In [14]:

```
import numpy as np
import matplotlib.pyplot as plt
```

In [15]:

```
Velo = [2, 5, 8]
```

In [16]:

```
for v in Velo:
    A= np.array([[0, 1], [0, 0]])
    print(A.shape)
    C = np.identity(2)
    B = np.expand_dims(np.array([0, 1/1888.6]).T, axis =1)


    P = np.hstack((B, A@B, A@A@B, A@A@A@B))
    #print(P.shape)

    Q = np.vstack((C, C@A, C@A@A, C@A@A@A))
    #print(Q.shape)
    rankP = np.linalg.matrix_rank(P)
    rankQ = np.linalg.matrix_rank(Q)
    print(f'For Velocity = {v} m/s\n')
    print(f'Rank of Controllability matrix = {rankP}')
    print(f'Rank of Observability matrix = {rankQ} \n')
```

```
(2, 2)
For Velocity = 2 m/s

Rank of Controllability matrix = 2
Rank of Observability matrix = 2

(2, 2)
For Velocity = 5 m/s

Rank of Controllability matrix = 2
Rank of Observability matrix = 2

(2, 2)
For Velocity = 8 m/s

Rank of Controllability matrix = 2
Rank of Observability matrix = 2
```

In [17]:

```
for v in Velo:
    A= np.array([[0, 1, 0, 0],
                 [0, -42.359/v, 42.359, -3.388/v ],
                 [0, 0, 0, 1],
                 [0 , -0.2475/v, 0.2475, -6.7/v]])
    C = np.identity(4)
    B = np.expand_dims(np.array([0, 21.1795, 0, 2.398]).T, axis =1)

    P = np.hstack((B, A@B, A@A@B, A@A@A@B))
    #print(P.shape)

    Q = np.vstack((C, C@A, C@A@A, C@A@A@A))
    #print(Q.shape)
    rankP = np.linalg.matrix_rank(P)
    rankQ = np.linalg.matrix_rank(Q)
    print(f'For Velocity = {v} m/s\n')
    print(f'Rank of Controllability matrix = {rankP}')
    print(f'Rank of Observability matrix = {rankQ} \n')
```

```
For Velocity = 2 m/s

Rank of Controllability matrix = 4
Rank of Observability matrix = 4

For Velocity = 5 m/s

Rank of Controllability matrix = 4
Rank of Observability matrix = 4

For Velocity = 8 m/s

Rank of Controllability matrix = 4
Rank of Observability matrix = 4
```

From the results above we can see the system is controllable and observable in all the velocity

Part 2

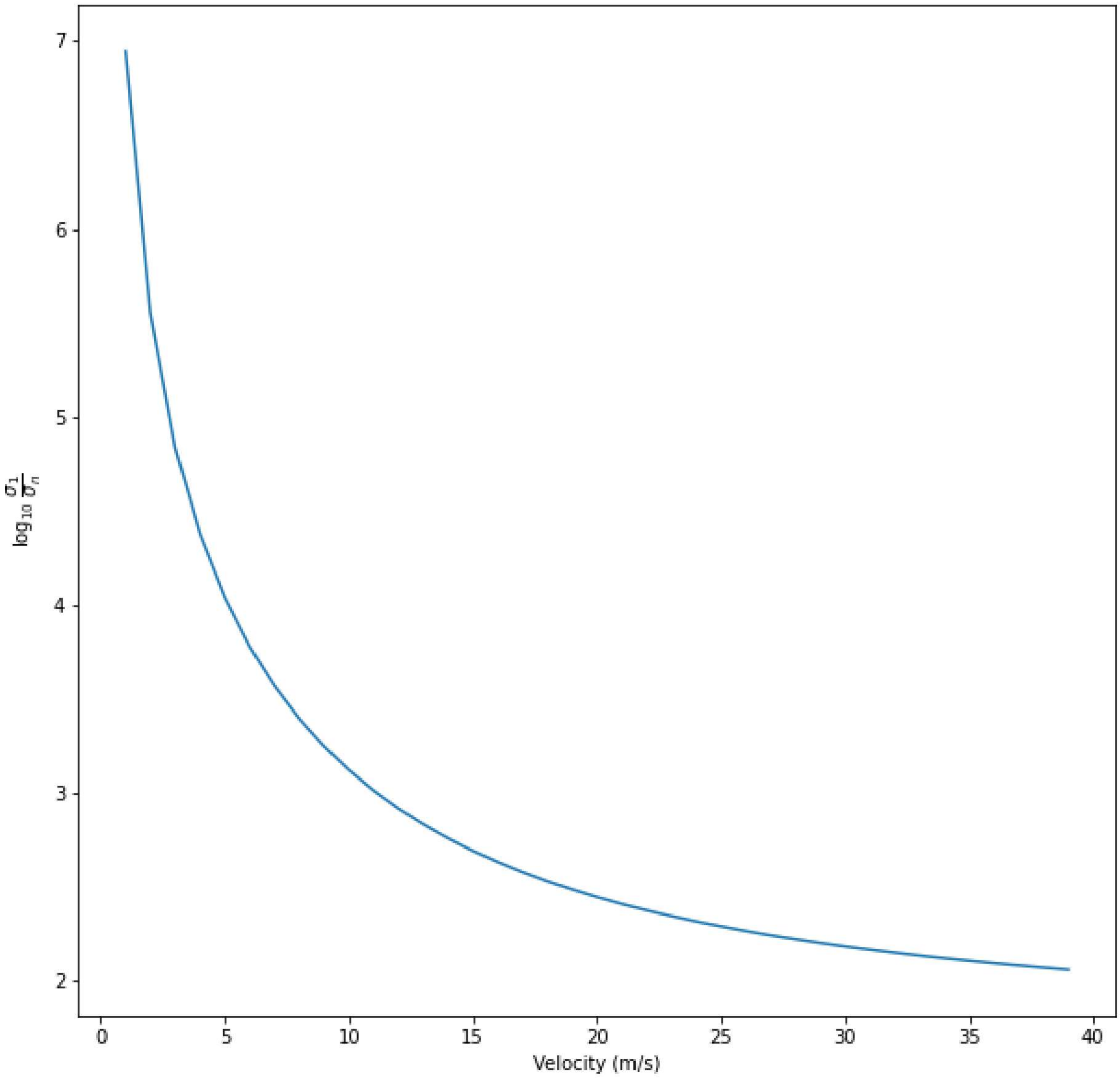
In [18]:

```
pole1 =[]
pole2 =[]
pole3 =[]
pole4 =[]
log =[]
for i in range(1, 40):
    v = i
    A= np.array([[0, 1, 0, 0],
                 [0, -42.359/v, 42.359, -3.388/v ],
                 [0, 0, 0, 1],
                 [0 , -0.2475/v, 0.2475, -6.7/v]])
    B = np.expand_dims(np.array([0, 21.1795, 0, 2.398]).T, axis =1)

    C = np.array([1, 1, 1, 1])
    #C = np.identity(4)

    D = np.array([0])
    #D = np.zeros((4,4))
    P = np.hstack((B, A@B, A@A@B, A@A@A@B))
    #print(P.shape)

    Q = np.vstack((C, C@A, C@A@A, C@A@A@A))

    U, sigma, V = np.linalg.svd(P)
    sigma1 = np.max(sigma)
    sigman = np.min(sigma)
    log.append(np.log10(sigma1 / sigman))
    Eig = np.linalg.eig(A)
    #print(Eig[0])
    #print(Eig[1])
    #S = control.StateSpace(A, B, C, D)
    #poles = control.pole(S)

    pole1.append(Eig[0][0].real)
    pole2.append(Eig[0][1].real)
    pole3.append(Eig[0][2].real)
    pole4.append(Eig[0][3].real)
```

Sigma plot

In [19]:

```
plt.figure(figsize = (10, 10))
plt.plot(np.arange(1, 40), log)
plt.xlabel('Velocity (m/s)')
plt.ylabel('$\log_{10}$ $\dfrac{\sigma_1}{\sigma_n}$')
plt.show()
```



Poles plot

In [20]:

```
fig, ax = plt.subplots(2, 2, figsize=(10,10))


ax[0,0].set_title('pole 1')
ax[0,0].plot(np.arange(1, 40), pole1)



ax[1,0].set_title('pole 2')
ax[1,0].plot(np.arange(1, 40), pole2)


ax[0,1].set_title('pole 3')
ax[0,1].plot(np.arange(1, 40), pole3)

ax[1,1].set_title('pole 4')
ax[1,1].plot(np.arange(1, 40), pole4)
for a in ax.flat:
    a.set(xlabel='Velocity', ylabel='Real part of poles')
```