

ECE 141 Homework 4

Lawrence Liu

June 2, 2022

Problem 1

We have that

$$\beta = \arctan\left(\frac{l_r}{l_r + l_f} \tan(u)\right)$$

therefore

$$\tan(\beta) = \frac{l_r}{l_r + l_f} \tan(u)$$

therefore since the range of \tan is $-\infty$ to ∞ for any β we can find a u that satisfies the equation.

Problem 2

We have that

$$\frac{d}{dt}y = v \sin(\psi + \beta)$$

$$\frac{d}{dt}\psi = \frac{v}{l_R} \sin(\beta)$$

$$\beta = \arctan\left(\frac{l_r}{l_r + l_f} \tan(u)\right)$$

Linearizing around $\psi = 0$ $\beta = 0$, we have

$$\frac{d}{dt}y = v(\psi + \beta)$$

$$\frac{d}{dt}\psi = \frac{v}{l_R}\beta$$

therefore taking the laplace transform we have

$$sY = v(\psi + \beta)$$

$$s\psi = \frac{v}{l_r}\beta$$

Therefore we get

$$sY = v\left(\frac{v}{l_r s} + 1\right)\beta$$

Therefore the transfer function is

$$\frac{Y(s)}{\beta} = \frac{v(v + l_r s)}{l_r s^2}$$

So now with a controller $D_c(s)$ and unity feedback we have that the transfer function is

$$\frac{Y}{R} = \frac{D_c(s) \frac{v(v+l_r s)}{l_r s^2}}{1 + D_c(s) \frac{v(v+l_r s)}{l_r s^2}}$$

If the controller is a simple portional controller then the transfer function becomes

$$\frac{Y}{R} = \frac{k_p v(v + l_r s)}{l_r s^2 + k_p v(v + l_r s)}$$

Therefore the transfer function for the error is

$$\frac{E}{R} = 1 - \frac{k_p v(v + l_r s)}{l_r s^2 + k_p v(v + l_r s)}$$

$$\frac{E}{R} = \frac{l_r s^2}{l_r s^2 + k_p v(v + l_r s)}$$

Therefore for unit step input, which is what we will have, we have that the steady state error is

$$\lim_{t \rightarrow \infty} e(\infty) = \lim_{s \rightarrow 0} \frac{l_r s^2}{l_r s^2 + k_p v (v + l_r s)}$$

$$\lim_{t \rightarrow \infty} e(\infty) = 0$$

Therefore we must have characteristic polynomial

$$l_r s^2 + k_p v (v + l_r s) = 0$$

$$s^2 + k_p v s + \frac{k_p v^2}{l_r} = 0$$

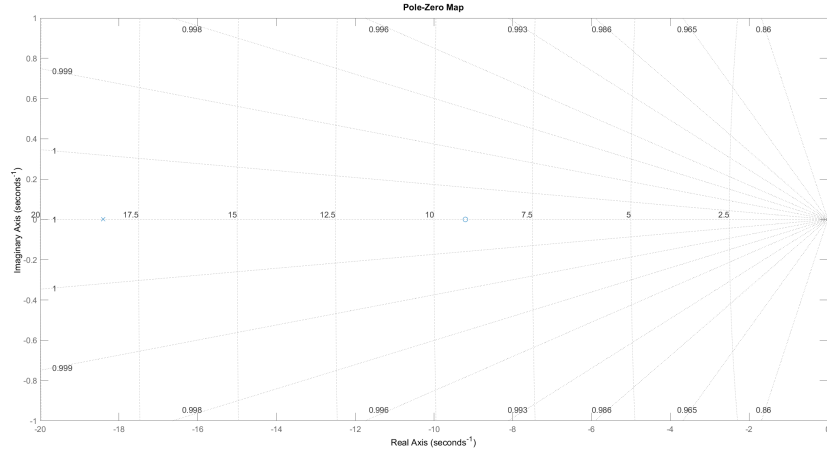
In order for this system to be critically damped we must have that

$$2v \sqrt{\frac{k_p}{l_r}} = k_p v$$

therefore we have that $\sqrt{k_p} = \frac{2}{\sqrt{l_r}}$, and thus $k_p = \frac{4}{l_r}$. To confirm that this is indeed stable, and that the system is critically damped, we plot out the poles and zeros of the system. Using the following matlab code, we get the pole zero plots

```
v=15.6464;
l_r=1.7;
k_p=4/l_r;

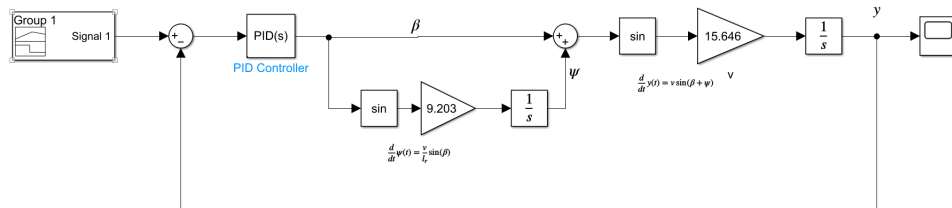
sys = tf([k_p*v*l_r k_p*v^2],[l_r k_p*v*l_r k_p*v^2]);
h = pzplot(sys);
grid on
```



As we can see, the system is critically damped since the poles are at the same place.

Problem 3

The blockdiagram for the nonlinear system looks like the following



We can modify the initial conditions by modifying the initial conditions for the integrator blocks. We define acceptable behaviour such that the absolute value of y is less than 0.1, ie the car, if oscillating, cannot have a amplitude greater than 0.1 or 10cm which would be barely noticable.

Using the following matlab code we can plot out where the initial values of y and ψ result in acceptable behaviour

```

psi_sweep=-pi/2:0.1:pi/2;
y_sweep=-3:0.1:-1;
t_final=40;
valid = zeros(length(psi_sweep),length(y_sweep));
for i=1:length(psi_sweep)
    psi_initial=psi_sweep(i);
    i
    psi_initial
    for j=1:length(y_sweep)
        y_initial=10^(y_sweep(j));
        a=sim("Problem3.slx");

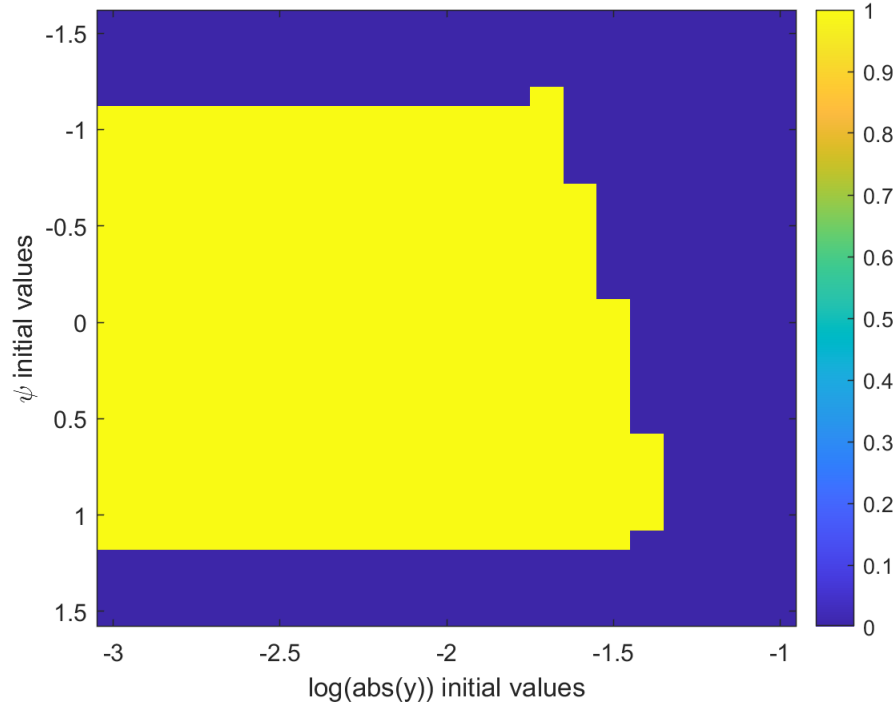
        data=a.yout.getElement("y");
        if max(abs(data.Values.Data))<=0.1
            valid(i,j)=1;
        end
    end
end

end

imagesc(y_sweep,psi_sweep,valid)
colorbar
ylabel("\psi initial values")
xlabel("log(abs(y)) initial values")

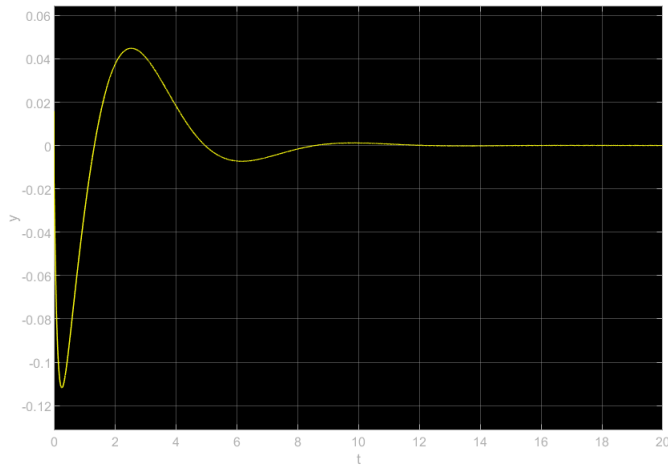
```

which outputs the following plot



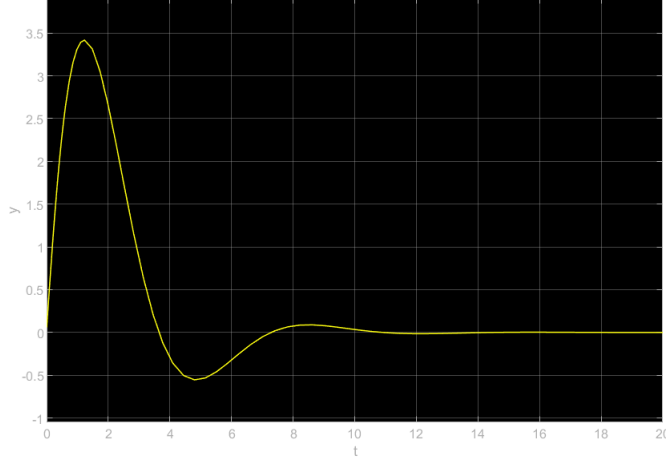
Where the yellow block is the acceptable region, and the blue blocks are the unacceptable regions.

To check this, let us plot out the behaviour of the system for the initial conditions $y(0) = 10^{-1.8} = 0.0158$ and $\psi(0) = -1.2831$



As we can see, for these initial conditions, the maximum deviation is greater

than -0.1 so it is not acceptable performance.



As we can see for these initial conditions, the maximum deviation is around 3.5 so the controller has inadequate performance for these initial conditions. Both of these discovers agree with the plot of initial conditions that result in adequate performance that we generated

Problem 4

We have

$$\begin{aligned}\frac{d}{dt}v &= a \\ sv &= a \\ \frac{v}{a} &= \frac{1}{s}\end{aligned}$$

therefore with a pid controller the transfer function from the input target velocity to the output velocity is

$$\frac{(k_p + k_d s + k_i \frac{1}{s}) \frac{1}{s}}{1 + (k_p + k_d s + k_i \frac{1}{s}) \frac{1}{s}}$$

Therefore the characteristic polyomial is

$$s^2 + (k_p s + k_d s^2 + k_i) = 0$$

, Setting $k_d = 0$ for now, we focus on making this system critically damped as well, since we would want the car to match velocity as quickly as possible, but not for it to oscillate then we have

$$s^2 + k_p s + k_i = 0$$

in order for this system to be critically damped, we want the

therefore picking $k_p = k_i = k_d = 1$ we get the following routh array from this matlab code

```
syms EPS;
k_d=1;
k_p=1;
k_i=1;
ra=routh([1+k_d k_p k_i],EPS)
```

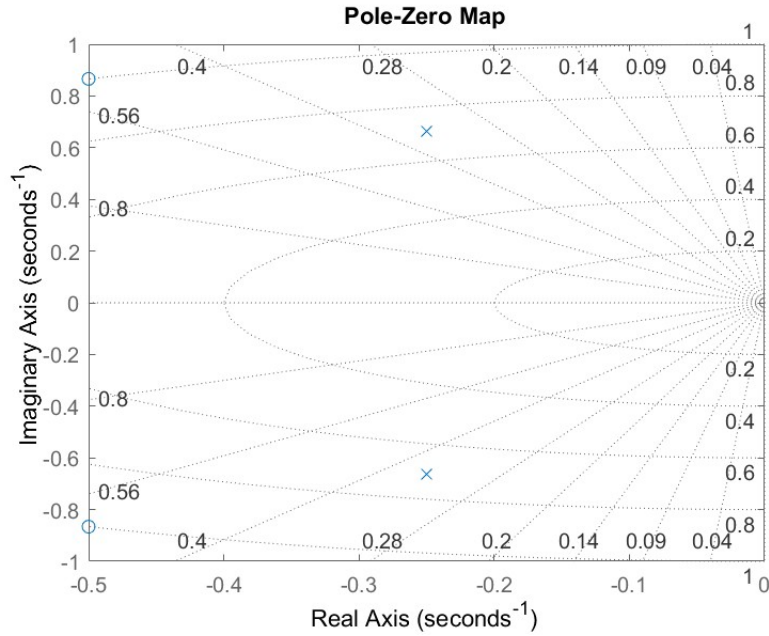
```

2  1
1  0
1  0
```

Therefore this controller stabilizes the system, to double check let us use the following matlab code to plot the poles and zeros of the transfer function

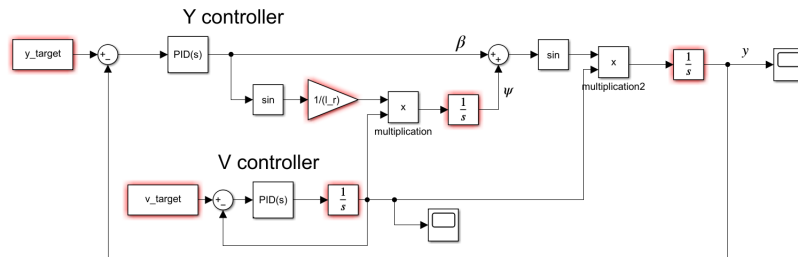
```
k_d=1;
k_p=1;
k_i=1;
sys = tf([k_d k_p k_i], [1+k_d k_p k_i]);
h = pzplot(sys);
grid on
```

which outputs the following pole zero plot



Problem 5

The blockdiagram for the nonlinear system looks like the following



Using the following code we plot out the initial values of y and ψ result in acceptable behaviour, when v initially is $10m/s$, $15.6464m/s$, and $20m/s$

```

        v_target=15.6464;
y_target=0;
l_r=1.7;

v_sweep=[0 15.6464 20];
log_y_sweep=-3:0.1:-1;
psi_sweep=0:0.25:2*pi;

for i=1:length(v_sweep)
v_initial=v_sweep(i)
f = figure;
valid = zeros(length(psi_sweep),length(log_y_sweep));
for j=1:length(psi_sweep)
    psi_initial=psi_sweep(j);
    for k=1:length(log_y_sweep)
        y_initial=10^(log_y_sweep(k));

        a=sim("Problem3.slx");

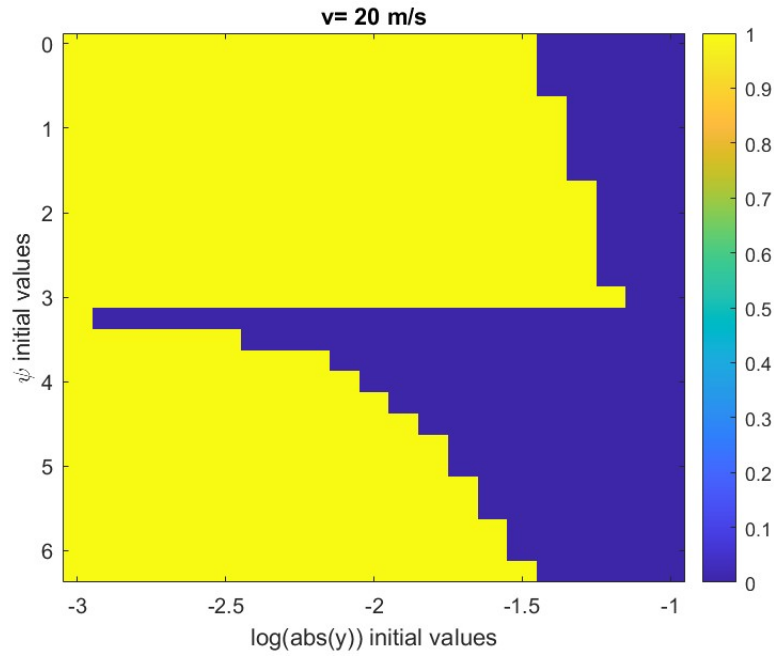
        data=a.yout.getElement("y");

        if max(abs(data.Values.Data))<0.6
            valid(j,k)=1;
        end
    end
end

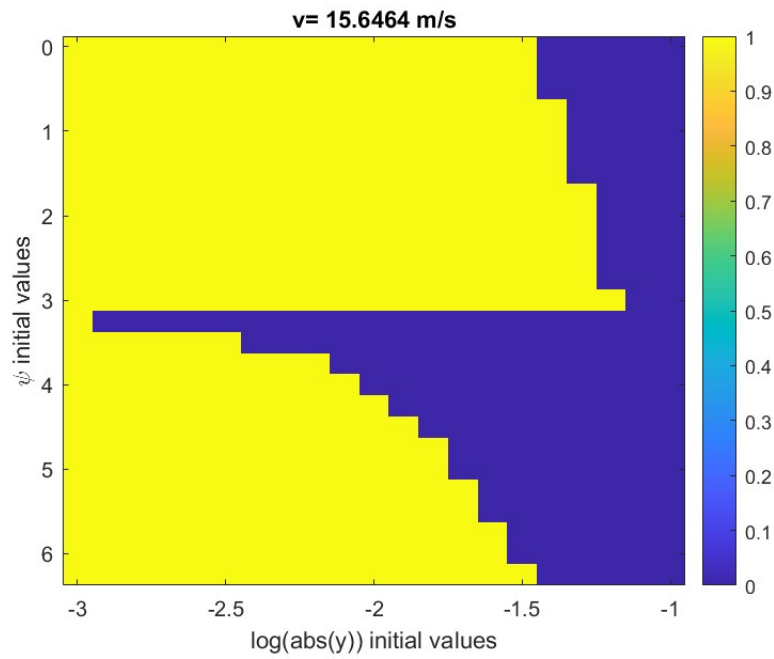
end
imagesc(log_y_sweep,psi_sweep,valid);
colorbar
ylabel("\psi initial values")
xlabel("log(abs(y)) initial values")
title(['v= ',num2str(v_initial),' m/s'])
end

```

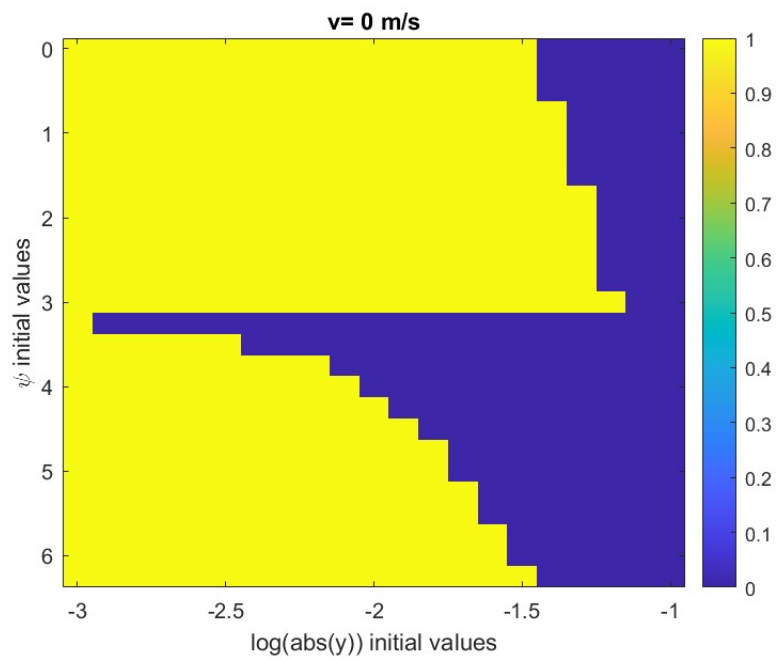
which results in the following plots, for $v = 20\text{m/s}$



for $v = 15.6464 m/s$



and for $v = 0 m/s$



As we can see changing the initial v does not seem to matter at all

Problem 6