For lateral control:

$$\frac{d}{dt}\begin{bmatrix} e_1 \\ \dot{e}_1 \\ e_2 \\ \dot{e}_2 \end{bmatrix} = \underbrace{\begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & -\frac{4C_\alpha}{m\dot{x}} & \frac{4C_\alpha}{m} & \frac{-2C_\alpha(l_f - l_r)}{m\dot{x}} \\ 0 & 0 & 0 & 1 \\ 0 & \frac{-2C_\alpha(l_f - l_r)}{I_z \dot{x}} & \frac{2C_\alpha(l_f - l_r)}{I_z} & -\frac{2C_\alpha(l_f^2 + l_r^2)}{I_z \dot{x}} \end{bmatrix}}_{A_{lat}} \begin{bmatrix} e_1 \\ \dot{e}_1 \\ e_2 \\ \dot{e}_2 \end{bmatrix} + \underbrace{\begin{bmatrix} 0 & 0 \\ \frac{2C_\alpha}{m} & 0 \\ 0 & 0 \\ \frac{2C_\alpha l_f}{I_z} & 0 \end{bmatrix}}_{B_{lat}} \begin{bmatrix} \delta \\ F \end{bmatrix} + \begin{bmatrix} \cdot \\ \cdot \\ \cdot \end{bmatrix} \cancel{\dot{\psi}_{des}}^{\,0}$$

For lateral control:

$$\frac{d}{dt}\begin{bmatrix} x \\ \dot{x} \end{bmatrix} = \underbrace{\begin{bmatrix} 0 & 1 \\ 0 & 0 \end{bmatrix}}_{A_{long}} \begin{bmatrix} x \\ \dot{x} \end{bmatrix} + \underbrace{\begin{bmatrix} 0 & 0 \\ 0 & \frac{1}{m} \end{bmatrix}}_{B_{long}} \begin{bmatrix} \delta \\ F \end{bmatrix} + \begin{bmatrix} 0 \\ \cancel{\dot{\psi}\dot{y} - fg}^{\,0} \end{bmatrix}$$

$e_1 \rightarrow$ distance between vehicle COM and trajectory.
$e_2 \rightarrow$ orientation error with respect to trajectory.

Exercise 1

1) Since the A and B matrices for the longitudinal controller are constant, the controllability matrix does not depend on the longitudinal speed.

$$P_{long} = \begin{bmatrix} B_{long} & A_{long} B_{long} \end{bmatrix} = \begin{bmatrix} 0 & 0 & 0 & \frac{1}{m} \\ 0 & \frac{1}{m} & 0 & 0 \end{bmatrix}$$

$$\text{rank}(P_{long}) = 2 \rightarrow \text{full rank}$$

Thus, the longitudinal position and velocity are <u>controllable</u> at all speeds.

For the lateral controller:

$$P_{lat} = \begin{bmatrix} B_{lat} & A_{lat} B_{lat} & A^2_{lat} B_{lat} & A^3_{lat} B_{lat} \end{bmatrix}$$

To check for controllability, the rank of this matrix can be computed at $\dot{x} = 2, 5$ and $8\ m/s$.

From programming, we get:

$$\text{rank}(Q_{lat,\dot{x}=2}) = \text{rank}(Q_{lat,\dot{x}=5}) = \text{rank}(Q_{lat,\dot{x}=8}) = 4$$

We get: the system describing the lateral error dynamics is <u>controllable</u> at $\dot{x} = 2, 5$ and $8\ m/s$.

To check observability, assuming $C = [Id]$, i.e., we are able to observe all the states, since we have all the sensing data (GPS, IMU etc.) at this stage.

$\therefore$ For longitudinal dynamics: $C_{long} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \Rightarrow Q_{long} = \begin{bmatrix} C \\ CA \end{bmatrix} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \\ 0 & 0 \\ 0 & 0 \end{bmatrix}$

$\text{rank}(Q_{long}) = 2 \Rightarrow$ observable

For the lateral dynamics, substituting $\dot{x} = 2, 5$ & 8 m/s and computing $Q_{lat} = [C \;\; CA \;\; CA^2 \;\; CA^3]^T$:

$\text{rank}(Q_{lat, \dot{x}=2}) = \text{rank}(Q_{lat, \dot{x}=5}) = \text{rank}(Q_{lat, \dot{x}=8}) = 4$

$\Rightarrow$ The system is fully observable.

Python code attached below.

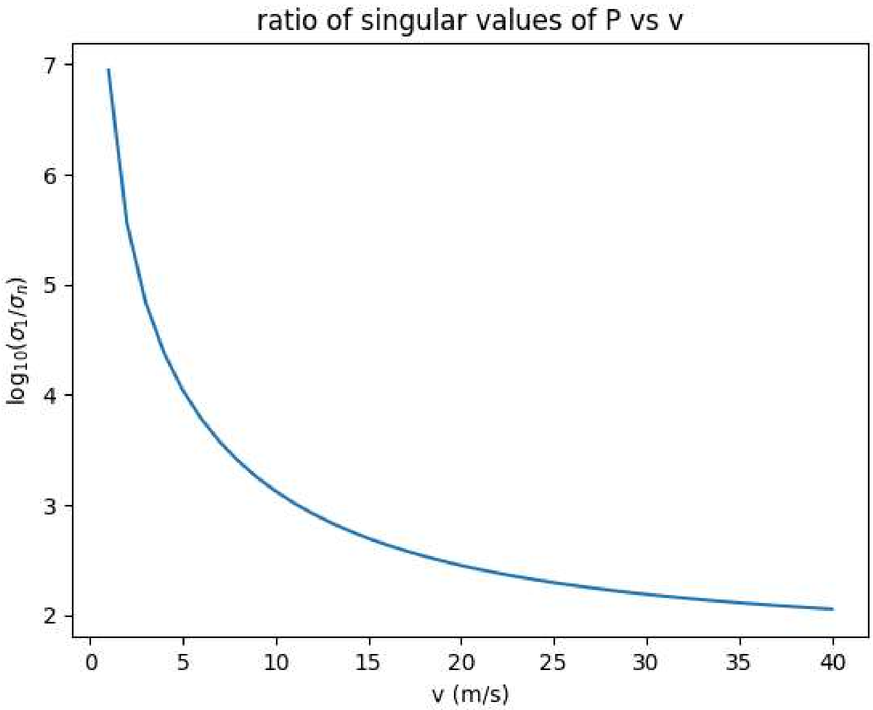
In [19]:
```
import numpy as np
from scipy.signal import ss2tf
import matplotlib.pyplot as plt

# defining constants
m = 1888.6
lr = 1.39
lf = 1.55
C_alpha = 20000
Iz = 25854
f = 0.019

# for longitudinal dynamics
A_long = np.array([[0, 1],
                   [0, 0]])
B_long = np.array([[0, 0],
                   [0, 1/m]])
C_long = np.eye(2)

P_long = np.hstack([B_long, A_long@B_long])
Q_long = np.vstack([C_long, C_long@A_long])
print("Longitudinal Error Dynamics at x_dot = 2, 5 and 8 m/s")
print("Rank of controllability matrix:", np.linalg.matrix_rank(P_long))
print("Rank of observability matrix:", np.linalg.matrix_rank(Q_long))


x_dot = [2, 5, 8]
B_lat = np.array([[0, 0],
                  [2*C_alpha/m, 0],
                  [0, 0],
                  [2*C_alpha*lf/Iz, 0]])
C_lat = np.eye(4)
D_lat = np.zeros([4, 2])

for dx in x_dot:
    A_lat = np.array([[0, 1, 0, 0],
                      [0, -4*C_alpha/(m*dx), 4*C_alpha/m, -2*C_alpha*(lf - lr)/(m*dx)],
                      [0, 0, 0, 1],
                      [0, -2*C_alpha*(lf - lr)/(Iz*dx), 2*C_alpha*(lf - lr)/Iz, -2*C_alpha*(lf**2 + lr**2)/(Iz*dx)]])

    P_lat = np.hstack([B_lat, A_lat@B_lat, A_lat@A_lat@B_lat, A_lat@ A_lat@A_lat@B_lat])
    Q_lat = np.vstack([C_lat, C_lat@A_lat, C_lat@A_lat@A_lat, C_lat@A_lat@A_lat@A_lat])
    print("\nLateral Error Dynamics at x_dot = ", dx, "m/s")
    print("Rank of controllability matrix:", np.linalg.matrix_rank(P_lat))
    print("Rank of observability matrix:", np.linalg.matrix_rank(Q_lat))
```

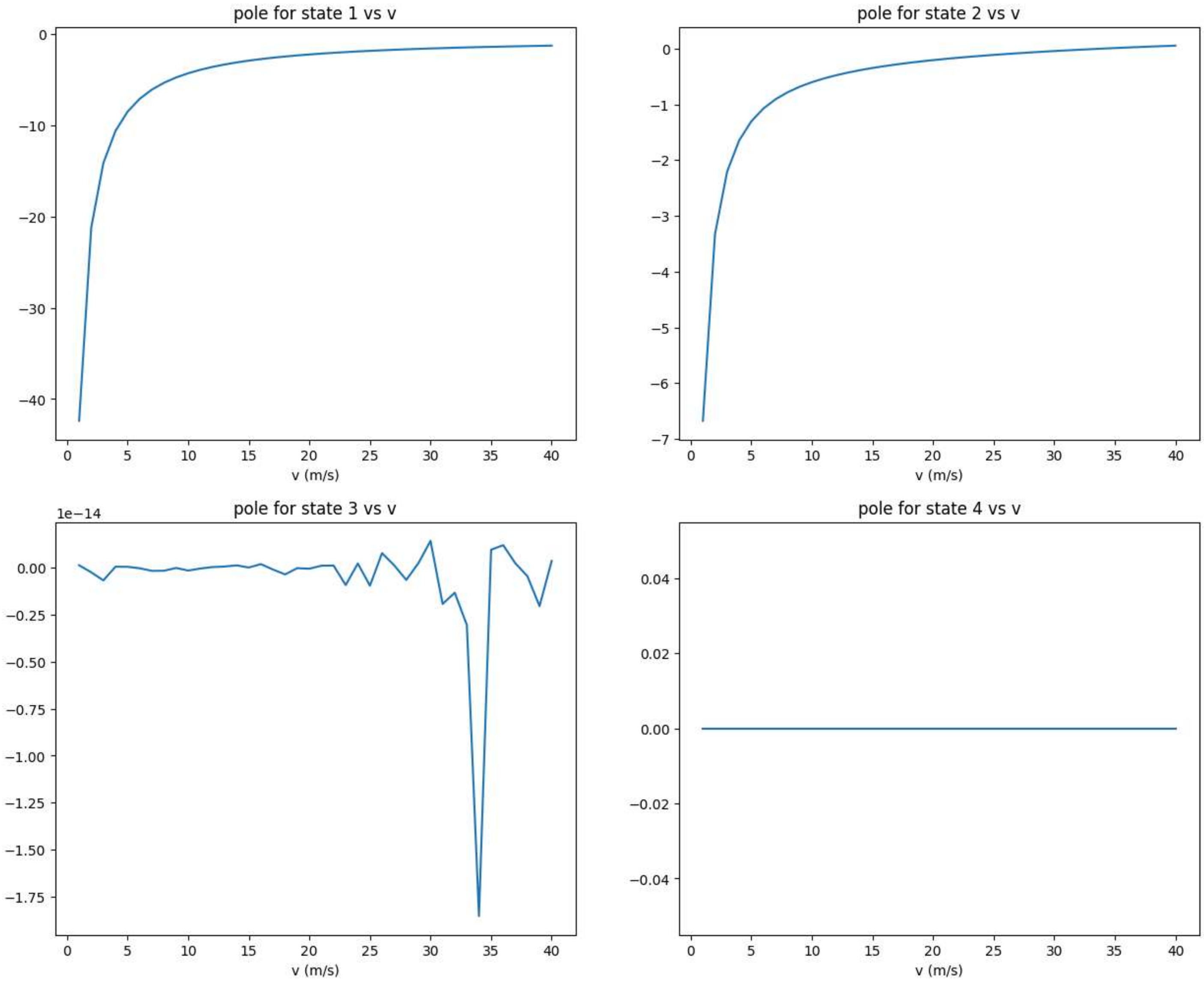
```
Longitudinal Error Dynamics at x_dot = 2, 5 and 8 m/s
Rank of controllability matrix: 2
Rank of observability matrix: 2

Lateral Error Dynamics at x_dot =  2 m/s
Rank of controllability matrix: 4
Rank of observability matrix: 4

Lateral Error Dynamics at x_dot =  5 m/s
Rank of controllability matrix: 4
Rank of observability matrix: 4

Lateral Error Dynamics at x_dot =  8 m/s
Rank of controllability matrix: 4
Rank of observability matrix: 4
```

In [26]:
```
v = np.linspace(1, 40, 40)
log_svd = []
poles_real = []

for vel in v:
    A_lat = np.array([[0, 1, 0, 0],
                      [0, -4*C_alpha/(m*vel), 4*C_alpha/m, -2*C_alpha*(lf - lr)/(m*vel)],
                      [0, 0, 0, 1],
                      [0, -2*C_alpha*(lf - lr)/(Iz*vel), 2*C_alpha*(lf - lr)/Iz, -2*C_alpha*(lf**2 + lr**2)/(Iz*vel)]])
    B_lat = np.array([[0, 0],
                      [2*C_alpha/m, 0],
                      [0, 0],
                      [2*C_alpha*lf/Iz, 0]])

    P_lat = np.hstack([B_lat, A_lat@B_lat, A_lat@A_lat@B_lat, A_lat@ A_lat@A_lat@B_lat])

    _, sig, _ = np.linalg.svd(P_lat)
    log_svd.append(np.log10(sig[0]/sig[-1]))
```

```python
    _, den = ss2tf(A_lat, B_lat, C_lat, D_lat)
    poles_real.append(np.roots(den).real)

poles_real = np.array(poles_real)

plt.plot(v, log_svd)
plt.xlabel("v (m/s)")
plt.ylabel("log$_1$$_0$($\sigma$$_1$/$\sigma$$_n$)")
plt.title("ratio of singular values of P vs v")
plt.show()

plt.figure(figsize = [15, 12])
plt.subplot(2, 2, 1)
plt.plot(v, poles_real[:, 0])
plt.xlabel("v (m/s)")
plt.title("pole for state 1 vs v")

plt.subplot(2, 2, 2)
plt.plot(v, poles_real[:, 1])
plt.xlabel("v (m/s)")
plt.title("pole for state 2 vs v")

plt.subplot(2, 2, 3)
plt.plot(v, poles_real[:, 2])
plt.xlabel("v (m/s)")
plt.title("pole for state 3 vs v")

plt.subplot(2, 2, 4)
plt.plot(v, poles_real[:, 3])
plt.xlabel("v (m/s)")
plt.title("pole for state 4 vs v")

plt.show()
```

```
<>:27: SyntaxWarning: invalid escape sequence '\s'
<>:27: SyntaxWarning: invalid escape sequence '\s'
C:\Users\barat\AppData\Local\Temp\ipykernel_7032\1166234373.py:27: SyntaxWarning: invalid escape sequence '\s'
  plt.ylabel("log$_1$$_0$($\sigma$$_1$/$\sigma$$_n$)")
```





Conclusions

From singular value plot

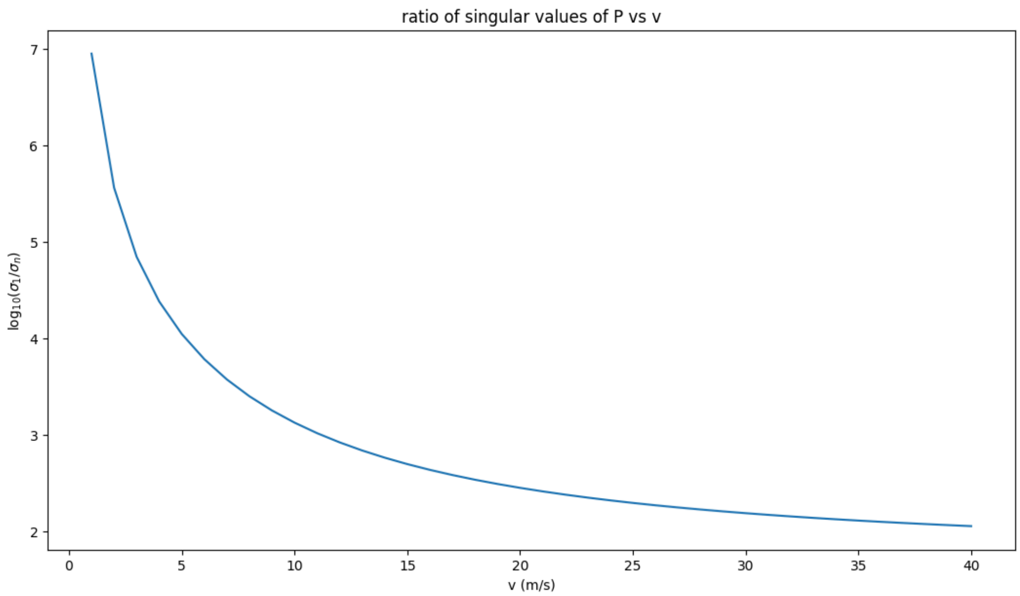
As v increases, the ratio of singular values decreases, indicating that the controllability matrix becomes less well-conditioned at higher speeds. A high condition number (i.e., large difference between singular values) implies that we have stronger controllability. As the singular values get closer to each other, the system may become less controllable, or certain states might be harder to control at higher speeds.

From eigenvalue plot

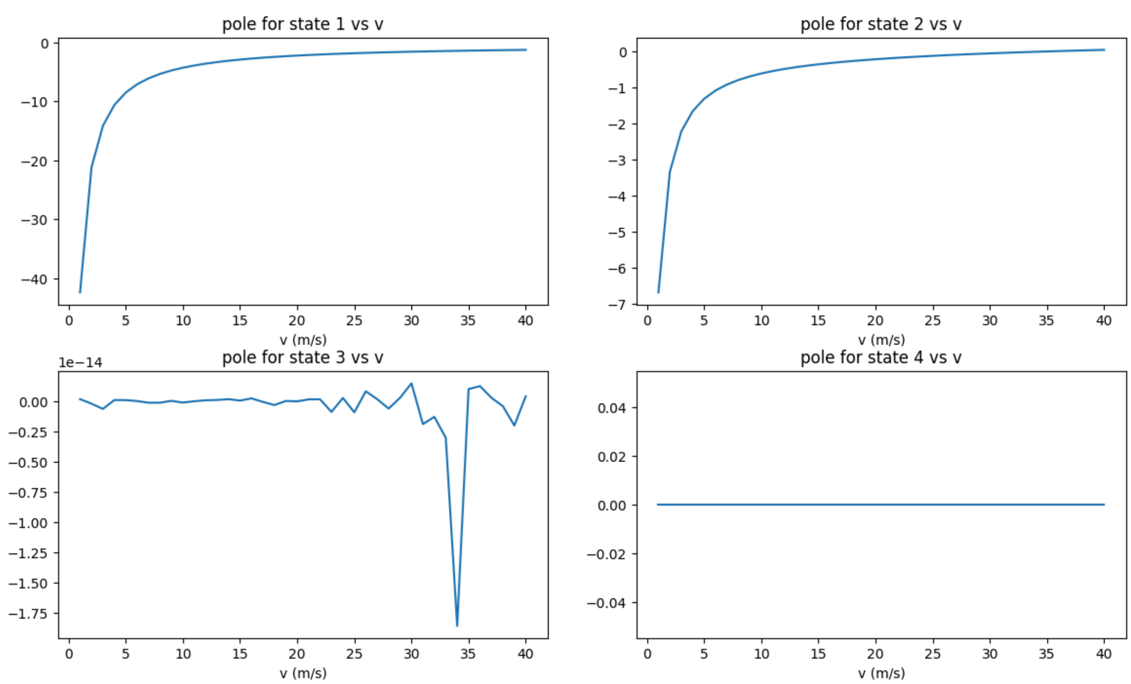
The shift of some eigenvalues towards zero suggests that the system might become marginally stable or potentially uncontrollable at higher velocities. For State 3, the eigenvalue remains close to zero but exhibits some fluctuation, especially around v = 30 m/s. This could indicate a nearly uncontrollable or unobservable mode that becomes more prominent around this velocity. For State 4, the eigenvalue stays close to zero without much change, suggesting a static mode that may be inherently difficult to control or observe across all speeds.

Overall, these plots indicate that both controllability and observability might be compromised as the system operates at higher velocities, with certain modes potentially becoming unmanageable.

Plot of largest singular value by the smallest vs velocity



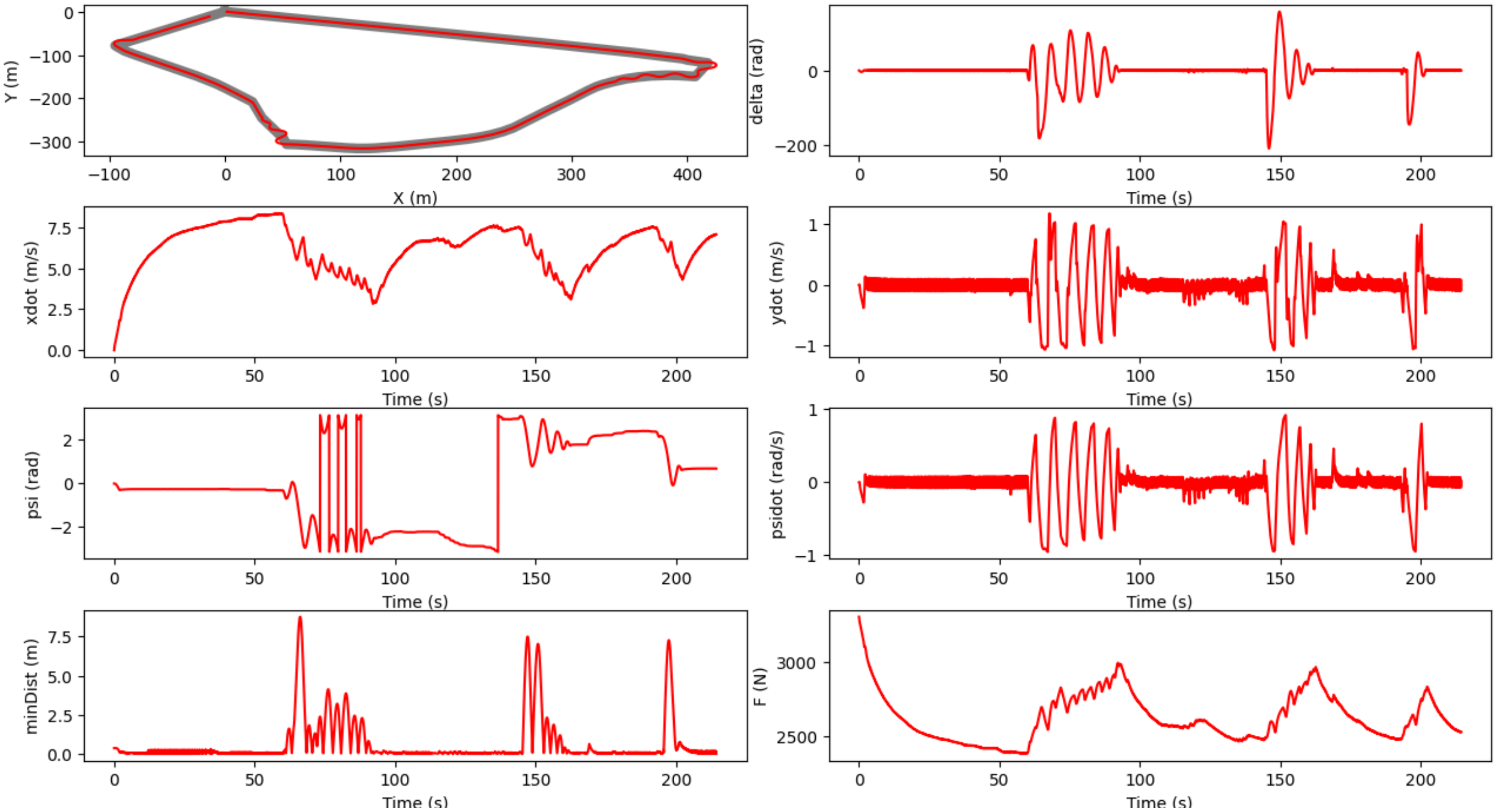
Plot of the poles of the system vs velocity



Name: Barathkrishna Satheeshkumar

Andrew ID: bsathees

Performance plot



Evaluation

```
Evaluating...
Score for completing the loop: 30.0/30.0
Score for average distance: 30.0/30.0
Score for maximum distance: 30.0/30.0
Your time is  214.272
Your total score is : 100.0/100.0
total steps:  214272
maxMinDist:  8.776031840267327
avgMinDist:  0.6536481573324382
```