**MOBRO**

Lab 3

By :

Raghuveer Siddaraboina

**(2,0)Differential Drive Robot**

**Static Feedback Based Control**

For static feedback, we need to follow some steps to achieve the desired control law.

Given that

and.

.

The matrix can be found out by finding the coefficients of and implementing the kinematic model.

So,

.

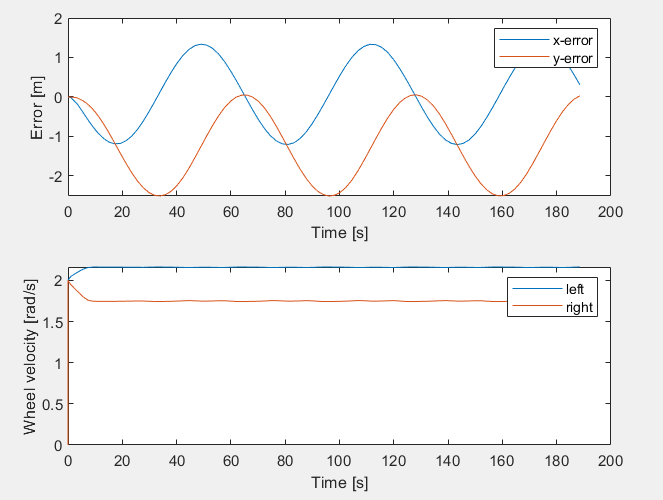
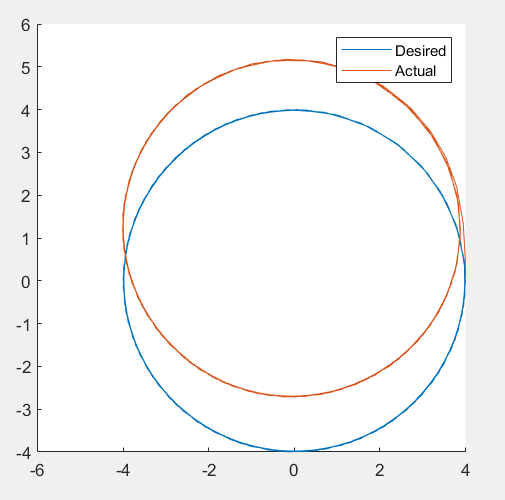
To satisfy the above equation, K matrix should be invertible or it should have non-zero determinant. So,we choose.

Also given

So,

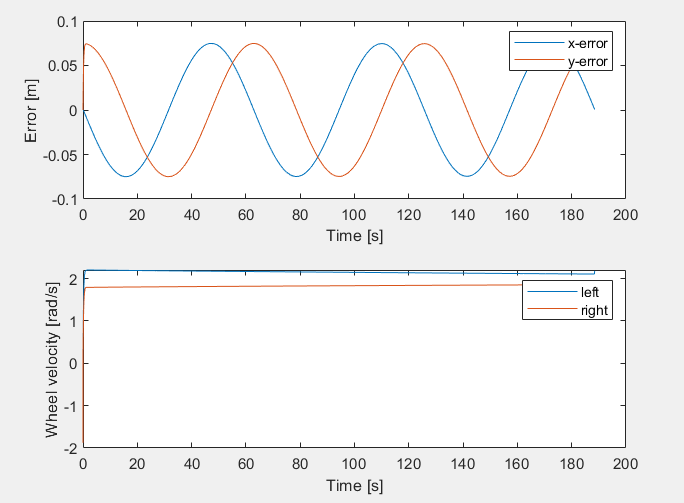
As is invertible, we can define the control law as: -

**Observations for different values**



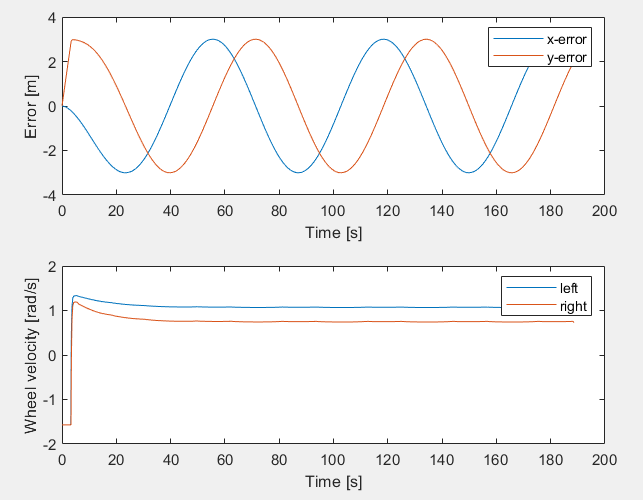
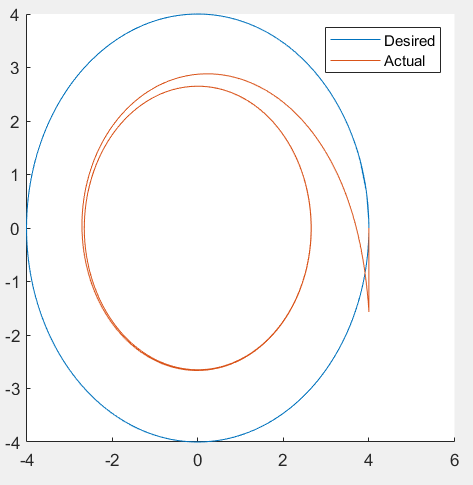
Kp = 0 and d = 1

This is an open loop static feedback and here we do not take the error into consideration i.e , the term xr – xp has no effect and control is generated assuming that the path is perfect,which is not the case.Hence the trajectory drifts and we see a lot of errors



Ideal Kp = 5 and d = 0.075

This is for ideal conditions ,Here we see that control is correct in the error so we get good trajectory



Kp = 5 and d = 3 and Vel = 70 velocity saturation

In this case the value of d is very high therefore the control point xp is far away from the robot frame and all control signals are generated from this point hence we see errors in the trajectory

**Lyapunov based control**

The classical function for (2,0) robot is given by: -

Here is the posture error in robot frame.

We know that, where depends on the choice of W and depends on the control input .

For lyapunov control, it’s necessary to have as negative definite, i.e. , must be <0 so that the error converges to 0.

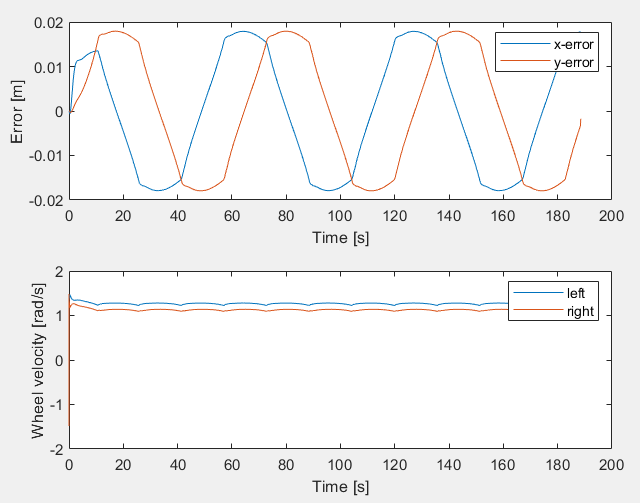
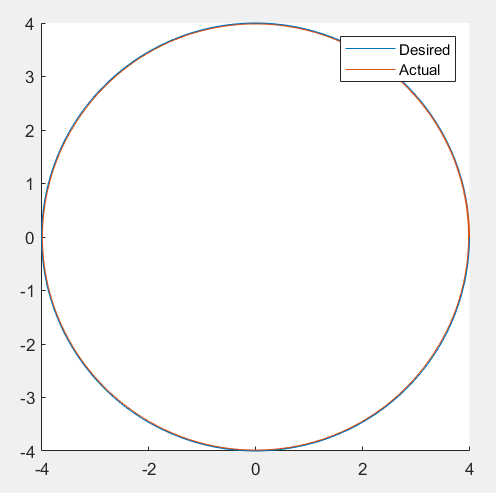
Change of error with respect to time is: -

[using the model]

[using the model]

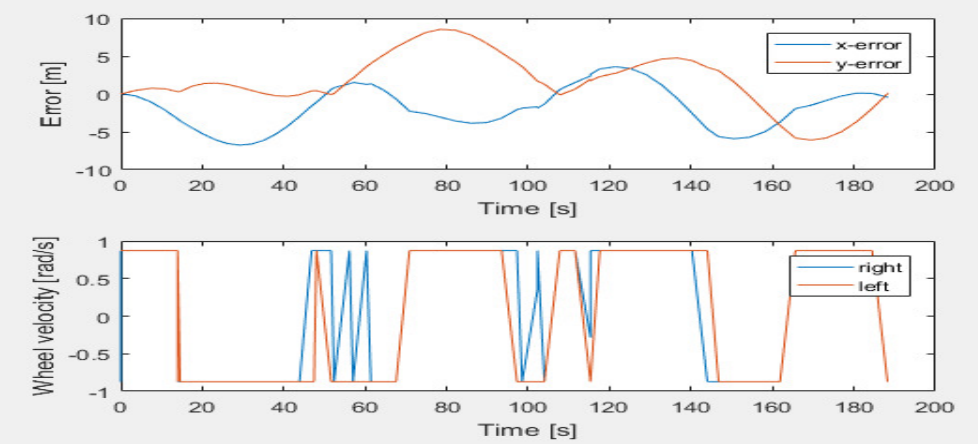
Therefore, W is a lyapunov function as

The lyapunov function has been implemented on Simulink too. The nature of the graph can be observed from the scope.



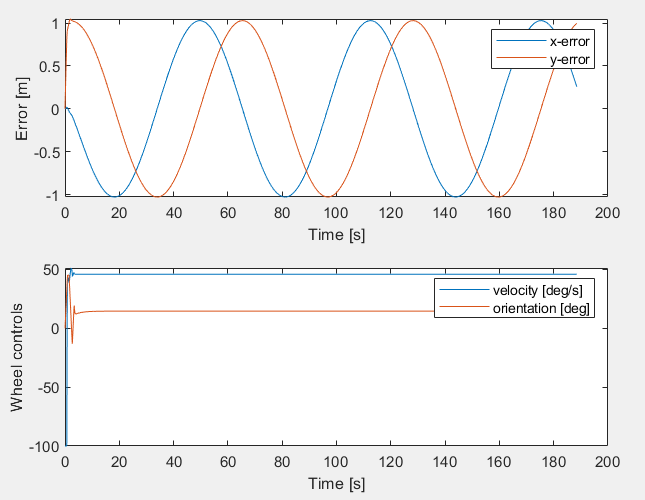
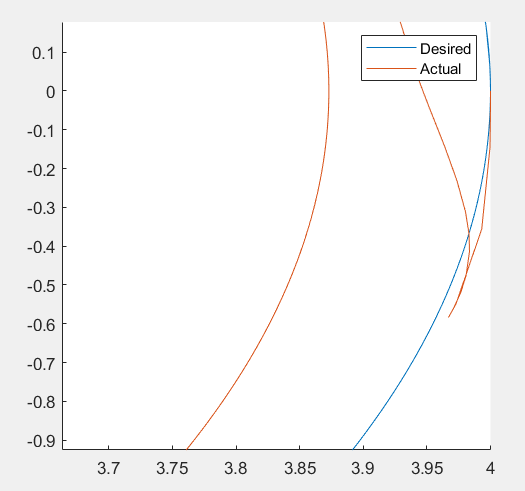
Kx = 15 , Ky = 15 and k theta = 2

Kx is effects the correction along y direction along robot frame and Ky effects the horizontal velocity k thete effects the ability of the robot to turn along the trajectory



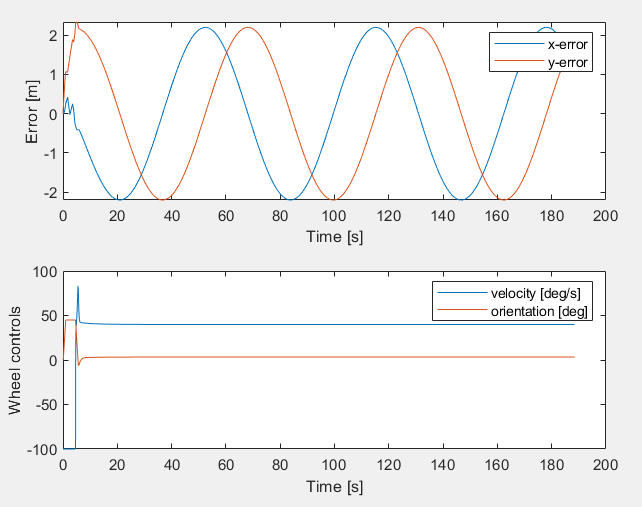
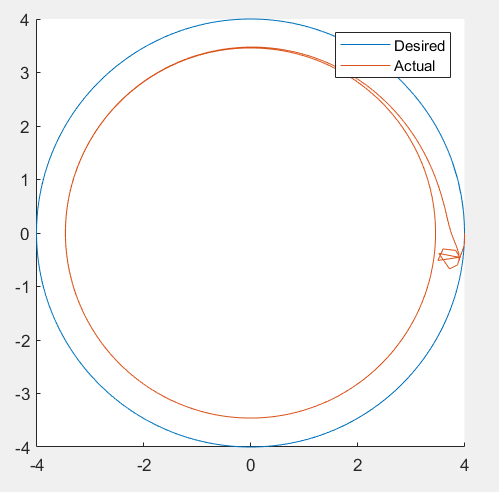
This is the condition for which velocity saturation happens and with this we will not able to follow the trajectory and in the above graph we can see that left and right velocities abruptly

**(1,1)Robot**



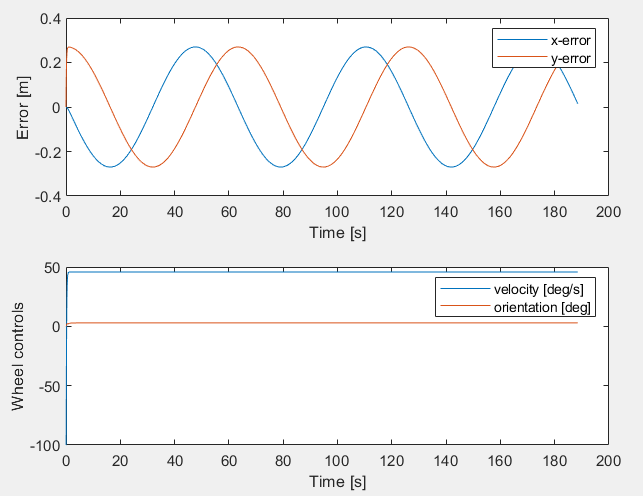
d = 0.03 Kp = 10 with L = 1

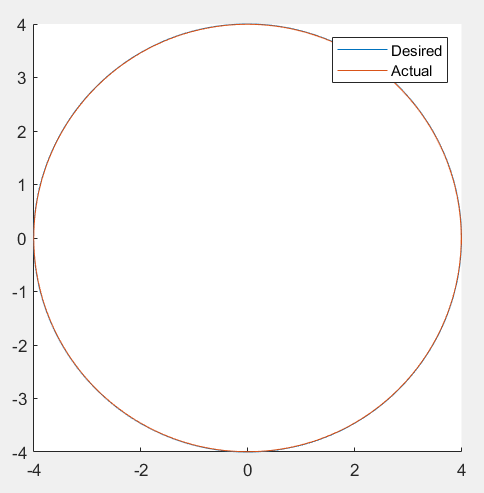
These are the results for proper tuning parameters however we see that there is always an error of 1 meter this because the control poin xp is calculated from front streable wheel which is at a distance of L (1m) from robot frame.it is also observed that the robot initially tries to go in opposite direction as the control point is ahead of distance L(1m)



d = 2 Kp = 7 with L = 0.2

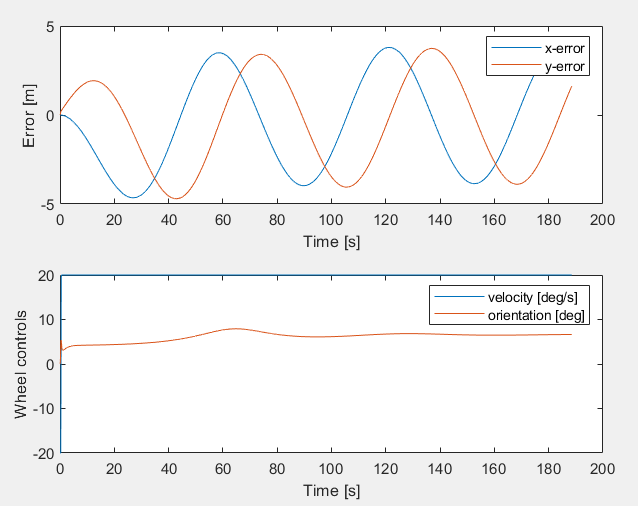
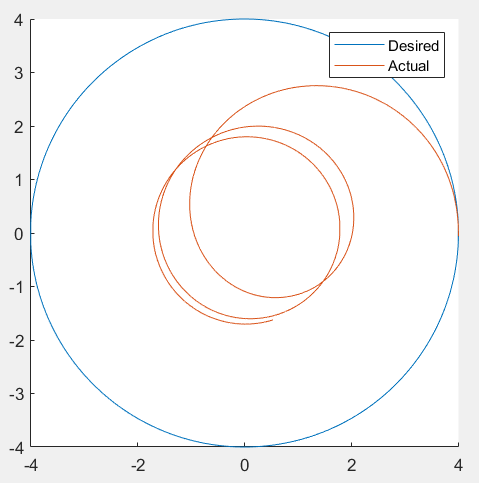
As L is very small and d is very far away So at the beginning to adjust the orientation of robot it will slightly reverse back as we can observer the robot is rotating in one point





Ideal Kp = 7 d = 0.07 L = 0.2

This is for ideal conditions ,Here we see that control is correct in the error so we get good trajectory



Rotation velocity = 20

This is similar to wheel saturation of 2-0. In this case the value of d is very high therefore the control point xp is far away from the robot frame and all control signals are generated from this point hence we see errors in the trajectory

**Lyapunov based control**

The classical function for (2,0) robot is given by: -

Here is the posture error in robot frame.

We know that, where depends on the choice of W and depends on the control input .

For lyapunov control, it’s necessary to have as negative definite, i.e. , must be <0 so that the error converges to 0.

Change of error with respect to time is: -

[using the model]

=

[using the model]

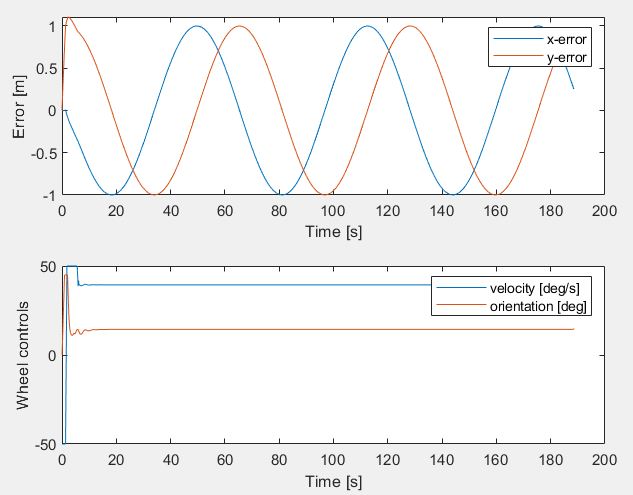
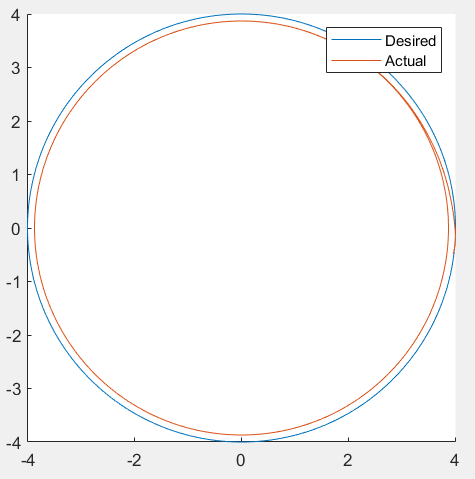
<0

<0

For worst case, when max orientation, ,

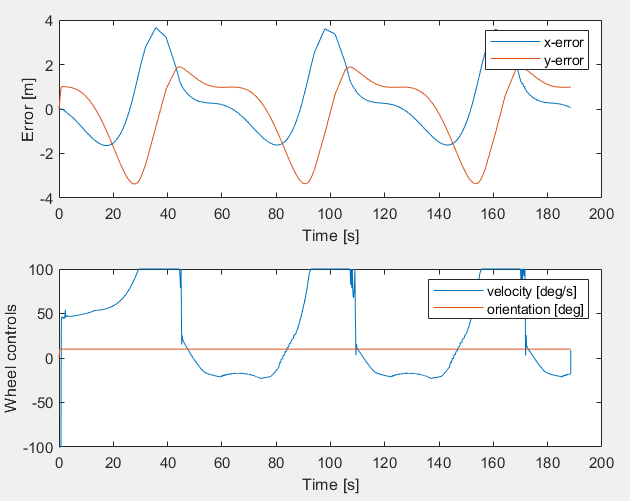
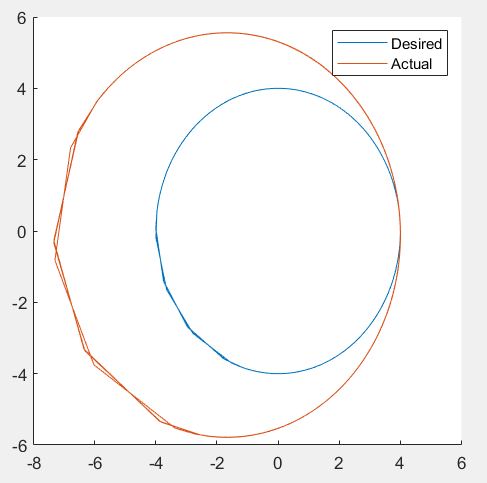
Therefore, W is a lyapunov function as

The lyapunov function has been implemented on Simulink too. The nature of the graph can be observed from the scope.



Kx = 30 , Ky = 30 and k theta = 1

Kx is effects the correction along y direction along robot frame and Ky effects the horizontal velocity k thete effects the ability of the robot to turn along the trajectory



This is the case with the orientation saturation.this will not follow the trajectory as the wheel is not able to turn enough for the curvature