ME 425 HW 1

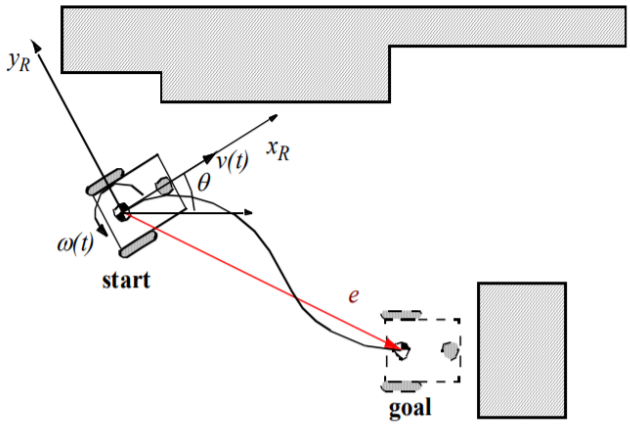
Demir Demirel 20586

***Abstract*— This report include implementation of kinematic position control and motion control of differential robot in a nutshell. Implementations were done by Simulink, MATLAB. Designed system consist of both Simulink blocks and MATLAB functions. The aim of this assignment is similar with the problem which known as parking of two wheeled robot in literature. Most of the equations referred by the course book.**

I. INTRODUCTION

In this assignment we are expected to move our virtual robot to the goal point which is G(0,0) in the Cartesian plane. We will assign some points around the G point. The problem is a format of car parking problem and we need our error e $\lim_{t\to\infty} e(t)$. For that purpose we will asked to design MIMO state feedback control system.

Fig. 1. Parking Problem



II. PROCEDURE

As I said before the goal represented as cartesian plane however kinematic position control is done by the polar coordinates. Hence we need to determine the values of the ρ, α, β in terms of x, y, Θ. The following formulas gives us the ρ, α, β .

$$\rho = \sqrt{\Delta x^2 + \Delta y^2}$$

$$\alpha = -\Theta + atan2(\Delta y, \Delta x) \tag{1}$$

$$\beta = -\Theta - \alpha$$

After finding the values for initial conditions of the ρ, α, β , we need to determine , , in terms of v and ω. For α is between $(-\frac{\pi}{2}, \frac{\pi}{2}]$ we need to apply following equations.

$$\begin{bmatrix} \dot{\rho} \\ \dot{\alpha} \\ \dot{\beta} \end{bmatrix} = \begin{bmatrix} -cos\alpha & 0 \\ \frac{sin\alpha}{\rho} & -1 \\ -\frac{sin\alpha}{\rho} & 0 \end{bmatrix} \begin{bmatrix} v \\ \omega \end{bmatrix} \tag{2}$$

For α is between $(-\pi, \frac{\pi}{2}] \cup (\frac{\pi}{2}, \pi]$ we need to apply following equations.

$$\begin{bmatrix} \dot{\rho} \\ \dot{\alpha} \\ \dot{\beta} \end{bmatrix} = \begin{bmatrix} cos\alpha & 0 \\ -\frac{sin\alpha}{\rho} & -1 \\ \frac{sin\alpha}{\rho} & 0 \end{bmatrix} \begin{bmatrix} v \\ \omega \end{bmatrix} \tag{3}$$

In 1^{st} and 4^{th} quadrants we expect robot to move forward direction. On the other hand for 2^{nd} and 3^{rd} quadrant we expect to move in bacward direction. Since moving backward is not a small problem in some of my systems, controlling is not as easy as I expect. Our system would not work only that block since we haven't design a controller for that purpose. Control law for kinematic position control with local stability could be determined with the equations below.

$$v = k_\rho \rho \quad \omega = k_\alpha \alpha + k_\beta \beta \tag{4}$$

$k_\rho, k_\alpha, k_\beta$ values should satisfy the rules listed below:

- $k_\rho > 0$
- $k_\beta < 0$
- $k_\alpha > k_\rho$

And referred book advise values as $k_\rho = 3, k_\alpha = 8, k_\beta = -1.5$.

In this lab we were required to construct a two wheeled robot. The aim of this group project was to observe and gather information on the motion of the robot which we provided after implementing our MATLAB code to gain access to the robots motor control. We were required to drive, observe and plot the received information about how robots linear and circular motion changed due to different speeds of each motor, with the help of a sensor mounted under the constructed vehicle. Rest of the report we will be explaining the construction, electrical connections and the programming of the robot and how did the change in speed affected our desired trajectory.

III. RESULT

From the given formulas I obtain the Simulink model in Figure:8 and try the system for different cases. First attempts were not as good as I expect. The robot couldn't move with the shortest or optimal trajectory. The turnings were not sharp as it can seen in Figure :2. I try to see trajectories from $0, \pi/2, \pi, -\pi/2, -\pi$. We can see that the robot couldn't make backward movement. The problem is based on the error that we have. Since in first Simulink model of mine has limited

control on the α and β this graph look very normal in that condition.

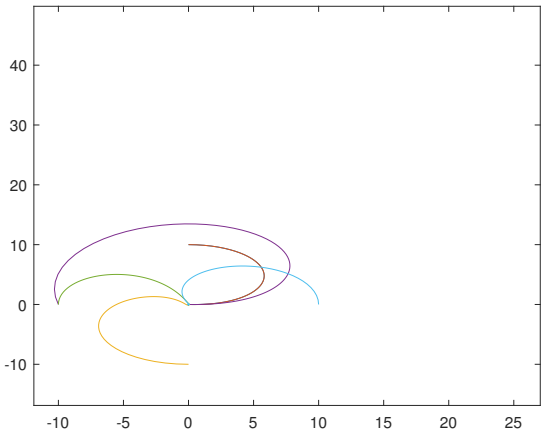


Fig. 2. First Attempt

With some improvements in the model I get the result as seen in Figure: 3 . This figure represent the movement only in the 1^{st} quadrant and the trajectory of the robot seems very well. The robot did not diverge any other way. When

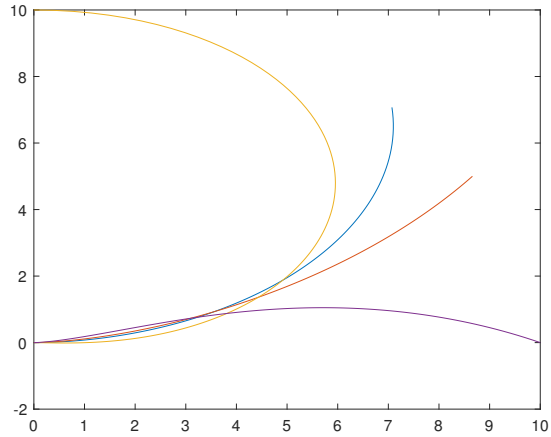


Fig. 3. First Quadrant

I try the same model into 1^{st} and 4^{th} it would also perform well as could be seen in Figure:4. But apply it to the 2^{nd} and 3^{rd} quadrant wont satisfy the expectations since robot follow circular trajectory until it gets 1^{st} quadrant. So it makes me push to review model one more time.

For the further graphs the finalize version of the Simulink model is represented. For the gains I apply $k_\rho = 3, k_\alpha = 8, k_\beta = -2$ as controller gains.

For the positions at (10,0) and (-10,0) robot tend follow bigger circular trajectory. Even if I change the controller gains it would not move directly to the towards. But the other trajectories from 1^{st} and 4^{th} quadrant are satisfactory.

When I try with different orientations, from each 4 quadrants

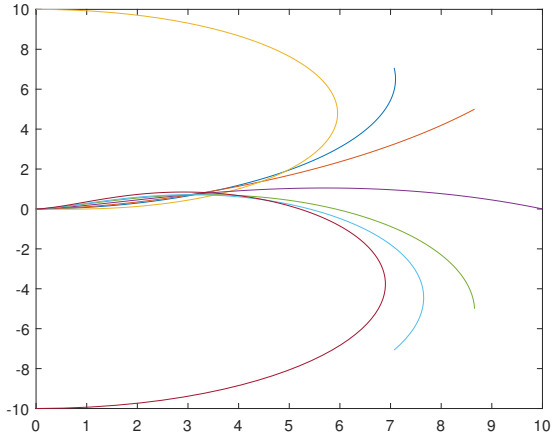


Fig. 4. 1^{st} and 4^{th} Quadrants

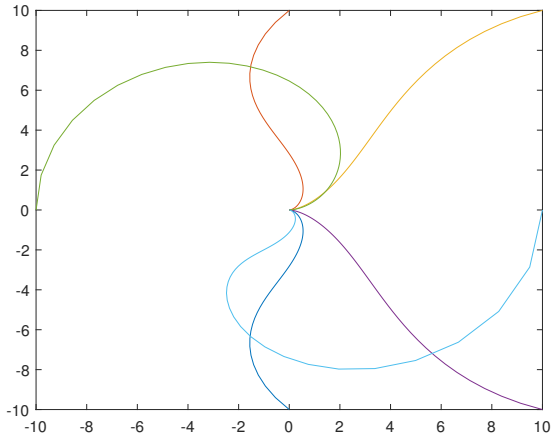


Fig. 5. Finalize Model Results

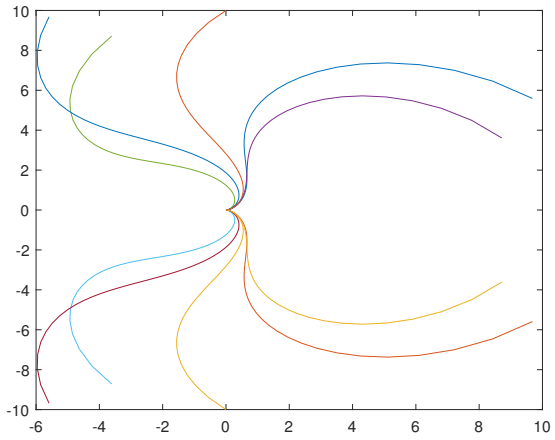


Fig. 6. Finalize Model Results

even the backward movement satisfy the criterion. But the movements from the points lay on the X-axis are not satisfy

expectations of mine. We can observe the data of ρ, α, β in

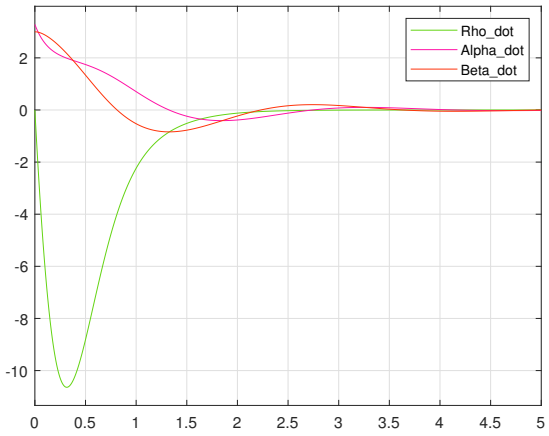


Fig. 7. ρ, α, β

the Fig:7. And it is obvious that ρ diverge to 0 as we expect. For the α and β I observe undershoot for both. After a while it also diverge to 0 but before that it travel around -1.

A. CONCLUSION

As a result the robot decide the possible trajectory with the given equation of motion and applied control law. With sophisticated control methods it would give much more accurate results. For instance the robot could approach to the target first, after coming closer the angular velocity would get into the game. There is also a trick could be apply like when finding the angle of α we can check it with atan2($sin(\theta), cos(\theta)$) to dismiss the unexpected errors. I couldn't succeed the backward movement, I try some techniques for that purpose but at the end of the day it wouldn't perform well.

APPENDIX

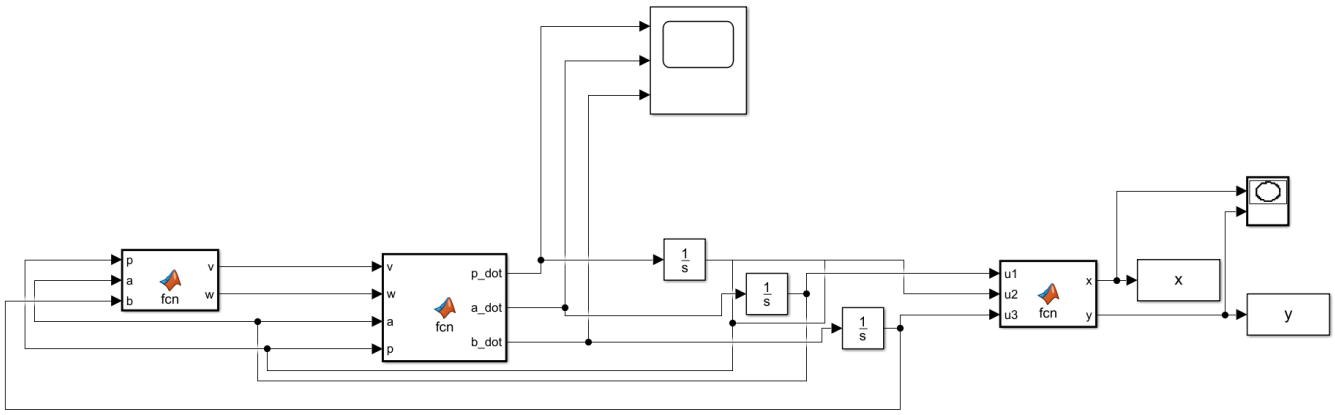


Fig. 8. Simulink Model

Listing 1. Controller

```
function [v,w]= fcn(p,a,b)
v= p* 3;
w=(a*8) + (b*(-2));
end
```

Listing 2. Kinematic Model

```
function [p_dot , a_dot, b_dot]= fcn(v,w,a,p)
if -pi/2<a && a≤pi/2
    p_dot = - cos(a) * v;
    a_dot =  v*(sin(a)/p) - w;
    b_dot = (-sin(a)/p)*v ;

else
    p_dot = - cos(a) * v;
    a_dot =  (-sin(a)/p)*v - w;
    b_dot = v*(sin(a)/p) ;
end
end
```

Listing 3. Initialization file

```
clear all;
x1=10;
y1=0;
kp=3;
theta1=0;

rho0=sqrt(x1^2+y1^2);
alpha0= -theta1 + atan2(y1,x1);

if -pi> alpha0
    alpha0=alpha0+pi;
end
if pi≤alpha0
    alpha0=alpha0+2*pi;
end
beta0 = -theta1 -alpha0;

sim('simulink_file_name.slx');
plot(x,y)
hold on
```