{1.2} < 0 \end{cases}$$

Now we are also done with the x-axis. At this point we are able to perform movements along the z- and x-axis simultaneously.

## 2.3 Y-Axis

Finally we will calculate the last the parameters of the y-axis. This is rather complicated compared to the other ones. This responsible for moving the quadruped's body sideways.

The following picture illustrates what we will calculate in the next step. The main problem is that if we shift the body sideways, the legs of one side of the robot get shorter and the other get longer. Therefore we need to calculate two new  $Z$  values, we will call them  $z_1$  and  $z_2$ . Indeed we will also calculate the angle  $alpha_3$  which is our hip angle.



In order to calculate all the desired values we some more details about the construction of the quadruped. As you can see in the picture above we 2 length of the hip joint, we will call them  $off_0$  and  $off_1$ .  $off_0$  is the vertical distance between the middle of the hip pivot and the shoulder pivot.  $off_1$  is the horizontal distance between the hip pivot and the middle of the upper leg. In order to make the robot's leg standing straight we need to include the  $off_1$  to our  $Y_{input}$ . Also keep in mind that the the y-axis can also either positive or false.

$$Y = Y_{input} + off_1 \quad (5)$$

We start by calculating some lengths of the triangles and their corresponding angles.

$$h_2 = \sqrt{Z^2 + Y^2} \quad (6)$$

$$h_3 = \sqrt{off_0^2 + off_1^2} \quad (7)$$

$$\beta_0 = \arctan \frac{Y}{Z} \quad (8)$$

$$\beta_1 = \arctan \frac{off_1}{off_0} \quad (9)$$

$$\beta_2 = \arctan \frac{off_0}{off_1} \quad (10)$$

$$\beta_3 = \arcsin \frac{h_3 * \sin(\beta_2 + 90)}{h_2} \quad (11)$$

$$\beta_4 = 90 - (\beta_2 + \beta_3) \quad (12)$$

$$\beta_5 = \beta_1 - \beta_4 \quad (13)$$

The following calculation returns  $\alpha_0$ , which is the shoulder angle.

$$\alpha_0 = \beta_0 - \beta_5 \quad (14)$$

The following steps are necessary to calculate the two different  $Z$  values. These values are necessary for calculating the calculations of the other axis.

$$\beta_6 = 90 - (\beta_1 - \alpha_0) \quad (15)$$



$$\beta_7 = 90 - \beta_6 \quad (16)$$

$$\beta_8 = 90 - (\beta_6 + \beta_7) \quad (17)$$

$$off_2 = h_3 * \cos(\beta_7) \quad (18)$$

$$z_1 = \frac{h_3 * \sin(\beta_4)}{\sin(\beta_8)} \quad (19)$$

$$z_2 = \frac{Z - off_2}{\cos(\beta_8)} \quad (20)$$

In case that  $Y$  is negative we will use  $z_1$  for the right side of the quadruped and  $z_2$  for the left side, otherwise we use them the other way around.

Now we are done with all the calculations. The next step is to combine all this steps to a final model.

## 2.4 Final Leg Model

Due to the two different  $Z$  values from the y-axis calculations, we need to replace the calculations of the x- and z-axis by the following functions. We will use this calculations additionally to the y-axis calculations. Basically we need to distinguish if we are calculating the left or the right side of the quadruped. Therefore we will introduce some new angles, namely  $\alpha_{1left}$ ,  $\alpha_{1right}$ ,  $\alpha_{2left}$ ,  $\alpha_{2right}$ ,  $\alpha_{1.1left}$ ,  $\alpha_{1.1right}$ ,  $\alpha_{1.2left}$  and  $\alpha_{1.2right}$  and some new lengths.

$$\alpha_{1.1left} = \begin{cases} \arccos \frac{u^2 + z_1^2 - l^2}{2 * u * z_1} & Y \geq 0 \\ \arccos \frac{u^2 + z_2^2 - l^2}{2 * u * z_2} & Y < 0 \end{cases}$$

$$\alpha_{1.1right} = \begin{cases} \arccos \frac{u^2 + z_2^2 - l^2}{2 * u * z_2} & Y \geq 0 \\ \arccos \frac{u^2 + z_1^2 - l^2}{2 * u * z_1} & Y < 0 \end{cases}$$

$$\alpha_{1.2left} = \begin{cases} \arccos \frac{z_1}{\sqrt{X^2 + z_1^2}} & Y \geq 0 \\ \arccos \frac{z_2}{\sqrt{X^2 + z_2^2}} & Y < 0 \end{cases}$$

$$\alpha_{1.2right} = \begin{cases} \arccos \frac{z_2}{\sqrt{X^2 + z_2^2}} & Y \geq 0 \\ \arccos \frac{z_1}{\sqrt{X^2 + z_1^2}} & Y < 0 \end{cases}$$

$$\alpha_{1left} = \begin{cases} \alpha_{1.1left} + \alpha_{1.2left} & \alpha_{1.2left} \geq 0 \\ \alpha_{1.1left} - \alpha_{1.2left} & \alpha_{1.2left} < 0 \end{cases}$$

$$\alpha_{1right} = \begin{cases} \alpha_{1.1right} + \alpha_{1.2right} & \alpha_{1.2right} \geq 0 \\ \alpha_{1.1right} - \alpha_{1.2right} & \alpha_{1.2right} < 0 \end{cases}$$

$$\alpha_{2left} = \begin{cases} \arccos \frac{u^2 + l^2 - z_1^2}{2 * u * l} & Y \geq 0 \\ \arccos \frac{u^2 + l^2 - z_2^2}{2 * u * l} & Y < 0 \end{cases}$$

$$\alpha_{2right} = \begin{cases} \arccos \frac{u^2 + l^2 - z_2^2}{2 * u * l} & Y \geq 0 \\ \arccos \frac{u^2 + l^2 - z_1^2}{2 * u * l} & Y < 0 \end{cases}$$

## 3 Body Inverse Kinematics

### 3.1 Pitch

### 3.2 Yaw

### 3.3 Roll

### 3.4 Final Body Model

## 4 Outroduction

At this point we are done with all the calculations. Our Quadruped is now able to perform all necessary movements except walking. The next step will be to make the quadruped walk. Therefore, dynamic walking will be subject of the next paper.

In case that you are interested in the project or how all of this works applied to a real robot. Then just visit the following sites:

[www.enabling-intelligence.com](http://www.enabling-intelligence.com)

<https://github.com/c-schicho>

## 5 References

[https://en.wikipedia.org/wiki/Inverse\\_kinematics](https://en.wikipedia.org/wiki/Inverse_kinematics)

[https://www.adham-e.dev/pdf/IK\\_Model.pdf](https://www.adham-e.dev/pdf/IK_Model.pdf)

<https://github.com/adham-elarabawy/open-quadruped>