Systems Final

Walter Livingston, Cars Chandler, Will Thomas Sheffield, Matthew Covington

Physical Constants

```
clc, clear all, close all

dt = 0.01 ;
tspan = 0:dt:5;
yaw_0 = 0;
IC = [0 0 0 yaw_0];

syms d(t)
d(t) = 0.5*sin(2*t);

a = 1.257; % m
b = 1.593; % m
Cf = 120000; % N/rad
Cr = 184600; % N/rad
Iz = 4292; % kg*m^2
Vcar = 30; % m/s
m = 1856; % kg
```

Part (a)

State Space Model

See derivation for SS in Appendix A at the end of the file

```
W = [((-Cf-Cr)/(m*Vcar)) (((b*Cr - a*Cf)/(m*Vcar^2)) - 1)
     ((b*Cr - a*Cf)/(Iz)) ((-a^2 * Cf - b^2 * Cr)/(Iz*Vcar))]
```

```
W = 2×2
   -5.4705   -0.9143
   33.3709   -5.1107
```

```
X = [((Cf)/(m*Vcar))
     ((a*Cf)/(Iz))]
```

```
X = 2×1
    2.1552
   35.1445
```

```
Y = [Vcar 0
     0 1
     ((-Cr-Cf)/(m)) ((b*Cr-a*Cf)/(m*Vcar))]
```

```
Y = 3×2
   30.0000         0
         0    1.0000
```

```
  -164.1164    2.5723
```

```
Z = [0
     0
     ((Cf)/(m))]
```

```
Z = 3×1
         0
         0
   64.6552
```

```
full_sys = ss(W,X,Y,Z)
```

```
full_sys =

  A =
             x1       x2
   x1   -5.471  -0.9143
   x2    33.37   -5.111

  B =
          u1
   x1  2.155
   x2  35.14

  C =
           x1      x2
   y1      30       0
   y2       0       1
   y3  -164.1   2.572

  D =
          u1
   y1      0
   y2      0
   y3  64.66

Continuous-time state-space model.
```

```
[num, denom] = ss2tf(full_sys(2).A,full_sys(2).B,full_sys(2).C,full_sys(2).D)
```

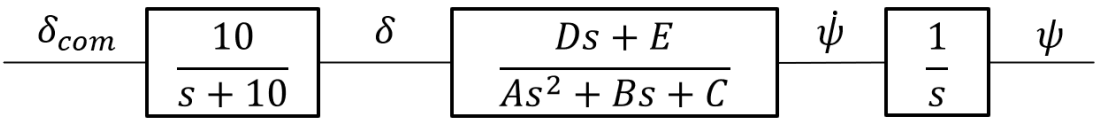
```
num = 1×3
         0   35.1445  264.1793
denom = 1×3
    1.0000   10.5813   58.4679
```

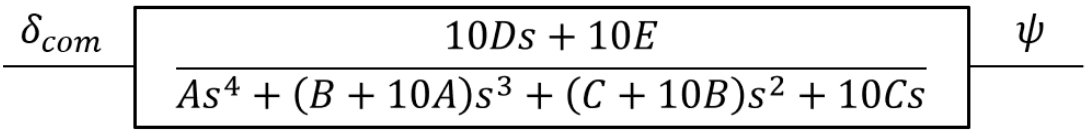
Coefficients A-E taken from State Space

```
A = denom(1); B = denom(2); C = denom(3); D = num(2); E = num(3);
```

Block Diagram

Accounting for the steering actuation dynamics as well as the fact that we are integrating the original yaw rate to yaw/heading, the block diagram for the system is as follows (see derivation for SS in Appendix):





Transfer functions for each block

Plant

```
plant = tf(num, denom)
```

```
plant =

     35.14 s + 264.2
  ---------------------
  s^2 + 10.58 s + 58.47

Continuous-time transfer function.
```

Steering Actuation Dynamics

```
del_sys = tf(10, [1 10])
```

```
del_sys =

    10
  ------
  s + 10

Continuous-time transfer function.
```

Pure Integrator to Integrate Yaw Rate to Yaw

```
yaw_sys = tf(1, [1 0])
```

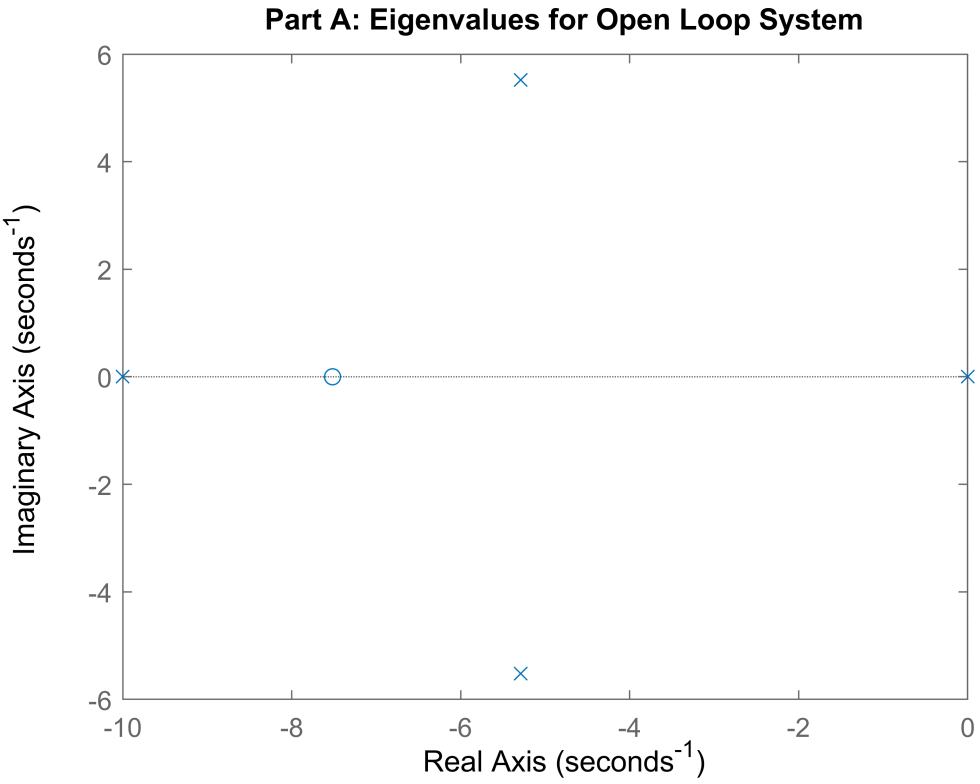
```
yaw_sys =

  1
  -
  s

Continuous-time transfer function.
```

Open Loop Transfer Functions

```
ol_yaw_rate = del_sys*plant;
ol_yaw = ol_yaw_rate*yaw_sys;
```

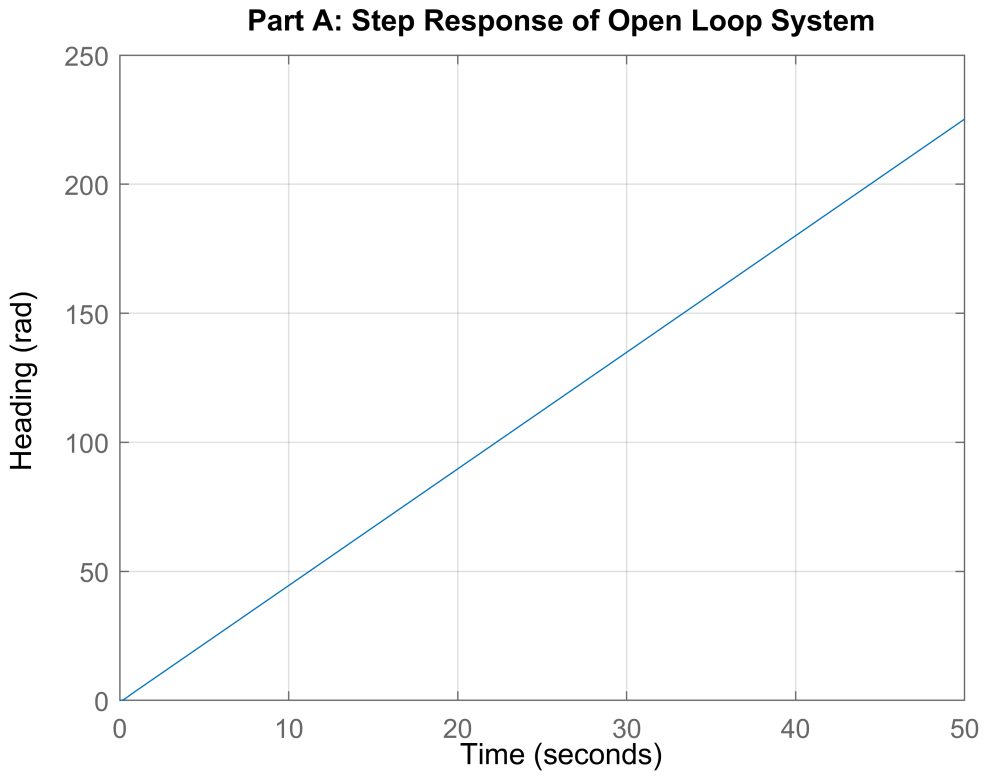
Plot Poles and Zeros

```
figure
pzplot(ol_yaw)
title('Part A: Eigenvalues for Open Loop System')
```



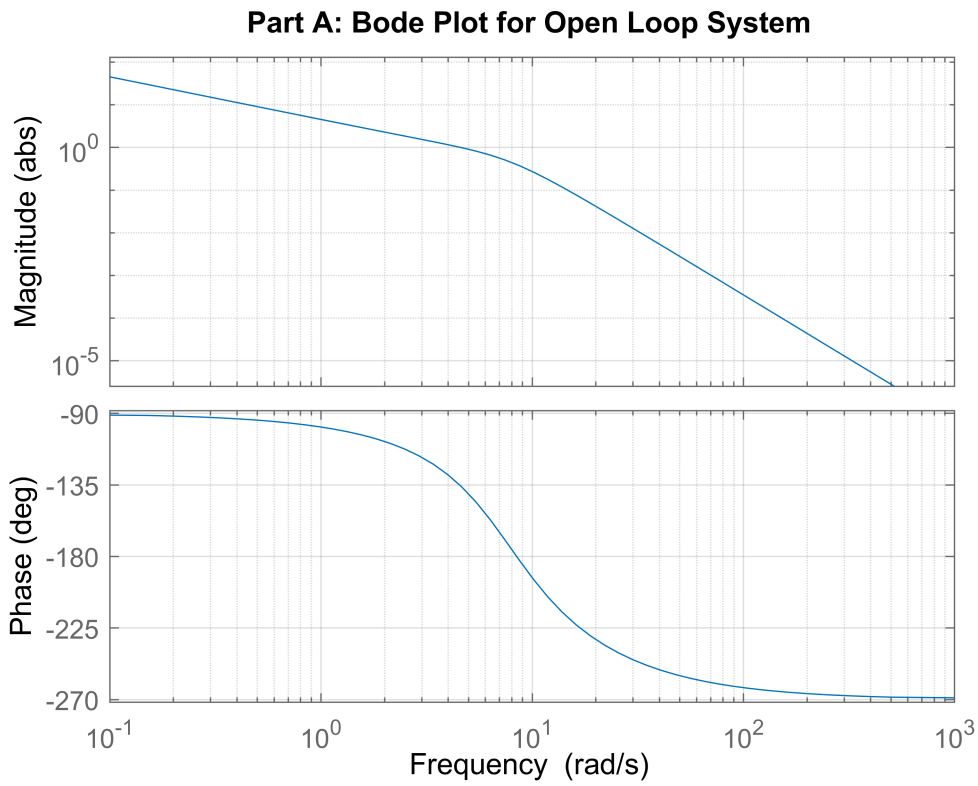
Plot Step Response

```
figure
step(ol_yaw)
grid on
title('Part A: Step Response of Open Loop System')
ylabel('Heading (rad)')
```



Plot Bode Diagram

```
figure
bode(ol_yaw)
grid on
title('Part A: Bode Plot for Open Loop System')
setoptions(gcr,'MagUnits','abs')
setoptions(gcr,'MagScale','log')
```

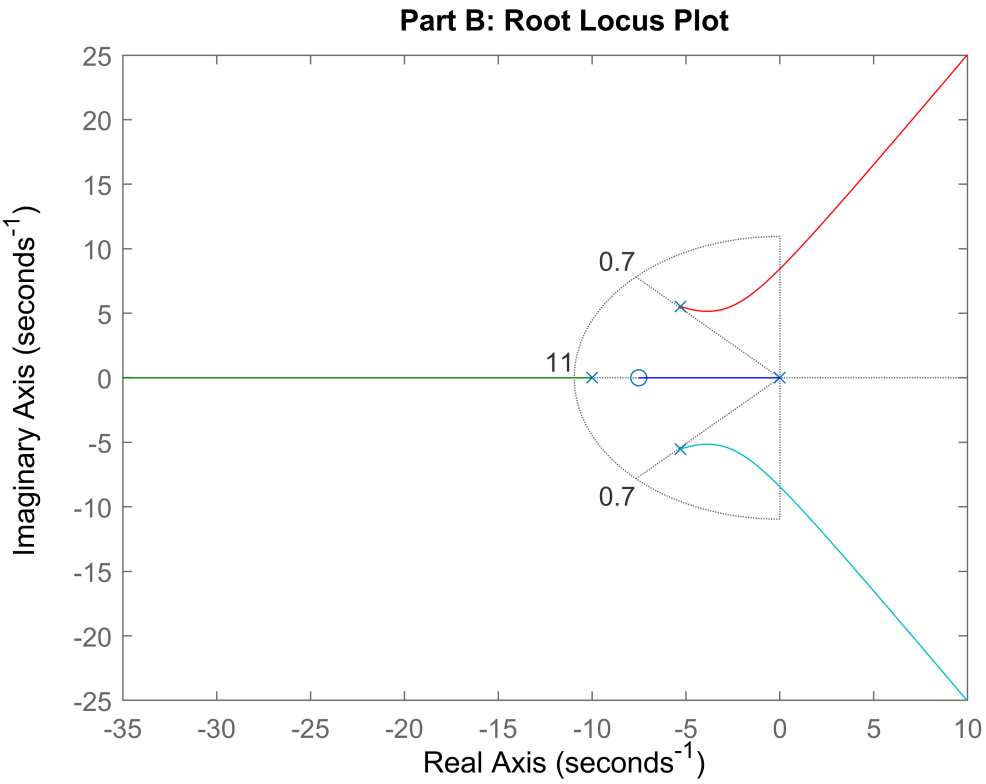


Part (b)

Root Locus Analysis

The `rlocus` and `sgrid` functions are used on the open loop system to determine how the poles and zeros should be adjusted. We know that we need to bring the poles to the left of the semicircle and within the cone formed by the two diagonal lines of consant zeta to achieve our system performance. We used this in tandem with the `controlSystemDesigner` tool to add two zeros and choose an appropriate compensator gain, which led us to our P-D-Double D controller.

```
figure
rlocus(ol_yaw)
sgrid(0.7,10.9524)
title('Part B: Root Locus Plot')
```



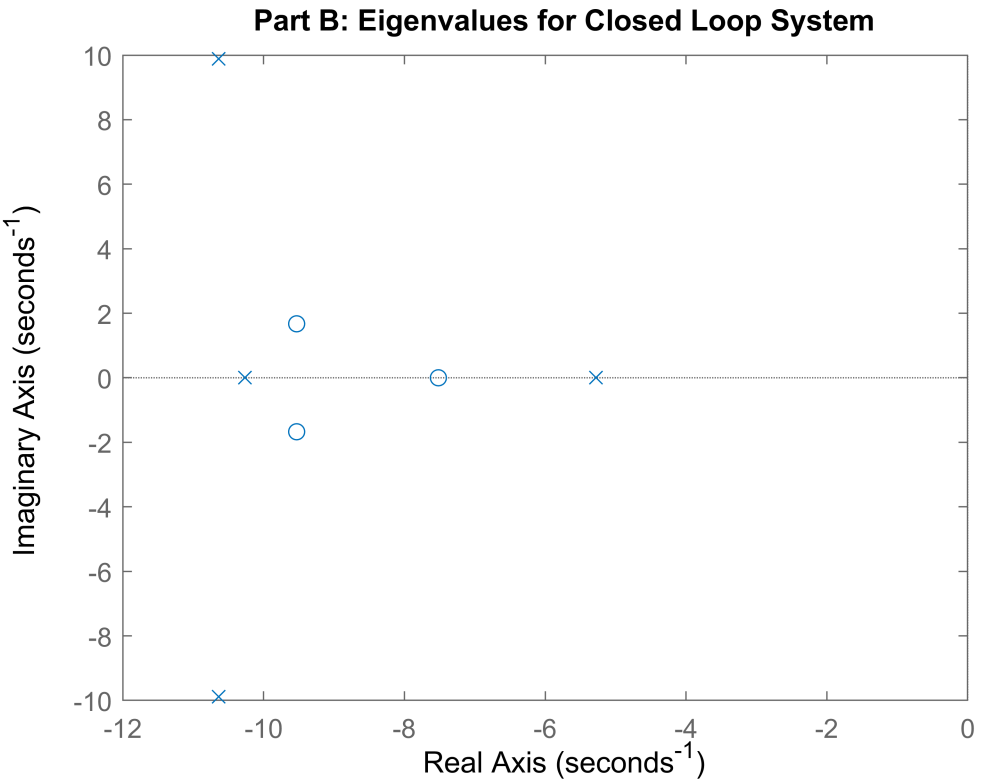

```
Kdd = 1;
Kd = 19.06;
Kp = 93.62;
k = 0.0462;
```

Determine Closed Loop System Equation

```
controller = tf([Kdd Kd Kp],1);
c_ol_sys = ol_yaw*controller;
c_cl_sys = feedback(k*c_ol_sys,1);
```

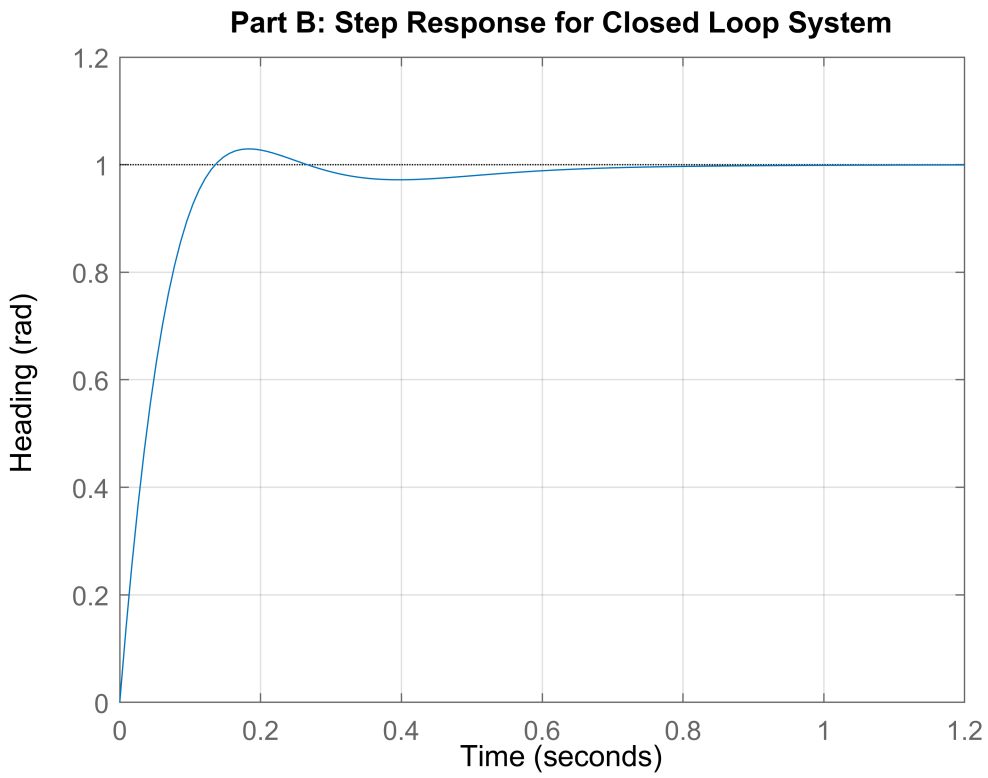
Plot Poles and Zeros

```
figure
pzplot(c_cl_sys)
title('Part B: Eigenvalues for Closed Loop System')
```



Plot Step Response

```
figure
step(c_cl_sys)
grid on
title('Part B: Step Response for Closed Loop System')
ylabel('Heading (rad)')
```

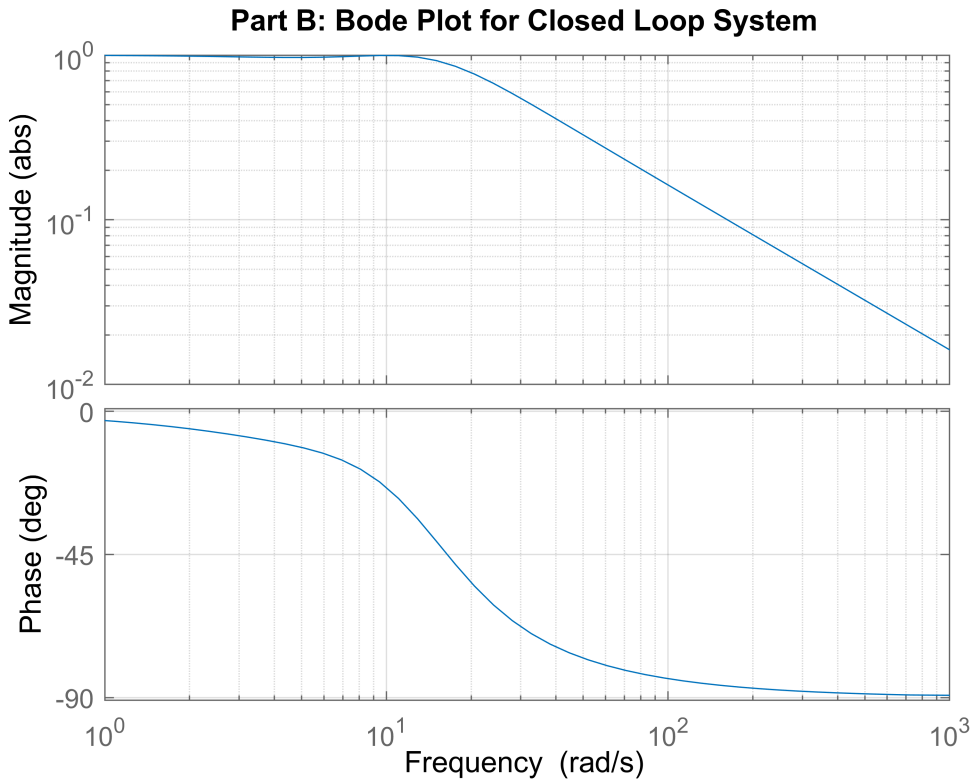



```
stepinfo(c_cl_sys)
```

```
ans = struct with fields:
         RiseTime: 0.0901
    TransientTime: 0.5058
     SettlingTime: 0.5058
      SettlingMin: 0.9266
      SettlingMax: 1.0294
        Overshoot: 2.9437
       Undershoot: 0
             Peak: 1.0294
         PeakTime: 0.1818
```

Plot Bode Diagram

```
figure
bode(c_cl_sys)
grid on
title('Part B: Bode Plot for Closed Loop System')
setoptions(gcr,'MagUnits','abs')
setoptions(gcr,'MagScale','log')
```



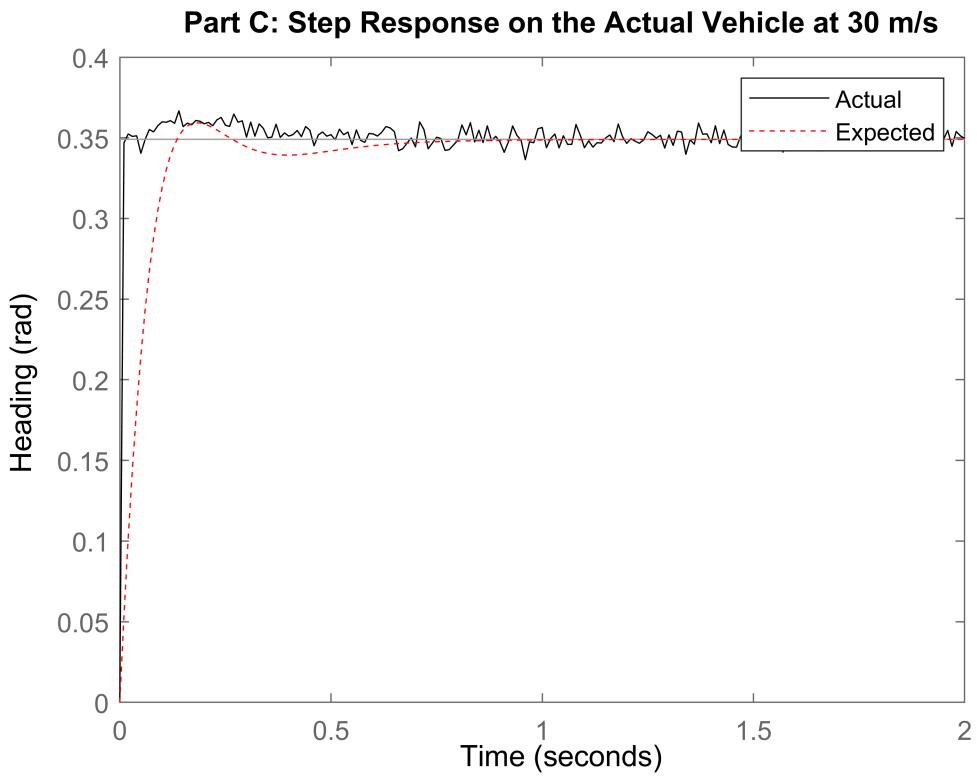
Part (c)

Apply Controller to Actual MKZ

```
tspan_c = 0:dt:2;
heading_des = zeros(1,length(tspan_c)) + 20*pi/180;
del_com(1) = 0;
for i=1:length(tspan_c)-1
    [GPS(i+1,:),yaw_gyro(i+1),del_meas(i+1),WP(i+1,:)]=run_MKZ_fp(del_com(i),Vcar,IC,0);
    e = heading_des(i) - GPS(i,3);
    e_dot = 0 - yaw_gyro(i);
    e_double_dot = 0 - ((yaw_gyro(i+1)-yaw_gyro(i))/dt);

    del_com(i+1) = (k*1)*e_double_dot + (k*19.06)*e_dot + (k*93.62)*e;
end
```

Plot Response of Actual Vehicle and Expected Response

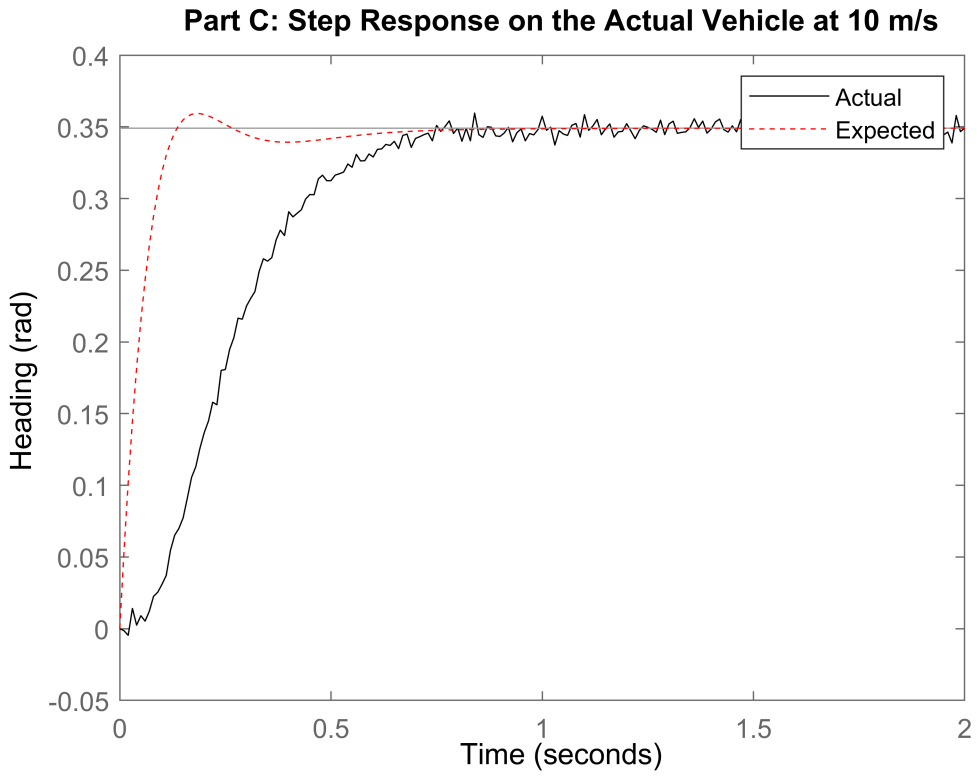
```
figure
plot(tspan_c, GPS(:,3),'k')
grid on, hold on
lsim(c_cl_sys, heading_des, tspan_c, 'r--')
title('Part C: Step Response on the Actual Vehicle at 30 m/s')
ylabel('Heading (rad)')
legend('Actual','Expected')
```




```
save tmp.mat
clear all
load tmp.mat
```

Plot Response of Actual Vehicle and Expected Response at 10 m/s

```
Vcar = 10; % m/s
heading_des = zeros(1,length(tspan_c)) + 20*pi/180;
del_com(1) = 0;
for i=1:length(tspan_c)-1
    [GPS(i+1,:),yaw_gyro(i+1),del_meas(i+1),WP(i+1,:)]=run_MKZ_fp(del_com(i),Vcar,IC,0);
    e = heading_des(i) - GPS(i,3);
    e_dot = 0 - yaw_gyro(i);
    e_double_dot = 0 - ((yaw_gyro(i+1)-yaw_gyro(i))/dt);

    del_com(i+1) = (k*1)*e_double_dot + (k*19.06)*e_dot + (k*93.62)*e;
end

figure
plot(tspan_c, GPS(:,3),'k')
grid on, hold on
lsim(c_cl_sys, heading_des, tspan_c, 'r--')
title('Part C: Step Response on the Actual Vehicle at 10 m/s')
ylabel('Heading (rad)')
legend('Actual','Expected')
```



The actual response takes longer to reach steady-state than the expected response does and appears more overdamped than the expected response does. The actual response obviously is much noiser than the expected response as well. This can be attributed to several factors: the input for the derivative control is a backward-looking approximation, there could be linearizations in the p-code, and the maximum steering angle may be reached using the controller.

The low-speed and high-speed responses are very similar, but the high-speed response tracks the expected response slightly better, which makes sense because our model was derived at a speed of 30 m/s.

The overshoot would decrease for a lower speed because the system does not have as much velocity/ momentum pushing it beyond the desired steering angle. The settle time would increase, however. At a high speed, the system responds quickly, but this also leads to overshoot. At a low speed, the system takes longer to respond, but this yields less overshoot.

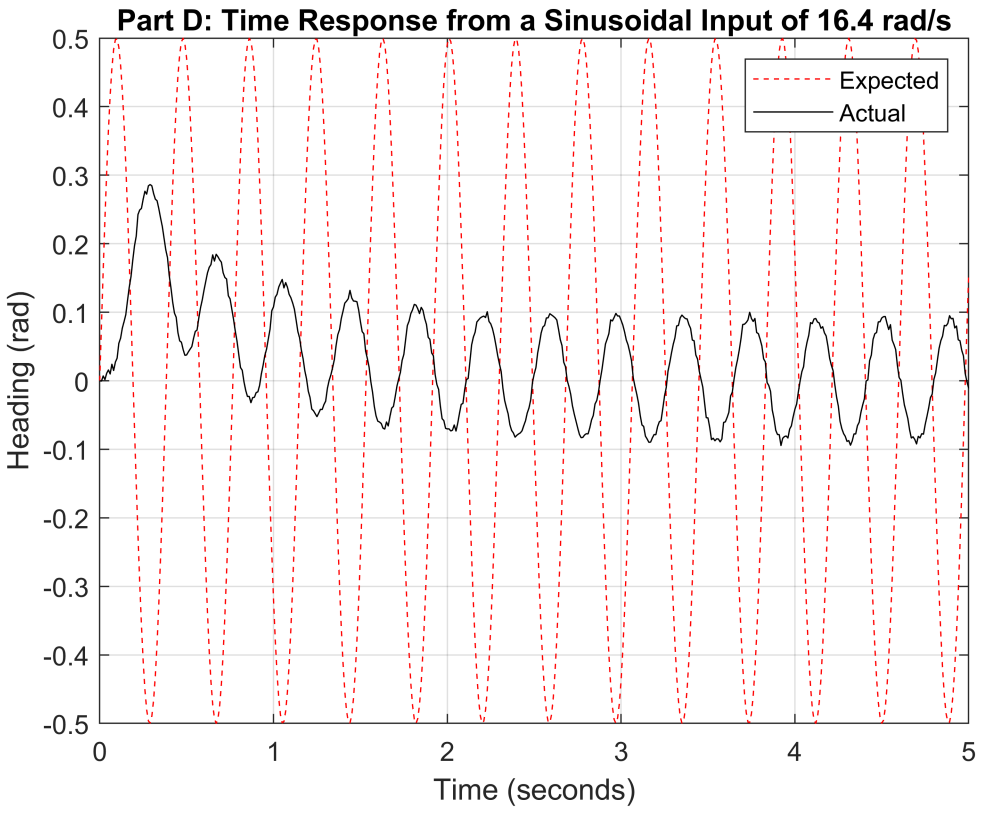
Part (d)

```
save tmp.mat
clear all
load tmp.mat
```

Evaluate Frequency Response of Control System on Actual MKZ at Various Frequencies

Response at 16.4 rad/s

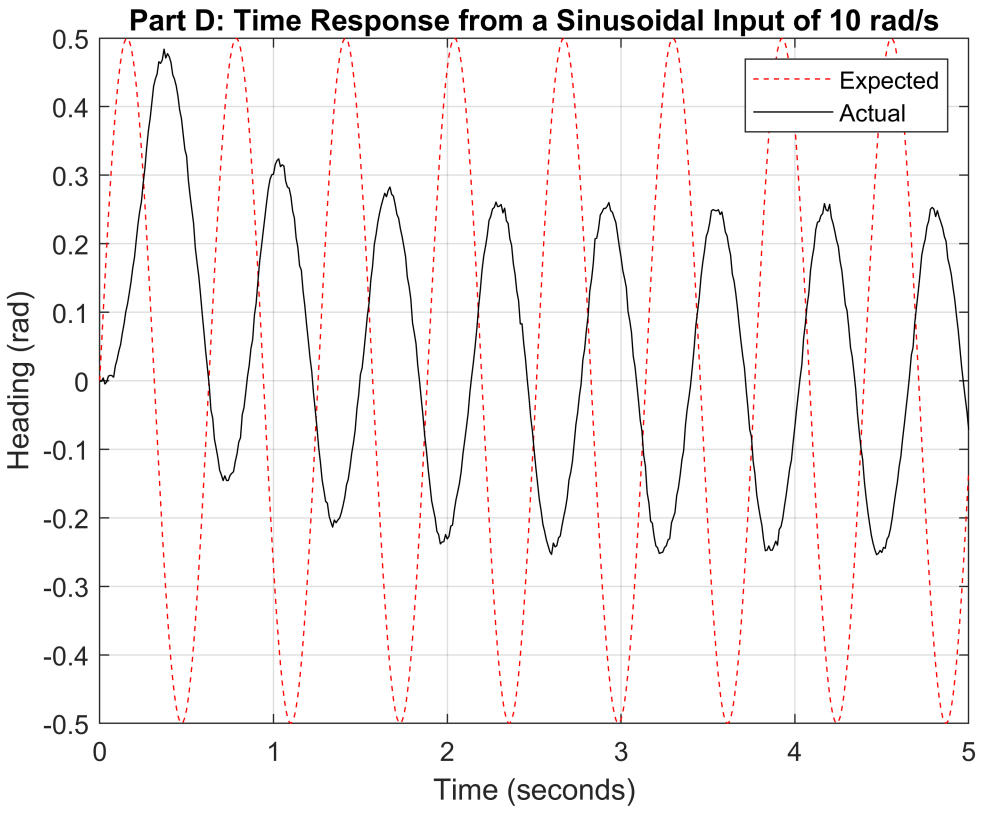
```
clear GPS
Vcar = 30; % m/s
del_com(1) = 0;
for i=1:length(tspan)-1
    [GPS(i+1,:),yaw_gyro(i+1),del_meas(i+1),WP(i+1,:)]=run_MKZ_fp(del_com(i),Vcar,IC,0);
    heading_des(i+1) = sin(16.4*tspan(i+1));
    e = heading_des(i) - GPS(i,3);
    e_dot = 0 - yaw_gyro(i);
    e_double_dot = 0 - ((yaw_gyro(i+1)-yaw_gyro(i))/dt);

    del_com(i+1) = (k*1)*e_double_dot + (k*19.06)*e_dot + (k*93.62)*e;
end

figure
plot(tspan, 0.5*sin(16.4*tspan),'r--', tspan, GPS(:,3),'k')
grid on
title('Part D: Time Response from a Sinusoidal Input of 16.4 rad/s')
xlabel('Time (seconds)')
ylabel('Heading (rad)')
legend('Expected','Actual')
```




```
save tmp.mat
clear all
load tmp.mat
```

Response at 10 rad/s

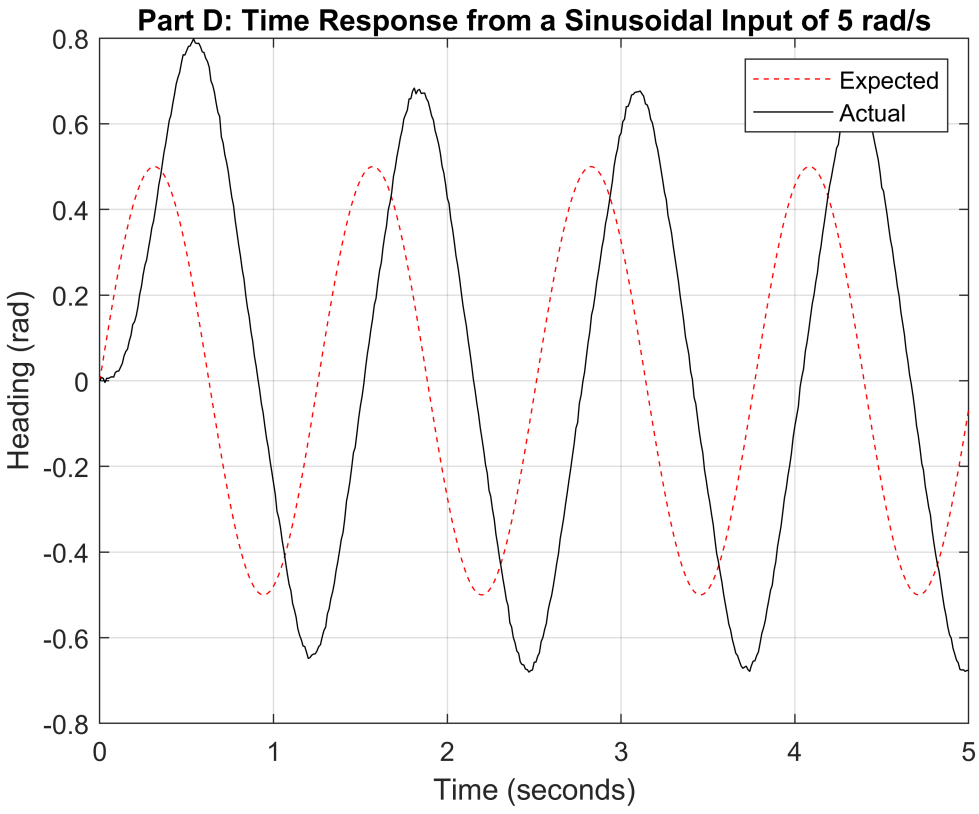
```
clear GPS
Vcar = 30; % m/s
del_com(1) = 0;
for i=1:length(tspan)-1
    [GPS(i+1,:),yaw_gyro(i+1),del_meas(i+1),WP(i+1,:)]=run_MKZ_fp(del_com(i),Vcar,IC,0);
    heading_des(i+1) = sin(10*tspan(i+1));
    e = heading_des(i) - GPS(i,3);
    e_dot = 0 - yaw_gyro(i);
    e_double_dot = 0 - ((yaw_gyro(i+1)-yaw_gyro(i))/dt);

    del_com(i+1) = (k*1)*e_double_dot + (k*19.06)*e_dot + (k*93.62)*e;
end

figure
plot(tspan, 0.5*sin(10*tspan),'r--', tspan, GPS(:,3),'k')
grid on
title('Part D: Time Response from a Sinusoidal Input of 10 rad/s')
xlabel('Time (seconds)')
ylabel('Heading (rad)')
legend('Expected','Actual')
```




```
save tmp.mat
clear all
load tmp.mat
```

Response at 5 rad/s

```
clear GPS
Vcar = 30; % m/s
del_com(1) = 0;
for i=1:length(tspan)-1
    [GPS(i+1,:),yaw_gyro(i+1),del_meas(i+1),WP(i+1,:)]=run_MKZ_fp(del_com(i),Vcar,IC,0);
    heading_des(i+1) = sin(5*tspan(i+1));
    e = heading_des(i) - GPS(i,3);
    e_dot = 0 - yaw_gyro(i);
    e_double_dot = 0 - ((yaw_gyro(i+1)-yaw_gyro(i))/dt);

    del_com(i+1) = (k*1)*e_double_dot + (k*19.06)*e_dot + (k*93.62)*e;
end

figure
plot(tspan, 0.5*sin(5*tspan),'r--', tspan, GPS(:,3),'k')
grid on
title('Part D: Time Response from a Sinusoidal Input of 5 rad/s')
xlabel('Time (seconds)')
ylabel('Heading (rad)')
legend('Expected','Actual')
```



The system achieves 50% of the desired heading amplitude at a frequency of ~10 rad/s.

There is a slight phase shift on all frequency responses, which is expected from the Bode plot. The gain starts to drop heavily around the 10 rad/s mark and by 30 rad/s the gain has decreased below 0.25.

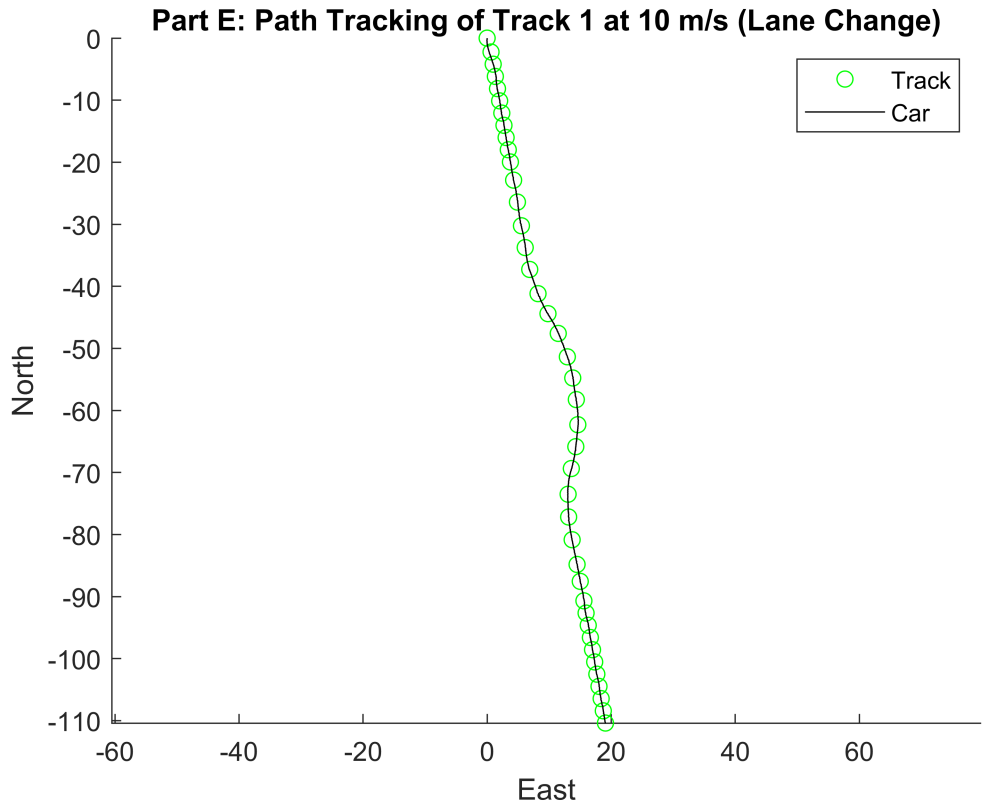
To make our controller track performance at higher frequencies, we would need to increase our corner frequency. In order to do this, we can increase our proportional gain. Increasing this proportional gain would decrease the damping ratio and would therefore increase the amount of overshoot. We would expect the settle time to remain roughly the same as it is inversely proportional to both zeta and natural frequency, so if these are changing in opposite directions, they would cancel out each other's action.

Part (e)

```
save tmp.mat
clear all
load tmp.mat
clear GPS
clear WP
```

Run the Lane Shift Track at 10 m/s

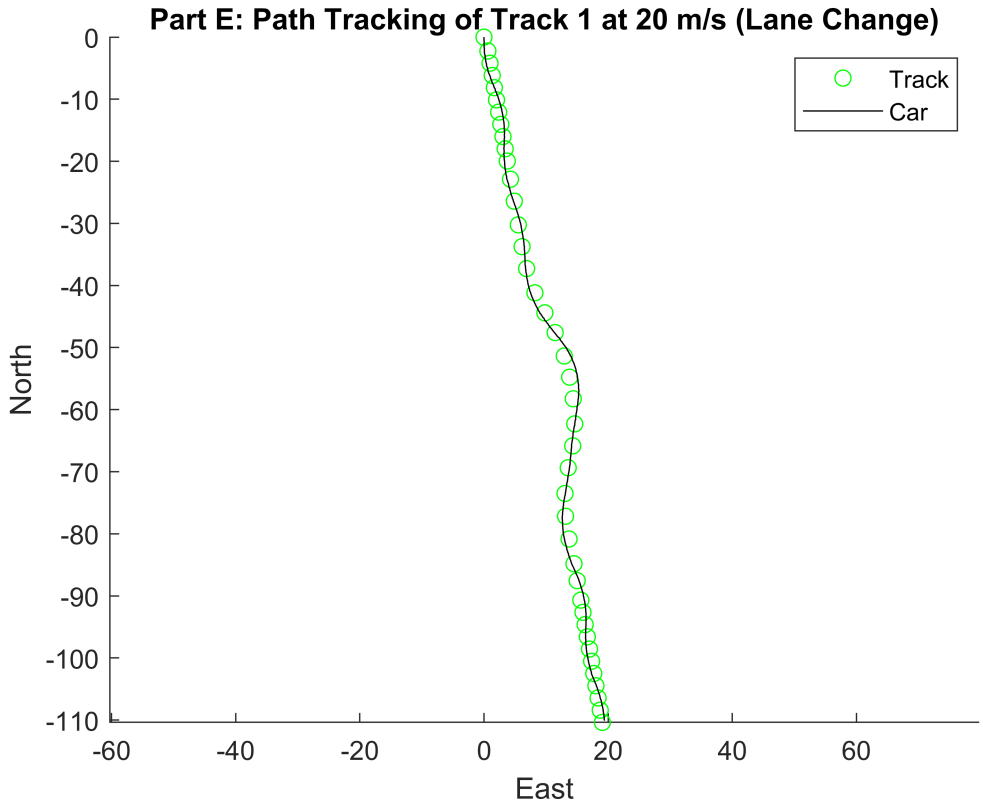
```
dt = 0.01;
t_f = 500;
tspan = 0:dt:t_f;
X0 = [0 0 pi 0];
Vel = 10;
WP_FILE = 1;

for i=1:length(tspan)-1
    [GPS(i+1,:),yaw_gyro(i+1),del_meas(i+1),WP(i+1,:)]=run_MKZ_fp(del_com(i),Vel,X0,WP_FILE);
    heading_des(i+1) = atan2((WP(i+1,1)-GPS(i+1,1)),(WP(i+1,2)-GPS(i+1,2)));
    e = wrap_angle(heading_des(i) - GPS(i,3),[-pi pi]);
    e_dot = 0 - yaw_gyro(i);
    e_double_dot = 0 - ((yaw_gyro(i+1)-yaw_gyro(i))/dt);

    del_com(i+1) = (k*1)*e_double_dot + (k*19.06)*e_dot + (k*93.62)*e;
    if isnan(WP(i+1)) == 1
       break
    end
end

figure
hold on
plot(WP(:,1),WP(:,2),"og")
plot(GPS(:,1),GPS(:,2),"-k")
legend('Track','Car')
xlabel('East')
ylabel('North')
axis equal
title('Part E: Path Tracking of Track 1 at 10 m/s (Lane Change)')
```



Run the Lane Shift Track at 20 m/s

```
save tmp.mat
clear all
load tmp.mat
clear GPS
clear WP

dt = 0.01;
t_f = 500;
tspan = 0:dt:t_f;
X0 = [0 0 pi 0];
Vel = 20;
WP_FILE = 1;

for i=1:length(tspan)-1
    [GPS(i+1,:),yaw_gyro(i+1),del_meas(i+1),WP(i+1,:)]=run_MKZ_fp(del_com(i),Vel,X0,WP_FILE);
    heading_des(i+1) = atan2((WP(i+1,1)-GPS(i+1,1)),(WP(i+1,2)-GPS(i+1,2)));
    e = wrap_angle(heading_des(i) - GPS(i,3),[-pi pi]);
    e_dot = 0 - yaw_gyro(i);
    e_double_dot = 0 - ((yaw_gyro(i+1)-yaw_gyro(i))/dt);

    del_com(i+1) = (k*1)*e_double_dot + (k*19.06)*e_dot + (k*93.62)*e;
    if isnan(WP(i+1)) == 1
       break
    end
end
```

```
figure
hold on
plot(WP(:,1),WP(:,2),"og")
plot(GPS(:,1),GPS(:,2),"-k")
legend('Track','Car')
xlabel('East')
ylabel('North')
axis equal
title('Part E: Path Tracking of Track 1 at 20 m/s (Lane Change)')
```

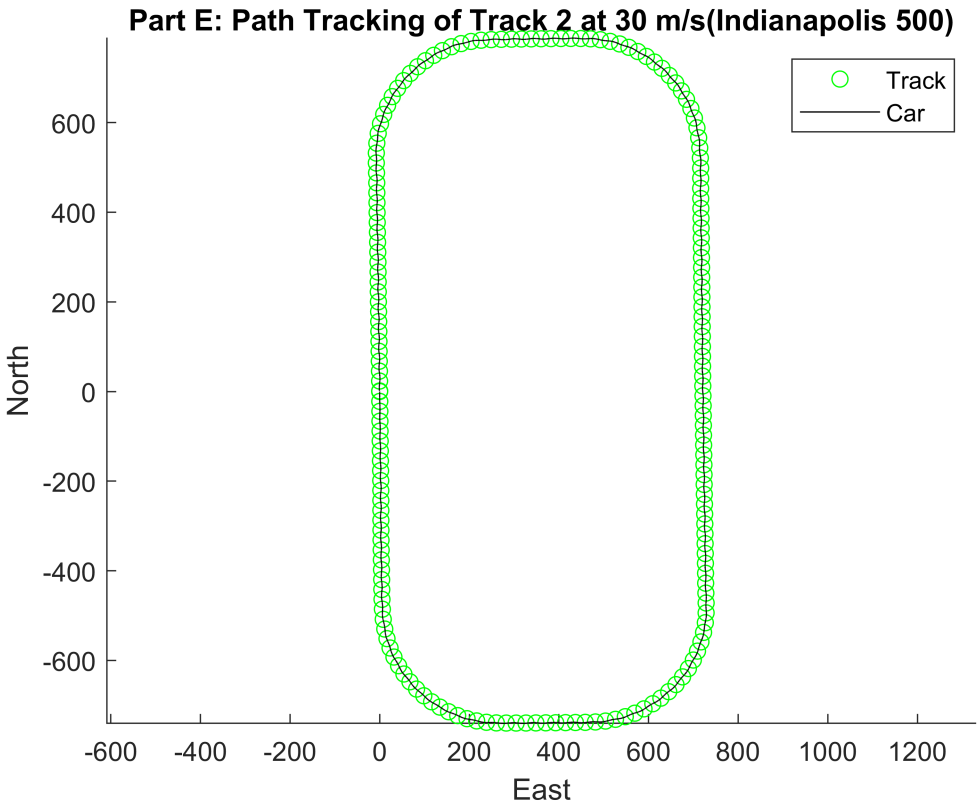



```
save tmp.mat
clear all
load tmp.mat
clear GPS
clear WP
```

Run the Indy 500 Track at 30 m/s

```
Vcar = 30; % m/s

WP_FILE = 2;
for i=1:length(tspan)-1
    [GPS(i+1,:),yaw_gyro(i+1),del_meas(i+1),WP(i+1,:)]=run_MKZ_fp(del_com(i),Vcar,X0,WP_FILE);
    heading_des(i+1) = atan2((WP(i+1,1)-GPS(i+1,1)),(WP(i+1,2)-GPS(i+1,2)));
    e = wrap_angle(heading_des(i) - GPS(i,3),[-pi pi]);
    e_dot = 0 - yaw_gyro(i);
    e_double_dot = 0 - ((yaw_gyro(i+1)-yaw_gyro(i))/dt);
```

```
    del_com(i+1) = (k*1)*e_double_dot + (k*19.06)*e_dot + (k*93.62)*e;
    if isnan(WP(i+1)) == 1
       break
    end
end

figure
hold on
plot(WP(:,1),WP(:,2),"og")
plot(GPS(:,1),GPS(:,2),"-k")
legend('Track','Car')
xlabel('East')
ylabel('North')
axis equal
title('Part E: Path Tracking of Track 2 at 30 m/s(Indianapolis 500)')
```

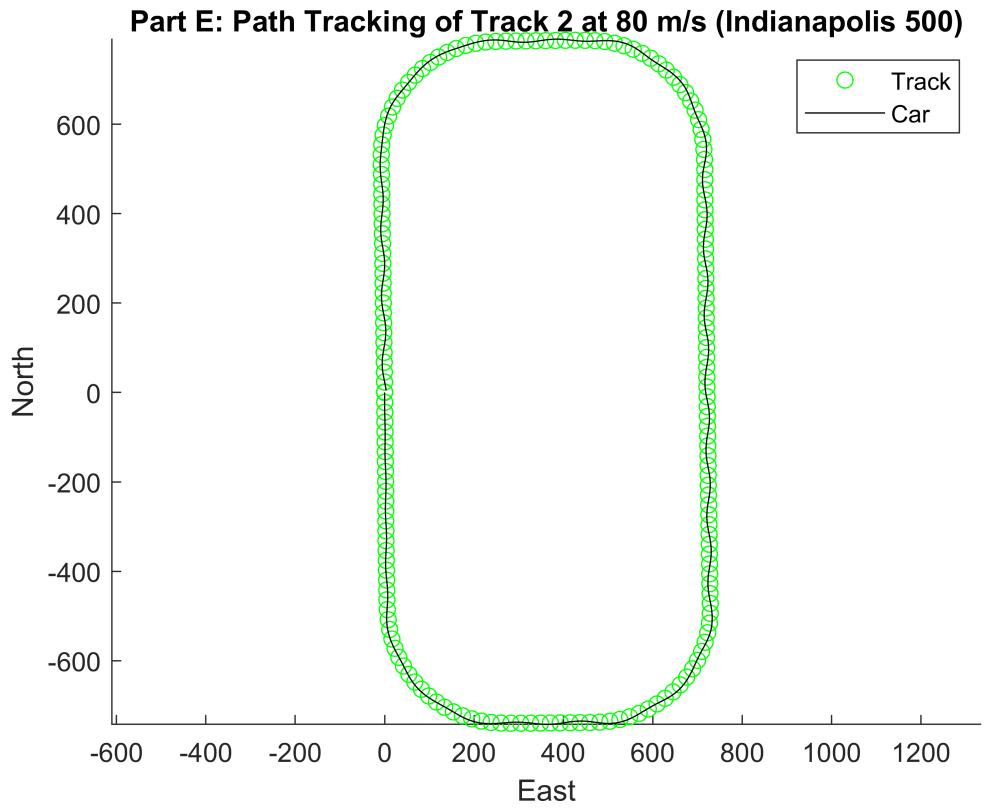



```
save tmp.mat
clear all
load tmp.mat
clear GPS
clear WP
```

Run the Indy 500 Track at 80 m/s

```
tspan_e = 0:dt:60;
Vcar = 80; % m/s
WP_FILE = 2;
for i=1:length(tspan_e)-1
```

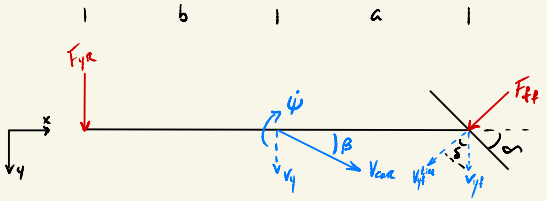
```
    [GPS(i+1,:),yaw_gyro(i+1),del_meas(i+1),WP(i+1,:)]=run_MKZ_fp(del_com(i),Vcar,X0,WP_FILE);
    heading_des(i+1) = atan2((WP(i+1,1)-GPS(i+1,1)),(WP(i+1,2)-GPS(i+1,2)));
    e = wrap_angle(heading_des(i) - GPS(i,3),[-pi pi]);
    e_dot = 0 - yaw_gyro(i);
    e_double_dot = 0 - ((yaw_gyro(i+1)-yaw_gyro(i))/dt);

    del_com(i+1) = (k*1)*e_double_dot + (k*19.06)*e_dot + (k*93.62)*e;
    if isnan(WP(i+1)) == 1
       break
    end
end

figure
hold on
plot(WP(:,1),WP(:,2),"og")
plot(GPS(:,1),GPS(:,2),"-k")
legend('Track','Car')
xlabel('East')
ylabel('North')
axis equal
title('Part E: Path Tracking of Track 2 at 80 m/s (Indianapolis 500)')
```



See Google Earth Results in Appendix B

Our vehicle is able to follow the desired path very well. On both of the trajectory paths provided, our vehicle is able to follow the path nearly perfectly at the appropriate speeds. The trajectory for the lane shift track is followed well up until ~20 m/s. For the Indy 500 track, 80 m/s was about as fast as the vehicle could traverse the track while still remaining on the track itself.

Appendix A: Derivations of State Space and Transfer Functions



$$V_y = V_{cog}\sin\beta \qquad \dot{V}_y = \dot{\beta} V_{cog}\cos\beta$$

$$V_{yr}^{line} = V_y - \dot{\psi} b = V_{cog}\sin\beta - \dot{\psi} b$$

$$V_{yf} = V_y + \dot{\psi} a = V_{cog}\sin\beta + \dot{\psi} a$$

$$F_{yr} = \frac{-C_r}{V_{cog}} V_{yr}^{line} \qquad F_{yf} = \frac{-C_f}{V_{cog}} V_{yf}^{line}$$

$$\Sigma M = I_z \ddot{\psi} = a F_{yf}\cos\delta - b F_{yr} = \frac{-aC_f}{V_{cog}}\left(V_{cog}\sin\beta + \dot{\psi} a - V_{cog}\sin\delta\right)\cos\delta - \frac{-bC_r}{V_{cog}}\left(V_{cog}\sin\beta - \dot{\psi} b\right)$$

$$\ddot{\psi} = \left(\frac{-a^2 C_f - b^2 C_r}{I_z V_{cog}}\right)\dot{\psi} + \left(\frac{-aC_f + bC_r}{I_z}\right)\beta + \left(\frac{aC_f}{I_z}\right)\delta$$

$$\Sigma F_y = m\ddot{y} = m\left(V_{cog}\dot{\psi} + \dot{V}_y\right) = m V_{cog}\left(\dot{\psi} + \dot{\beta}\cos\beta\right) = F_{yr} + F_{yf}\cos\delta$$

$$m V_{cog}\left(\dot{\psi} + \dot{\beta}\cos\beta\right) = \frac{-C_r}{V_{cog}}\left(V_{cog}\sin\beta - \dot{\psi} b\right) + \frac{-C_f}{V_{cog}}\left(V_{cog}\sin\beta + \dot{\psi} a - V_{cog}\sin\delta\right)\cos\delta$$

$$\dot{\beta} = \left(\frac{C_r b - C_f a}{m V_{cog}^2} - 1\right)\dot{\psi} + \left(\frac{-C_r - C_f}{m V_{cog}}\right)\beta + \left(\frac{C_f}{m V_{cog}}\right)\delta$$

$$\begin{bmatrix} \dot{\beta} \\ \ddot{\psi} \end{bmatrix} = \begin{bmatrix} \frac{-C_r - C_f}{m V_{cog}} & \frac{bC_r - aC_f}{m V_{cog}^2} - 1 \\ \frac{-aC_f + bC_r}{I_z} & \frac{-a^2 C_f - b^2 C_r}{I_z V_{cog}} \end{bmatrix} \begin{bmatrix} \beta \\ \dot{\psi} \end{bmatrix} + \begin{bmatrix} \frac{C_f}{m V_{cog}} \\ \frac{aC_f}{I_z} \end{bmatrix} \delta$$

$$\vec{x} = \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = \begin{bmatrix} \beta \\ \dot{\psi} \end{bmatrix} \qquad \vec{u} = [u_1] = \delta \qquad \vec{y} = \begin{bmatrix} V_y \\ \dot{\psi} \\ \ddot{y} \end{bmatrix}$$

$$V_y = V_{cog}\cdot\beta \qquad \dot{\psi} = \dot{\psi} \qquad \ddot{y} = \left(\frac{bC_r - aC_f}{m V_{cog}}\right)\dot{\psi} + \left(\frac{-C_r - C_f}{m}\right)\beta + \left(\frac{C_f}{m}\right)\delta$$

$$\begin{bmatrix} V_y \\ \dot{\psi} \\ \ddot{y} \end{bmatrix} = \begin{bmatrix} V_{cog} & 0 \\ 0 & 1 \\ \frac{-C_r - C_f}{m} & \frac{bC_r - aC_f}{m V_{cog}} \end{bmatrix} \begin{bmatrix} \beta \\ \dot{\psi} \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ C_f/m \end{bmatrix} \delta$$

$\tau = 0.1s \qquad G_{DC} = 1$

$A\dot{\delta} + B\delta = C\delta_{com} \qquad As + B = 0 \qquad s = \frac{-B}{A} \qquad \tau = \left|\frac{1}{s}\right| = \frac{A}{B} = 0.1 = \frac{1}{10}$

$A = 1$
$B = 10$

$\dot{\delta} + 10\delta = C\delta_{com} \qquad \Delta[s](s+10) = C\,\Delta_{com}[s]$

$T[s] = \frac{\Delta[s]}{\Delta_{com}[s]} = \frac{C}{s+10} \qquad G(\omega) = T[j\omega] = \frac{C}{j\omega+10}$

$G_{DC} = G(0) = \frac{C}{0+10} = 1 \qquad C = 10$

$$\boxed{\dot{\delta} + 10\delta = 10\delta_{com}}$$

$M\% = 5\% \rightarrow \zeta = 0.7$

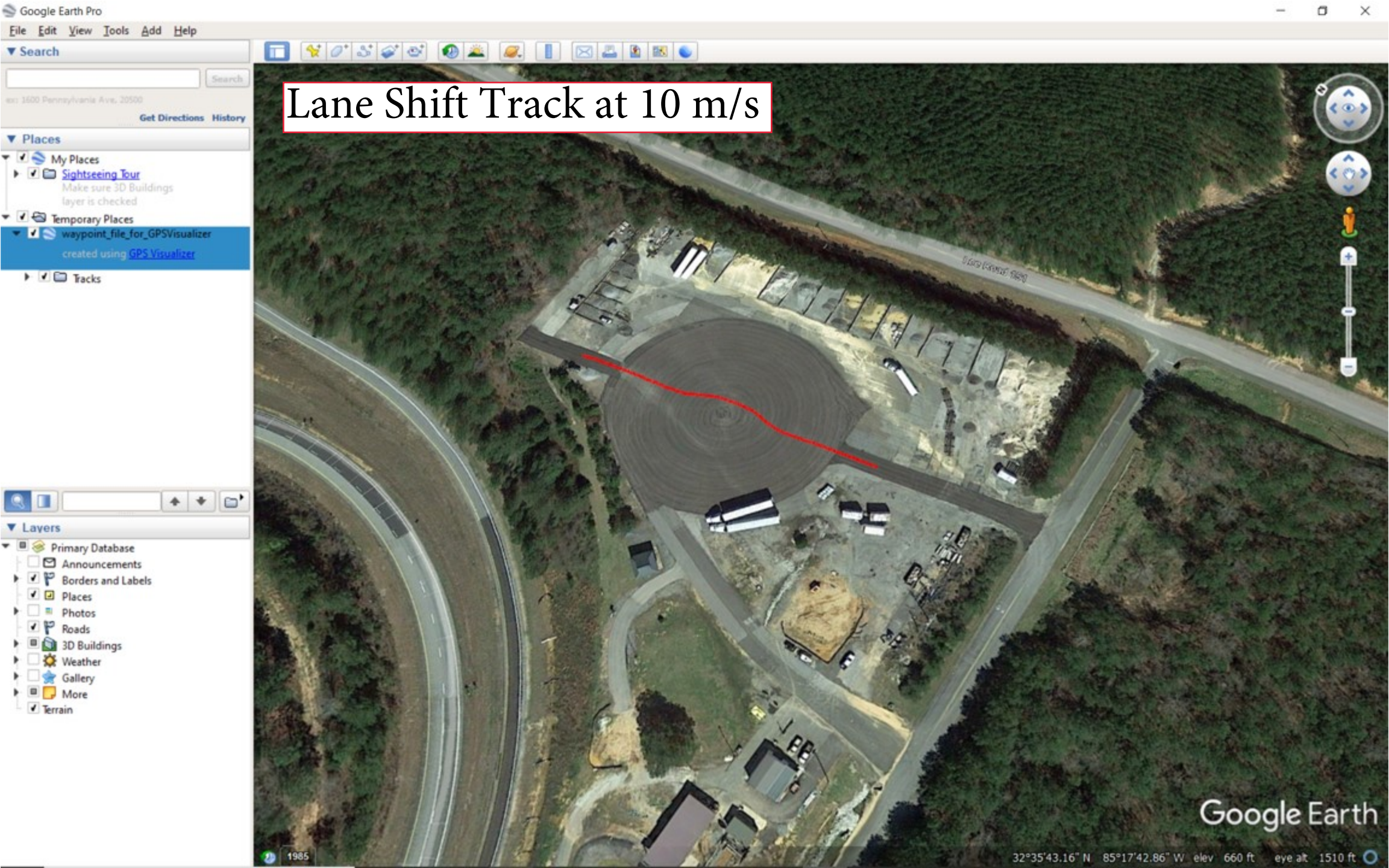
$A\ddot{z} + B\dot{z} + Cz = D\dot{\delta} + E\delta$

$As^2 + Bs + C = 0$
$s^2 + \frac{B}{A}s + \frac{C}{A} = 0$
$s^2 + 2\zeta\omega_n s + \omega_n^2 = 0$

$t_s = 0.6 = \frac{4.6}{\zeta\omega_n}$
$\omega_N = 10.9524$

$$\boxed{\zeta = 0.7 \qquad \omega_n = 10.9524}$$

Appendix B: Google Earth Track Run Results


Google Earth Pro
File Edit View Tools Add Help
Search
Get Directions History
Places
My Places
Sightseeing Tour
Make sure 3D Buildings layer is checked
Temporary Places
waypoint_file_for_GPSVisualizer
created using GPS Visualizer
Tracks
Layers
Primary Database
Announcements
Borders and Labels
Places
Photos
Roads
3D Buildings
Weather
Gallery
More
Terrain
Lane Shift Track at 10 m/s
Google Earth
1985
32°35'43.16" N 85°17'42.86" W elev 660 ft eye alt 1510 ft


Google Earth Pro
File Edit View Tools Add Help
Search
Get Directions History
Places
My Places
Sightseeing Tour
Temporary Places
waypoint_file_for_GPSVisualizer
created using GPS Visualizer
Tracks
waypoint_file_for_GPSVisualizer
created using GPS Visualizer
Tracks
Layers
Primary Database
Announcements
Borders and Labels
Places
Photos
Roads
3D Buildings
Weather
Gallery
More
Terrain
Lane Shift Track at 20 m/s
Google Earth


Google Earth Pro
File Edit View Tools Add Help
Search
Get Directions History
Places
My Places
Sightseeing Tour
Make sure 3D Buildings layer is checked
Temporary Places
waypoint_file_for_GPSVisualizer
created using GPS Visualizer
Tracks
Layers
Primary Database
Announcements
Borders and Labels
Places
Photos
Roads
3D Buildings
Weather
Gallery
More
Terrain
Indy 500 Track at 30 m/s
Indianapolis Motor Speedway
IMS Lot 6
Indianapolis Motor Speedway Museum
American Legion Post 500
Image Landsat / Copernicus
Google Earth


Google Earth Pro
File Edit View Tools Add Help
Search
Get Directions History
Places
My Places
Sightseeing Tour
Make sure 3D Buildings layer is checked
Temporary Places
waypoint_file_for_GPSVisualizer
created using GPS Visualizer
Tracks
Layers
Primary Database
Announcements
Borders and Labels
Places
Photos
Roads
3D Buildings
Weather
Gallery
More
Terrain
Indy 500 Track at 80 m/s
Indianapolis Motor Speedway Museum
Indianapolis Motor Speedway
Brickyard Crossing
Lacy Branch
Falcon Creek
Image Landsat / Copernicus
Google Earth