```
!pip install control
```

```
Collecting control
  Downloading https://files.pythonhosted.org/packages/e8/b0/32a903138505dd4ea523
     |████████████████████████████████| 256kB 4.6MB/s
Requirement already satisfied: numpy in /usr/local/lib/python3.6/dist-packages (
Requirement already satisfied: scipy in /usr/local/lib/python3.6/dist-packages (
Requirement already satisfied: matplotlib in /usr/local/lib/python3.6/dist-packa
Requirement already satisfied: python-dateutil>=2.1 in /usr/local/lib/python3.6/
Requirement already satisfied: kiwisolver>=1.0.1 in /usr/local/lib/python3.6/dis
Requirement already satisfied: pyparsing!=2.0.4,!=2.1.2,!=2.1.6,>=2.0.1 in /usr/
Requirement already satisfied: cycler>=0.10 in /usr/local/lib/python3.6/dist-pac
Requirement already satisfied: six>=1.5 in /usr/local/lib/python3.6/dist-package
Building wheels for collected packages: control
  Building wheel for control (setup.py) ... done
  Created wheel for control: filename=control-0.8.3-py2.py3-none-any.whl size=26
  Stored in directory: /root/.cache/pip/wheels/c2/d9/cc/90b28cb139a6320a3af22854
Successfully built control
Installing collected packages: control
Successfully installed control-0.8.3
```

```
import numpy as np
import matplotlib.pyplot as plt
from scipy.linalg import svd
from numpy.linalg import matrix_power
from numpy.linalg import matrix_rank
from scipy import signal
import control
```

```
m=1888.6
lr=1.39
lf=1.55
Ca=20000
Iz=25854
f=0.019
delT=0.032
Vx=6;

A=np.array([[0,1,0,0],
  [0,-(4*Ca)/(m*Vx),(4*Ca)/m,-(2*Ca*(lf-lr))/(m*Vx)],
  [0,0,0,1],
  [0,-1*(2*Ca*(lf-lr))/(Iz*Vx),(2*Ca*(lf-lr))/Iz,-1*(2*Ca*(lf**2+lr**2))/(Iz*Vx)]],

B=np.array([[0,0],
    [(2*Ca)/m,0],
    [0,0],
    [(2*Ca*lf)/Iz,0]], dtype=float)

C=np.identity(4)


```

```
D=np.zeros((4,2))


sys=control.StateSpace(A,B,C,D)
```

```
P=np.column_stack((B,A@B,matrix_power(A,2)@B,matrix_power(A,3)@B))
```

```
U, s, VT = svd(P)
s
```

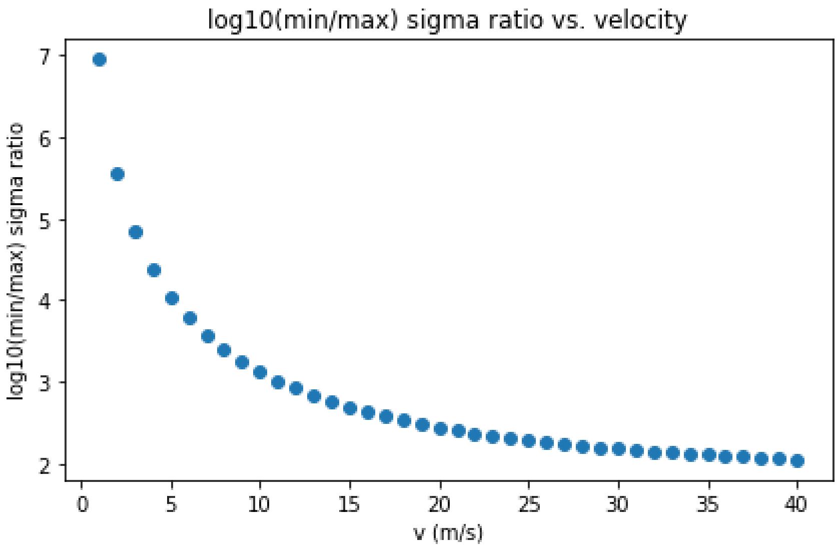
```
array([8.57208893e+03, 1.22408603e+01, 3.34276770e+00, 1.41586038e+00])
```

```
poles = control.pole(sys)
poles
```

```
array([ 0.00000000e+00, -7.10513903e+00, -1.07247654e+00, -1.74015667e-16])
```

```
x = np.arange(1,41)
ln_sigma=[]

for i in x:
  Vx=i
  A=np.array([[0,1,0,0],
    [0,-(4*Ca)/(m*Vx),(4*Ca)/m,-(2*Ca*(lf-lr))/(m*Vx)],
    [0,0,0,1],
    [0,-1*(2*Ca*(lf-lr))/(Iz*Vx),(2*Ca*(lf-lr))/Iz,-1*(2*Ca*(lf**2+lr**2))/(Iz*Vx)]

  B=np.array([[0,0],
    [(2*Ca)/m,0],
    [0,0],
    [(2*Ca*lf)/Iz,0]], dtype=float)

  C=np.identity(4)

  D=np.zeros((4,2))

  sys=control.StateSpace(A,B,C,D)

  P=np.column_stack((B,A@B,matrix_power(A,2)@B,matrix_power(A,3)@B))

  U, s, VT = svd(P)

  ln_sigma.append(np.log10(s[0]/s[3]))


y=ln_sigma

plt.plot(x, y, 'o')
plt.xlabel("v (m/s)")
```

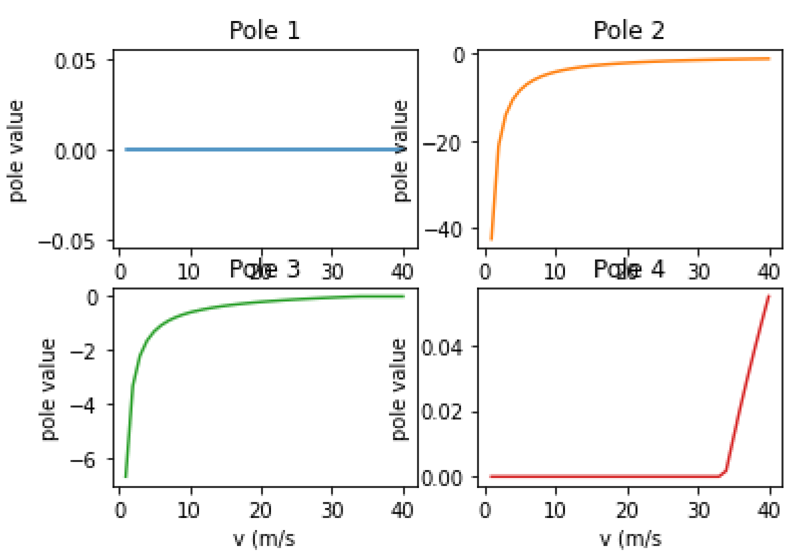
```
plt.ylabel("log10(min/max) sigma ratio")
plt.title("log10(min/max) sigma ratio vs. velocity")
plt.tight_layout()
plt.show()
```




```
x = np.arange(1,41)
pole1=[]
pole2=[]
pole3=[]
pole4=[]

for i in x:
  Vx=i
  A=np.array([[0,1,0,0],
    [0,-(4*Ca)/(m*Vx),(4*Ca)/m,-(2*Ca*(lf-lr))/(m*Vx)],
    [0,0,0,1],
    [0,-1*(2*Ca*(lf-lr))/(Iz*Vx),(2*Ca*(lf-lr))/Iz,-1*(2*Ca*(lf**2+lr**2))/(Iz*Vx)

  B=np.array([[0,0],
    [(2*Ca)/m,0],
    [0,0],
    [(2*Ca*lf)/Iz,0]], dtype=float)

  C=np.identity(4)

  D=np.zeros((4,2))

  sys=control.StateSpace(A,B,C,D)

  P=np.column_stack((B,A@B,matrix_power(A,2)@B,matrix_power(A,3)@B))

  poles=control.pole(sys)

  pole1.append(poles[0])
  pole2.append(poles[1])
```

```
pole3.append(poles[2])
pole4.append(poles[3])
```

```

fig, axs = plt.subplots(2, 2)
axs[0, 0].plot(x, pole1)
axs[0, 0].set_title('Pole 1')
#axs[0, 0].set_ylim([-10,10])
axs[0, 1].plot(x, pole2, 'tab:orange')
axs[0, 1].set_title('Pole 2')
#axs[0, 1].set_ylim([-10,10])
axs[1, 0].plot(x, pole3, 'tab:green')
axs[1, 0].set_title('Pole 3')
#axs[1, 0].set_ylim([-10,10])
axs[1, 1].plot(x, pole4, 'tab:red')
axs[1, 1].set_title('Pole 4')
#axs[1, 1].set_ylim([-10,10])

for ax in axs.flat:
    ax.set(xlabel='v (m/s', ylabel='pole value')

# Hide x labels and tick labels for top plots and y ticks for right plots.
#for ax in axs.flat:
   # ax.label_outer()
```




```
pole4[32] #first negative pole
```

```
-1.0910085295447952e-14
```

```
x[32] #corresponds to 33 m/s
```

```
33
```

Write Up...

Controllability: As seen in the svd log ratio plot, it can be inferred that as the speed increases, the system becomes ease of control increases.

Stability: As seen in the 4 pole plot, the system is stable for speeds below rougly 33 m/s. Once above this value the system presents an unstable pole.

The main takeaway I took from this is that stability and controllability for a system can have different relationships with the system.

```
1
```