ME145 Robotic Planning and Kinematics: Lab 3

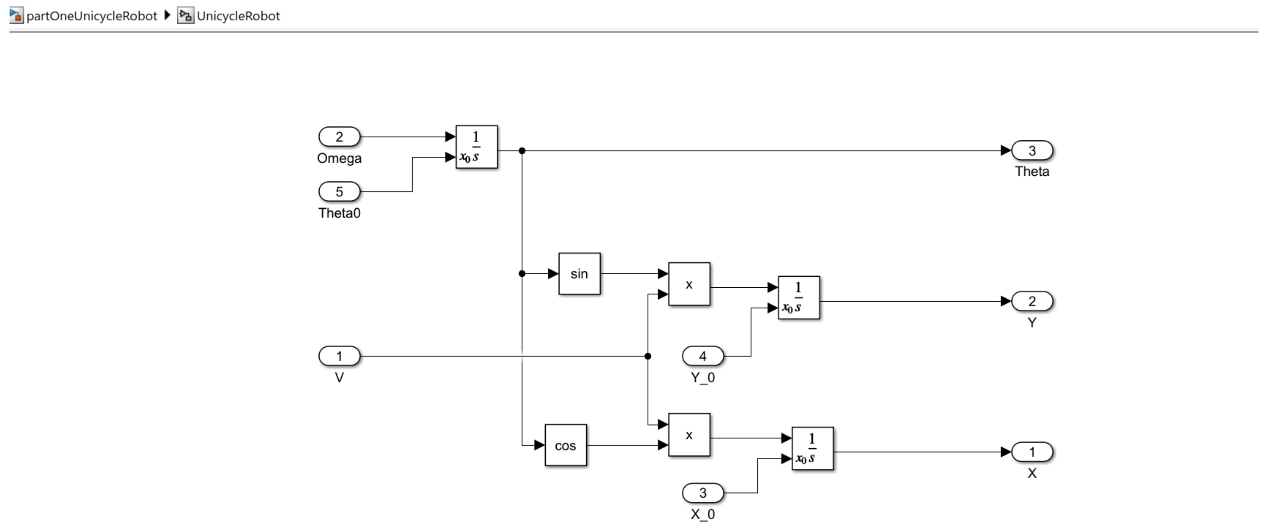
Line Following for Differential-Drive and Unicycle Robots

Eric Perez

Spring 2024

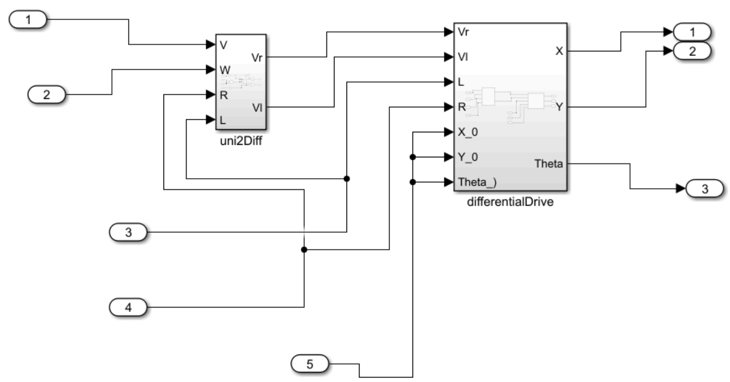
May 06, 2024

1) Simulink Model for Unicycle Robot

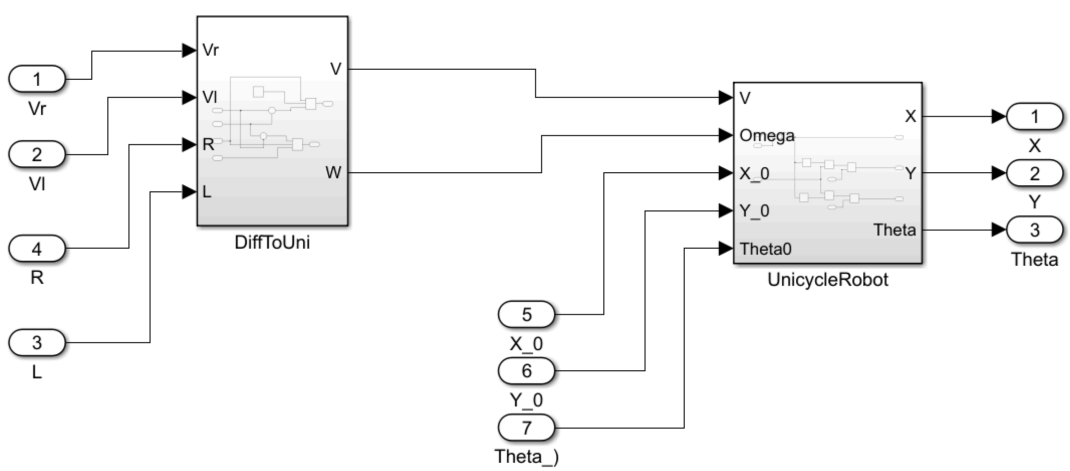


This model was created based on the equations derived in the lecture for a unicycle robot. Where $X' = V\cos(\theta)$, $Y' = V\sin(\theta)$, and $\theta' = \omega$. This model takes in omega, velocity, and initial conditions as inputs and produces x,y, and theta.

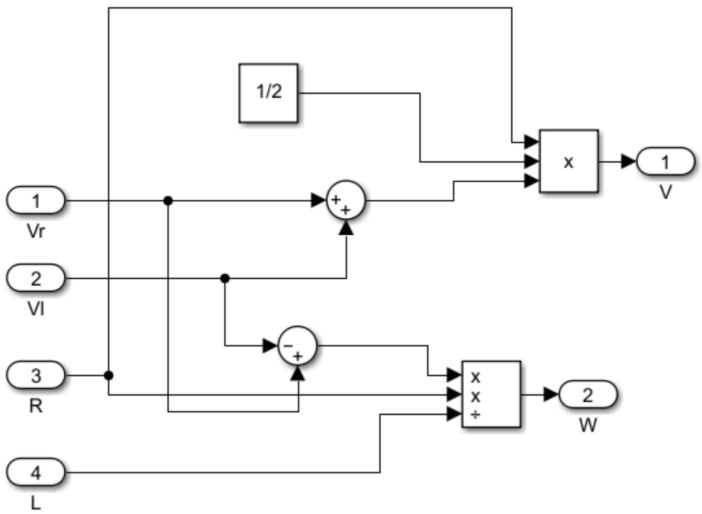
2) Simulink Model for a differential-drive robot



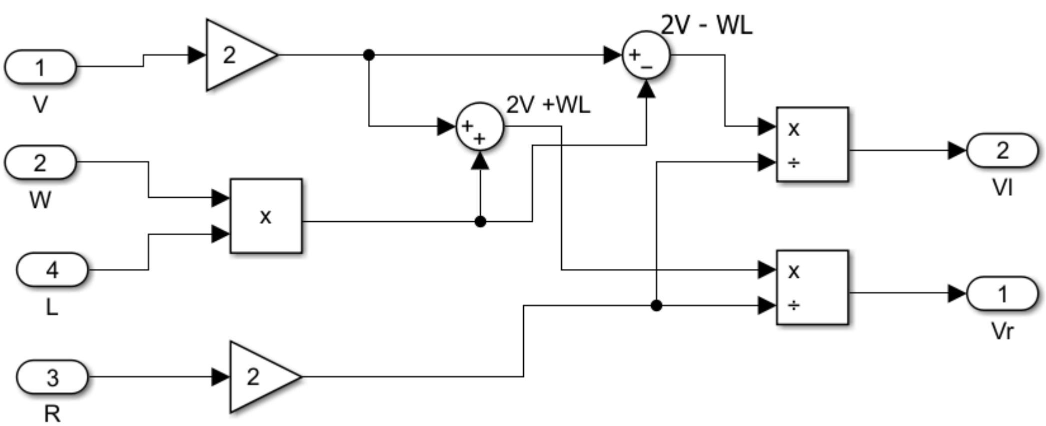
Inside differentialDrive subsystem:



Inside DiffToUni subsystem:



Inside UnitToDiff Subsystem:



This model was created using two subsystems. The differentialDrive subsystem contains two subsystems, DiffToUni and UnicycleRobot, and takes inputs Vr, Vl, R, L, and initial position.DifftoUni converts the differential drive inputs (Vr,Vl,R, and L) into acceptable inputs (V and omega) for the UnicycleRobot subsystem. The UniToDiff subsystem converts the initially given input into acceptable inputs for the differentialDrive subsystem. The transformation equations used were derived in class.

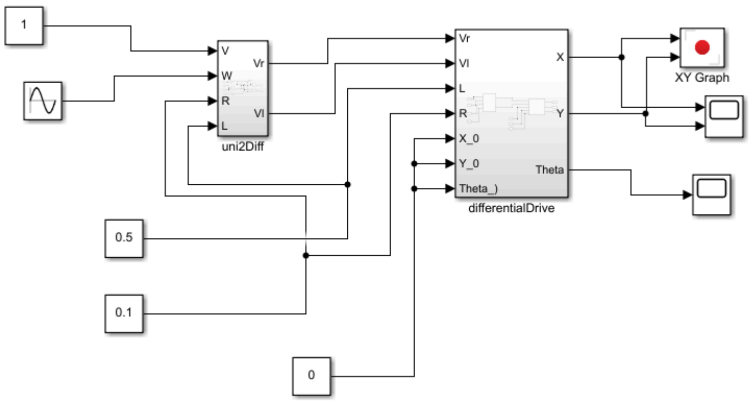
For UniToDiff:

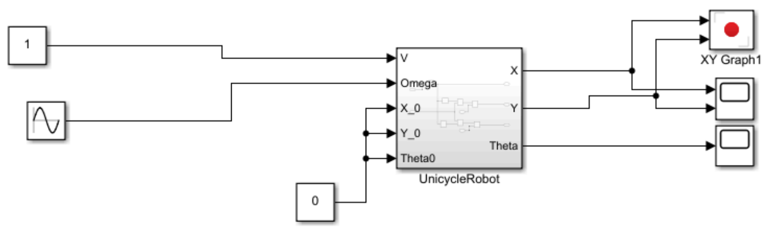
- $V_R = \frac{2V + \omega L}{2R}$ and $V_L = \frac{2V - \omega L}{2R}$

For DiffToUni:

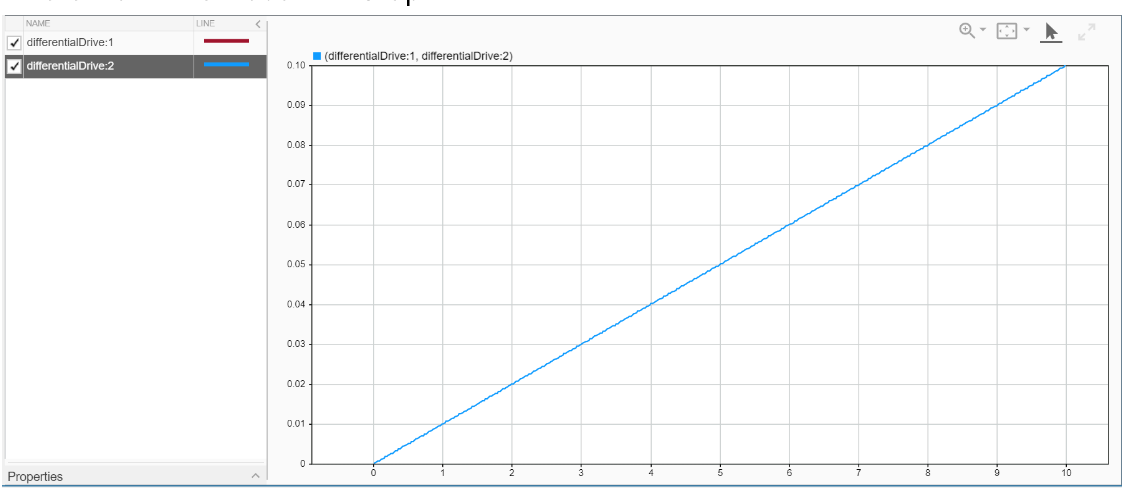
- $V = \frac{R}{2}(V_R + V_L)$
- $\omega = \frac{R}{L}(V_R - V_L)$

3) Testing the above models with V=1, Omega=sin(100t), R=0.1, L=0., and initial conditions set to zero. In the images below, the position and orientation of the robots are proven to be the same by using the scope block to plot the robot's x,y, and theta outputs in a 10-second span.

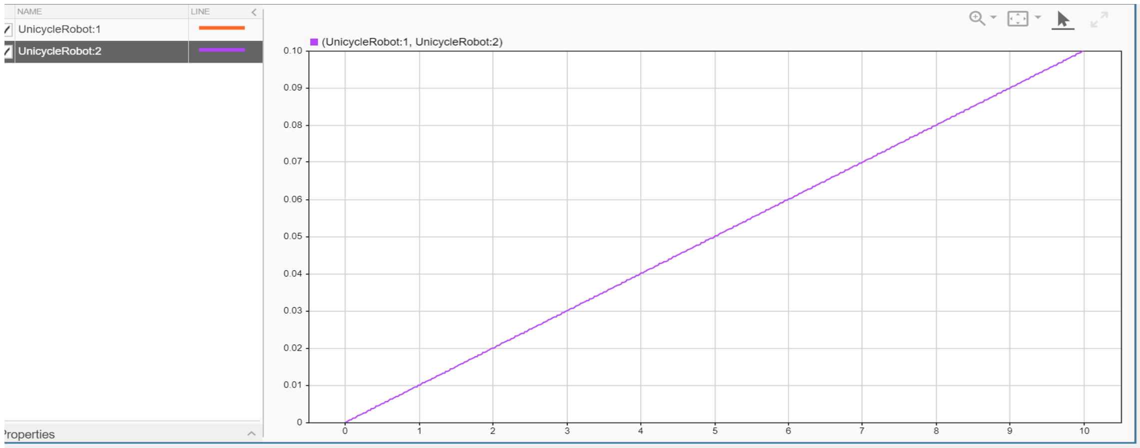




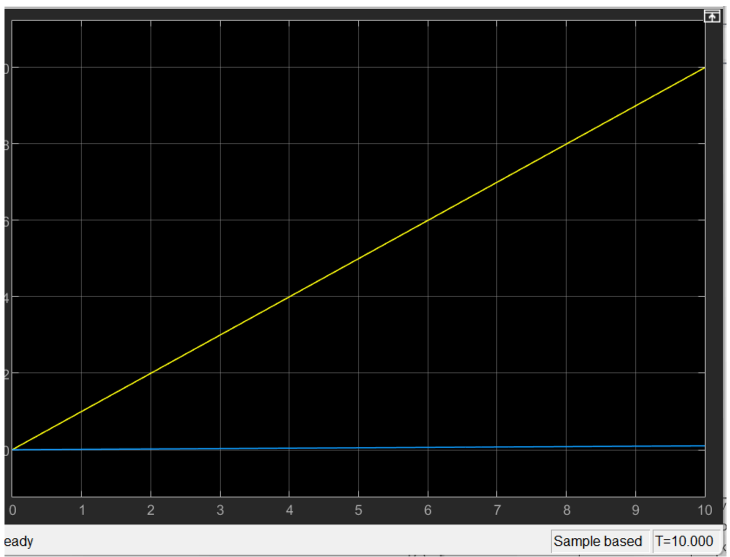
Differential Drive Robot XY Graph:



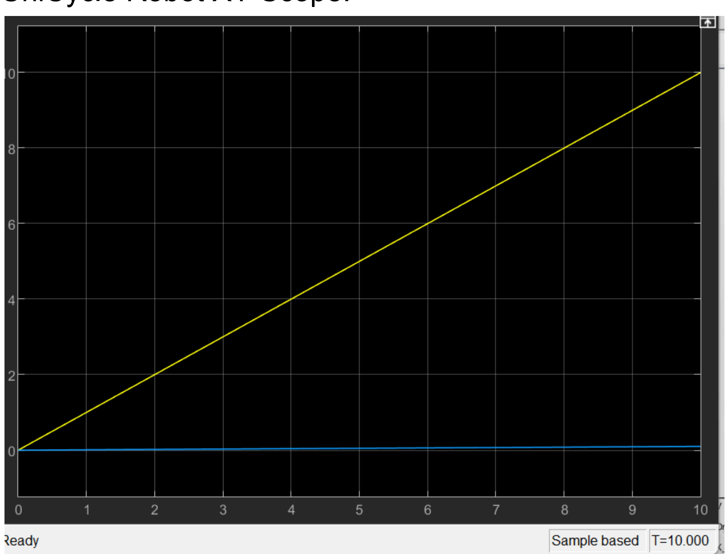
UniCycle Robot XY Graph:



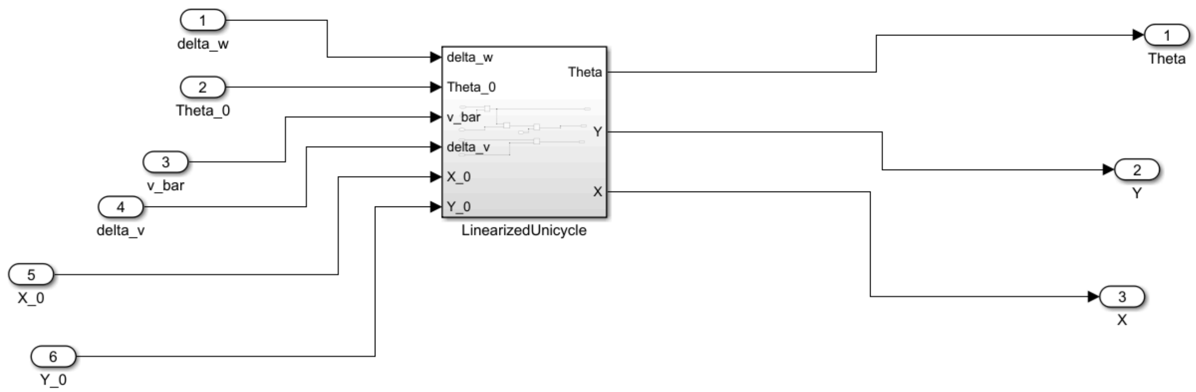
Differential Drive Robot XY Scope::

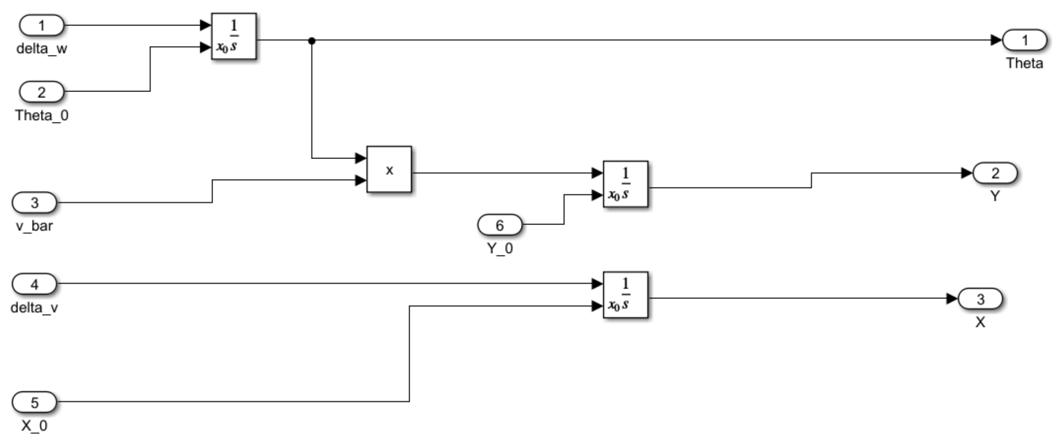


UniCycle Robot XY Scope:



4) Simulink Model for a Robot with Linearized Unicycle Dynamics:

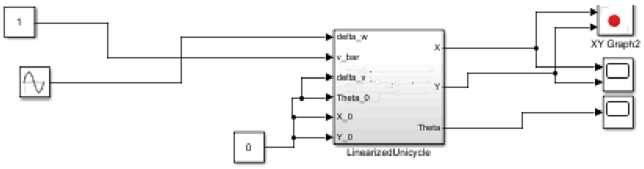


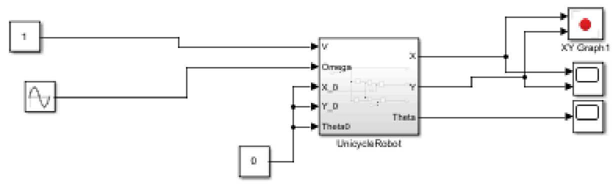


This model was created by deriving equations for each of the inputs. The equations were derived based on the desired X, Y, and theta (X_D= Vbar*t, Y_D = 0, $Theta_D$= 0). A matrix was formed based on the linearized unicycle dynamics around horizontal trajectory velocity (In lecture).The equations used to find the outputs are listed below.

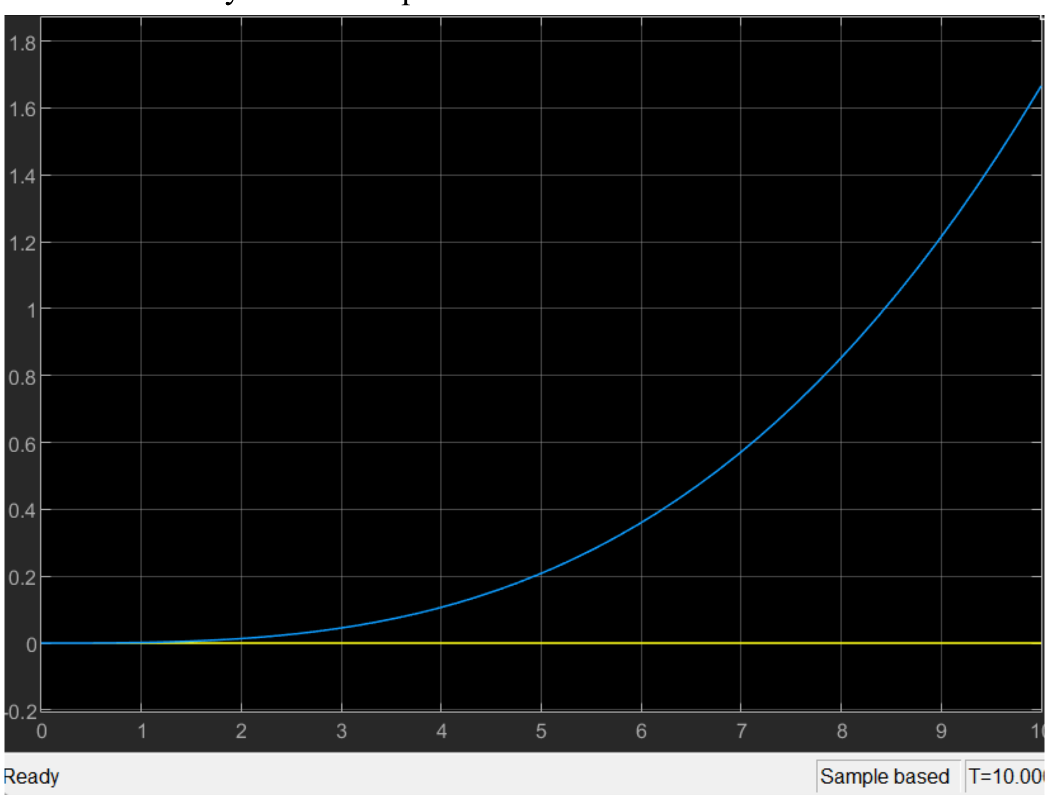
- $X' = \delta V$
- $Y' = Vbar*\theta$
- $\theta' = \delta\omega$

5) Comparing the unicycle robot with its linearized model. Inputs: V = 1 and Omega = sin(0.01*t).

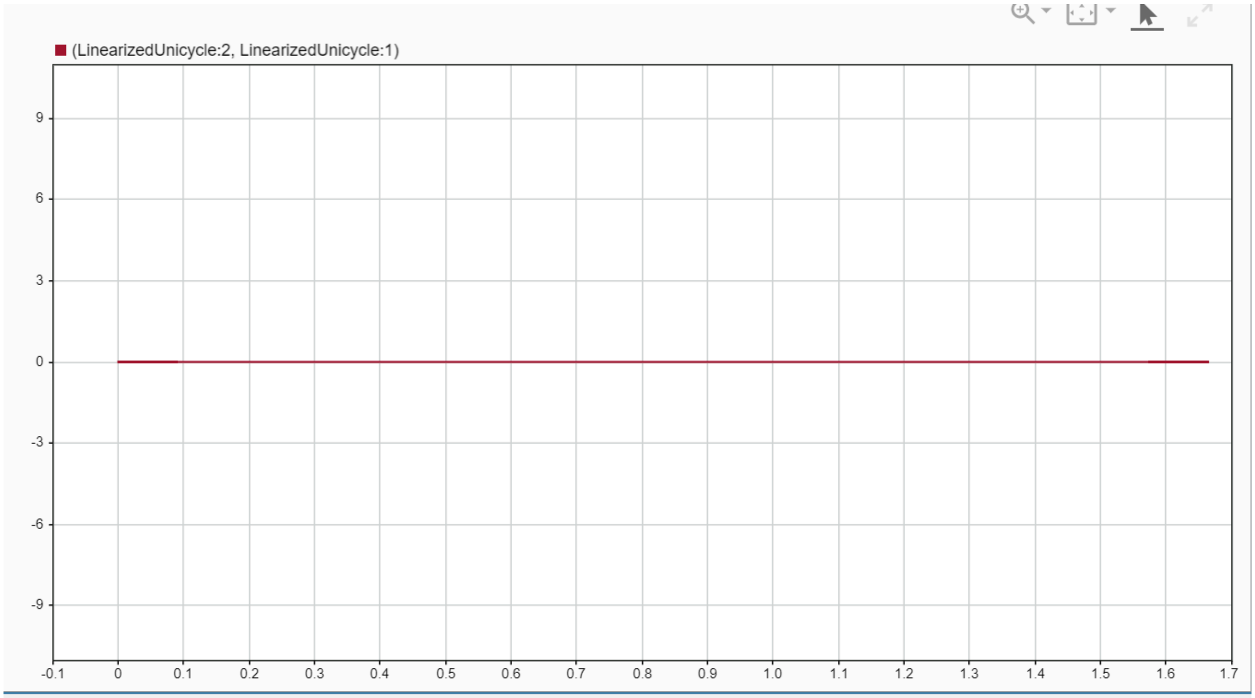




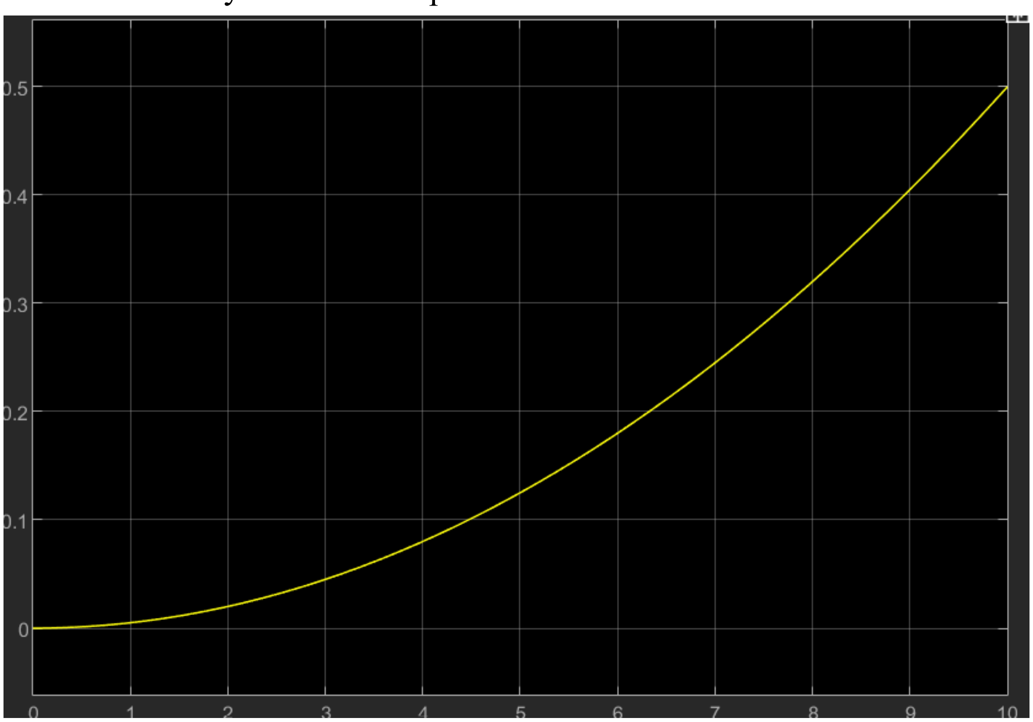
Linearized Unicycle XY Scope:



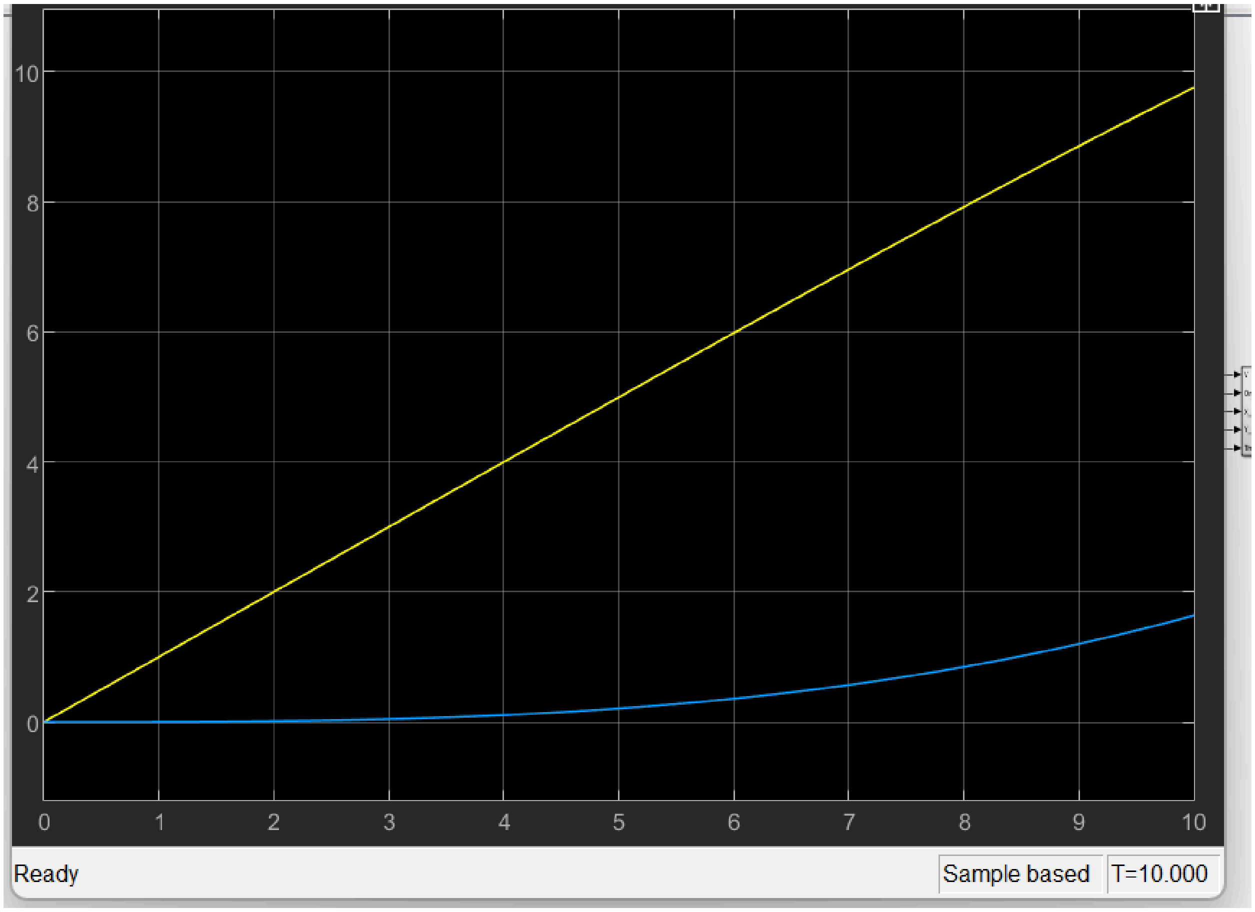
Linearized Unicycle XY Graph:



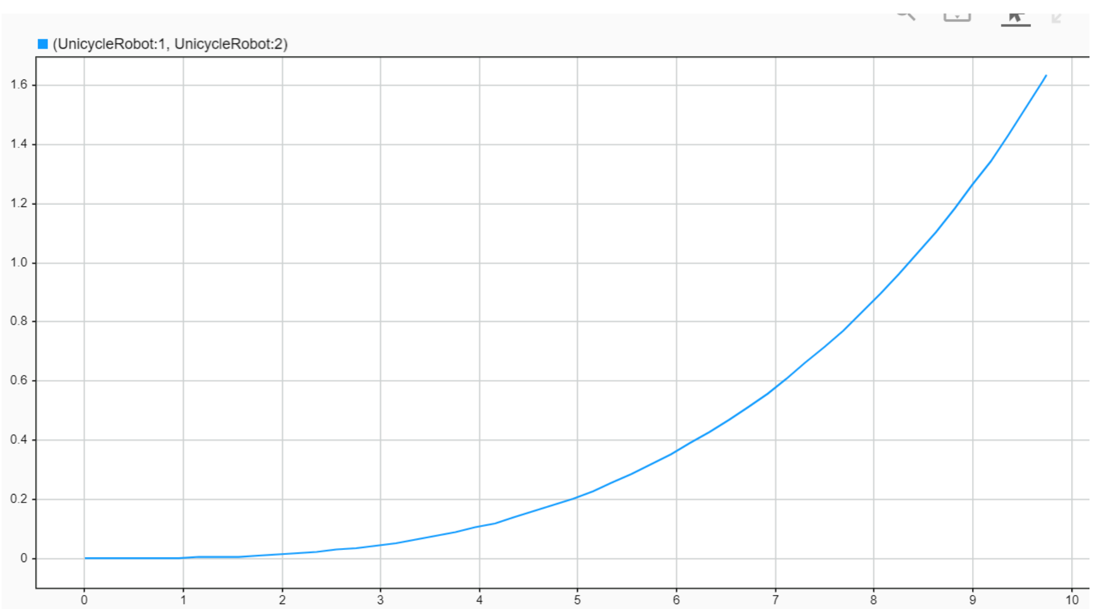
Linearized Unicycle Theta Scope:



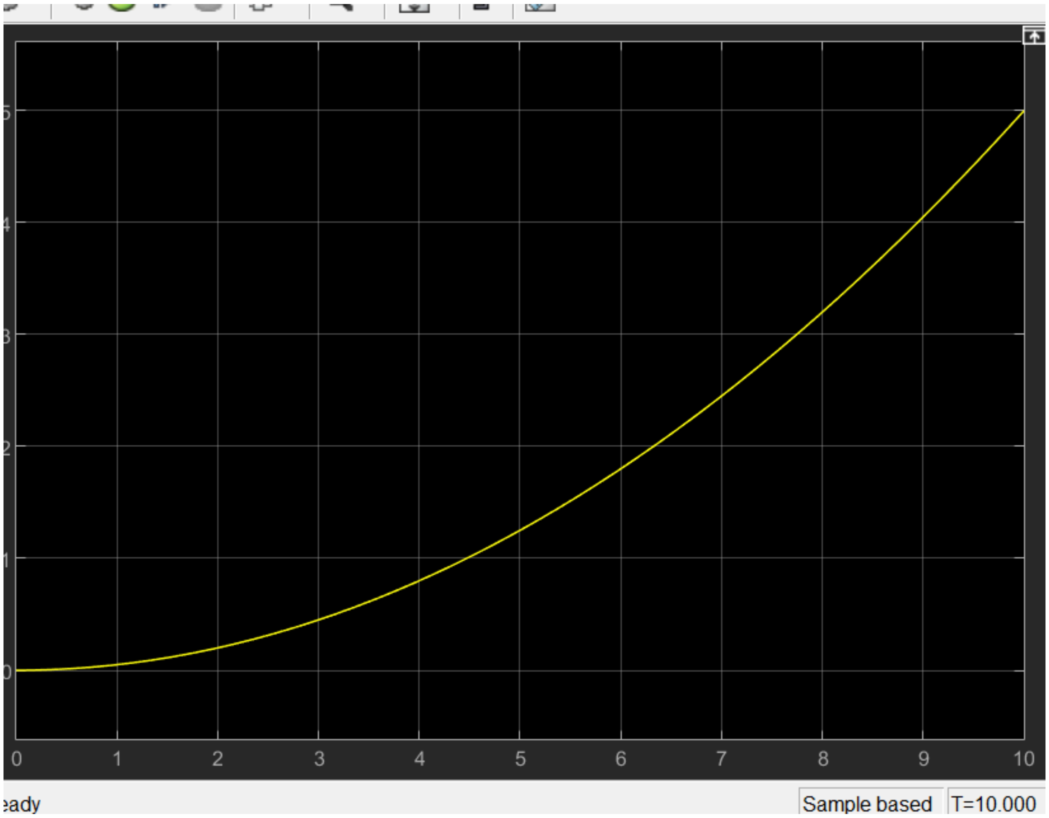
Unicycle Robot XY Scope:



Unicycle Robot XY Graph:

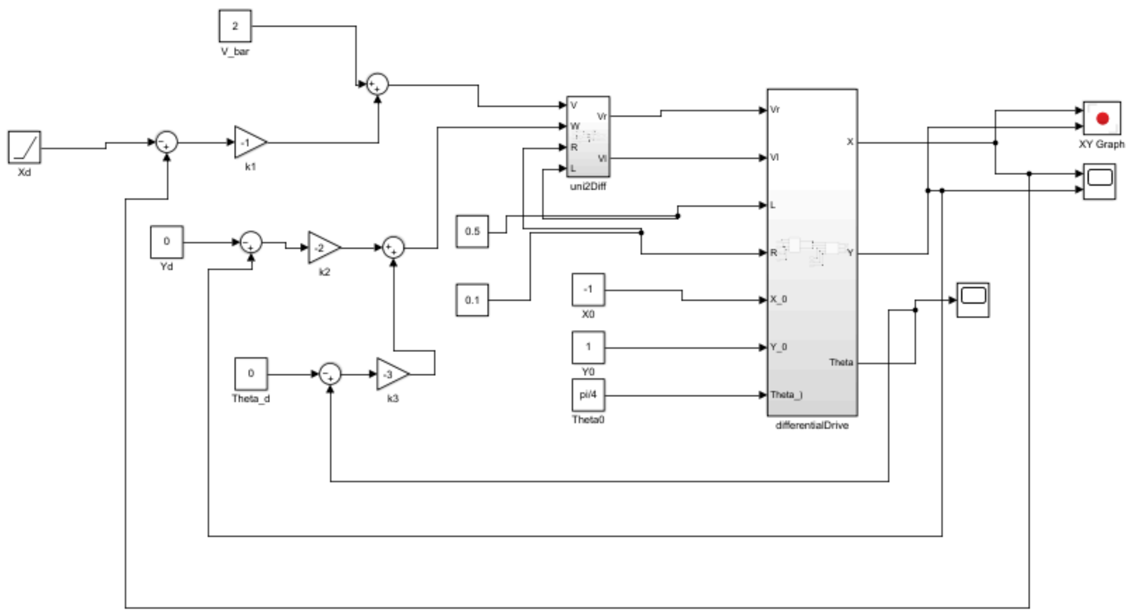


Unicycle Robot Theta scope:



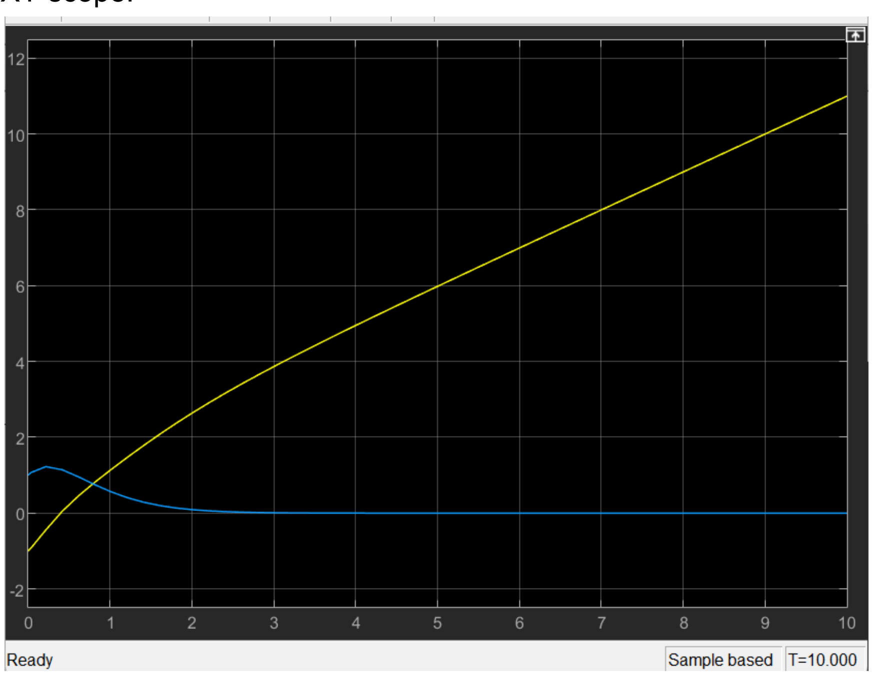
As seen in the images above, when testing, the robots have different orientations as the linearized robot only follows the horizontal direction.

6) Control law based on the linearized unicycle dynamics that force the differential drive robot to follow a straight line along the x direction with a velocity equal to 2 m/s:

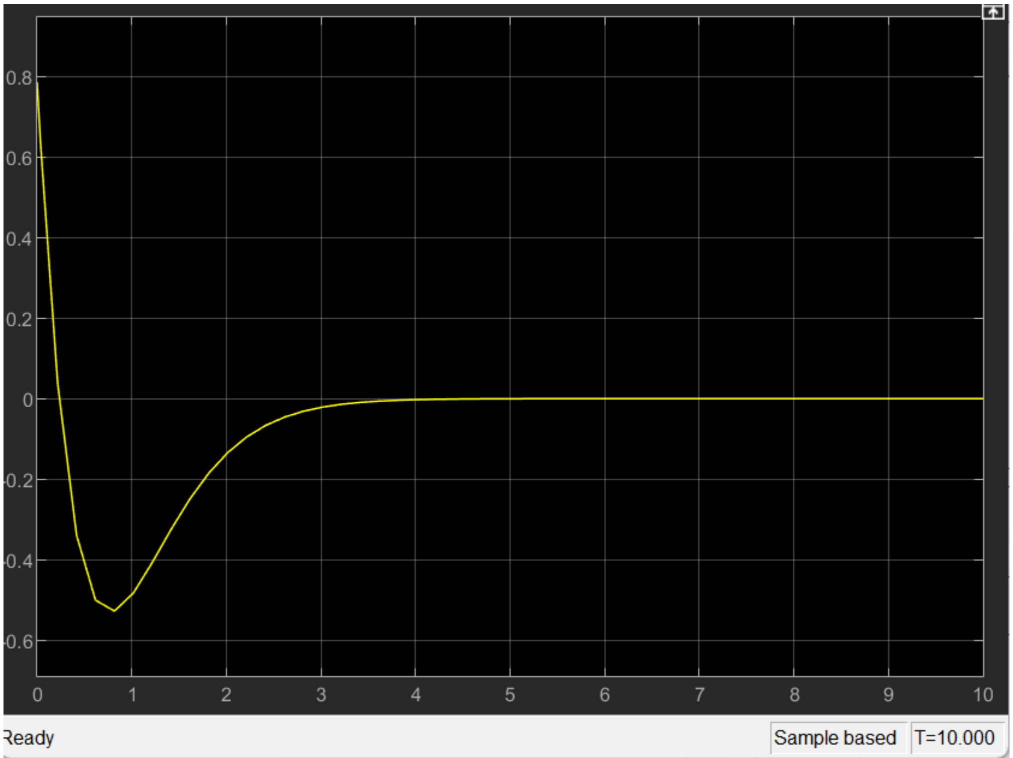


K values -1,-2, and -3 were selected to create a stable controller. These K values are used in the following equations: $\delta V = K_1 \cdot \delta X$ and $\delta\omega = K_2\delta y + K_3\delta\theta$. Using the uni2Diff subsystem allowed the user to put in V and omega as inputs and for the inputs to work with differential drive. Below are the images of the scopes with initial conditions X0 = -1, Y0 = 1 and Theta0 = pi/4.

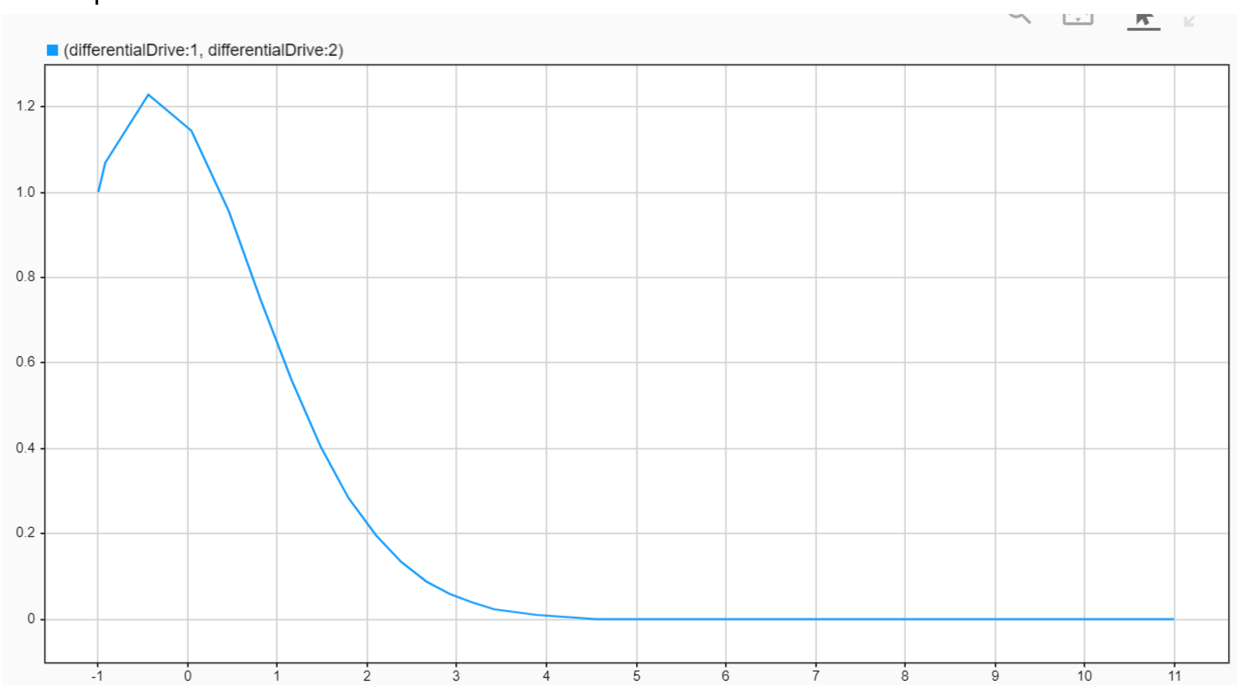
XY scope:



Theta Scope:

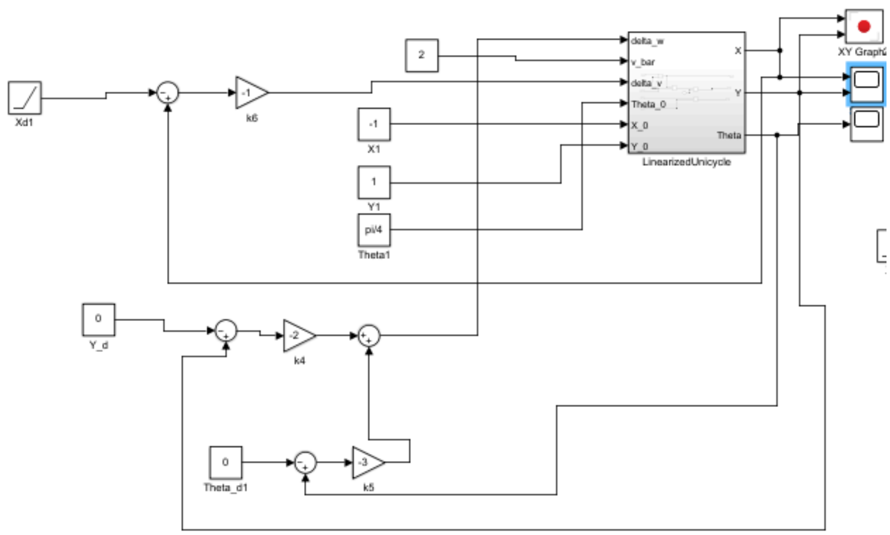


XY Graph:

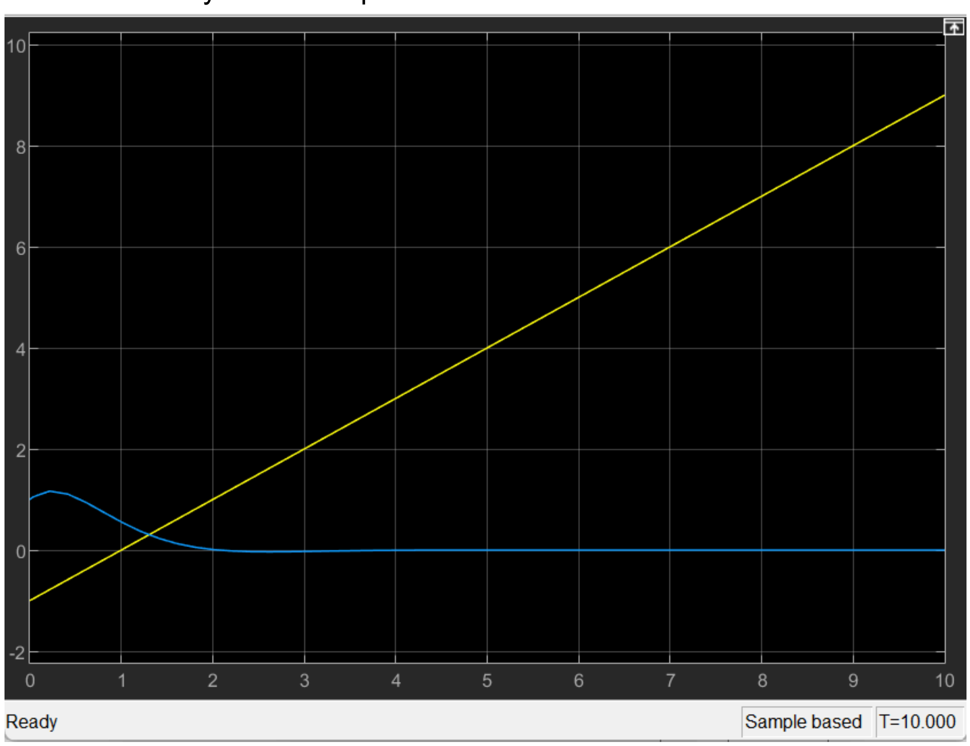


Below are images of the setup and results when comparing with the linearized unicycle model.

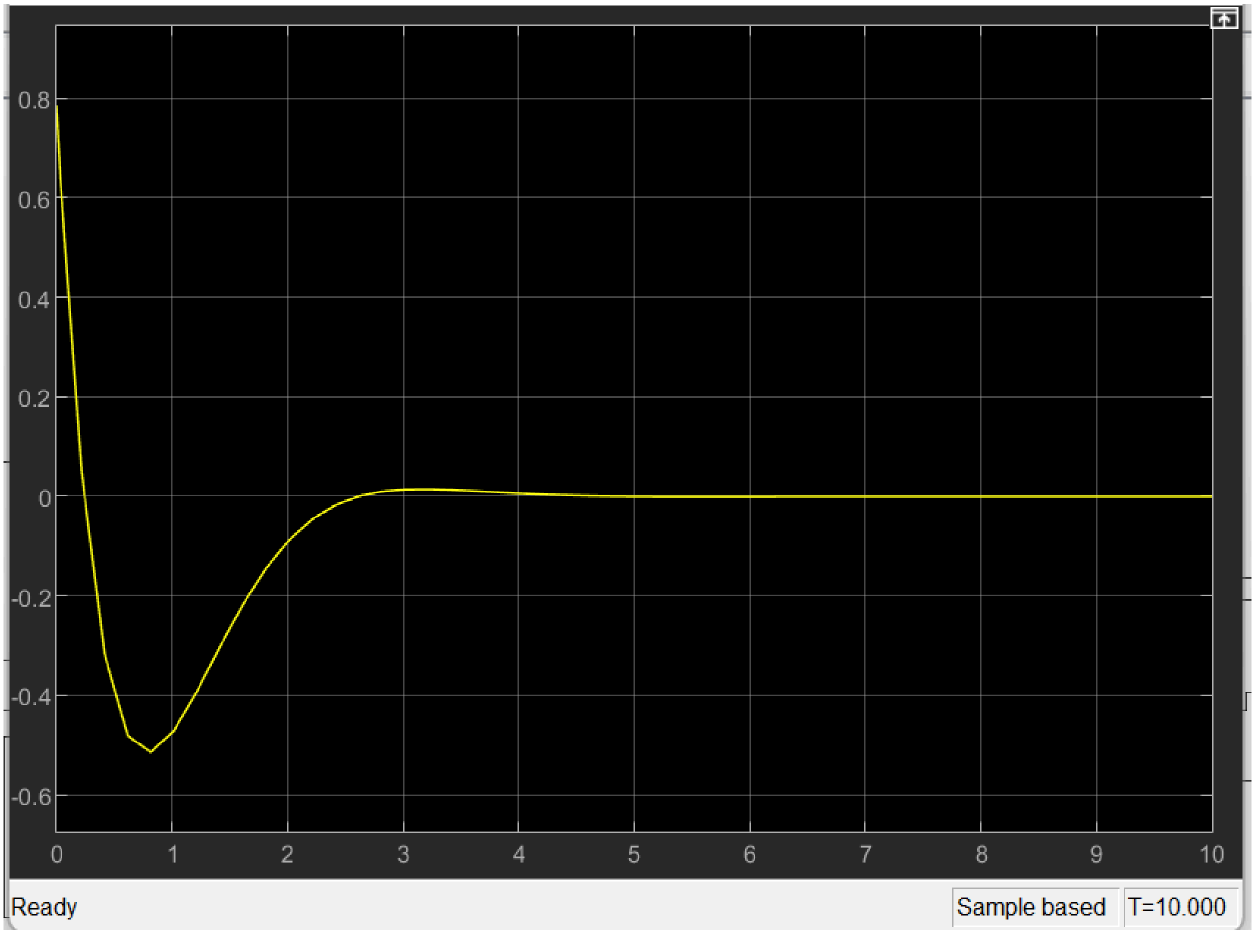
Simulink model of the linearized unicycle with the same inputs for comparison:



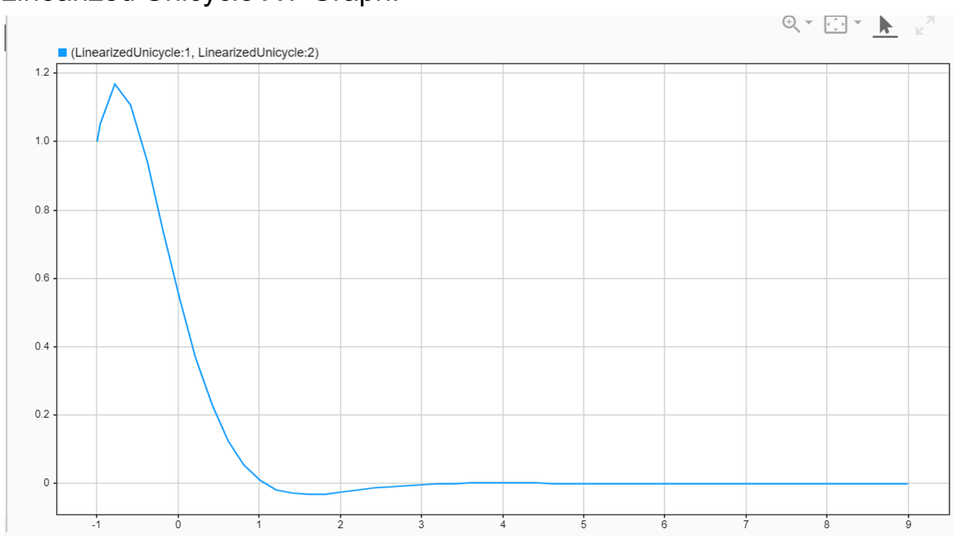
Linearized Unicycle XY Scope:



Linearized Unicycle theta Scope:



Linearized Unicycle XY Graph:

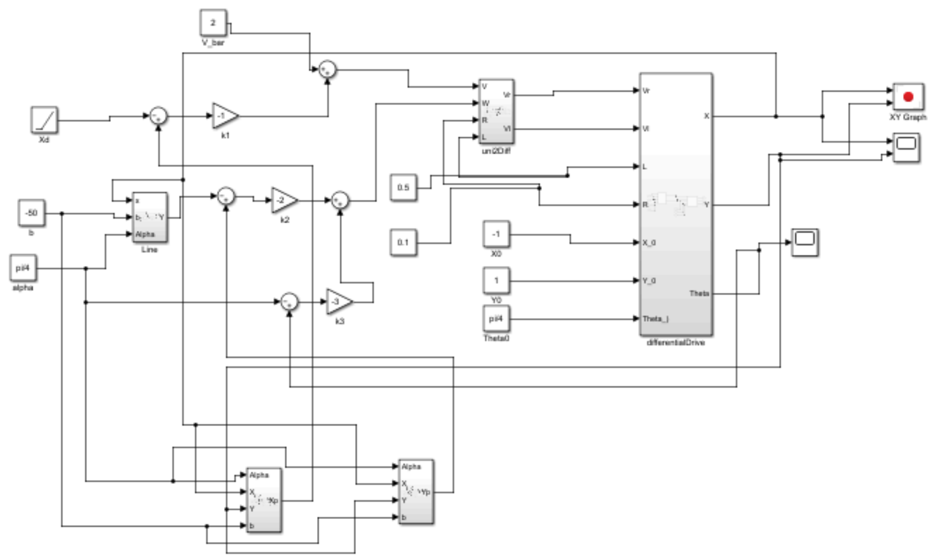


:

As seen in the images above, the trajectories overall match with slight differences.

7) Control law based on the linearized unicycle dynamics that force the differential drive robot to follow a straight line written as y = tan(Alfa)x + b with velocity equal to 2 m/s.

   Simulink model of the controller set up with differential drive robot:



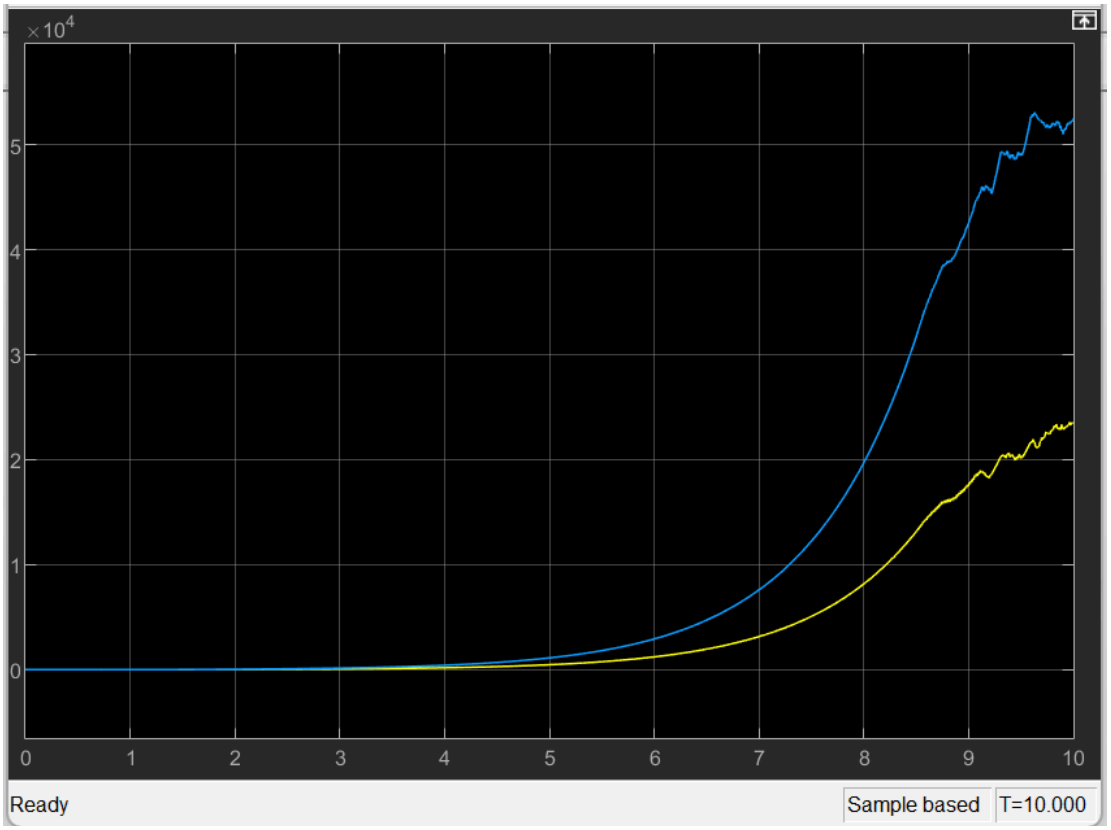
The desired y becomes tan(Alfa)x + b with alfa tested at pi/4 and b tested at -50. I used the same controller that was designed for the horizontal line. To do so, I made a change of coordinates (rotation) so that the oblique line becomes/coincide with the new "x" axis. This is done by taking the x,y, and theta coordinates of the actual unicycle and computing the following quantities:

- x_p = x*cos(alfa) + y*sin(alfa) - b*sin(alfa)
- y_p = -x*sin(alfa) + y*cos(alfa) - b*cos(alfa)
- theta_p = theta - alfa

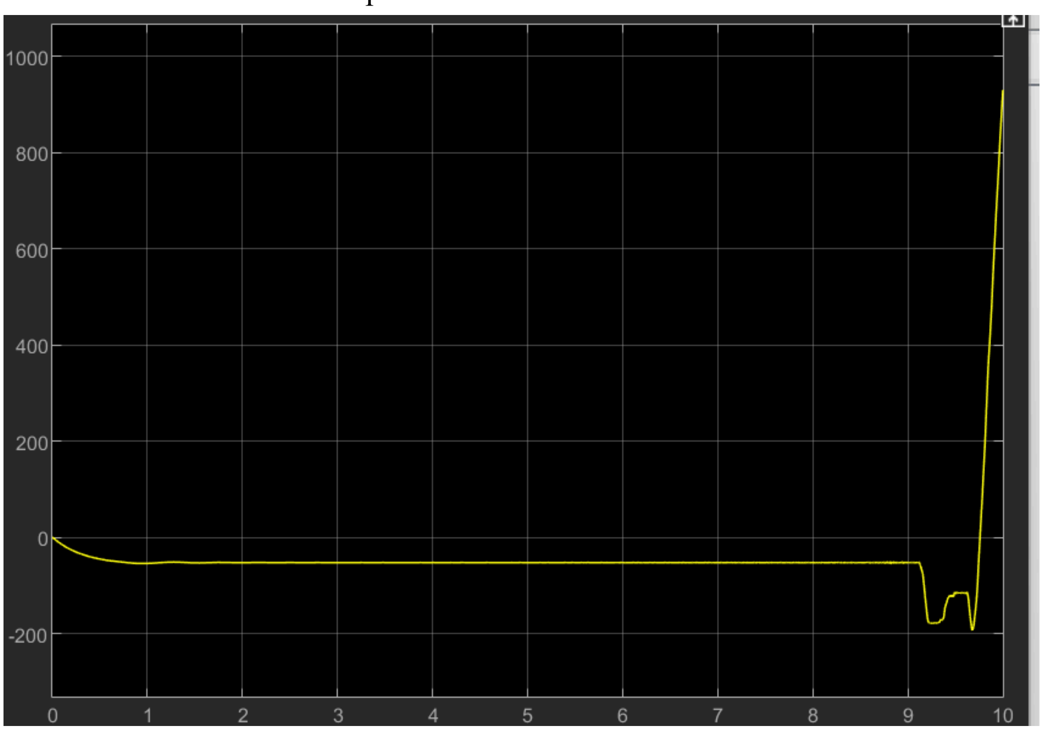
Now, x_p, y_p, and theta_p are the position and orientation of the unicycle in the new coordinate frame (where the x axis coincides with the oblique line).

Testing the controller: X0 = -1, Y0 = 1 and Theta0 = pi/4:

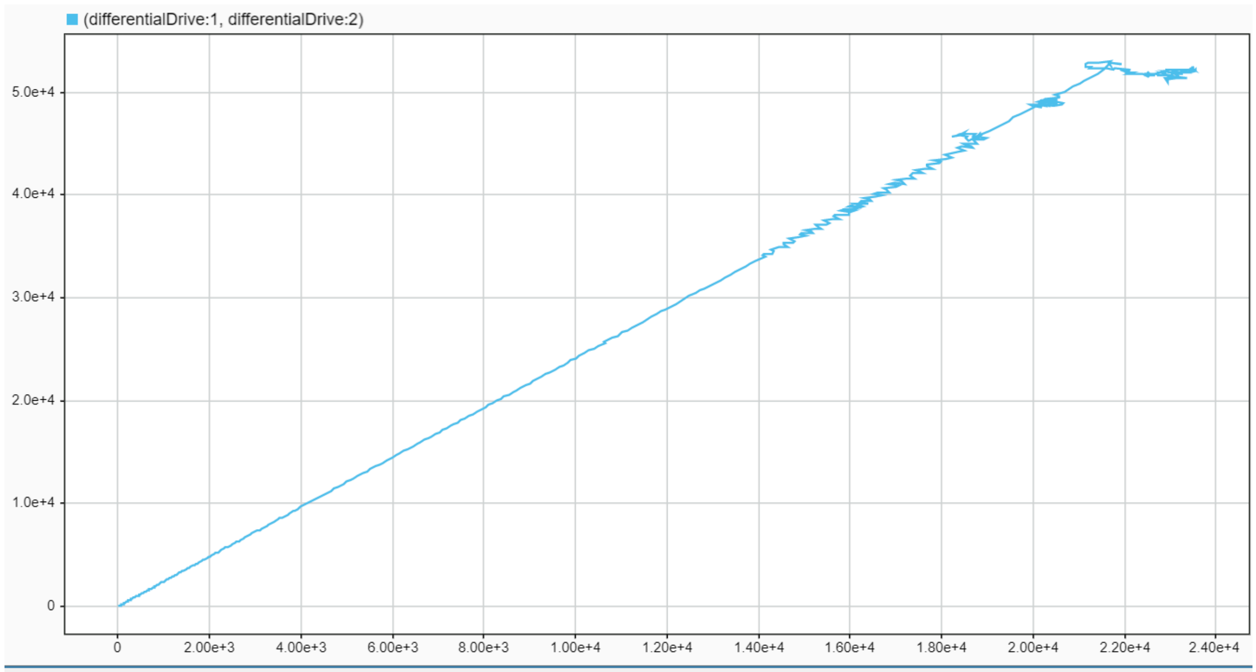
Differential Drive XY Scope:



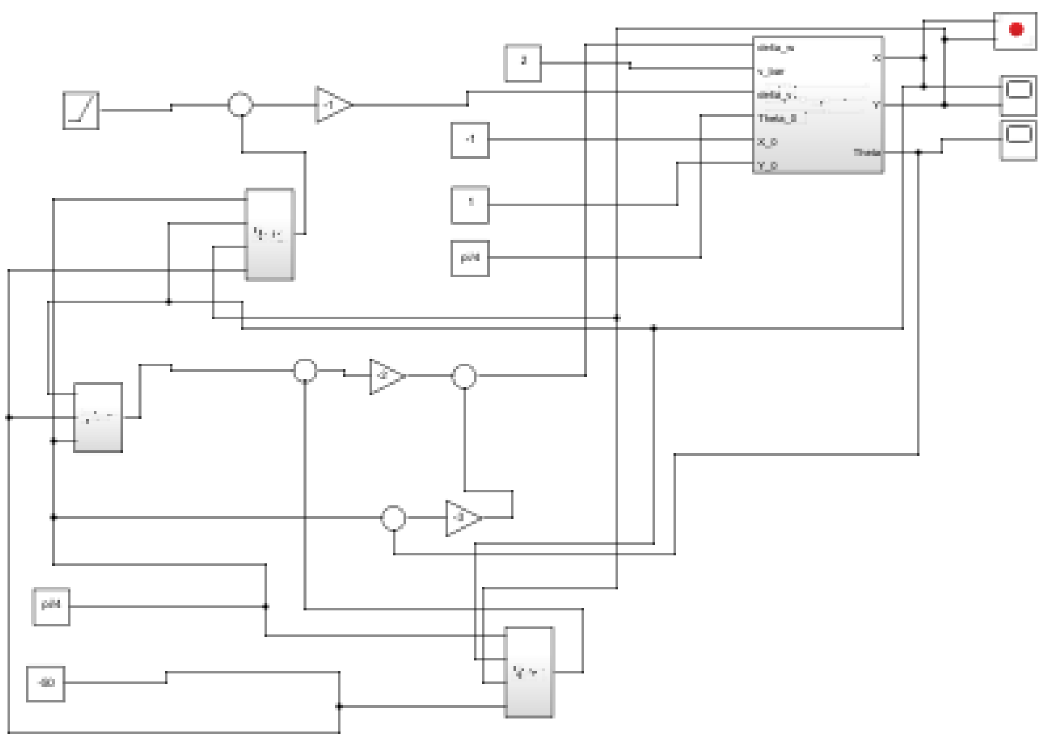
Differential Drive Theta Scope:



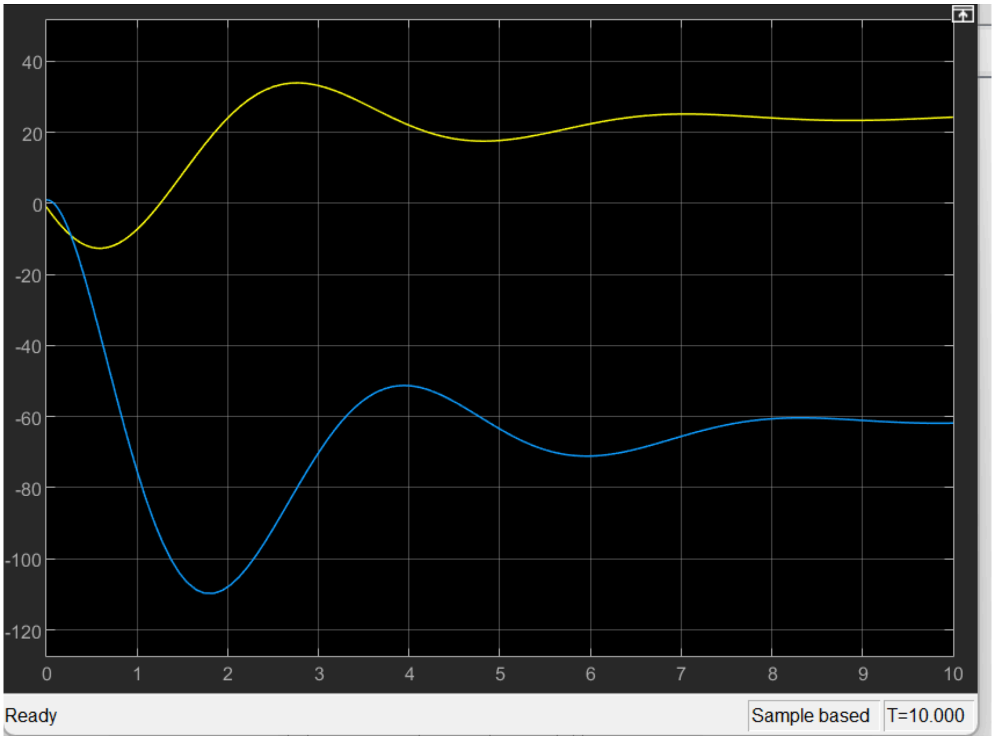
Differential Drive XY Graph:



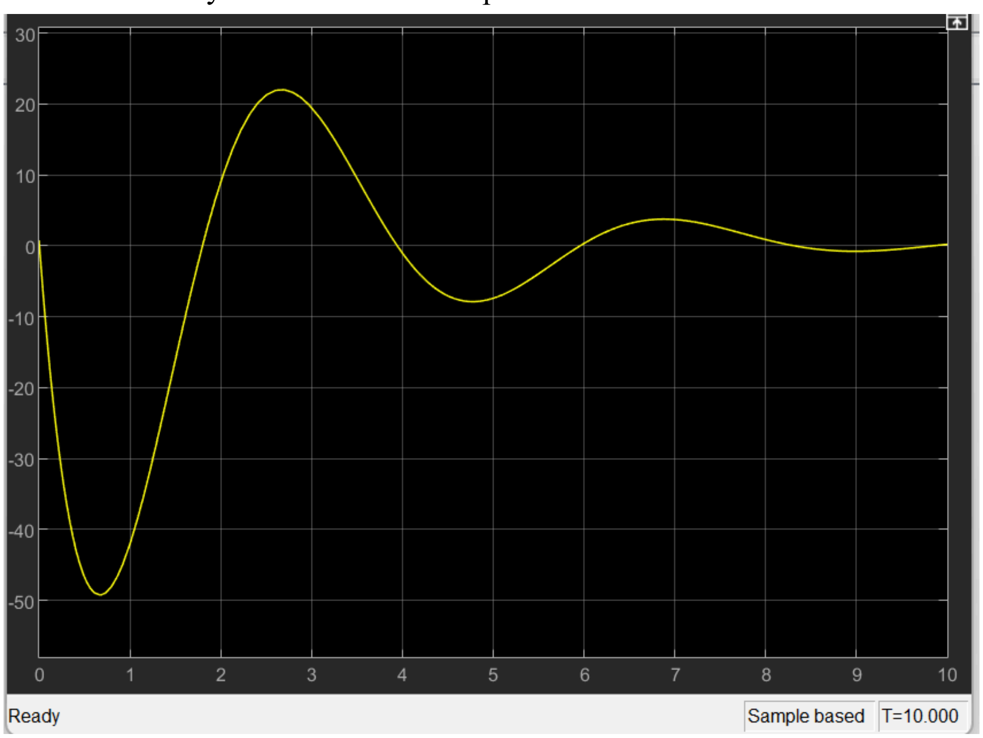
The controller was used on the linearized Unicycle Robot for comparison.
Simulink Model of the controller set up on the linearized Unicycle Robot:



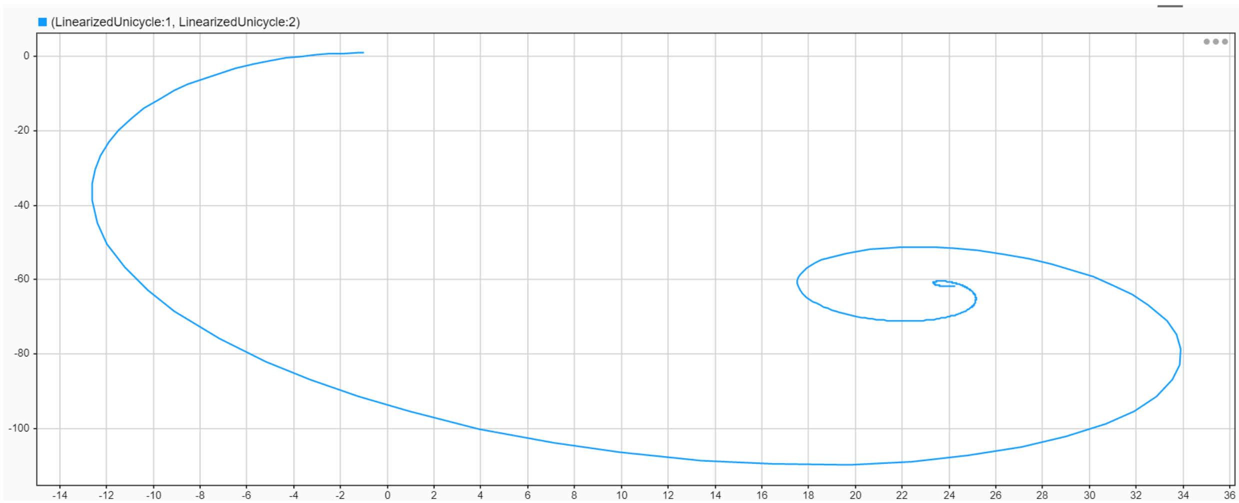
Linearized Unicycle Robot XY Scope:



Linearized Unicycle Robot Theta Scope:



Linearized Unicycle Robot XY Graph:



As seen in the images above, the two robots do not match and this may have been due to a controller setup error.