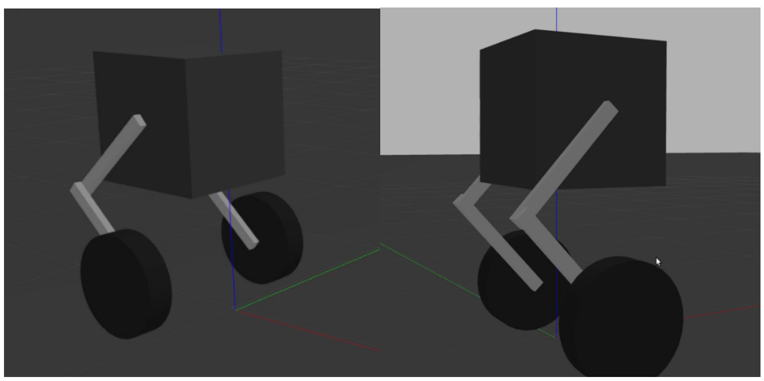
Legged wheeled robot **ABHIMANYU**

Introduction

The task is to build a two legged wheeled robot. We build a two wheeled legged robot named ABHIMANYU. It has got two revolute joints (thigh and knee) and a wheel at the bottom per leg.



Solution approach

We break the whole control of the body into tow parts.

- First part will simply control the wheels using LQR controller.
- Second to control the joint motors using PID controller.

1 Wheel controller

For this task we used Inverted pendulum analogy. referred to the Balance bot report. We calculated the height of the centre of mass from joint controller node(explained below). based on that height we calculate our state space model for balancing the robot.

$$q = \begin{bmatrix} X \\ \theta \\ \phi \end{bmatrix}$$

Here -
X - horizontal position.
θ - Pitch angle.
ϕ - Yaw angle.

State variables -

$$Y = \begin{bmatrix} \dot{q}^T & q^T \end{bmatrix}^T => Y = \begin{bmatrix} \dot{X} \\ \dot{\theta} \\ \dot{\phi} \\ X \\ \theta \\ \phi \end{bmatrix}$$

$$\dot{Y} = \begin{bmatrix} \ddot{q}^T & \dot{q}^T \end{bmatrix}^T => \dot{Y} = \begin{bmatrix} \ddot{X} \\ \ddot{\theta} \\ \ddot{\phi} \\ \dot{X} \\ \dot{\theta} \\ \dot{\phi} \end{bmatrix}$$

State space model -

$$\dot{Y} = AY + BU$$

For state space modelling, we linearize the system at its, equilibrium

$$Y := (0, 0, 0, 0, 0, 0).$$

Using Jacobian, we found A and B matrices in terms of height of centre of mass. (Used Matlab to calculate the Values of A and B, script can be found in the source code).

We then tuned the robot to get Q, R values as -

$$Q = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 10 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0.1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 100 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1000 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}, R = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$$

Based on these values we calculated the gain matrix K and then feed the value of output to the wheels

$$output = K.errorMatrix$$

Here errorMatrix is the matrix of difference of the state variables form their setpoints. For more details on this part refer to Balance bot report.

2 Joint controller

In this task, We are given the height of the robot. We had to angle the joint motors such that that robot finally stands at the given height. For this we break the total height into the height of each leg and performed the inverse kinematics to calculate the thigh and knee joint angles value. These values then feed into PID controller to control the joint angles.

2.1 Inverse kinematics

The inverse kinematics is very much similar to that of a 2R robot.

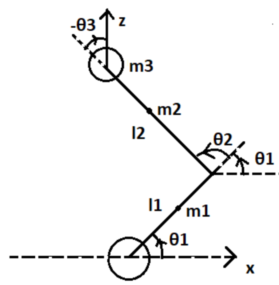


Fig 1 - line diagram of one leg of robot

Here -
m_1, m_2 - mass of lower and upper link respectively
m_3 - mass of upper body
l_1, l_2 - link length of lower and upper link respectively
θ_1 - base angle
θ_2 - knee joint angle
θ_3 - thigh joint angle

Using simple trigonometry, we can simply write the following equations -

$$X_a = 0, Z_a = 0$$

$$X_{m1} = X_a + \frac{l_1}{2}cos(\theta_1), Z_{m1} = Z_a + \frac{l_1}{2}sin(\theta_1)$$

$$X_b = X_a + l_1cos(\theta_1), Z_b = Z_a + l_1sin(\theta_1)$$

$$X_{m2} = X_b + \frac{l_2}{2}cos(\theta_1 + \theta_2), Z_{m2} = Z_b + \frac{l_2}{2}sin(\theta_1 + \theta_2)$$

$$X_c = X_{m_3} = X_b + l_2cos(\theta_1 + \theta_2), Z_c = Z_{m_3} = Z_b + l_2sin(\theta_1 + \theta_2)$$

Now, we have given the height H of the m_3 and the X_{com} should be zero i.e., on top of the base(wheel). based on these conditions we find the two equations

$$Z_{m3} = H$$

$$m_1 X_{m1} + m_3 X_{m3} + m_3 X_{m3} = 0$$

upon solving the above two equations we can obtain the value of θ_1 and θ_2 and also,

$$\theta_3 = \frac{\pi}{2} - \theta_1 - \theta_2$$

Now taking some approximation for easily solving θ_1 and θ_2, We neglected the mass of links. and both the link lengths are equal(l_0). we simply obtain -

$$\theta_1 = sin^{-1}\left(\frac{H}{2l_0}\right)$$

$$\theta_2 = \pi - 2\theta_1$$

$$\theta_3 = \frac{\pi}{2} - \theta_1 - \theta_2$$

2.2 Lean cornering motion

For leaning motion, we used Bicycle and motorcycle leaning dynamics. we calculated the lean angle based on the yaw velocity and forward velocity.

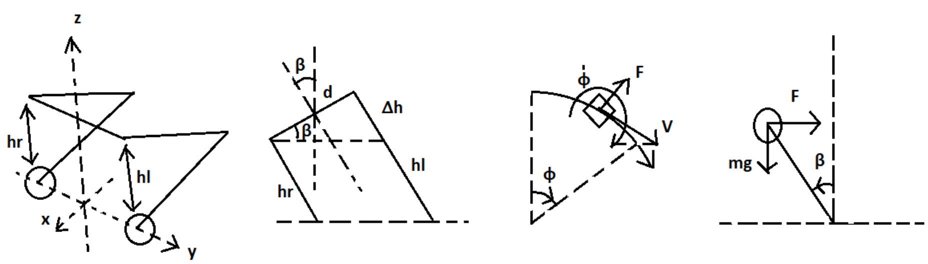


Fig 2a, 2b, 2c, 2d - lean motion

Here -
h_l, h_r - height of left and right leg respectively
Δh - difference in height $(h_l - h_r)$
d - width of the robot
β - roll angle
$\dot{\phi}$ - yaw velocity
V or $\dot{X}$ - forward velocity
F - centrifugal force.
m - mass of upper body
g = 9.80665 m/s^2

From fig 2b -

$$tan(\beta) = \frac{\Delta h}{d}$$

From fig 2c -

$$F = m\dot{X}\dot{\phi}$$

From fig 2d -

$$tan(\beta) = \frac{F}{mg} = \frac{\dot{X}\dot{\phi}}{g}$$

So we finally got the relation -

$$tan(\beta) = \frac{\Delta h}{d} = \frac{\dot{X}\dot{\phi}}{g}$$

$$\Delta h = \frac{\dot{X}\dot{\phi}}{g}d$$

After having Δh we can find $h_l and h_r$ by -

$$h_l = H + \frac{\Delta h}{2}, h_r = H - \frac{\Delta h}{2}$$

where H is the given height. and the $h_{com} = H = \frac{h_l+h_r}{2}$, here h_{com} is height of centre of mass from the wheel axis.

2.3 Roll disturbance rejection

For roll disturbance rejection we came up with a very simple solution. we go for P controller to control the difference of height of both the legs based on roll error.

$$\Delta h^{'} += K_p tan(\beta - \beta_0)d$$

and

$$\Delta h_{total} = \Delta h + \Delta h^{'}$$

After having Δh_{total} we now have $h_l and h_r$ by -

$$h_l = H + \frac{\Delta h_{total}}{2}, h_r = H - \frac{\Delta h_{total}}{2}$$

Here -
β_0 - roll setpoint
β - roll angle
d - robot width
K_p - proportionality constant

3 Jumping

For now, we went for a much naive approach to this problem. we implemented the jumping by controlling the rate of height of the robot. the process is pretty similar to how a person jump. he sit down a little then rapidly go up and then pull his leg up. this is how a human jump in a nut shell. we used this three phase to code the jumping of our robot.

Let me describe one simulation tested jump manoeuvre for our robot.

- First it decent it's height to 0.5m with the rate of 0.5m/s
- Then it go up till height of 0.8m with the rate of 2.5m/s. this sudden manoeuvre is able to make the robot airborne.
- Lastly as soon as it left the ground, it then pulls its legs upward with a rate of 2m/s until robot's height become 0.5m again.

Now the above manoeuvre made our robot jump vertically at it's position. now suppose we give the robot some forward velocity, it will then act as an projectile. it jumps upward while going forward due to it's forward velocity.

In upcoming stage to are intended to implement Spring Loaded Inverted Pendulum(SLIP) analogy to perform a much accurate and sophisticated jumping manoeuvre.

4 Future Work

- Improving the controller for the robot.
- Improving jumping using proper SLIP model.
- Filter designing ans noise adding.
- Hardware implementation.

References

[1] Modeling and control of two-legged wheeled robot, *by Adam Kollarčík January 2021*

[2] Dynamic Modeling of a Two-wheeled Inverted Pendulum Balancing Mobile Robot, *International Journal of Control, Automation, and Systems (2015)*

[3] Ascento: A Two-Wheeled Jumping Robot, *IEEE, ICRA 2019*

[4] LQR-Assisted Whole-Body Control of a Wheeled Bipedal Robot with Kinematic Loops, *IEEE ROBOTICS AND AUTOMATION LETTERS. PREPRINT VERSION. ACCEPTED JANUARY, 2020*

[5] Modelling and Control of a Hybrid Wheeled Jumping Robot, *2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*

[6] Hardware and control design of a ball balancing robot, *2019 22nd International Symposium on Design and Diagnostics of Electronic Circuits and Systems DDECS*

[7] Feedback Control of Dynamic Systems, *by Gene F. Franklin — J. David Powell — Abbas Emami-Naeini*

[8] MODERN ROBOTICS MECHANICS, PLANNING, AND CONTROL, *by Kevin M. Lynch and Frank C. Park1*