PROJECT REPORT

Project Title :	Design and Testing of a Wheeled Biped	
Team Members :	L Shiva Rudra	2020A8PS1246H
	Juloori Amogh	2020AAPS0422H
	Tanay Ranjan	2020A3PS0483H
Course Title:	ME F366 Formal Project	
Course Instructor:	Dr. Abhishek Sarkar	

Abstract

To build a biped on two wheels with self balance and jumping modes. It should be able to switch between both modes according to the necessity. Outline to achieve the objective:

- To establish a mathematical model for both modes based on the system kinematics and dynamics and make controller design for both cases and figure out a way to integrate them using sensors such as accelerometer. This is to be done using MATLAB/Simulink which can help in estimating specifications of motors and sensors.
- To estimate the essential values such as robot dimensions, mechanism design, CAD tool Onshape is to be used.
- Importing the CAD model from Onshape into Simulink allows for establishing the control in the model using MATLAB/Simscape. This also allows for simulating before real-time testing.
- Building the physical robot and performing real-time testing.

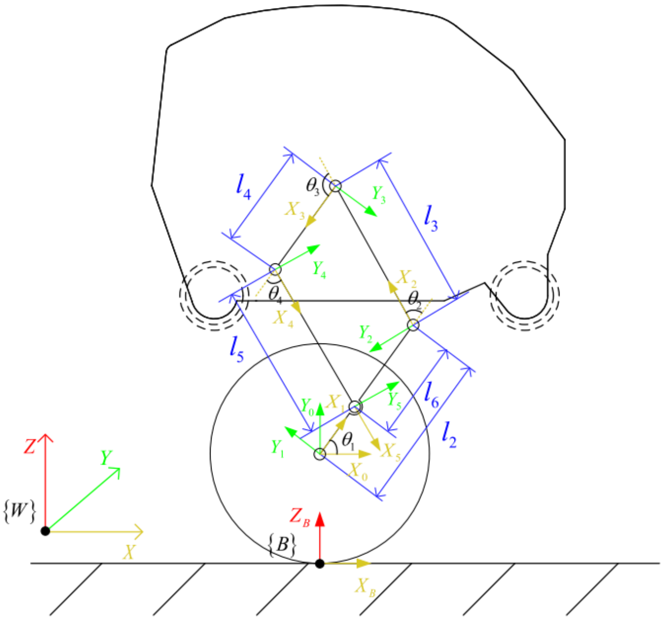
Literature Review

A parallel four-bar mechanism is being used as a new type of wheel-legged robot, and two controllers are designed to ensure stable motion - the linear quadratic regulator (LQR) and fuzzy proportion differentiation (PD) jumping controller. The robot is equipped with the ability to jump over obstacles and navigate rough terrain. The height of the jump trajectory adjusts based on changes in the link angle of the parallel four-bar linkage mechanism, enhancing the robot's ability to tackle vertical obstacles.

The two modes of operation of the system:

- **Self-balance:** using LQR controller.
  - To linearize the nonlinear state-space model about the equilibrium point(taking angular conditions as zero). Straight line motion(no steering can be assumed in the initial stage).
- **Jumping:** using a Fuzzy controller.
  - To use Fuzzy conditions to switch between Stance and Flight stages and get the controller output accordingly.

The kinematics of the robot are given as follows:



Self-Balance Mode:

The robot's balance and speed are regulated using the Linear Quadratic Regulator (LQR) technique. The equation for the state of the linear time-invariant system is: x_dot(t) = Ax(t) + Bu(t). The LQR controller works only on linear systems. But, our system is nonlinear with dynamics given as follows:

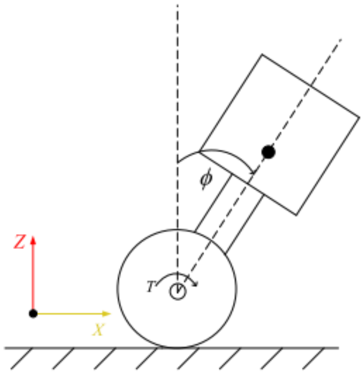
$$\ddot{\phi} = \frac{\begin{array}{c}(a_{21}\cos\phi - a_{11})(T_L + T_R) + \\ (a_{11}a_{24} - a_{15}a_{21}\cos\phi)\,\dot{x}_b - (a_{14}a_{21} + a_{11}a_{25})\,\dot{\phi} \\ + a_{13}a_{21}\dot{\phi}^2 \sin\phi\cos\phi + a_{11}a_{22}\sin\phi\end{array}}{a_{11}a_{23} + a_{12}a_{21}\cos^2\phi}$$

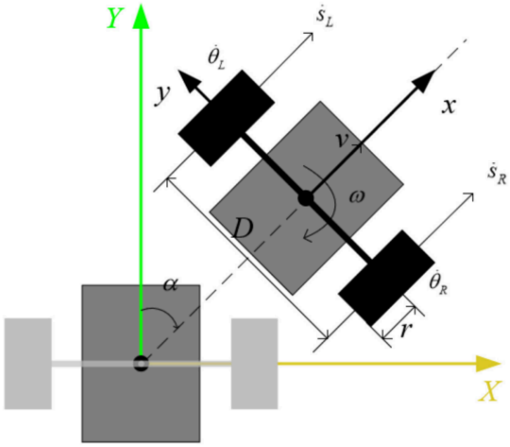
$$\ddot{x}_b = \frac{\begin{array}{c}-(a_{23} + a_{12}\cos\phi)(T_L + T_R) - a_{13}a_{23}\dot{\phi}^2 \sin\phi + \\ (a_{14}a_{23} - a_{12}a_{25}\cos\phi)\,\dot{\phi} \\ + (a_{15}a_{23} + a_{12}a_{24}\cos\phi)\,\dot{x}_b + a_{12}a_{22}\sin\phi\cos\phi\end{array}}{a_{11}a_{23} + a_{12}a_{21}\cos^2\phi}$$

$$\ddot{\alpha} = \frac{T_L - T_R}{a_{31}}$$

On converting it into matrix form by taking 6 states:

$$\begin{bmatrix} \dot{x}_b \\ \ddot{x}_b \\ \dot{\phi} \\ \ddot{\phi} \\ \dot{\alpha} \\ \ddot{\alpha} \end{bmatrix} = \begin{bmatrix} 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & A_{22} & A_{23} & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & A_{42} & A_{43} & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} x_b \\ \dot{x}_b \\ \phi \\ \dot{\phi} \\ \alpha \\ \dot{\alpha} \end{bmatrix} + \begin{bmatrix} 0 & 0 \\ B_2 & B_2 \\ 0 & 0 \\ B_4 & B_4 \\ 0 & 0 \\ B_6 & -B_6 \end{bmatrix} \begin{bmatrix} T_L \\ T_R \end{bmatrix}$$





where:

$$
\begin{aligned}
&\Delta = r^2(2M^2L^2 + 2m_bML^2 - 2m_bI_b - MI_b) + 2ML^2I_w - 2I_wI_b \\
&A_{22} = 2(bML^2 - bI_b + MLrb)/\Delta \\
&A_{23} = M^2gL^2r^2/\Delta \\
&A_{42} = (4brm_b + 2Mbr + 4bI_w/r - 2MbL)/\Delta \\
&A_{43} = (2Mm_bgLr^2 + M^2gLr^2 + 2I_wMgL)/\Delta \\
&B_2 = (-ML^2r + I_br - MLr^2)/\Delta \\
&B_4 = (-2m_br^2 - Mr^2 - 2I_w + MLr)/\Delta \\
&B_6 = Dr/(D^2I_w + m_bD^2r^2 + 2I_tr^2)
\end{aligned}
$$

Here,

- xb is the forward displacement of the point of contact with the ground
- alpha is the steering angle
- phi is the tilt angle of the robot w.r.t. the vertical axis.
- M is mass of the robot
- L is the length of the robot
- mb is the mass of the wheel
- Ib is the moment of inertia of the robot abt y-axis
- Iw is the moment of inertia of the wheel
- r is the radius of the wheel
- g is the acceleration due to gravity
- D is the wheel to wheel distance
- It is the moment of inertia of the robot about vertical direction

Jumping mode:

The robot's jump control utilizes fuzzy Proportional Differentiation (PD) control.

Model is to be built based on the dynamic equation of the jump stage:

$$M(q)\ddot{q} + \underbrace{V(\dot{q}, q) + G(q)}_{H(\dot{q},q)} = \tau$$

Mathematical Model for Jumping Mode

The following equations are used to find out the torque of the hip joint as a function of its angle:

$$\begin{cases} K_0 = \frac{1}{2} m_b \dot{X}_B^2 \\ K_1 = \frac{1}{2} m_{c1} (\dot{X}_{c1}^2 + \dot{Z}_{c1}^2) + \frac{1}{2} I_{c1} \dot{\theta}_1^2 \\ K_2 = \frac{1}{2} m_{c2} (\dot{X}_{c2}^2 + \dot{Z}_{c2}^2) + \frac{1}{2} I_{c2} (\dot{\theta}_1 + \dot{\theta}_2)^2 \\ K_3 = \frac{1}{2} m_{c3} (\dot{X}_{c3}^2 + \dot{Z}_{c3}^2) \\ K_4 = \frac{1}{2} m_{c4} (\dot{X}_{c4}^2 + \dot{Z}_{c4}^2) + \frac{1}{2} I_{c4} (\dot{\theta}_1 + \dot{\theta}_2 + \dot{\theta}_3 + \dot{\theta}_4)^2 \\ K_5 = \frac{1}{2} m_{c5} (\dot{X}_{c5}^2 + \dot{Z}_{c5}^2) + \frac{1}{2} I_{c5} \dot{\theta}_3^2 \end{cases}$$

$$V = 2m_{c1}gZ_{c1} + 2m_{c2}gZ_{c2} + m_{c3}gZ_{c3} + 2m_{c4}gZ_{c4} +$$

$$2m_{c5}gZ_{c5} + 2m_b g Z_b$$

$$L = 2K_0 + 2K_1 + 2K_2 + K_3 + 2K_4 + 2K_5 - V$$

Here :

- K0 is the kinetic energy of one wheel
- Ki is the kinetic energy of ‘ith’ link of a single 4-bar linkage
- K is the total kinetic energy of on each side(left and right) of the bot
- V is the total potential energy
- L is the Lagrangian

$$\tau = \left[\frac{d}{dt} \left(\frac{\partial K}{\partial \dot{\theta}_3} \right) - \frac{\partial K}{\partial \theta_3} + \frac{\partial V}{\partial \theta_3} \right] / 2$$

Where **τ** is the torque of the hip joint.

Methodology

Self-balance Mode

Calculating the length of the robot(L):

From the kinematics schematic of the robot, the following transformation matrices can be found out:

$$ {}^{0}_{1}T = \begin{bmatrix} C_1 & -S_1 & 0 & 0 \\ S_1 & C_1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}, {}^{0}_{2}T = \begin{bmatrix} C_{12} & -S_{12} & 0 & C_1 l_2 \\ S_{12} & C_{12} & 0 & S_1 l_2 \\ 0 & 0 & 1 & -d_1 - d_2 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (1) $$

$$ {}^{0}_{3}T = \begin{bmatrix} C_{123} & -S_{123} & 0 & C_1 l_2 + C_{12} l_3 \\ S_{123} & C_{123} & 0 & S_1 l_2 + S_{12} l_3 \\ 0 & 0 & 1 & -d_1 - d_2 - d_3 \\ 0 & 0 & 0 & 1 \end{bmatrix}, $$

$$ {}^{0}_{4}T = \begin{bmatrix} C_{1234} & -S_{1234} & 0 & C_1 l_2 + C_{12} l_3 + C_{123} l_4 \\ S_{1234} & C_{1234} & 0 & S_1 l_2 + S_{12} l_3 + S_{123} l_4 \\ 0 & 0 & 1 & -d_1 - d_2 - d_3 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (2) $$

$$ {}^{0}_{5}T = \begin{bmatrix} C_{1234} & -S_{1234} & 0 & C_1 l_2 + C_{12} l_3 + C_{123} l_4 + C_{1234} l_5 \\ S_{1234} & C_{1234} & 0 & S_1 l_2 + S_{12} l_3 + S_{123} l_4 + S_{1234} l_5 \\ 0 & 0 & 1 & d_5 - d_1 - d_2 - d_3 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (3) $$

As coordinate {3} represents the motor's position and {0} represents the wheel position, so, the length of the robot can be found by using the [0T3] matrix. (Assuming that center of mass of the robot lies near motor)

The fourth column in [0T3] represents the coordinates of {3} in {0}. So, using distance formula,

L=sqrt(sum of squares of first and second value in last column of [0T3]).

$$ \Rightarrow L^2 = \left[l_2 cos\theta_1 + l_3 cos\left(\theta_1 + \theta_2\right) \right]^2 + \left[l_2 sin\theta_1 + l_3 sin\left(\theta_1 + \theta_2\right) \right]^2 $$

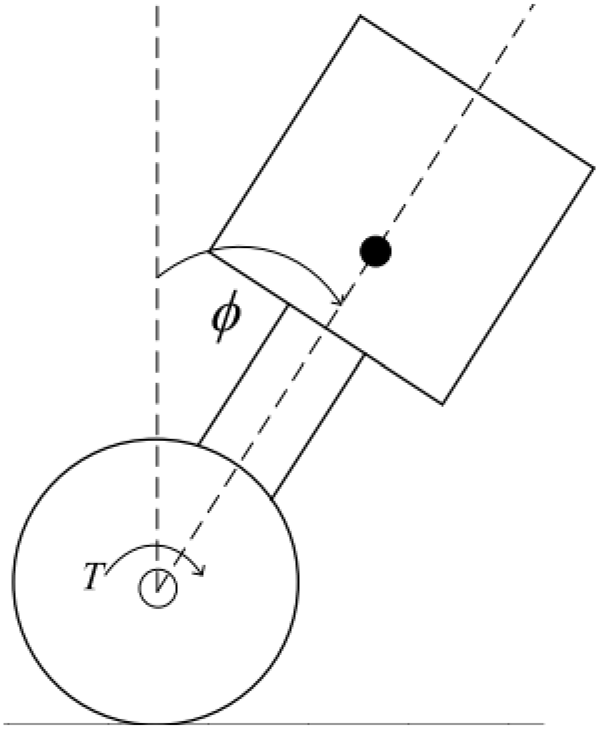
$$ \Rightarrow L^2 = l_2^2 + l_3^2 + 2 l_2 l_3 cos\theta_2 $$

Maximum value of L will occur at $\theta_2 = 0$

$$ \Rightarrow L_{max} = l_2 + l_3 = 0.28\, m $$

Calculating the mass of the robot (M):

The Mass of body is calculated as follows -



Upper Half of body consists of -

1. 2 T-Motors - $180\ g \times 2 = 360\ g$
2. Lipo 6s (As T Motors requires 6s Lipo) - $720\ g$
3. Microcontroller (Arduino) - $25\ g$
4. Microprocessor (Raspberry Pi) - $10\ g$
5. Buck Converter - $11\ g$
6. 2 40 A ESC - $30\ g \times 2 = 60\ g$

Lower Half of body consists of 2 Motors + Wheel (Negligible) = $360\ g$

Total Mass $M = 1546\ g$

Lower Body Mass $mb = 360\ g$

For safety factor, we will consider 2 times of the calculated mass

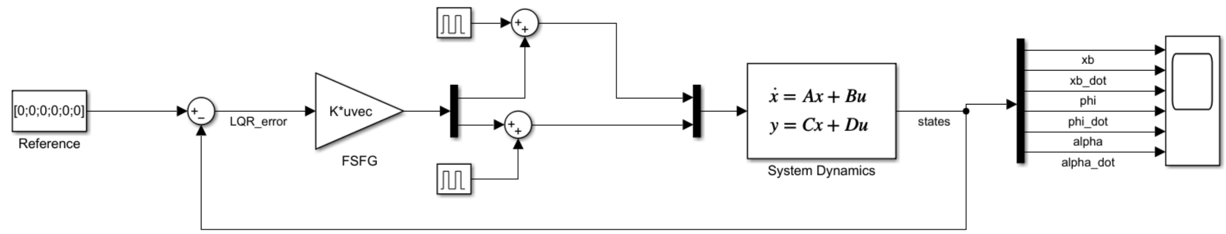
Therefore $M = 3\ kg,\ mb = 0.72\ kg$

So the values are taken as follows:

- r=0.03m

- M=3 kg
- mb=0.72 kg
- Ib=M*L*L
- Iw=4.5125*0.0001
- It=0.0225
- b=0.75
- g=9.82ms-2
- D=0.3m

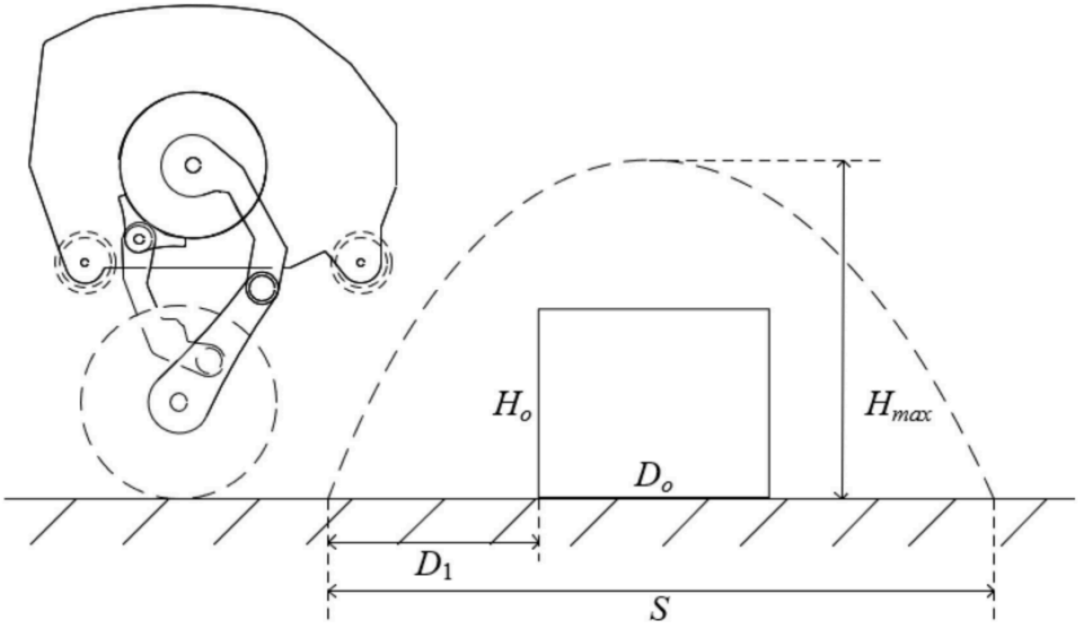
The following model is made for the self-balancing mode using LQR controller:



- Disturbance(using Pulse Generator) is added to the Plant input.
- The six states are observed in the o/p scope.

Jumping Mode

Trajectory Planning:



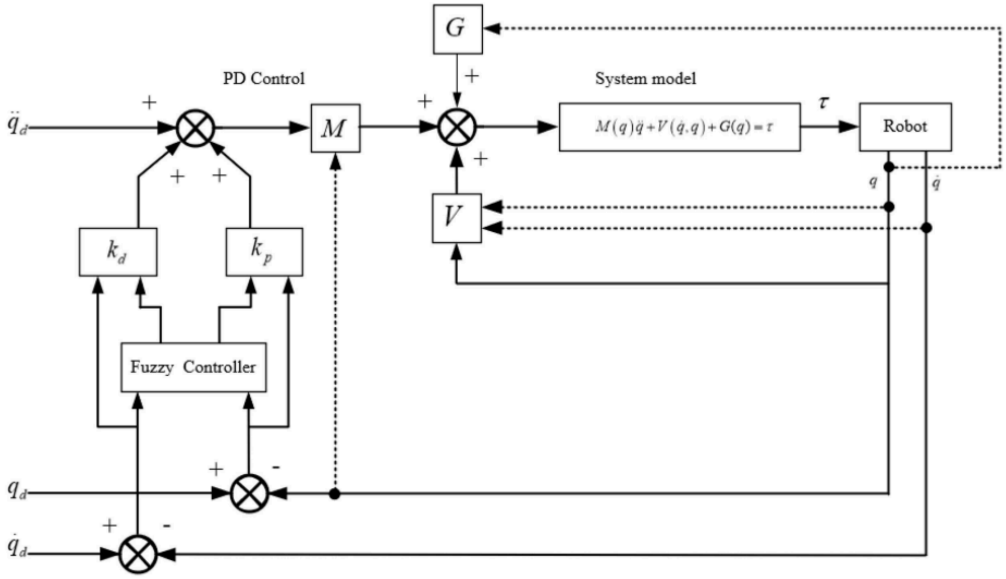
- We have taken Hmax as 0.5 m. This is used to estimate the vertical velocity of the robot when it is about to take off.
- S is taken as 0.2 m. This is used to estimate the horizontal velocity of the robot throughout the projectile.

$$H_{\max} = \frac{V_y^2}{2g} > H_o$$

$$y(t) = a_0 + a_1 t + a_2 + t^2 + a_3 t^3$$

- We used this as the trajectory of the vertical component of the COM(hip joint) during the takeoff.

Control Block Diagram for Jumping Mode:

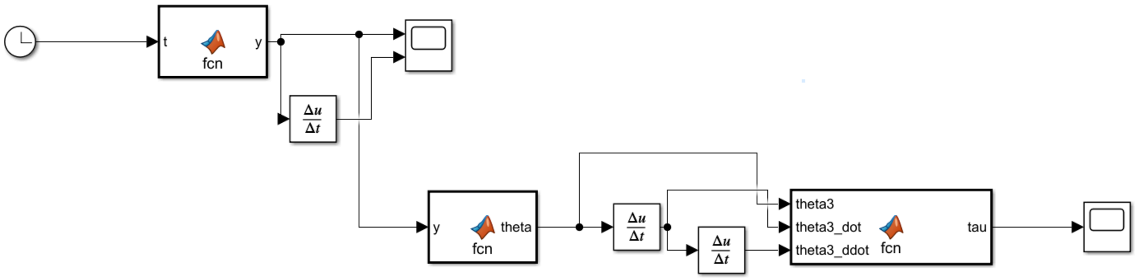


This is the control block diagram which we used to control the hip joint angle and its derivative during the flight phase.

- qd, qd_dot and qd_ddot are the reference inputs. Where q represents the hip joint angle.
- Fuzzy rules are made to tune the delta_Kp and delta_Kd values which get added to constant Kp0 and Kd0 values to get PD gains.
- Torque of the system while performing this control system during the takeoff is given by:

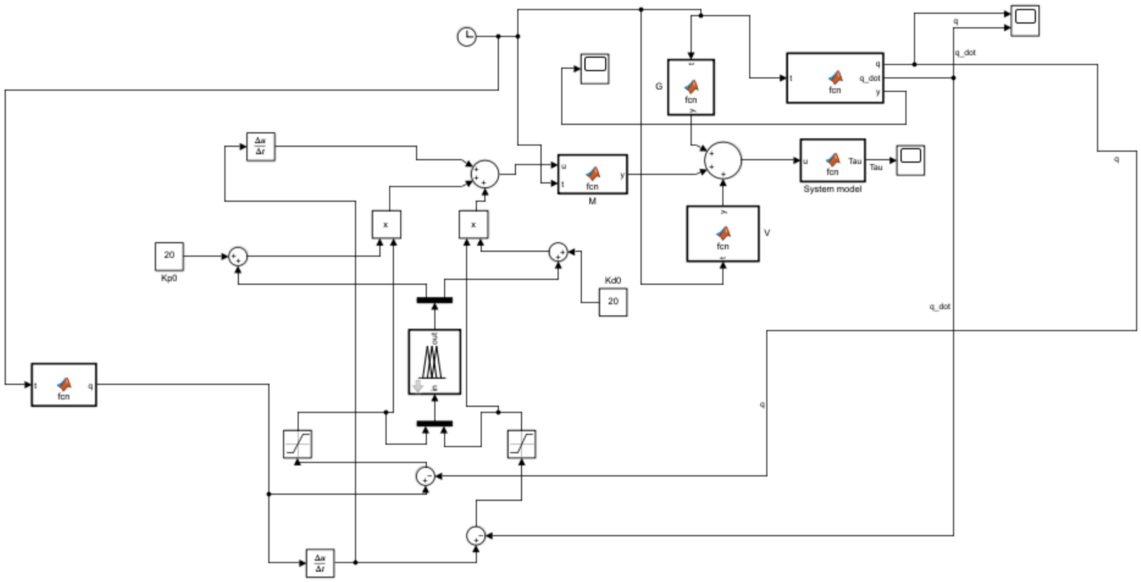
Steps followed to establish this system:

- Obtained the Lagrangian from the KE and PE equations. This involves finding the transformation matrices of the COM of all the links wrt the world frame {W}.
- Next, we derived the equation of torque in terms of theta3, theta3_dot and theta3_ddot where theta3 is the hip joint angle.
- We segregate the terms in the torque equation to find the following matrices:
  - Inertia matrix: M(q)
  - Coriolis force matric: V(q,q_dot)
  - Gravitational force matric: G(q)
- We used this information to build the following Simulink models:



This estimates the torque as a function of time taking takeoff time as 0.45 seconds.

Control Design for q, q_dot control during takeoff:

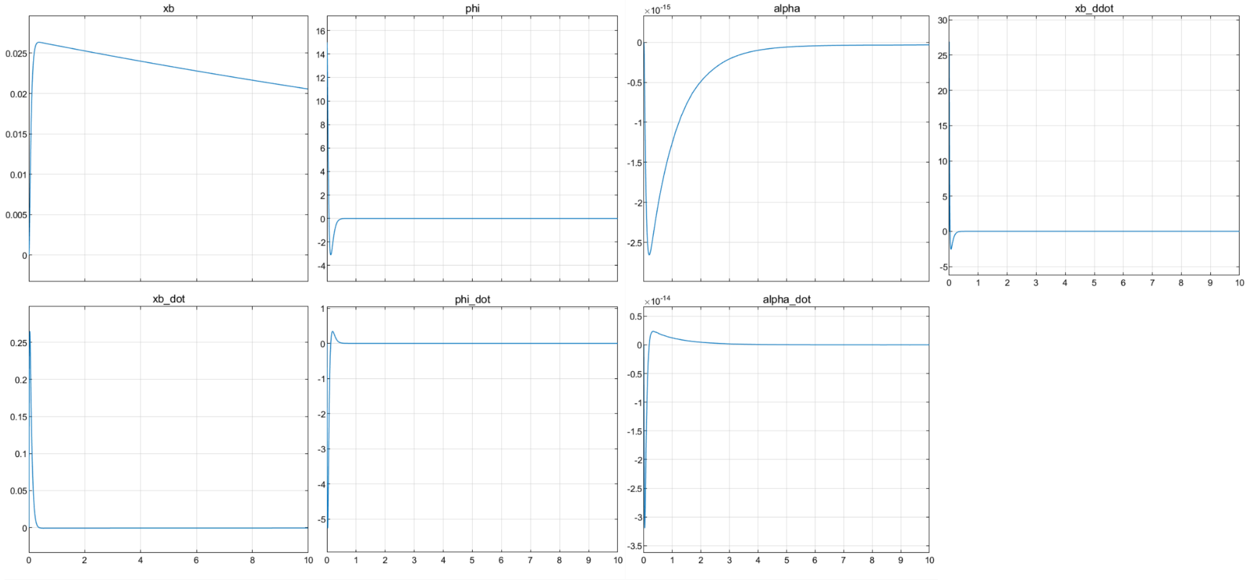


We built this system to control the q, q_dot while takeoff where q is the hip joint angle. But due to certain issues like the number of rules and membership functions of the fuzzy controller and due to the lack of proper reference trajectory equations, we could not get proper results for the torque and the control variables:

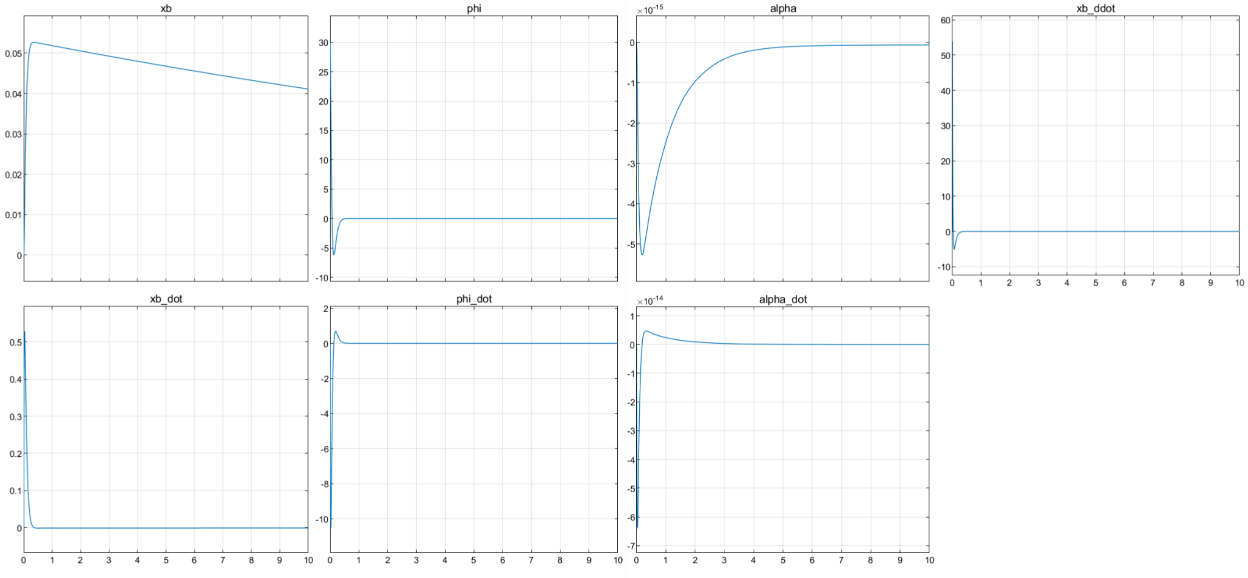
Results

For testing, we set initial angles of robot as 15°, 30°, and 45°, and then we observed how it recovers by accelerating its bottom wheel

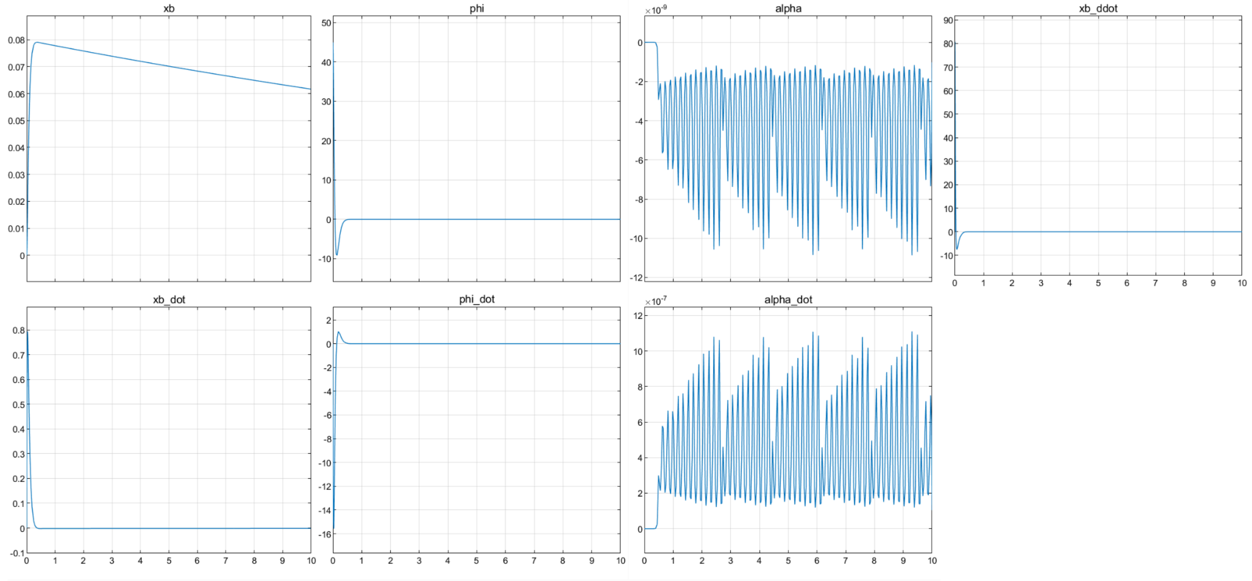
15° -



30° -



45° -

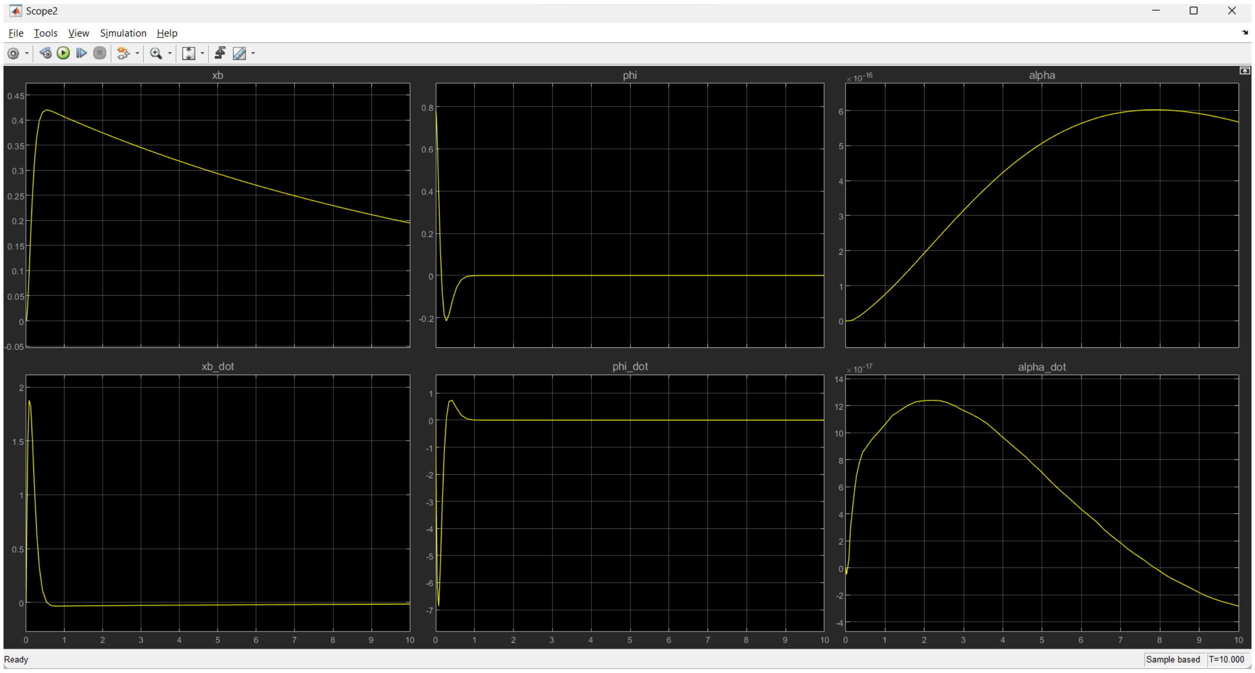


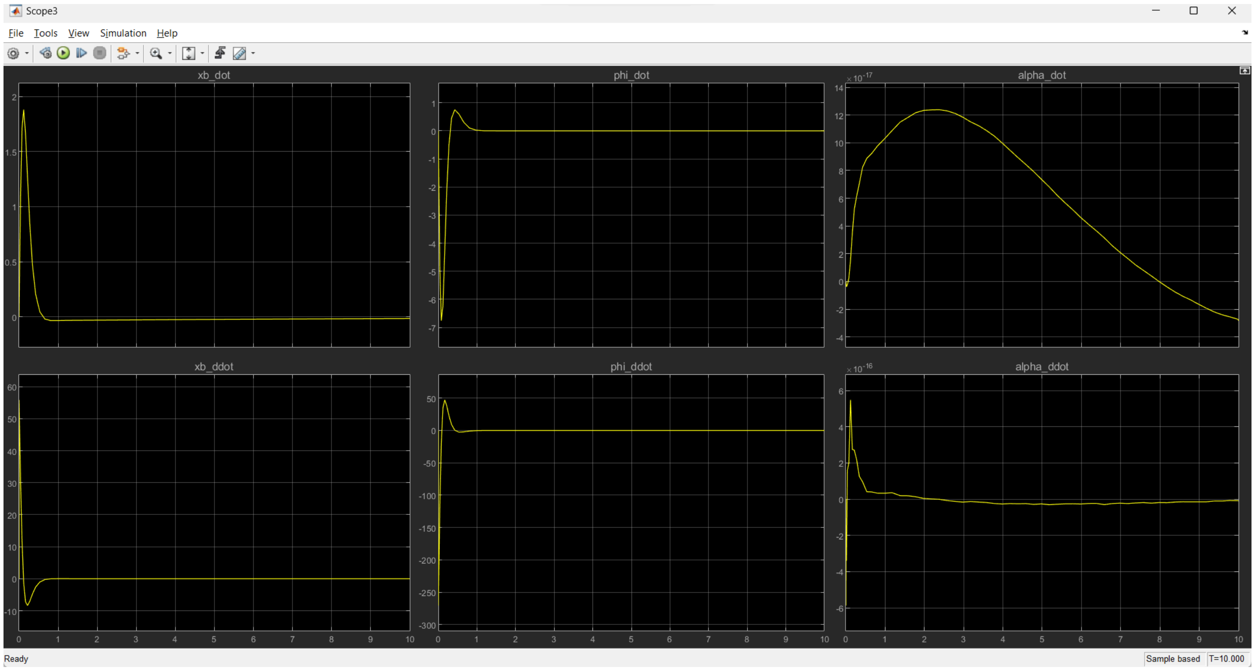
Using these results and aiming that the robot should be able to recover from 45° with max leg length, the specifications of motors for forward translational motion are found as:

- Max RPM = 254.77 RPM
- Max Torque = 9 N-m

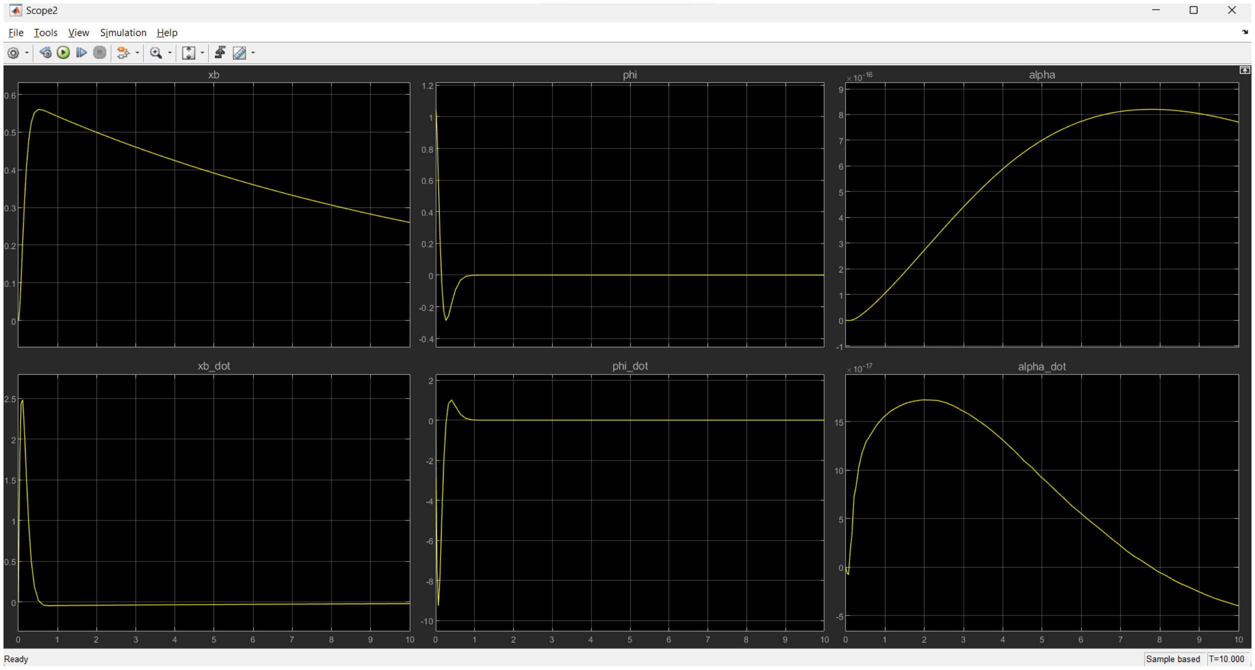
Self balance Mode

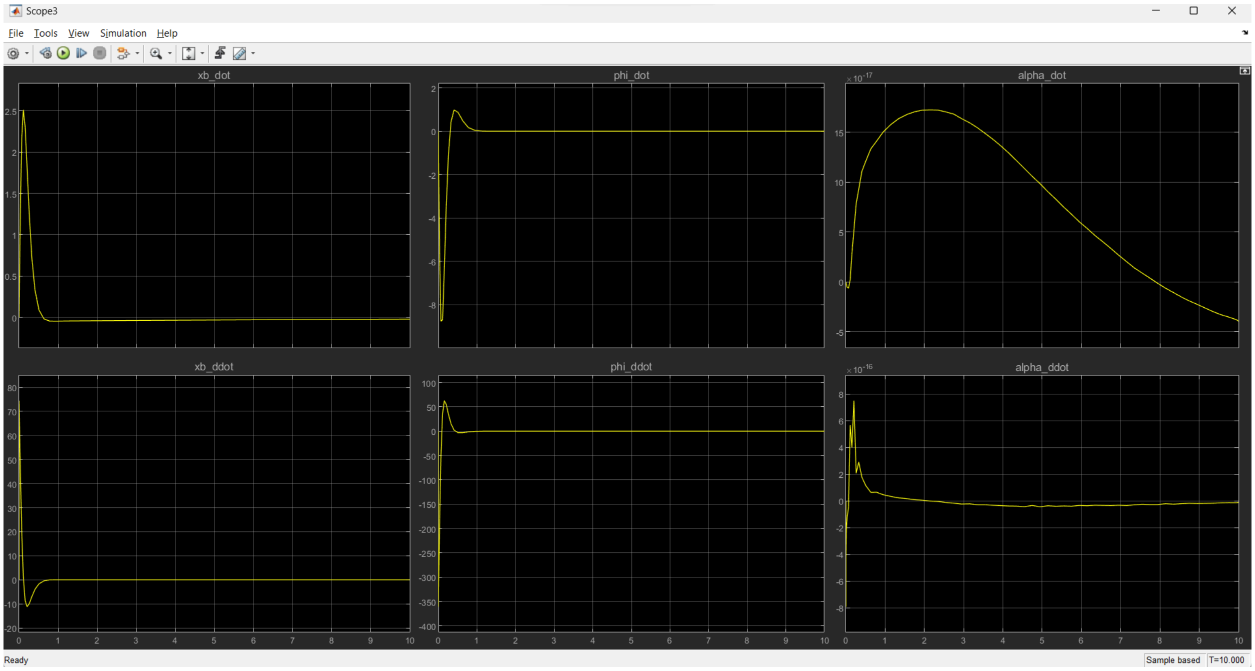
Initial Angle=45 degrees, Hip angle=120 degrees





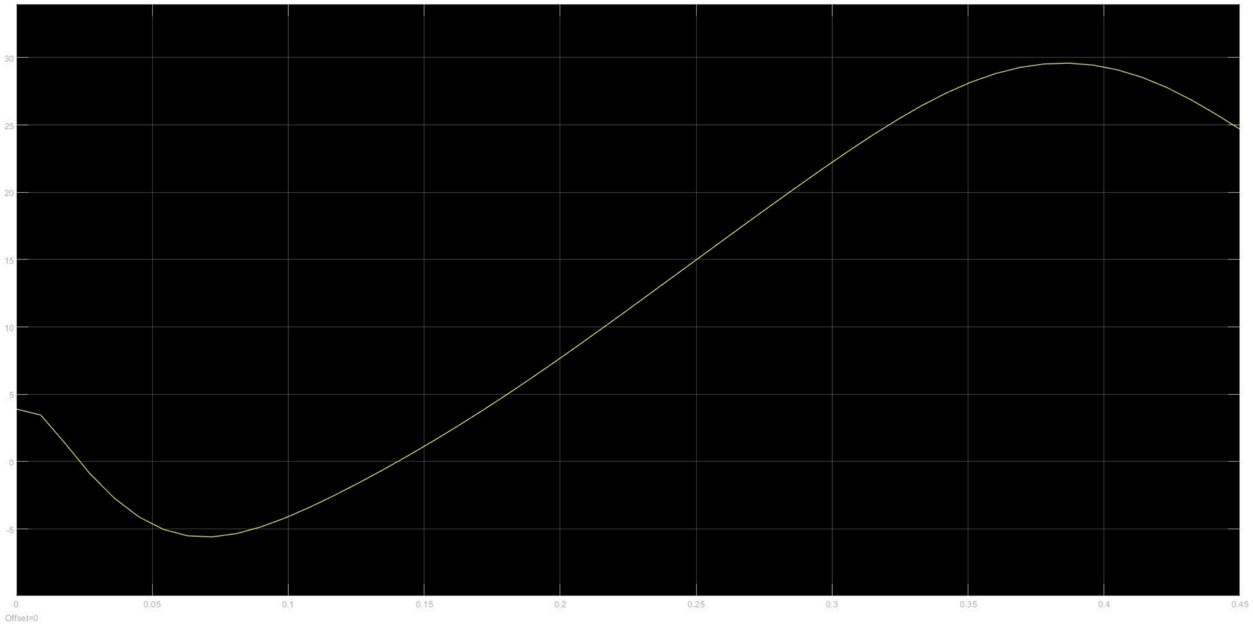
Initial Angle=60 degrees, Hip angle=120 degrees



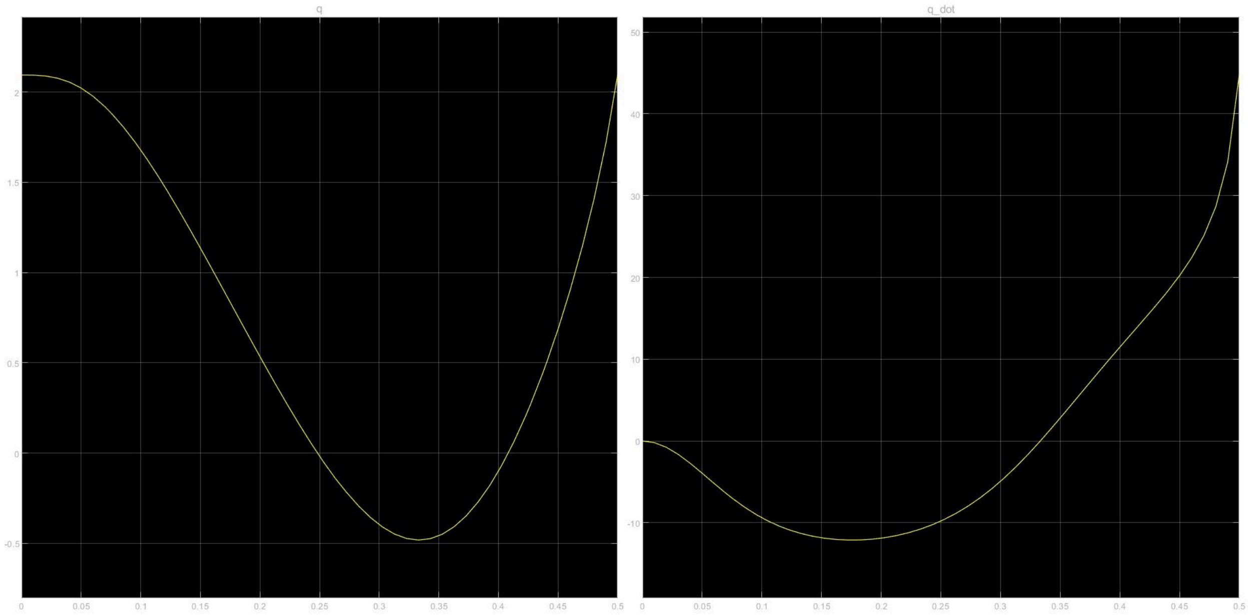


The simulation results can be used to obtain the motor specifications.

Jumping Mode



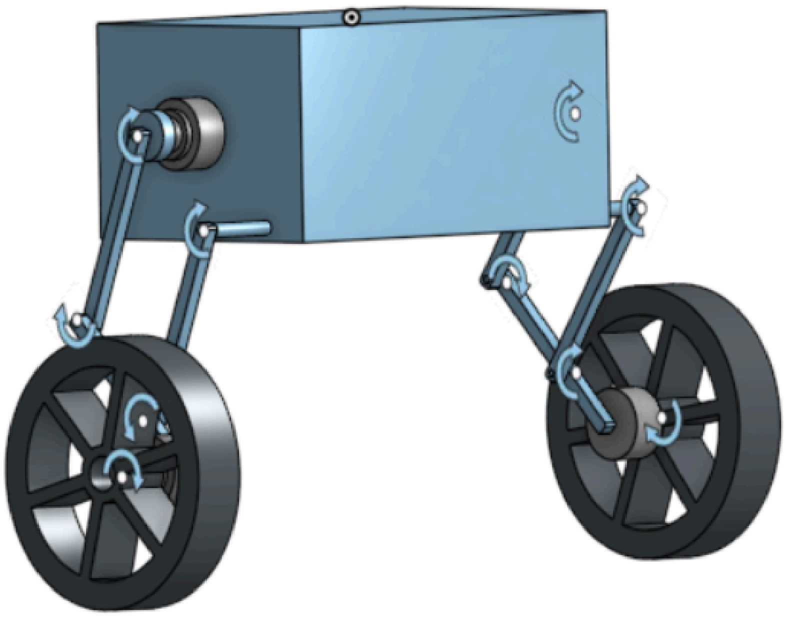
This is the variation of the torque. We can clearly see that the torque increases in the negative direction while retraction and then increases in the positive while jumping. (Note that the jumping phase involves retraction followed by jumping).

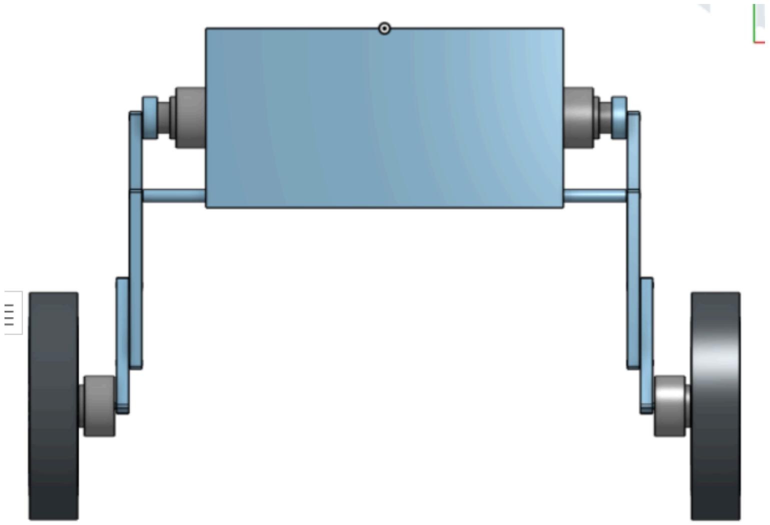


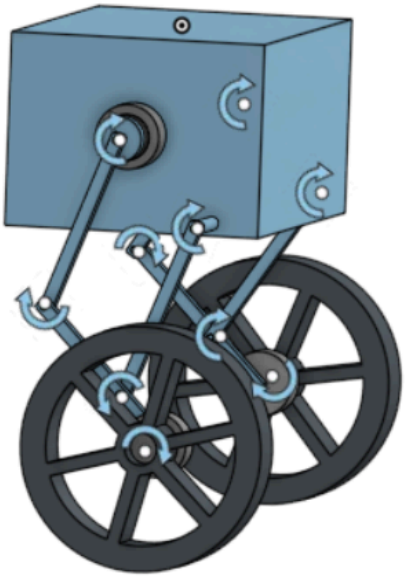
These are the plots for the q and q_dot. The parabolic projectile reference is taken into consideration.

CAD

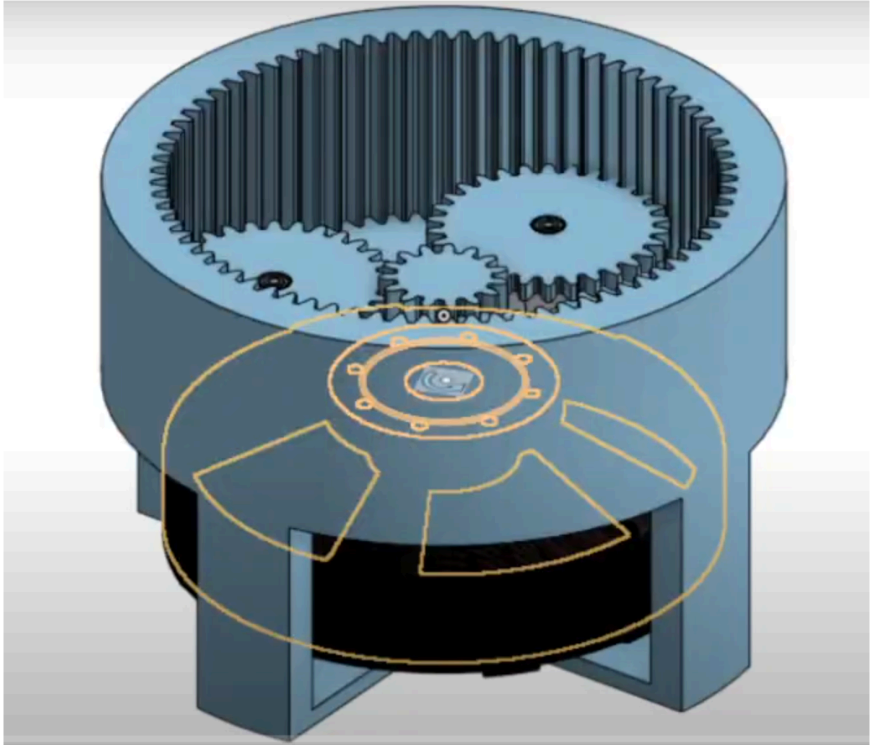
The CAD model for the robot is as follows:







Gearbox



Conclusion

- We learned the mathematical modeling of the wheeled biped and its implementation using LQR and Fuzzy PD controllers.
- We would like to work on the simulation of the landing phase and integrate both the Self balancing and Jumping modes in a more robust manner.
- We also plan to obtain proper values of moment of inertias, and masses of various links to obtain the desired control and minimize the error of trajectory while considering the physical constraints.
- The fabrication and testing of the physical system of the robot would be our target for the future.

References

- Design and dynamic analysis of jumping wheel-legged robot in complex terrain environment