EXERCISE NO. :6

SIMULATION OF TWO WHEEL DIFFERENTIAL DRIVE MOBILE ROBOT

DATE: 03-10-2020

Reg. No. : RA1711018010101

LAB PREREQUISITES:

Introduction to Physical Modelling-Simscape

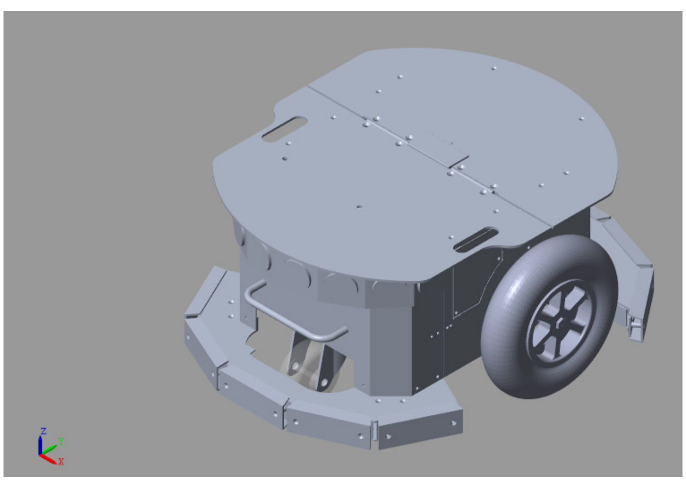
PREREQUISITE KNOWLEDGE:

Fundamentals of SIMULINK modelling and simscape (physical) modelling of physical systems.

OBJECTIVES:

- Given the simscape model of the mobile robot (generated model from the CAD file),understand the effect of gravity axis on the robot dynamics
- Drive the robot wheel with an angular velocity set at each wheel and observe the behavior
- To develop a simulink model of motion constraint equation for the given robot and add in the model. Observe the result when each of the wheel are actuated as
  $W_r = W_l > 0$ (W_r- angular velocity of right wheel; W_l- angular velocity of left wheel)
  $W_r = W_l < 0$
  $W_r = -W_l$
  $W_r > W_l > 0$
  $W_l > W_r > 0$
- Drive a robot to a set linear distance (with zero heading angle) given robot linear velocity and angular velocity as design parameter.

THEORY



Dimensions of the Robot

Physical Parameter	**Specification**
Radius of wheel	0.0950 m
Wheel to wheel distance	0.330 m

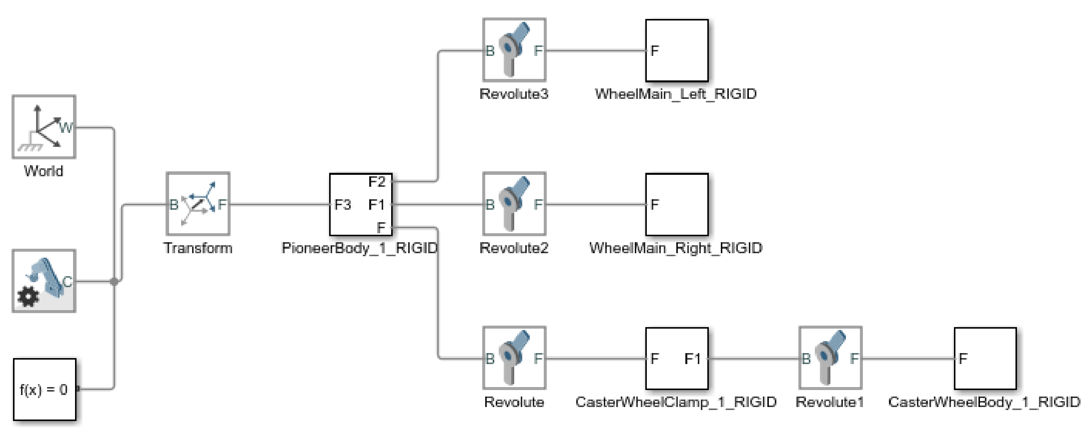
EXERCISE TASKS

Task 1: Effect of gravity axis in the dynamics of robot

- Simulate the robot model to view the physical model of robot in the explorer
- Change the robot gravity axis to [0,-9.81,0]
- Observe the castor wheel motion
- Set the gravity axis to [0,0,-9.81] for rest of the tasks

Task 2: Drive the robot wheels with different set of angular velocity

- Create the subsystem for left, right and castor wheel
- Select the revolute joint actuation → Torque- automatically computed and motion –provided by input
- Use integrator to convert angular velocity input by user to the wheel position (input to the revolute joint)
- Sense the angular velocity of the wheel
- Observe the motion of robot when the wheels are set with defined angular velocity



Task 3: Develop a simulink model of motion constraint equation for the given robot and add in the model.

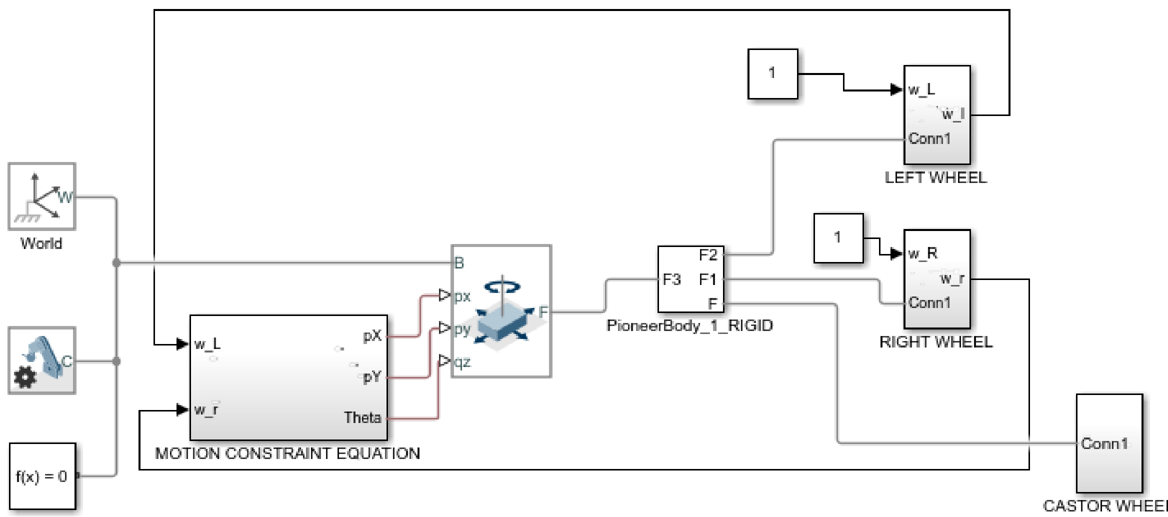
- Develop a Simulink model for the motion constraint equation for the given mobile robot

$$P_X = \int V \sin\theta$$

$$P_Y = \int V \cos\theta$$

$$\theta = \int \omega \ dt$$

- Observe the motion of robot for the set of angular velocities defined in the objective



Task 4: Drive the robot to a set linear distance given robot linear and angular velocity

- Modify the simscape model to compute the angular velocity of the wheel from robot linear and angular velocity
- Set the distance to be travelled by the robot
- Observe the motion of the robot

DELIVERABLES

Task 1

- Image of the SimMechanics explorer window showing the model of the robot
- Plot the castor wheel clamp(revolute joint) velocity for the gravity axis chosen along Y axis

Task 2

- Image of the SimMechanics model after creating the subsystem and input port for angular velocity
- Plot the robot angular velocity VS. time

Task 3

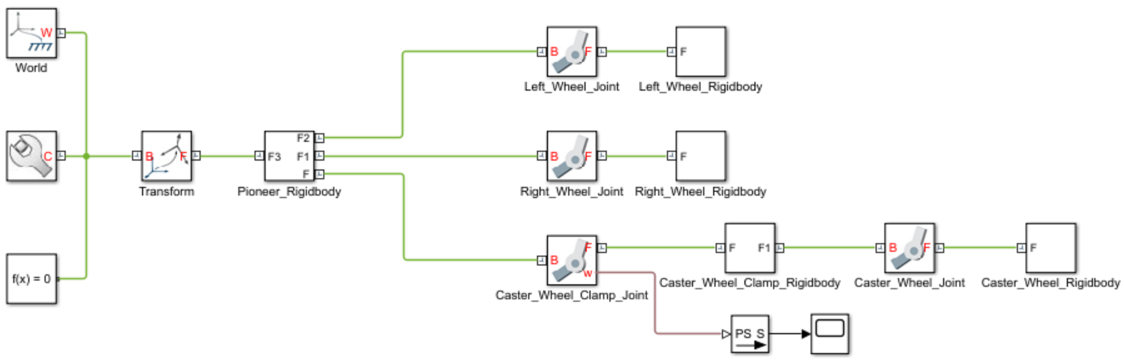
- Write the equation to find the robot linear and angular velocity from the wheel angular velocity
- Image of the Simulink model developed
- Plot the robot linear and angular velocity Vs time for the various angular velocity input set
- $W_r=W_l>0$ (W_r- angular velocity of right wheel; W_l- angular velocity of left wheel)
- $W_r=W_l<0$
- $W_r=-W_l$
- $W_r>W_l>0$
- $W_l>W_r>0$

Task 4

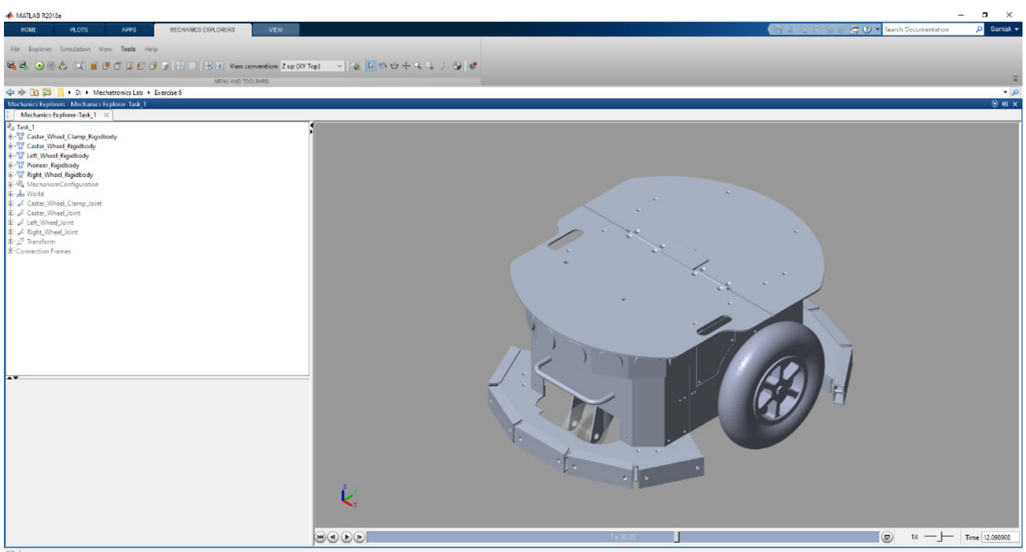
- Image of the simmechanics model developed with modifications to input robot linear and angular velocity as input
- Write the equation to compute the angular velocity of the wheel from the robot linear and angular velocity
- Plot the robot linear distance ,set distance Vs time
- Plot the angular velocity of wheel Vs time

Task 1

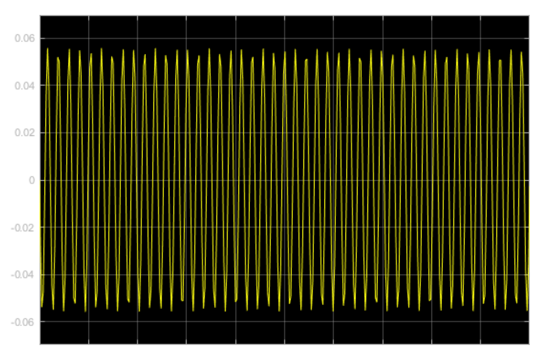
1. SimMechanics model block diagram:



2. Image of the SimMechanics explorer window showing the model of the robot:

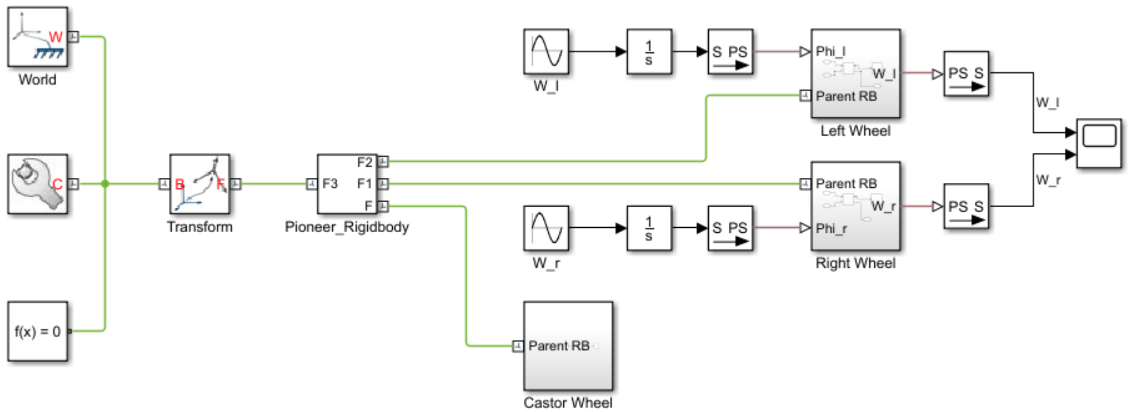


3. Plot of the castor wheel clamp (revolute joint) velocity for the gravity axis chosen along Y axis:

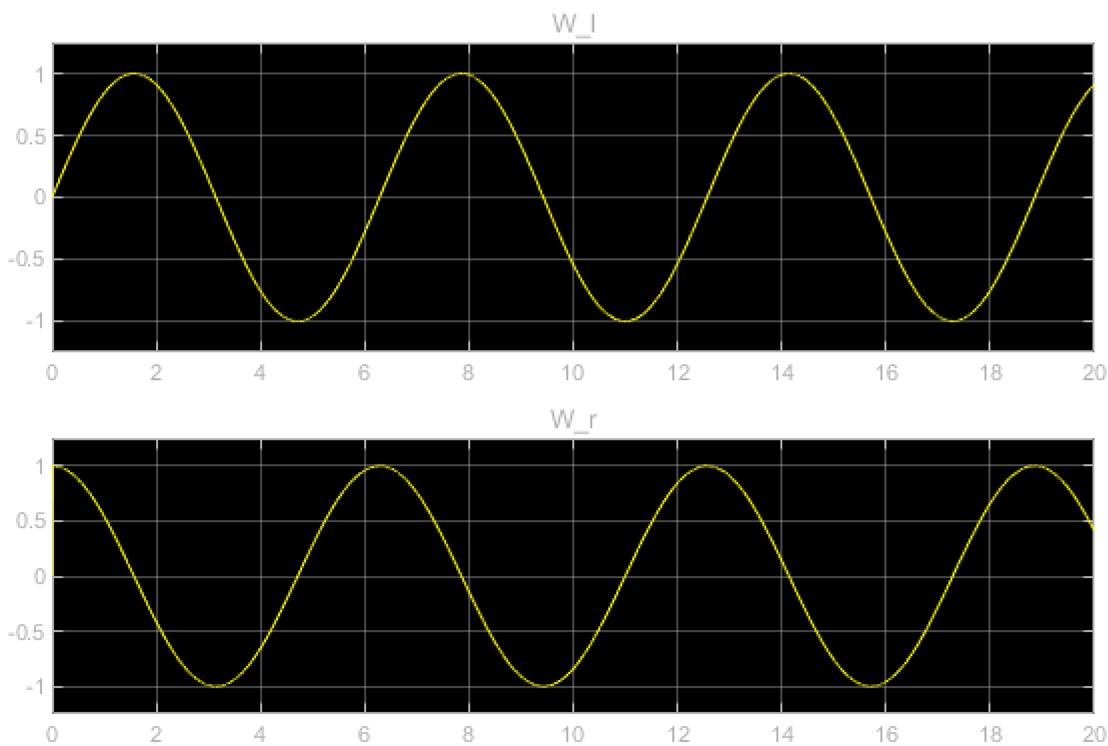


Task 2

1. SimMechanics model block diagram:



2. Plot of the robot wheel angular velocities v/s time:



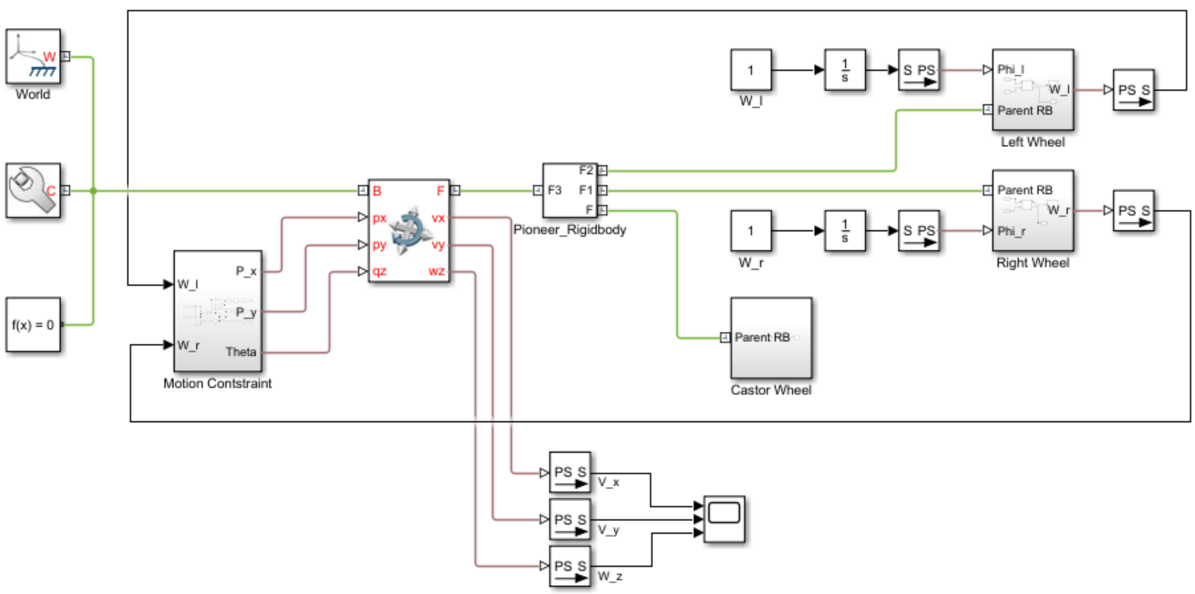
Task 3

1. Equation to find the robot linear and angular velocity from the wheel angular velocity:

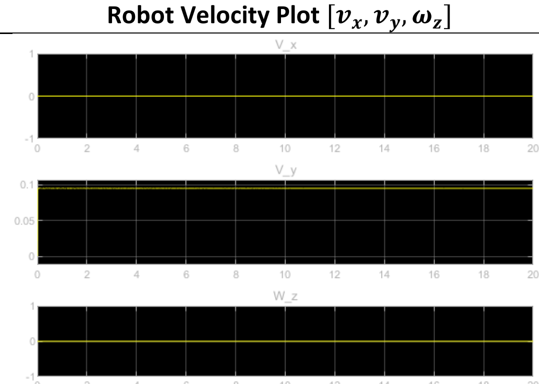
$$v = \frac{r_l * \omega_l + r_r * \omega_r}{2}$$

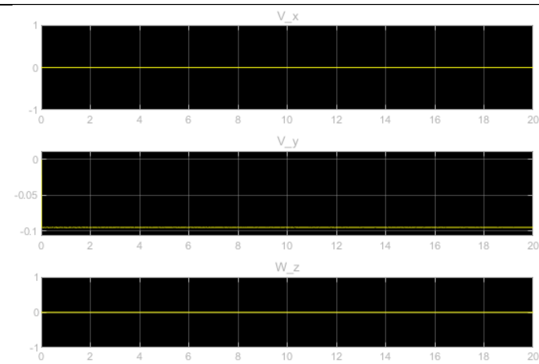
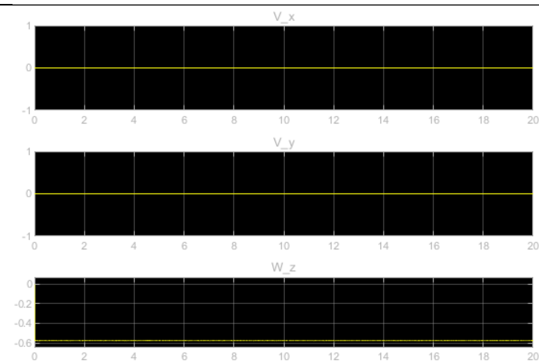
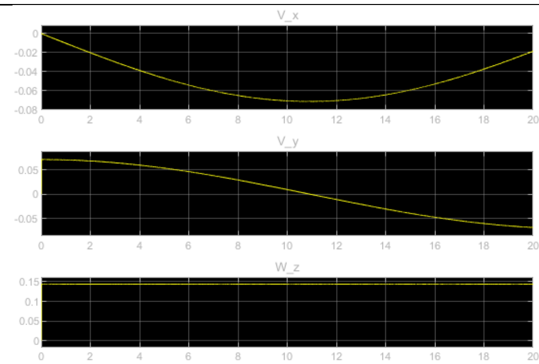
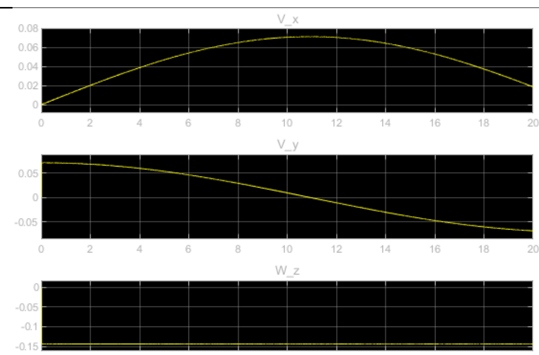
$$\omega = \frac{r_l * \omega_l - r_r * \omega_r}{2 * l}$$

2. SimMechanics model block diagram:



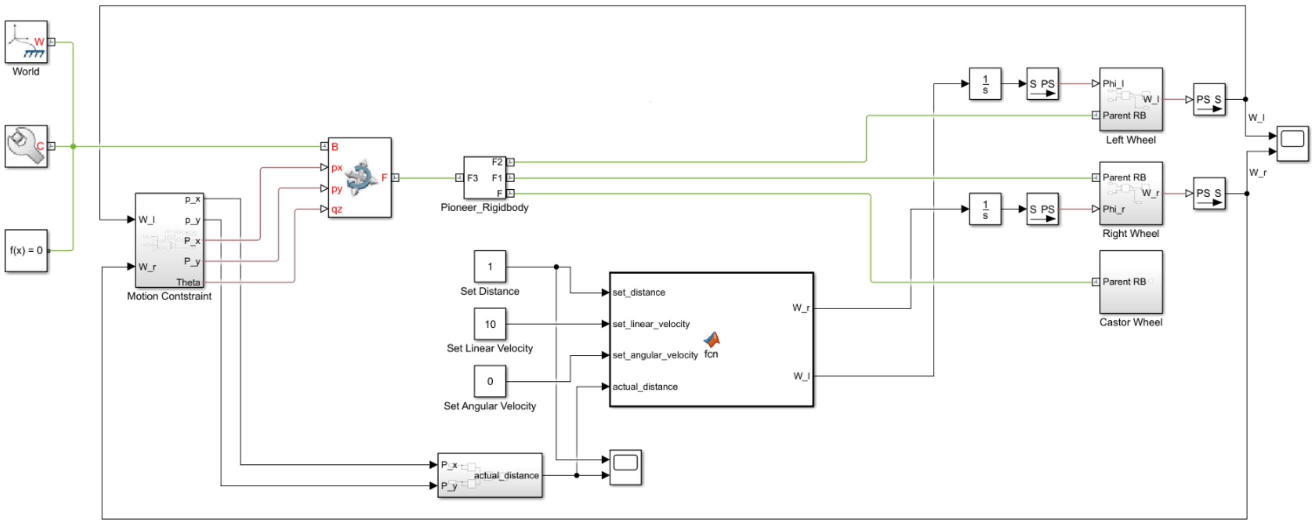
3.

ω_l (rad/s)	ω_r (rad/s)	Robot Velocity Plot $[v_x, v_y, \omega_z]$
1	1	

-1	-1	
1	-1	
0.5	1	
1	0.5	

Task 4

1. SimMechanics model block diagram:

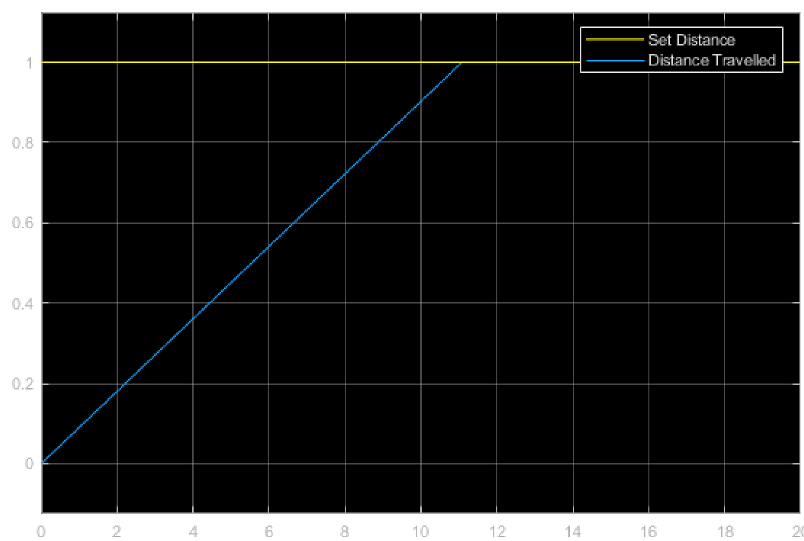


2. Equation to compute the angular velocity of the wheels from the robot linear and angular velocity:

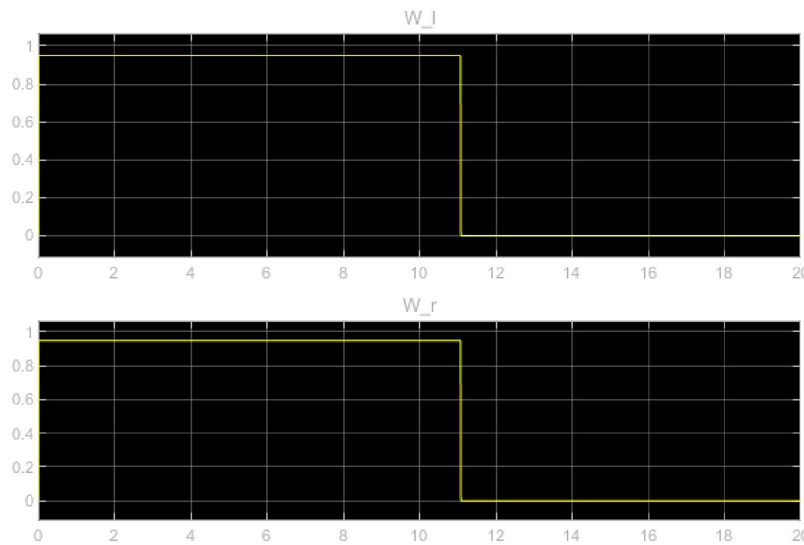
$$\omega_l = \frac{2 * v_{set} - \omega_{set} * l}{2 * r_r}$$

$$\omega_r = \frac{2 * v_{set} + \omega_{set} * l}{2 * r_r}$$

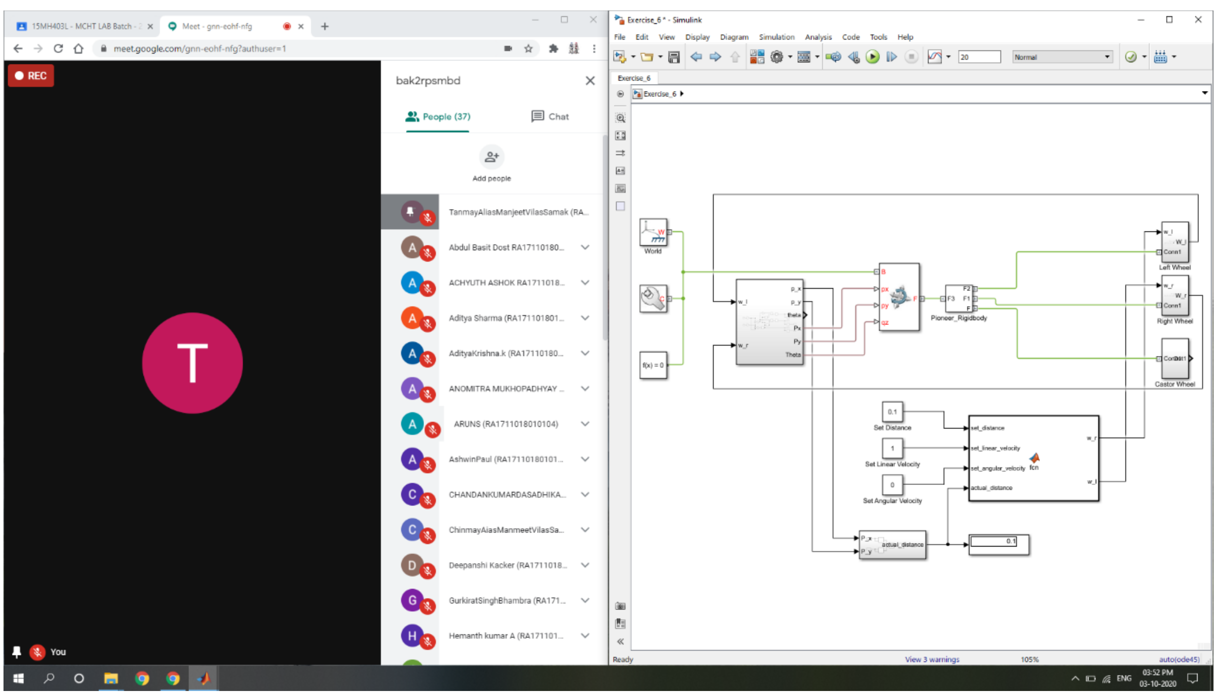
3. Plot of robot linear distance and set distance v/s time:



4. Plot of angular velocity of wheels v/s time:



Lab Session Screenshot



Inference

This experiment revised the concepts of physical modelling using SimMechanics. It further gave a deeper understanding of modelling a mobile robot in MATLAB Simulink along with the motion constraints and linear drive control for a preset distance.

The effect of gravity on physical models was observed in Task 1 by changing the gravity axis from Z to Y. The oscillations of castor wheel clamp indicated instability in the joint. Next, in Task 2, the left and right wheels were actuated to understand the effect of integrator to convert angular velocity commands into position control commands. Task 3 comprised of developing motion constraint equation and incorporating it into the SimMechanics model. The robot velocity was observed for various combinations of wheel velocities to understand their relation intuitively. Finally, in Task 4, the linear drive control was implemented using MATLAB Function block to drive the robot to a set linear distance (with zero heading angle) given robot linear velocity and angular velocity as design parameter.

This experiment gave a hands-on experience of modelling and controlling a differential drive mobile robot using MATLAB Simulink and will prove to be a stepping stone towards future mobile robotics projects.