OCTOBOTICS ASSIGNMENT REPORT

Goal1: Creating a controller package

To set parameter of the pendulum using the provide ROS service server "/inverted_pendulum/set_params" first, I create the ros package name controller_pkg with the appropriate dependencies.

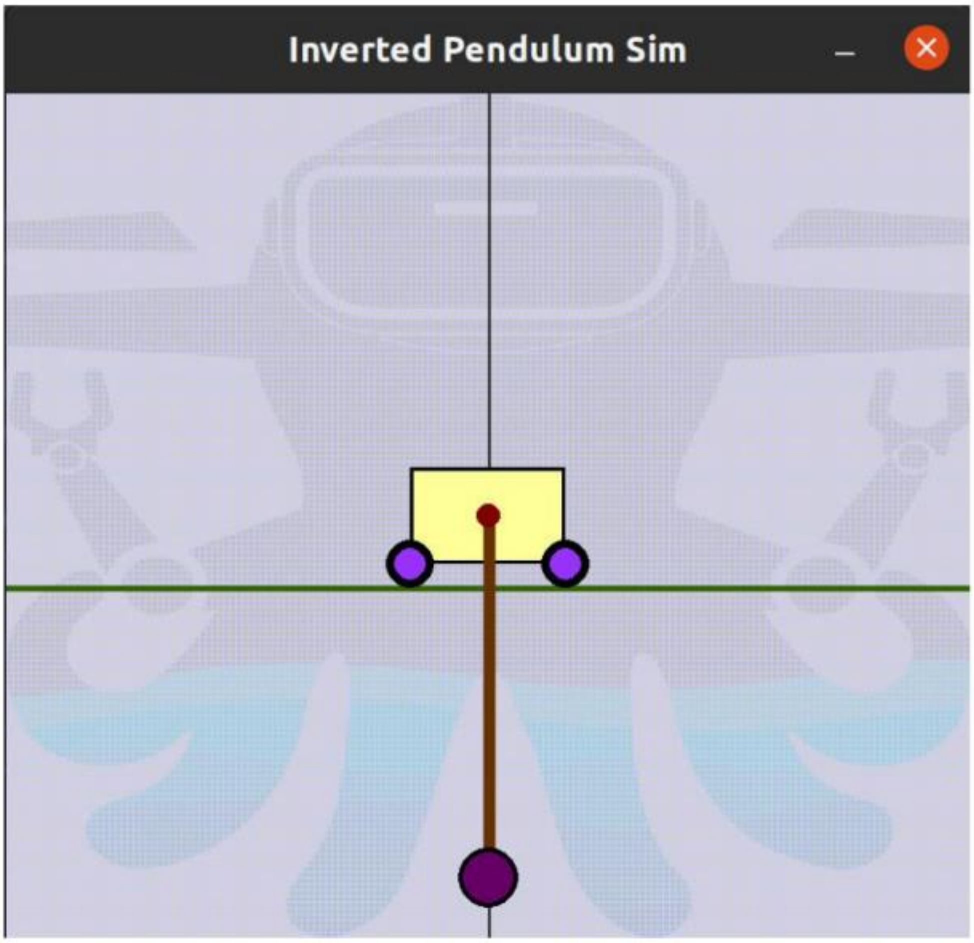
And to run the pendulum, I set the initial the params as mention in problem statement

pendulum weight = 2kg, cart weight = 0.5kg, pendulum length = 300units, cart position = in the centre, pendulum orientation = vertical down pendulum length = 300units)

To run the task1

roslaunch controller_pkg task1.launch

Output –



Goal2: Send control input to the pendulum

To input a sinusoidal force input to the cart, I created a publisher node "Task2_control_input" which publish the sinusoidal force msg on the topic "inverted_pendulum/control_force"

To calculate the sinusoidal force, I use the equation

y(t) = Amplitude * sin(2*pi*Frequency + phase_angle)

phase_angle = 0 , in case on sin wave

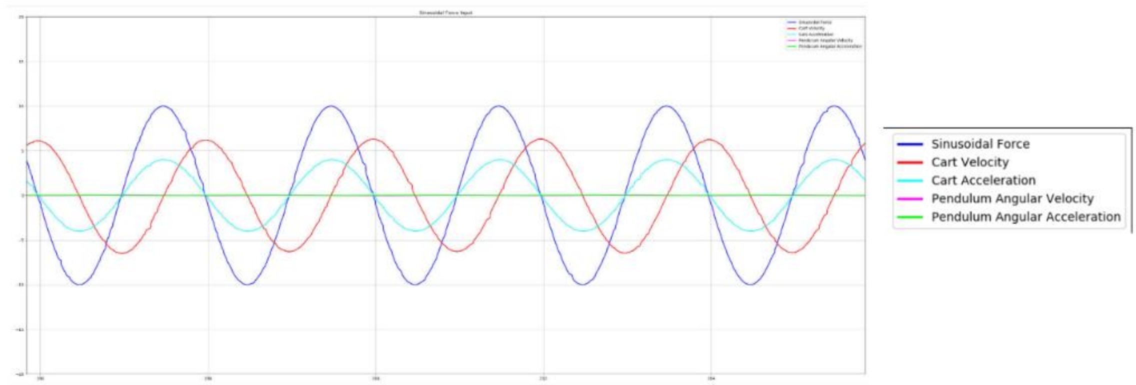
To change the amplitude and frequency, I add a two methods in the python script which change the frequency and amplitude in the runtime.

Also I plotted the cart velocity, acceleration, as well as pendulum angular velocity and acceleration,

To run the task 2

roslaunch controller_pkg task2.launch

Output –



Goal3: Balance the inverted pendulum

Here, for implementing the balancing of inverted pendulum, I used the PID algorithm.

I created the control node python node which subscribe the "**inverted_pendulum/current_state/curr_theta**" ros topic as a feedback message and used the simple pid python module to calculates the pid output of sinusoidal force to move the cart to maintain the pendulum vertically upward.

Also, I plotted the graphs for theta is plotted using rqt_plot which shows the theta position tending to pi rad.

To run the task3

roslaunch controller_pkg task3.launch

OUTPUT –