Modeling and Control of Robots – Project 2

Forward and inverse kinematics of Hexapod

Team Members : Ezhilan Veluchami, Kirthik Roshan Nagaraj, Shiva Sam Kumar, Suriya Prakash Murugan, Jyothsna Suresh

Team number : 21

Arizona State University

Inverse Kinematics

In this project, there are two parts in algorithm :

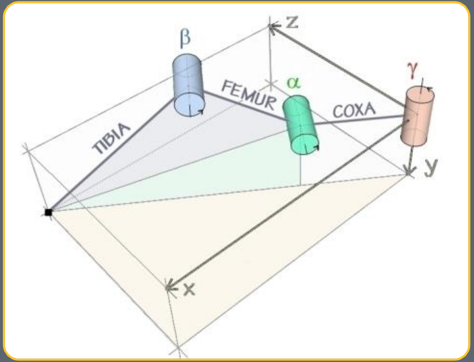
- Inverse Kinematics of Leg
- Inverse Kinematics of Body

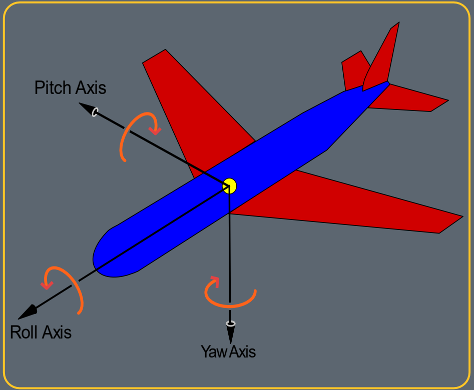
Changing the body's center will alter the coordinates of the feet, which will alter the servo angles.

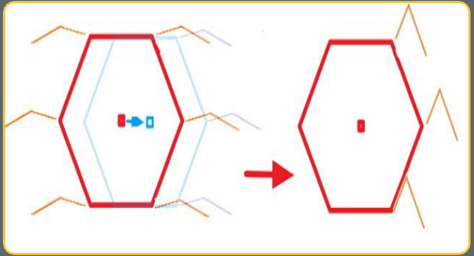
In Airplanes, generally there are 3 types of body movements

- Roll
- Pitch
- Yaw (Rotate)   Actually, there is one more for Hexapod Robot
- Translate
- We are moving the robot to the right (Centre of robot moving from red to the blue pt.), the end of feet coordinates would shift to the right of the same distance as the Centre of the body. The feet are stationary, so moving the body won’t change the absolute positions of the feet.

Taking these changes into account, we compute the latest relative coordinates using Body IK procedure, then pass that results to Leg IK algorithm.

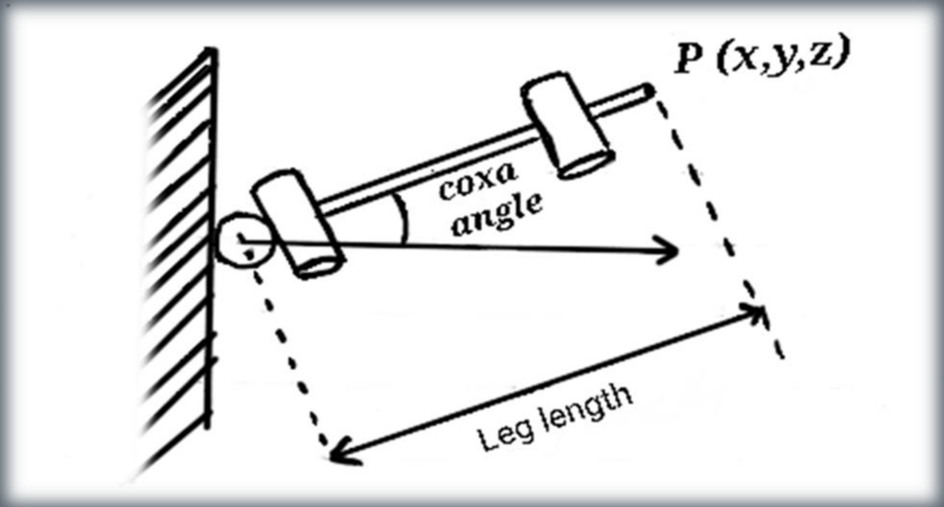


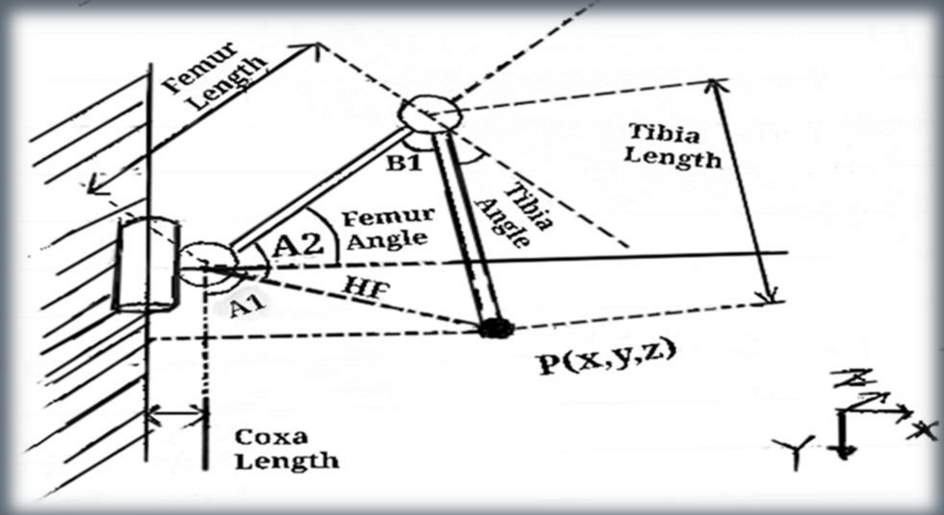




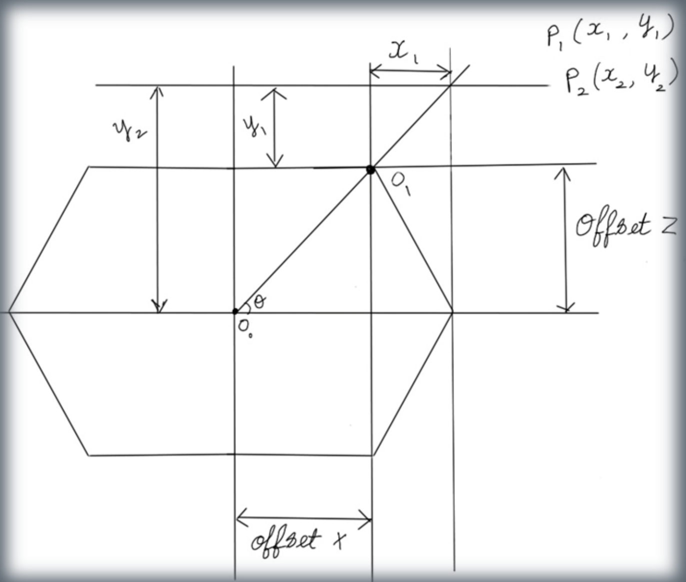
Inverse Kinematics - Leg

1. Leg Length $= \sqrt{(X^2 + Z^2)}$
2. HF $= \sqrt{((\text{Leg Length} - \text{Coxa Length})^2 + Y^2)}$
3. A1 $= \text{ArcTan}\left(\frac{(\text{Leg Length} - \text{Coxa Length})}{Y}\right)$
4. A2 $= \text{ArcCos}\left(\frac{(\text{TLength}^2 - \text{FLength}^2 - \text{HF}^2)}{-2 \cdot \text{Flength} \cdot \text{HF}}\right)$
5. F.Angle $= 90 - (A1 + A2)$
6. B1 $= \text{ArcCos}\left(\frac{(\text{HF}^2 - \text{TLength}^2 - \text{FLength}^2)}{-2 \cdot \text{Flength} \cdot \text{TLength}}\right)$
7. T.Angle $= 90 - B1$
8. C.Angle $= \text{ArcTan}\left(\frac{Z}{X}\right)$





Inverse Kinematics - Body



Compute Initial Position Parameter: P_1 and P_2 w.r.t coxa O_1 *and* O_0

$$x_2 = x_1 + \text{Offset X}$$

$$y_2 = y_1 + \text{Offset Z}$$

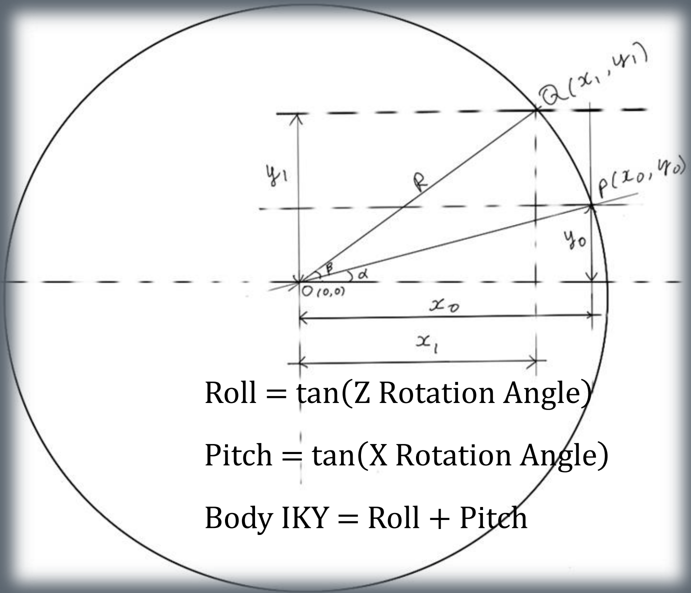
The distance of end point of feet to center of the body(O_0), **d=** $\sqrt{((x_2)^2 + (y_2)^2)}$

We can get the angle between $P_1 O_0$ **and x-axis :** $\theta = \operatorname{atan}\left(\frac{y_2}{x_2}\right)$

When we are rotating around Y axis (vertical to the ground)

The new point is (x_1, y_1) and old point is (x_0, y_0) : Body IK Position X $= x_1 - x_0$, Body IK Position Z $= y_1 - y_0$

Known P (x_0, y_0) α, β, R.	-->	To find Q(x_1, y_1)
$\tan\alpha = \left(\frac{y_0}{x_0}\right)$	-->	$\alpha = \operatorname{atan}\left(\frac{y_0}{x_0}\right)$
$\tan(\alpha+\beta) = \left(\frac{y_1}{x_1}\right)$	-->	$y_1 = x_1 \tan(\alpha+\beta)$
$x_1^2 + y_1^2 = R^2$	-->	$x_1^2 + x_1^2\tan^2(\alpha+\beta) = R^2$
$x_1^2 = \left(\frac{R^2}{1+\tan^2(\alpha+\beta)}\right)$	-->	$x_1 = R * \cos(\alpha+\beta)$
$y_1^2 = R^2 - x_1^2$	-->	$y_1 = R * \sin(\alpha+\beta)$



Roll = tan(Z Rotation Angle)

Pitch = tan(X Rotation Angle)

Body IKY = Roll + Pitch

Improvements In Inverse Kinematics

- $R_x(\theta) = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos\theta & -\sin\theta \\ 0 & \sin\theta & \cos\theta \end{bmatrix}$ $\quad R_z(\alpha) = \begin{bmatrix} \cos\alpha & 0 & \sin\alpha \\ 0 & 1 & 0 \\ -\sin\alpha & 0 & \cos\alpha \end{bmatrix}$ $\quad R_y(\beta) = \begin{bmatrix} \cos\beta & -\sin\beta & 0 \\ \sin\beta & \cos\beta & 0 \\ 0 & 0 & 1 \end{bmatrix}$
- $R = R_x \cdot R_z \cdot R_y = \begin{bmatrix} \cos\alpha \cdot \cos\beta & -\cos\alpha \cdot \sin\beta & \sin\alpha \\ \sin\theta \cdot \sin\alpha \cdot \cos\beta + \cos\theta \cdot \sin\beta & \sin\theta \cdot \sin\alpha \cdot \sin\beta + \cos\theta \cdot \cos\beta & -\sin\theta \cdot \cos\alpha \\ -\cos\theta \cdot \sin\alpha \cdot \cos\beta + \sin\theta \cdot \sin\beta & \cos\theta \cdot \sin\alpha \cdot \sin\beta + \sin\theta \cdot \cos\beta & \cos\theta \cdot \cos\alpha \end{bmatrix}$
- $\begin{bmatrix} X' \\ Z' \\ Y' \end{bmatrix} = R \begin{bmatrix} x \\ z \\ y \end{bmatrix} = \begin{bmatrix} x(\cos\alpha \cdot \cos\beta) - z(\cos\alpha \cdot \sin\beta) + y(\sin\alpha) \\ x(\sin\theta \cdot \sin\alpha \cdot \cos\beta + \cos\theta \cdot \sin\beta) + z(\sin\theta \cdot \sin\alpha \cdot \sin\beta + \cos\theta \cdot \cos\beta) - y(\sin\theta \cdot \cos\alpha) \\ x(-\cos\theta \cdot \sin\alpha \cdot \cos\beta + \sin\theta \cdot \sin\beta) + z(\cos\theta \cdot \sin\alpha \cdot \sin\beta + \sin\theta \cdot \cos\beta) + y(\cos\theta \cdot \cos\alpha) \end{bmatrix}$
- We need to do transformation matrix to move from frame x_0, z_0 to frame x_1, z_1
- $\begin{bmatrix} \cos\theta & 0 & \sin\theta \\ 0 & 1 & 0 \\ -\sin\theta & 0 & \cos\theta \end{bmatrix} \begin{bmatrix} x \\ z \\ y \end{bmatrix} = \begin{bmatrix} \cos\theta x + \sin\theta z \\ y \\ -\sin\theta x + \cos\theta z \end{bmatrix}$ $\quad$ Where, $\theta = -30^\circ$

Web Application

- Why Web app
  - Cross platform and can be ran in any operating system
  - No installation of any Software/Library to use the visualizer
- UI Libraries
  - ReactJs – Front end SPA library to create a web app
  - Plotty – 3D Graphing Library used to simulate the links of the hexapod
- Steps to run code in local server
  - Install NodeJs (Version 14 or above) from https://nodejs.org/en/
  - In the Root folder run command "npm install" wait till the installation is completed and node_modules folder is created
  - Run command "npm start" in the root folder
  - The web app will be loaded in "http://localhost:3000/"

Visit our web application here

Or

Visit : https://hexapod-mae547.web.app/

Inverse Kinematics

- For each leg: 1. Derive a few properties about the leg given what you already know which you'd later (see computeInitialProperties() for details
- This includes the coxiaPoint. If this coxiaPoint is below the ground - then there is no solution. Early exit.
- 2. Compute the alpha of this leg. see (computeAlpha()) If alpha is not within range, then there is no solution. Early exit.
- 3. Solve for beta and gamma of this leg (see LegIKSolver module) If a problem was encountered within this module, then there is no solution. Early exit.
- If the beta and gamma are not within range, then there is no solution, early exit.
- 4. Sometimes the LegIKSolver module would return a solution where the leg would not reach the target ground contact point. (this leg would be on the air) If the combination of the legs in the air would produce an unstable pose (e.g 4 legs are in the air or all left legs are in the air)
- Then there is no solution. Early exit. (see also HexapodSupportChecker)
- If no problems are encountered, we have found a solution! Return!

Team Contributions

Name	Tasks	Contribution Percentage
Ezhilan Veluchami		20%
Kirthik Roshan Nagaraj		20%
Shiva Sam Kumar		20%
Suriya Prakash Murugan		20%
Jyothsna Suresh		20%

Introduction

- No solutions exists for inverse kinematics for the below conditions
  - Coxia Point is below the ground
  - Alpha, Beta, Gamma is not within range
  - Legs are in the air or all left legs are in the air

Thank you!

Team 21

Arizona State University