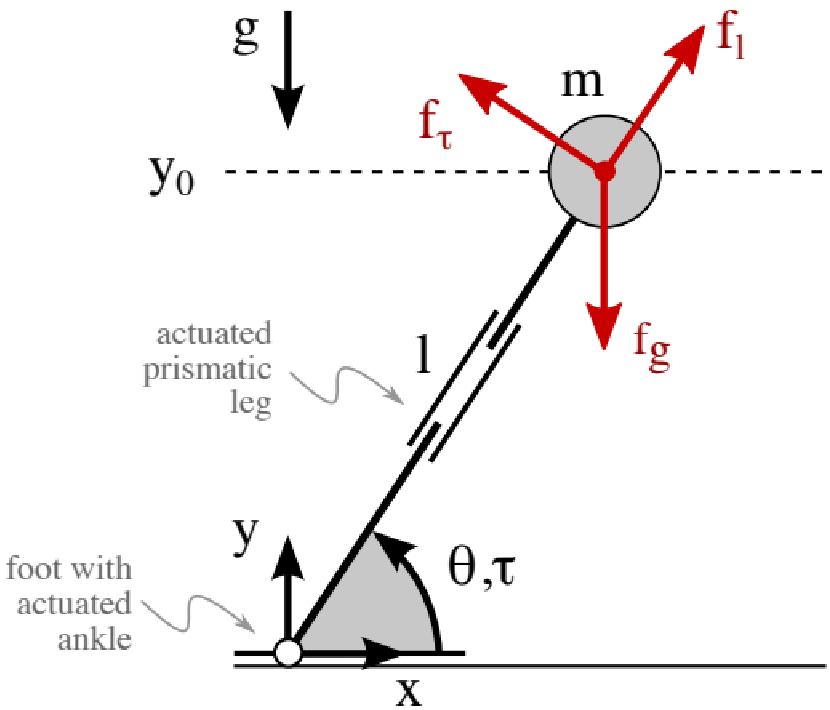
Submitted by: Shahram Najam Syed **Dated: 30th November 2023**
AndrewID: snsyed **Homework # 4 (16-665 Robot Mobility)**

Part 1:

Q1.

Ans: The figure shown below, is the diagram of a Linear Inverted Pendulum Model (LIPM) with an actuated foot.



To derive the equations of motion for this system, we will analyze the forces acting on the point mass m in both the horizontal (x) and vertical (y) directions.

i. Decompose the actuator force f_l:
The actuator force f_l is applied along the length of the leg. To decompose it into f_{lx} and f_{ly}, we use vector decomposition:

$$f_{lx} = f_l \cos(\theta)$$
$$f_{ly} = f_l \sin(\theta)$$

Here, θ is the angle of the leg with the horizontal.

ii. Decompose force f_τ:
The force f_τ resulting from the actuator torque at the angle, and can be decomposed into its components of x and y as follows:

$$f_{\tau x} = f_\tau \cos(90 - \theta) = f_\tau \sin(\theta)$$
$$f_{\tau y} = f_\tau \sin(90 - \theta) = f_\tau \cos(\theta)$$

iii. The force due to gravitational force acting directly downwards with a magnitude of m.g, and thus has no component in the x-direction since it acts vertically. Therefore,

$$f_{gx} = 0$$
$$f_{gy} = mg$$

The net forces in the x direction can then be written as:

$$f_x = f_{lx} - f_{\tau x}$$

$$f_x = f_l \cos(\theta) - f_\tau \sin(\theta)$$

Similarly, the net forces in the y direction can then be written as:

$$f_y = f_{ly} + f_{\tau y} - f_{gy}$$

$$f_y = f_l \sin(\theta) + f_\tau \cos(\theta) - mg$$

Given the condition that the vertical position stays constant at $y = y_0$, this implies that the net force in the y-direction must be zero to prevent vertical acceleration. Therefore, $f_y = 0$ which leads to:

$$f_l \sin(\theta) + f_\tau \cos(\theta) - mg = 0$$

Q2.

Ans: To express the net forces f_x and f_y as a function of $x, y_0, l, m, g, f_l, and\ \tau$, lets start by defining $\cos(\theta)$ and $\sin(\theta)$ in terms of the geometry of the inverted pendulum.

Given that θ is the angle between the leg and the horizontal, we can define:

$$\cos(\theta) = \frac{x}{l}$$

$$\sin(\theta) = \frac{y_0}{l}$$

Also, the relationship between the torque τ and the force f_τ is given by the lever arm equation:

$$\tau = f_\tau . l$$

Where, l is the length from the point where the force is applied (the foot) to the axis of rotation (the ankle). This means that,

$$f_\tau = \frac{\tau}{l}$$

Now using the requisite derived equations above, we can express f_x and f_y using these relationships:

$$f_x = \frac{f_l\, x}{l} - \frac{\tau\, y_0}{l^2}$$

$$f_y = \frac{f_l\, y_0}{l} + \frac{\tau\, x}{l^2} - mg$$

Given the condition that the vertical position stays constant at $y = y_0$, this implies that the net force in the y-direction must be zero to prevent vertical acceleration. Therefore, $f_y = 0$ which leads to:

$$\frac{f_l\, y_0}{l} + \frac{\tau\, x}{l^2} = mg$$

Q3:

Ans: Solving for f_l using the horizontal motion constraint that requires zero net force in the vertical direction ($f_y = 0$), we can use the following equation (derived in the previous part of this question):

$$\frac{f_l\, y_0}{l} + \frac{\tau\, x}{l^2} = mg$$

$$\frac{f_l\, l\, y_0 + \tau\, x}{l^2} = mg$$

$$f_l\, l\, y_0 + \tau\, x = mgl^2$$

$$f_l\, l\, y_0 = mgl^2 - \tau\, x$$

$$f_l = \frac{mgl^2 - \tau\, x}{l\, y_0}$$

Q4:

Ans: Since we determined the expression for f_x as follows:

$$f_x = \frac{f_l\, x}{l} - \frac{\tau\, y_0}{l^2}$$

To find the expression for f_x as a function of $x, y_0, m.g, and\ \tau$, we will start by substituting our expression for f_l into the equation for f_x:

$$f_x = \frac{\left(\frac{mgl^2 - \tau\, x}{l\, y_0}\right) x}{l} - \frac{\tau\, y_0}{l^2}$$

$$f_x = \frac{(mgl^2 - \tau\, x)\, x}{l^2 y_0} - \frac{\tau\, y_0}{l^2}$$

$$f_x = \frac{(mgl^2 - \tau\, x)\, x}{l^2 y_0} - \frac{\tau\, y_0}{l^2}$$

$$f_x = \frac{(mgl^2 x - \tau\, x^2)}{l^2 y_0} - \frac{\tau\, y_0}{l^2}$$

$$f_x = \frac{mgx}{y_0} - \frac{\tau\, x^2}{l^2 y_0} - \frac{\tau\, y_0}{l^2}$$

Factoring out τ and l^2:

$$f_x = \frac{mgx}{y_0} - \frac{\tau}{l^2}\left(\frac{x^2}{y_0} + y_0\right)$$

$$f_x = \frac{mgx}{y_0} - \frac{\tau}{l^2}\left(\frac{x^2 + {y_0}^2}{y_0}\right)$$

$$f_x = \frac{mgx}{y_0} - \frac{\tau}{l^2}\left(\frac{l^2}{y_0}\right)$$

$$f_x = \frac{mgx}{y_0} - \frac{\tau}{y_0}$$

This is the reduced expression for the horizontal force f_x that is a function of the distance x, the constant height y_0, the gravitational force $m.g$, and the torque τ.

Q5:

Ans: Since, in the previous question we determined:

$$f_x = \frac{mgx}{y_0} - \frac{\tau}{y_0}$$

The equation of motion becomes:

$$mx'' = f_x$$

$$mx'' = \frac{mgx}{y_0} - \frac{\tau}{y_0}$$

Since,

$$p = \frac{\tau}{mg}$$

So,

$$\tau = pmg$$

Hence,

$$mx'' = \frac{mgx}{y_0} - \frac{pmg}{y_0}$$

Factoring out $\frac{mg}{y_0}$,:

$$mx'' = \frac{mg}{y_0}(x - p)$$

Q6:

Ans: Since we are given that:

$$f_x = mx'' = \frac{mg}{y_0}(x - p)$$

And work done, W is given by:

$$W = \int_{-x_T}^{0} \frac{mg}{y_0} \ (x - p) \ dx$$

$$W = \frac{mg}{y_0} \int_{-x_T}^{0} (x - p) \ dx$$

$$W = \frac{mg}{y_0} \left(\frac{x^2}{2} - p\,x \right) \Big|_{-x_T}^{0}$$

$$W = \frac{mg}{y_0} \left(\frac{x_T^2}{2} + p\,x_T \right)$$

And Kinetic energy is given by:

$$K.E = \frac{1}{2} m{v_0}^2$$

According to the work-energy principle

$$W = K.E$$

So,

$$\frac{mg}{y_0} \left(\frac{x_T^2}{2} + p\,x_T \right) = \frac{1}{2} m{v_0}^2$$

$$\frac{mg}{y_0} \left(\frac{x_T^2 + 2p\,x_T}{2} \right) = \frac{1}{2} m{v_0}^2$$

$$mg(x_T^2 + 2p\,x_T) = m{v_0}^2 y_0$$

$$mgx_T^2 + 2mgpx_T = m{v_0}^2 y_0$$

$$mgx_T^2 + 2mgpx_T - m{v_0}^2 y_0 = 0$$

Using quadratic formula:

$$x_T = \frac{-2mgp \ \pm \sqrt{(2mgp)^2 - 4(mg)(-m{v_0}^2 y_0)}}{2\,mg}$$

$$x_T = \left(\frac{-mgp \ \pm \sqrt{m^2 g^2 p^2 + (m^2 g {v_0}^2 y_0)}}{mg} \right)$$

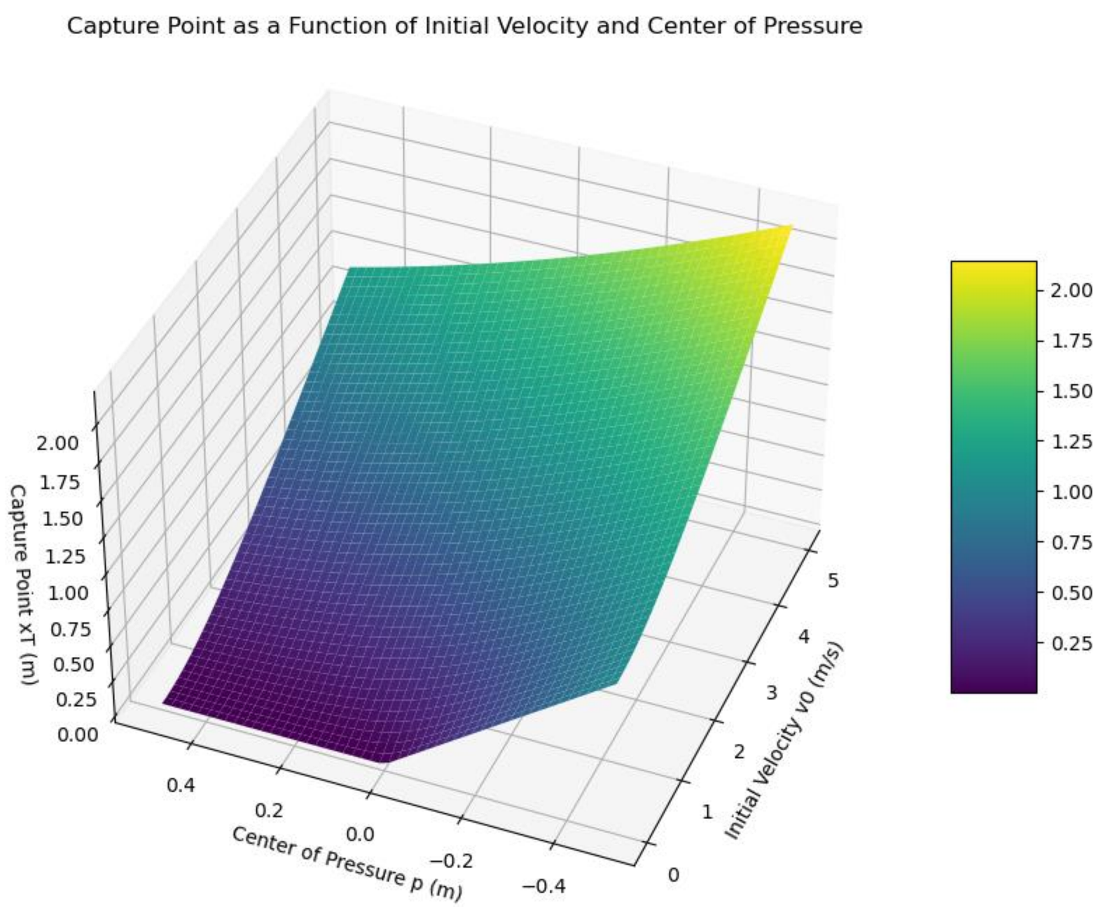
Ignoring the solution yielding $x_T < 0$

$$x_T = \left(\frac{-mgp + \sqrt{m^2 g^2 p^2 + (m^2 g {v_0}^2 y_0)}}{mg} \right)$$

Factoring out m:

$$x_T = \frac{-gp + \sqrt{g^2 p^2 + (g {v_0}^2 y_0)}}{g}$$

Assuming the value for g =9.81 m/s^2 and y_0 = 1.0 meter, following is the graph:



Several key assumption are made when deciding whether a humanoid robot should use an ankle strategy or a stepping strategy based on the capture point plot and its foot geometry, which are as follows:

1. The humanoid and its segments behave as rigid bodies, ignoring compliance in structure and joints.
2. Assumes the robot is on a flat rigid surface.
3. Assumes a static base of support based solely on robot’s foot geometry, neglecting dynamic factors like weight distribution.

4. Treats the CoM as a single point, overlooking the distributing mass across the body.

The decision for for a humanoid robot to use an ankle strategy or a stepping strategy for balance recovery is a critical aspect in the context of dynamic stability. The 3D plot presented above of the capture point x_T against the initial velocity v_0 and center of pressure p provides essential insights for making this decision.

1. Polygon of Support and Foot Geometry:
   i. The Polygon of Support (PoS) is primarily defined by the area of the robot's feet. A larger foot size increases the PoS, providing more room for the robot to manipulate the center of mass (CoM) within the bounds of its feet using ankle strategies.
   ii. The shape and orientation of the feet also play a role, as they can affect the effective distribution of pressure and that range of stable postures the robot can assume.
2. Ankle Strategy:
   i. The ankle strategy involves making small adjustments at the robot's ankle joints to control its CoM.
   ii. This strategy is viable when the capture point x_T, given the humanoid's current state (initial velocity v_0 and center of pressure p), falls within its current polygon of support.
   iii. The robot can employ ankle torques to realign its CoM over its PoS, thus regaining stability without needing to step. This is typically feasible for small perturbations or low initial velocities.
3. Stepping Strategy:
   i. A stepping strategy becomes necessary when the capture point x_T lies outside the humanoid's current PoS. This situation occurres under larger perturbations, reflected by higher initial velocities v_0 or significant shifts in the center of pressure p.
   ii. The robot must take a step to reposition its PoS such that the capture point falls within this PoS. This strategy is essential for handling larger disturbances where ankle torques alone are insufficient for balance recovery.
4. Transition between strategies:
   i. The transition from using ankle strategy to taking a step is not always abrupt. As the disturbance increases, the efficacy of the ankle strategy diminishes until it becomes clear that a step is necessary.
   ii. The plot can be used to set thresholds based on the humanoid's foot geometry and dynamic capabilities. For instance, a humanoid with larger feet might handle higher initial velocities with ankle strategies compared to one with smaller feet.

In conclusion, the choice between ankle and stepping strategies for a humanoid robot is highly dependent on the dynamic state of the robot (as indicated by the capture point), the geometry of its feet (which defines its PoS), and the specific conditions of its environment and task.

Part2

Q1:

Ans: The motor part number I selected is 581294, from the Maxon EC90 series which has the following specifications:

- The rotor inertia J_m is given as 5301 gcm^2. To convert this to kgm^2, which is the SI unit for inertia, we multiply with 10^{-7}. This comes out to be:
$$J_m = 5.301 \times 10^{-4} kgm^2$$
- The maximum continuous torque τ_m is 968 mNm, i.e.
$$\tau_m = 968 \times 10^{-3}\ Nm$$

To model the mechanical dynamics of the selected Maxon EC90 48V (part number 581294) without considering the series spring, we can use the mechanical dynamics equation for the motor as follows:

$$J_m \theta'' = \tau_m - \tau_{ext}$$

Here, J_m is the motor inertia, θ'' is the angular acceleration, τ_m is the motor torque, and τ_{ext} is the external torque. Since we are not considering the series spring, hence $\tau_{ext} = 0$

Substituting the values for rotor inertia and motor torque into the above equation to solve for angular acceleration θ'':

$$(5.301 \times 10^{-4})\theta'' = (968 \times 10^{-3}) - 0$$

$$\theta'' = 1826.07\ rads/sec^2$$

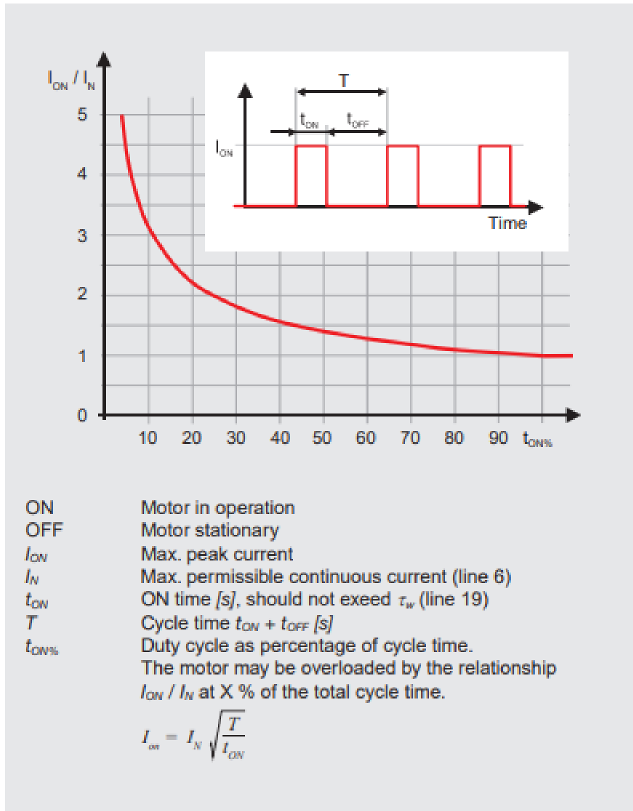
Which represents the maximum acceleration that the motor shaft can achieve under the provided conditions.

Stock program
Standard program
Special program (on request)

Part Numbers					
with Hall sensors		588849	542099	581294	581295
Motor Data					
Values at nominal voltage					
1 Nominal voltage	V	18	30	48	60
2 No load speed	rpm	2110	2120	1990	2020
3 No load current	mA	831	502	285	232
4 Nominal speed	rpm	1810	1820	1700	1730
5 Nominal torque (max. continuous torque)	mNm	1030	992	968	968
6 Nominal current (max. continuous current)	A	12.4	7.24	4.16	3.37
7 Stall torque	mNm	15700	15100	13500	13700
8 Stall current	A	195	113	59.5	48.8
9 Max. efficiency	%	87.6	87.3	86.8	86.9
Characteristics					
10 Terminal resistance phase to phase	Ω	0.0923	0.266	0.807	1.23
11 Terminal inductance phase to phase	mH	0.12	0.334	0.964	1.47
12 Torque constant	mNm/A	80.7	134	228	281
13 Speed constant	rpm/V	118	71.4	42	34
14 Speed/torque gradient	rpm/mNm	0.135	0.142	0.149	0.149
15 Mechanical time constant	ms	7.59	7.88	8.26	8.25
16 Rotor inertia	gcm^2	5301	5301	5301	5301

Q2:

Ans: Since we are given that the SEA is actuating one leg, and a 50% duty cycle of motor operation can be expected hence using the formula and graph provided over [here](#) on page # 44 (image attached below):



$$\tau^*_{max} = \left(Nominal\ current\ \times \sqrt{2} - no\ load\ current\right)(torque\ constant)$$

$$\tau^*_{max} = \left(4.16 \times \sqrt{2} - 0.285\right)(0.228)$$

$$\tau^*_{max} = 1.28$$

Now, since

$$\frac{M \times g \times r}{N} = \tau^*_{max}$$

So,

$$N = \frac{1.37 \times M \times g \times r}{\tau^*_{max}}$$

$$N = \frac{1.37 \times 80 \times 9.8 \times 0.05}{1.28}$$

$$N \approx 42$$

Q3:

Ans: Since we are given that:

$$F_s = K\,\Delta l$$

$$\Delta l = \Delta l_y - \Delta l_m$$

$$\Delta l_y = y_0 - y$$

$$\Delta l_m = \frac{r}{N}\theta_m$$

So,

$$\Delta l = y_0 - y - \frac{r}{N}\theta_m$$

Substituting value of Δl to the expression of F_s:

$$F_s = K\,(y_0 - y - \frac{r}{N}\theta_m)$$

Since,

$$\frac{F_s r}{N} = \tau_{ext}$$

So,

$$\tau_{ext} = \frac{K\,(y_0 - y - \frac{r}{N}\theta_m)r}{N}$$

Since,

$$J_m\theta'' = \tau_m - \tau_{ext}$$

Hence,

$$J_m\theta'' = \tau_m - \frac{K\,(y_0 - y - \frac{r}{N}\theta_m)r}{N}$$

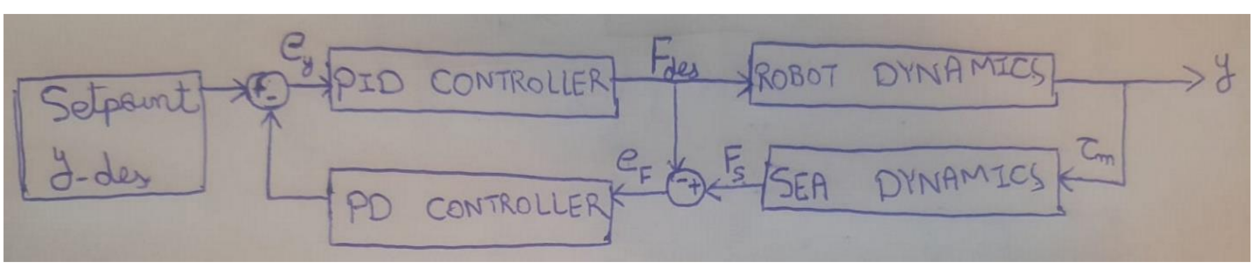
Q4:

Ans: In this particular question we were asked to design a cascaded controller for a humanoid robot to stabilize at a reference height. In order to achieve that we require two distinct control loops: an outer-loop for the high-level behavior control and an inner loop for the actuator control.

1. Outer-loop behavior controller (PID):
   a. Objective: To maintain the robot at a desired height y_{des}.
   b. Design choices:
      i. PID Control: Chosen for its ability to handle a simple heigh control problem effectively based on the current error (proportional), past errors (integral), and future errors (derivative).
      ii. Gains (Kp, Ki, Kd): Selected to ensure a balance between fast response and minimal overshoot, where the proportional and integral gains are tuned to correct steady-state errors, and a derivative gain is used to dampen the response.
2. Inner-loop SEA Controller (PD):
   a. Objective: To control the motor's torque to achieve the force determined by the outer loop.
   b. Design choices:
      i. PD Control: Used for its effectiveness in controller systems with a well-defined force output since the PD controller responds to the error in force and its rate of change, providing a torque output that tracks the desired force.
      ii. Gains (Kp_inner, Kd_inner): This has to be tuned to provide a quick response to changes in desired force without causing instability. The derivative gains is particularly important for smoothing out the response and avoiding oscillations.

The cascaded design allows for a clear separation of concerns, i.e. the outer loop focuses on the high-level goal (maintaining height), while the inner loop focuses on the immediate actiation needed to achieve that goal. The separation simplifies the design and tuning of the controllers since each can be tuned somewhat independently. The outer loop is tuned for overall system performance, and the inner loop is tuned for responsiveness and stability of the actuation system.

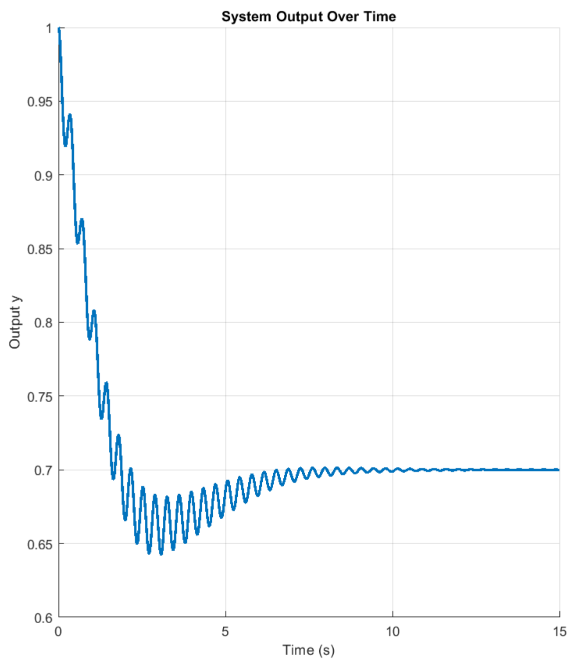
Following is the controller diagram for the same:

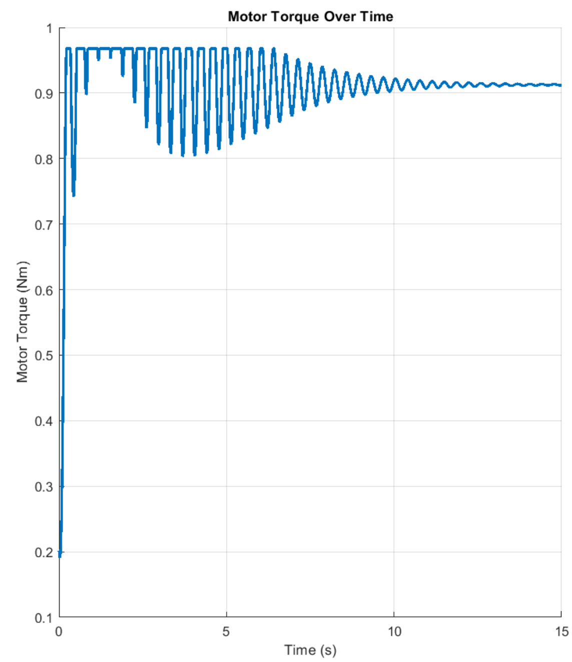


Q5:

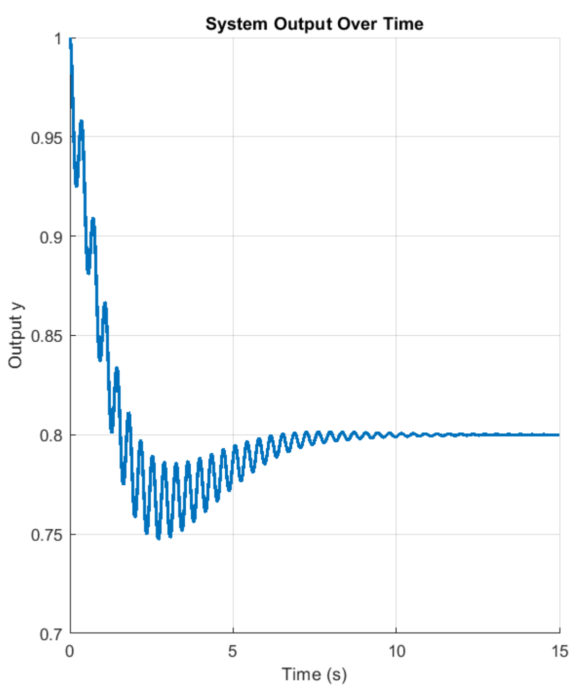
Ans: Implementing the cascaded controller with inner and outer loops for controlling the actuator force and heigh achieved "y" respectively, and with provided initial condition, following are the height and torque profiles for varying setpoints:

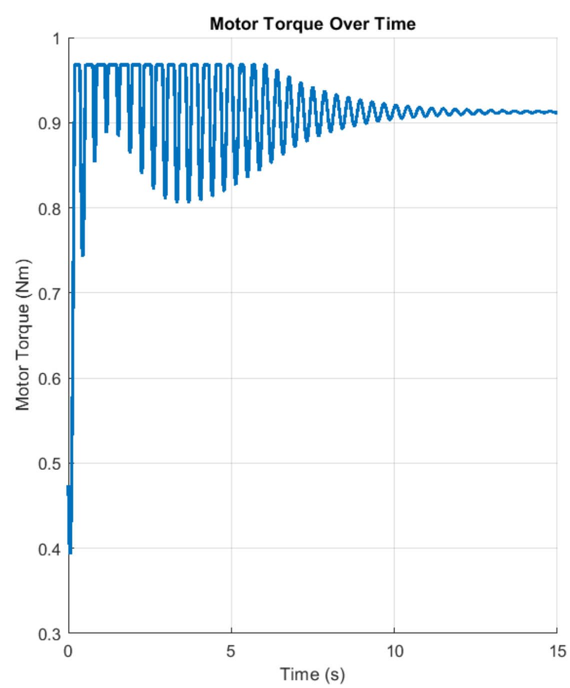
For y_des = 0.7:



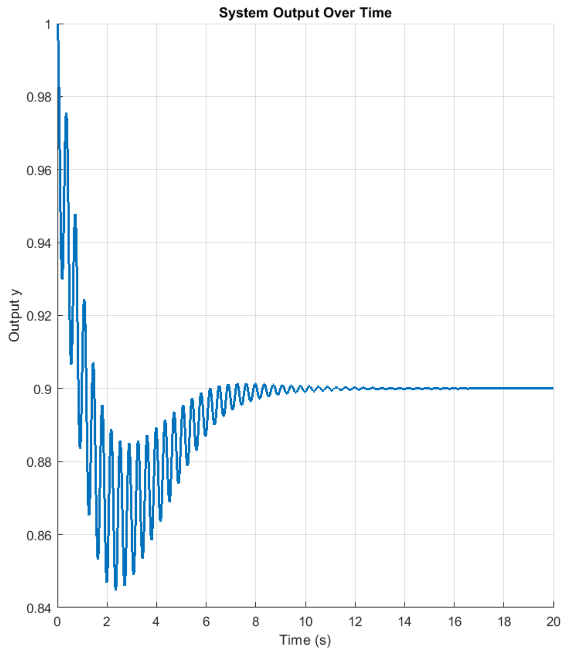


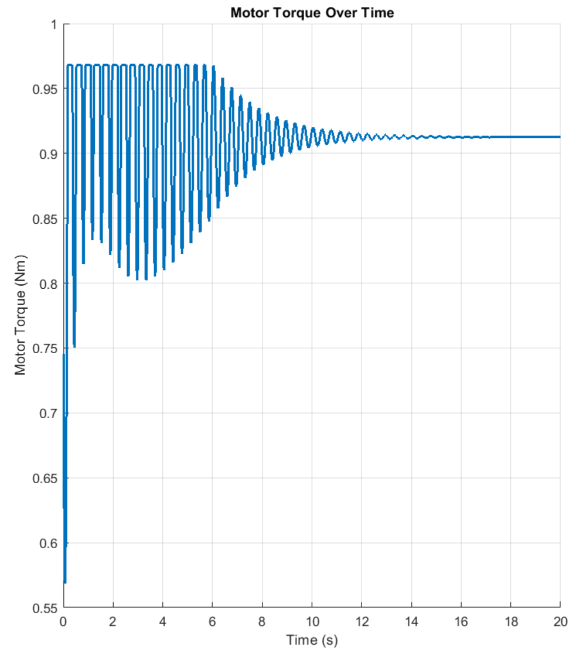
For y_des = 0.8:





Y_des = 0.9:


System Output Over Time
Output y
Time (s)


Motor Torque Over Time
Motor Torque (Nm)
Time (s)

Part-3

Q1:

Ans: For this part of the assignment, the control system is designed to manage the dynamics of the robot during a stance phase, specifically focusing on three key elements:

1. Center of Mass (CoM)
2. The Foot Point (FP)
3. The Trunk (TR)

Now going into a bit more detail:

1. Center of Mass (CoM) planning:
    a. The CoM's trajectory is computed using the Linear Inverted Pendulum Model (LIPM), which simplifies the humanoid model into an inverted pendulum with a point mass (representing the CoM) atop a massless, rigid leg without considering the effects of ankle torque. This approach lays the foundation for planning the CoM trajectory, ensuring dynamic stability and balance throughout the stance phase.
    b. The LIPM-based planning involves calculating the CoM's position and velocity over time, critical for maintaining the robot's upright posture and enabling a controlled transition between steps.
2. Foot Point (FP) Planning:
    a. The FP refers to the swing foot's trajectory, which is pivotal for the robot's locomotion. The planning of the FP involves determining the path that the swing foot will follow to achieve an optimal step placement at the end of the swing phase.
    b. The element of control is essential for ensuring stability upon foot landing and for preparing the robot for subsequent steps. The FP trajectory is influenced by factors like desired step length, foot clearance, and timing of the swing phase.
3. Trunk (TR) Planning:
    a. The TR component entails controlling the motion of humanoid's upper body or torso. Effective trunk control is pivotal for overall balance.
    b. The trunk's orientation and movement are adjusted to counteract the robot's linear and angular momentum. This adjustment aids in stabilizing the robot, especially when dealing with variations in walking speed, direction changes, and external disturbances.
    c. The trunk's planned motion is aligned with lower body's dynamics to achieve a coordinated and efficient walking gait.

The control strategy in the humanoid robot simulation exemplifies the intricacies of bipedal robot control. By carefully planning the motion of the CoM, FP, and TR, the control system can address the critical challenges of humanoid locomotion, namely balance, stability, and smooth transitioning between steps.

Q2:

Ans: In the provided code (MATLAB scripts), the plans for the Foot Point (FP) and Trunk (TR) are modified reactively during the stance phase, similar to how the Center of Mass (CoM) plan is adjusted. Here is how these modification take place in the provided code:

- Foot Point (FP) Plan Modification:
    - The 'PD_TrackSwingFP' function is used to modify the swing foot (FP) plan at each time step.
    - This function implements a PD control strategy, where the current position ('xFP') and velocity ('dxFP') of the swing foot are calculated based on the joint angles ('q') and velocities ('dq').
    - The desired acceleration ('FP_acc') for the swing foot is updated by comparing the current state (position and velocity) with planned state ('FPpos' and 'FPvel'). The differences are multiplied by the proportional ('kp_sw') and derivative ('kd_sw') gains to compute the correction needed to keep the swing foot on its planned trajectory.

- Trunk (TR) Plan Modfication:
    - For the trunk (TR), the script adjusts the desired trunk acceleration ('TR_ades') using a similar PD control strategy.
    - Specfici proportional ('kp_tr') and derivative ('kd_tr') gains are set for proportional control.
    - The desired trunk acceleration is updated based on the difference between planned ('TR_pdes', 'TR_vdes') and the current trunk state (position and velocity derived from 'HMD.q' and 'HMD.dq').
    - The adjustment ensures that the trunk maintain its desired orientation and position, contributing to the overall balance and posture of the robot.

In the main control loop these modifications are applied at each time step. The adjusted plans for the CoM, FP, and TR are then used in a Quadratic Programmig (QP) problem to compute the joint accelerations, torques, and ground reaction forces required to achieve these modified plans.

Q3:

Ans: A key aspect of maintaining stability is adhering to the friction cone requirement. This constraint ensures that the horizontal ground reaction force GRFx remains within limits determined by the friction coefficient μ and the vertical ground reaction force GRFy. Mathematically, the constraint is represented as:

$$-\mu\ GRF_y \ \leq GRF_x \leq \mu\ GRF_y$$

To implement this in MATLAB, we employ the quadprog function, a tool for quadratic programming optimization. The function requires that inequality constraints be presented in the format:

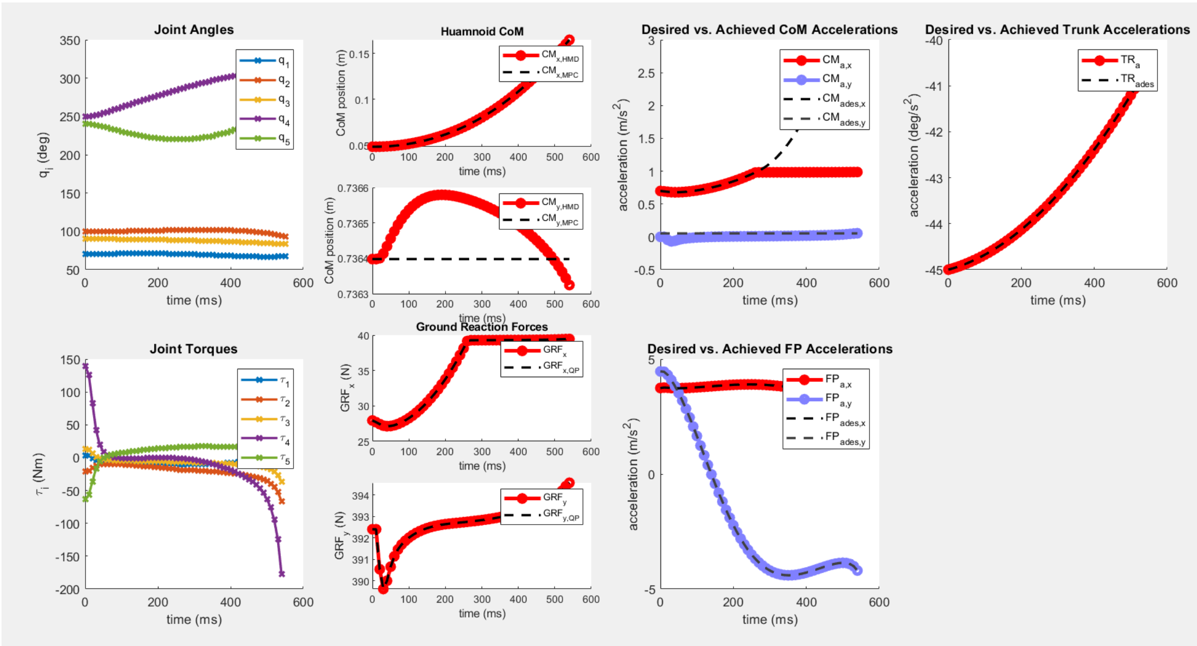
$$A_{ineq}\,.x \leq b_{ineq}$$

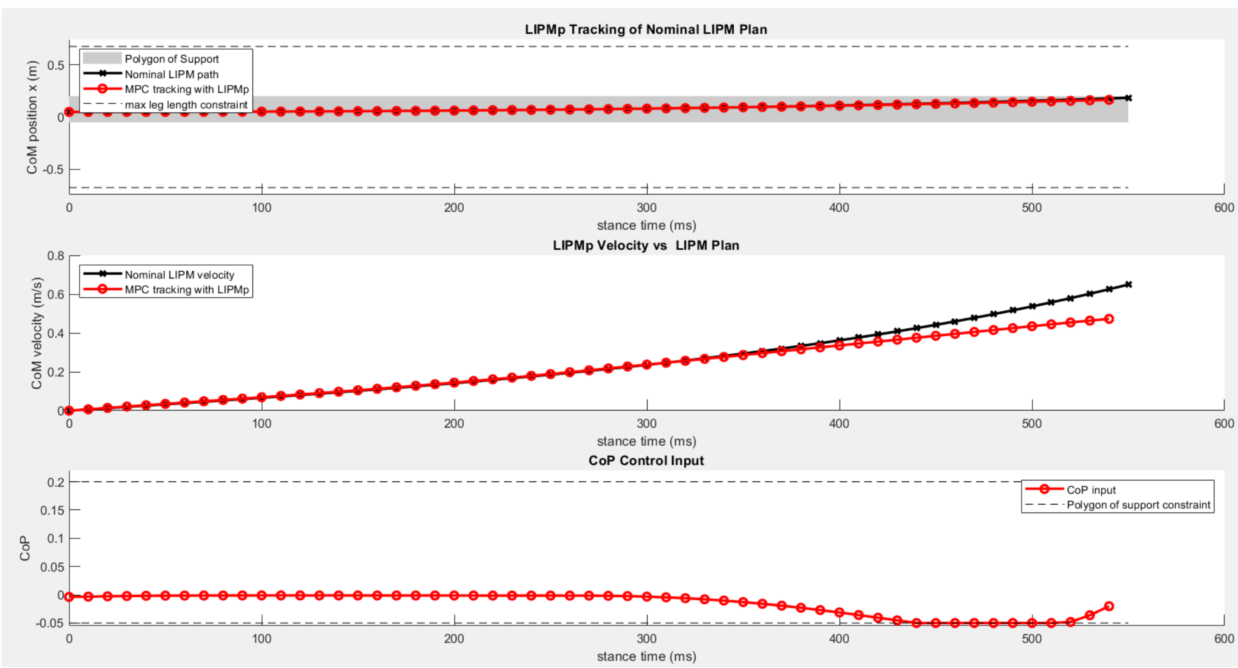
Where x represents the vector of optimization variables.

The friction cone constraints are reformulated into a matrix compatible format for quadprog. The inequalities $GRF_x \leq \mu\ .\ GRF_y$ and $-\mu.\ GRF_y \leq GRF_x$ are translated into $[1, -\mu].\,[GRF_x, GRF_y]^T \leq 0$ and $[-1, -\mu].\,[GRF_x, GRF_y]^T \leq 0$ respectively.

Q4:

Ans: Following are the results when μ is set to 0.1:



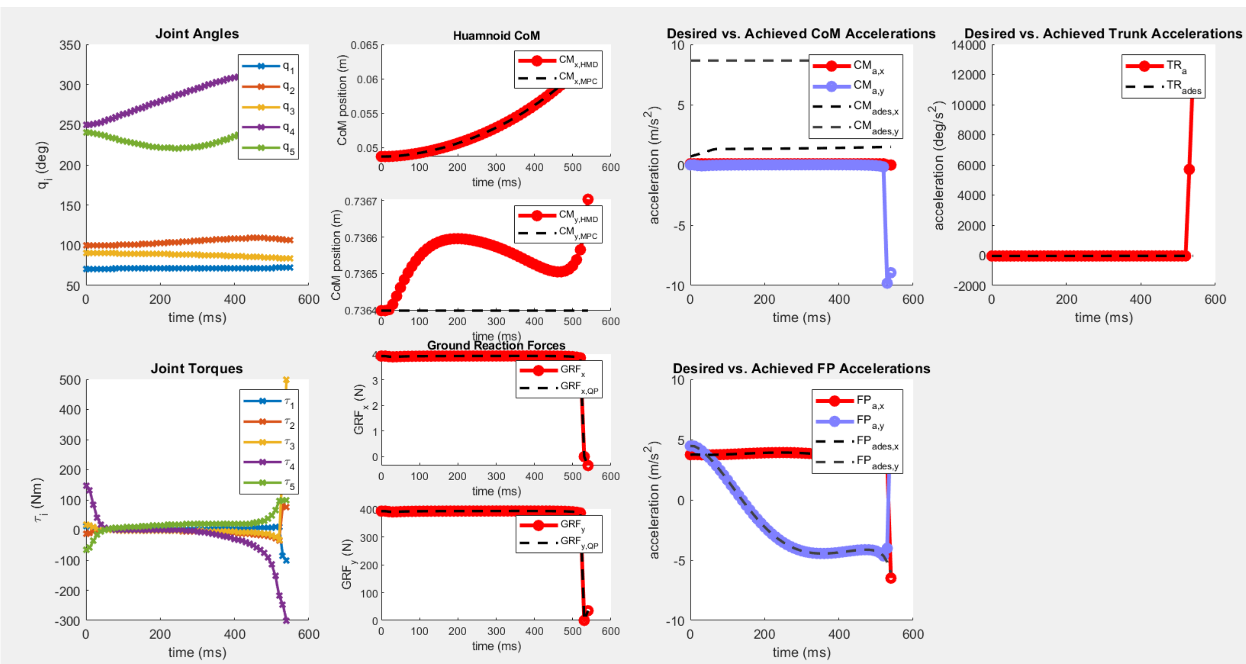


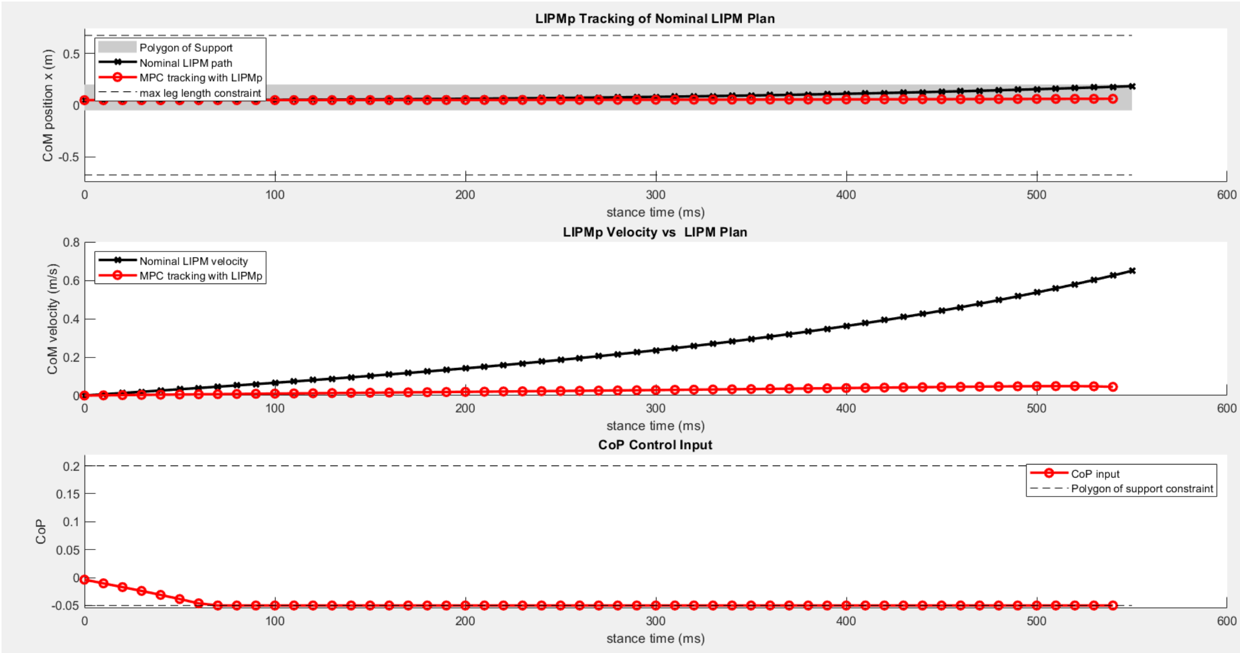
Following are my observations for the results presented above:

a. The plot indicate that the horizontal ground reaction force (GRFx) remain well below the expected maximum of 40N, and the vertical ground reaction force (GRFy) remain well below the expected maximum of 400N suggesting that the robot is able to generate sufficient lateral forces for stability without reaching the friction limit.

b. The top plot in figure 2 shows the CoM position tracking, where the red circles represent that actual path tracked by the Model Predictive Control (MPC) using LIPM with predictive control against the nominal (desired) LIPM path. The tracking appears to be very close to the nominal path, indicating that the robot is able to maintain its balance and follow the planned trajectory even with a reduced friction coefficient of 0.1.
c. The middle plot in figure 2 represents the CoM velocity, with the red circles donating MPC tracking outcome and the black line indicating the nominal velocity. The actual velocities follow the desired velocity profile closely throughout the stance phase, suggesting effective control over th humnaoid's forward progression.
d. The bottom plot of figure 2 shows CoP control input, where the red circles indicate the actual CoP input within the constraints of the polygon of support (marked by the dashed lines). The CoP remains well within the boundaries of the support polygon, which suggests that the humanoid is able to maintain a stable stance without exceeding the maximum allowable CoP range. This also indicates that the control system is not reaching the limits of the friction cone, allowing for stable support.

And following is the result when μ is set to 0.01:





Following are my observations for the results presented above:

a. With a much lower friction coefficient, the horizontal GRF (GRFx) remains bekiw the expected limit of 4N. This extreme reduction in friction means the robot cannot rely on horizontal forces to stabilize itself or change direction effectively.

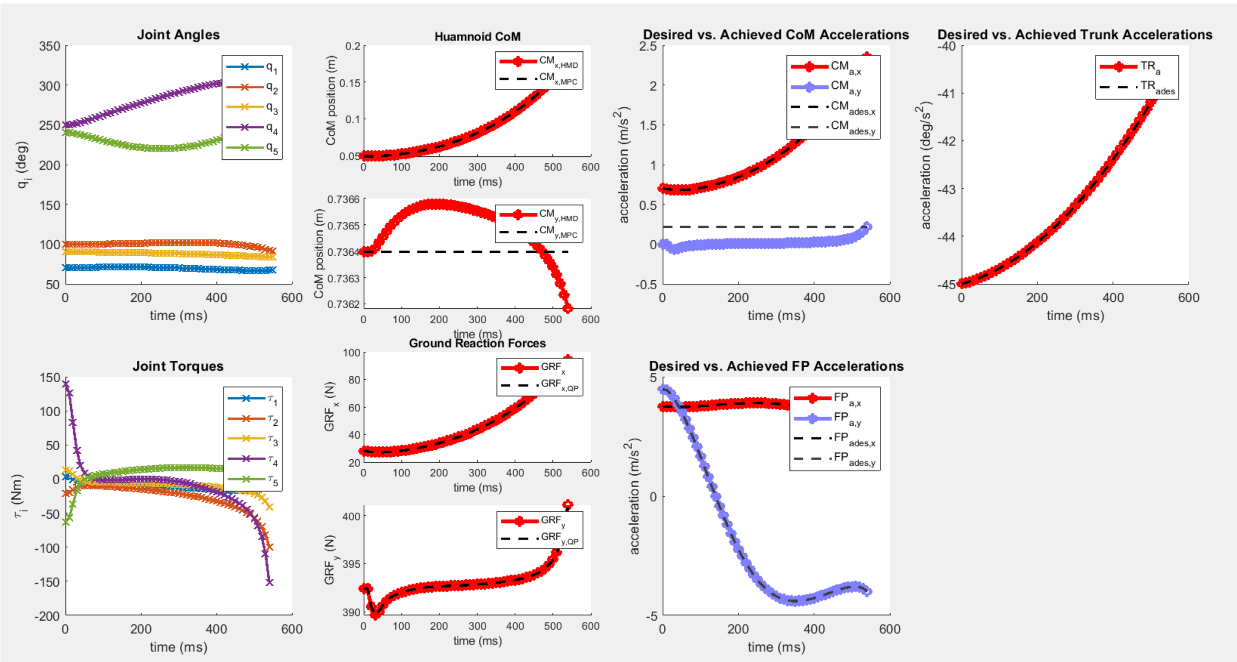
b. The top plot in figure 2 shows the CoM position tracking, where the red circles represent that actual path tracked by the Model Predictive Control (MPC) using LIPM with predictive control against the nominal (desired) LIPM path. The tracking appears to be very close to the nominal path during the first ~300 msec, indicating that the robot is able to maintain its balance and follow the planned trajectory even with a reduced friction coefficient of 0.01, but then the tracking starts to diverge indicating as time progresses the robot is having difficulty maintaining its balance.

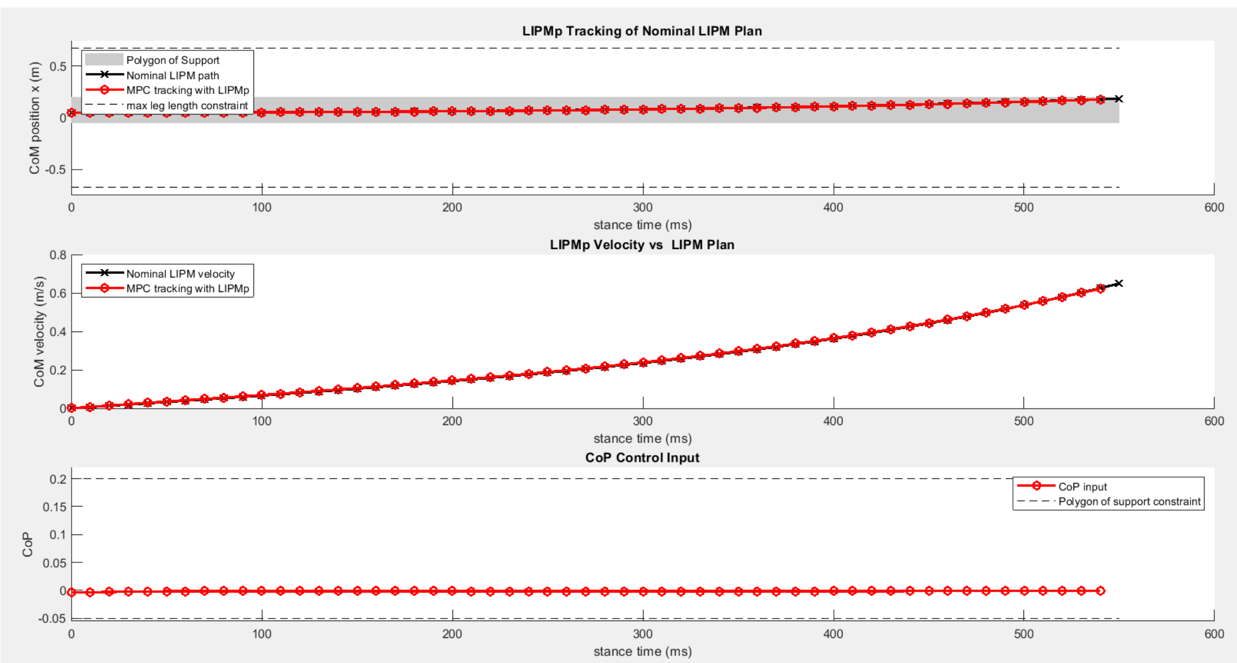
c. The middle plot in figure 2 represents the CoM velocity, with the red circles donating MPC tracking outcome and the black line indicating the nominal velocity. The actual velocities diverge from the desired velocity profile throughout the stance phase, suggesting ineffective control over the humanoid's forward progression.

d. The bottom plot of figure 2 shows CoP control input, where the red circles indicate the actual CoP input within the constraints of the polygon of support (marked by the dashed lines). The CoP remains well within the boundaries of the support polygon, which suggests that the humanoid is able to maintain a stable stance without exceeding the maximum allowable CoP range. This also indicates that the control system is not reaching the limits of the friction cone, allowing for stable support.

Q5:

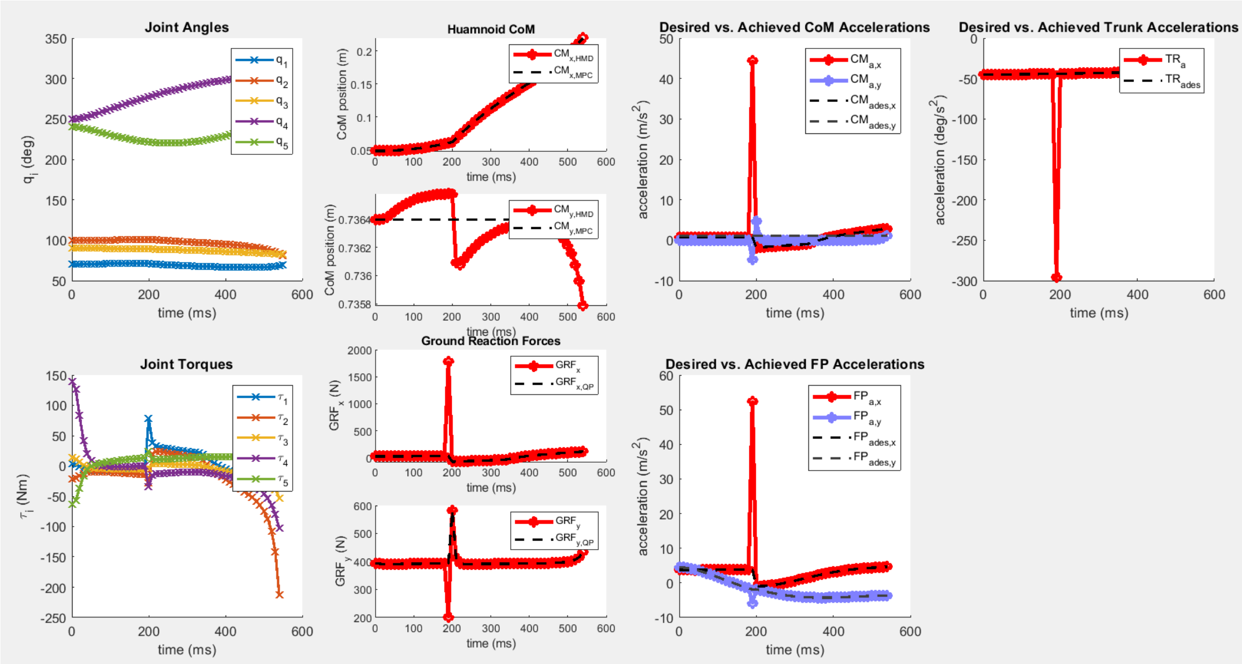
Ans: Setting the $\mu = 0.8$ and disturbance impulse set at 0 Ns and actuator noise set at 0 we get:

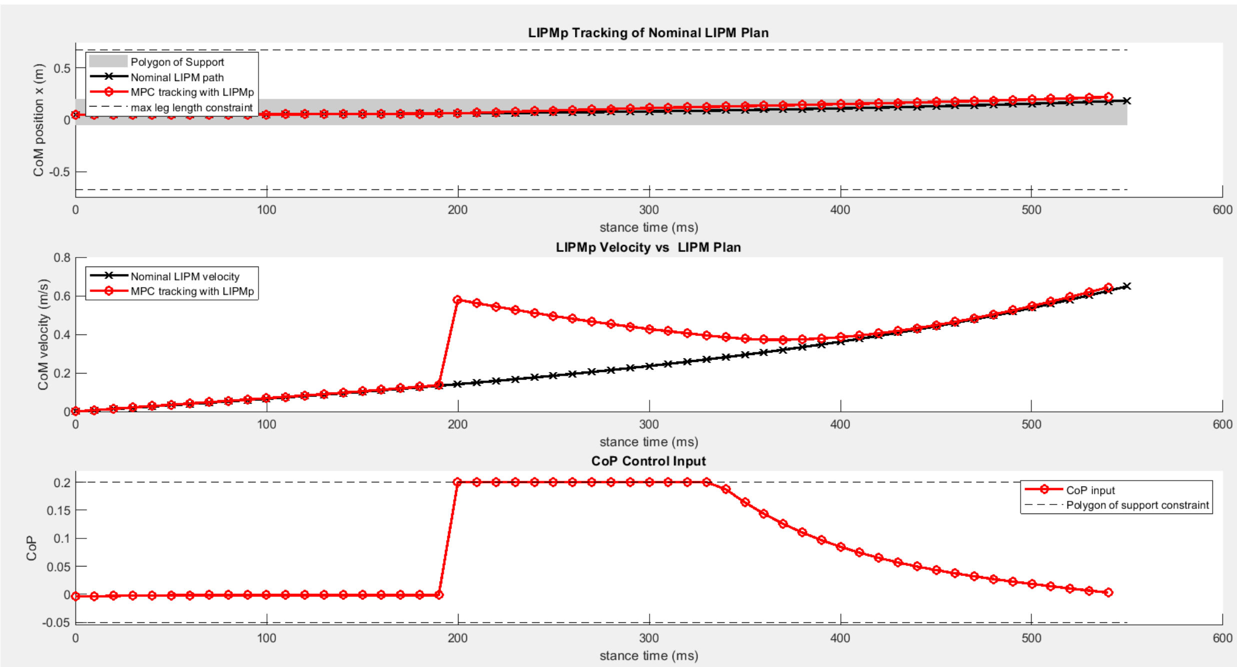




The above set of graphs show the system's behavior in an ideal condition. The CoM tracking is precise and the Ground Reaction Forces (GRFs) are stable, indicating that the robot is following its intended plan accurately.

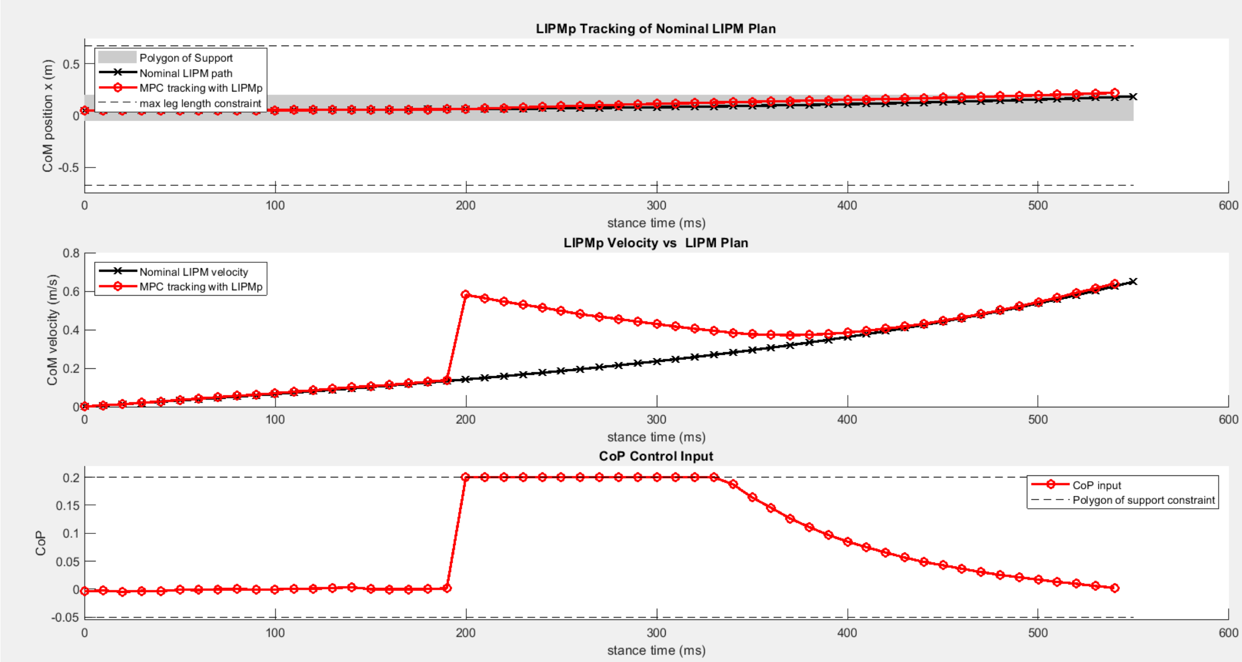
Setting the $\mu = 0.8$ and disturbance impulse set at 20 Ns and actuator noise set at 0 N we get:

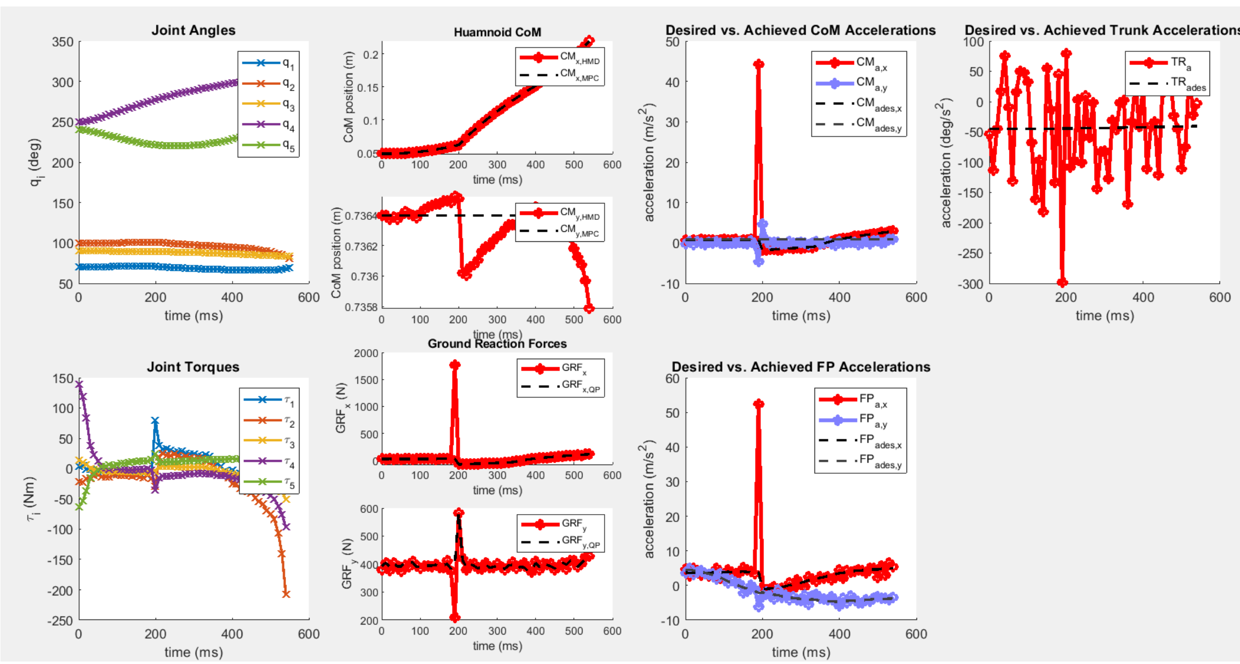




The above set of graphs show the system's response to a horizontal push applied to the CoM. The noiticeable spikes in the GRFs indicate the robot's response to the disturbance. Despite the disturbance, the robot's CoM tracking appears to remain close to the nominal path, suggesting the control system can compensate for the push and maintain the intended plan.

Finally, adding in actuator noise we get:





Finally, when the actuator noise is introduced in addition to the horizontal disturbance, there is an increase in the variability of the GRFs, joint torques, and especially in the trunk accelerations. The noise represents the uncertainty in the motor control, which seems to cause significant deviations in trunk behavior, likely due to the control system’s efforts to counteract the noise. However, the CoM tracking still appears to be robust against these perturbations, as indicated by the minimal deviation from the nominal path.

Code:

Part 2 Question 5:

```
% System parameters
M = 80; g = 9.81; k = 20000; J_m = 0.0005301; r = 0.05; N = 42;
tau_max = 0.968; % Maximum allowable torque
y_des = 0.7; % Desired height

% PID controller gains for outer loop (Tune these gains)
Kp_outer = 25; Ki_outer = 3; Kd_outer = 10;
integral_error = 0; derivative_error = 0; previous_error = 0;

% PD controller gains for inner loop (Tune these gains)
Kp_inner = 5; Kd_inner = 2;
previous_F_error = 0;

% Simulation parameters
dt = 0.01; % Time step
T = 20; % Total simulation time
t = 0:dt:T; % Time vector

% Initialize state variables
y = 1; y_dot = 0; theta_m = 0; theta_dot_m = 0;
F_s = 0; % Spring force

% Preallocate arrays for recording results
y_record = zeros(1, length(t));
tau_m_record = zeros(1, length(t));

% Main simulation loop
for i = 1:length(t)
    % Outer loop (PID controller for height control)
    error = y_des - y;
    integral_error = integral_error + error * dt;
    derivative_error = (error - previous_error) / dt;
    F_des_s = Kp_outer * error + Ki_outer * integral_error + Kd_outer *
derivative_error;
    previous_error = error;

    % Inner loop (PD controller for SEA control)
    F_error = F_des_s - F_s;
    tau_m = Kp_inner * F_error + Kd_inner * (F_error - previous_F_error) /
dt;
    previous_F_error = F_error;

    % Apply torque saturation
    tau_m = max(min(tau_m, tau_max), -tau_max);

    % Update system states using Euler integration
    y_ddot = (F_des_s - k * (y - y0)) / M - g;
    y_dot = y_dot + y_ddot * dt;
    y = y + y_dot * dt;

    theta_ddot_m = (tau_m - k * (r/N) * theta_m) / J_m;
    theta_dot_m = theta_dot_m + theta_ddot_m * dt;
    theta_m = theta_m + theta_dot_m * dt;
```

```
    F_s = k * (r/N) * theta_m; % Update spring force based on new motor angle

    % Record results for plotting
    y_record(i) = y;
    tau_m_record(i) = tau_m;
end

figure('Position', [100, 100, 1200, 600]); % Set figure size

% Subplot for Output y
subplot(1,2,1)
hold on; grid on;
plot(t, x_actual(1,:), 'LineWidth', 2);
xlabel("Time (s)"), ylabel("Output y"), title("System Output Over Time");

% Subplot for Motor Torque
subplot(1,2,2)
hold on; grid on;
plot(t, u_final, 'LineWidth', 2);
xlabel("Time (s)"), ylabel("Motor Torque (Nm)"), title("Motor Torque Over
Time");

% Additional plot settings
set(gcf,'Color','w'); % Set background color to white
sgtitle('System Response and Control Effort'); % Overall title for all
subplots
```