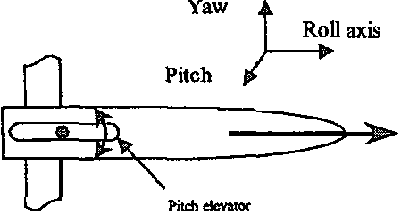
# INFORMATION:

An Unmanned Free-Swimming Submersible (UFSS) vehicle is shown in Figure 1. The depth of the vehicle is controlled as follows. During forward motion, an elevator surface on the vehicle is deflected by a selected amount. This deflection causes the vehicle to rotate about the pitch axis. The pitch of the vehicle creates a vertical force that causes the vehicle to submerge or rise. The pitch control system for the vehicle is used here. The block diagram for the pitch control system is shown in Figure 2*.* In this case study, we investigate the time response of the vehicle dynamics that relate the pitch angle output to the elevator deflection input.



# Diagram Description automatically generatedControl Loop:

# MATLAB requirements:

## Part 2

* Using Laplace transforms, find the analytical expression for the response of the pitch angle to a step input in elevator surface deflection. [Use MATLAB to find the time domain expression by taking inverse Laplace transform].

### Code:

syms s;

z1 = -0.435; p1 = [-1.23 -0.113-1j\*0.0642 -0.113+1j\*0.0642]; k1 = -0.125;

Tf = (-0.125\*(s+0.435)/((s+1.23)\*(s^2+0.226\*s+0.0169)));

% for step response

Tf\_s=(1/s)\*Tf;

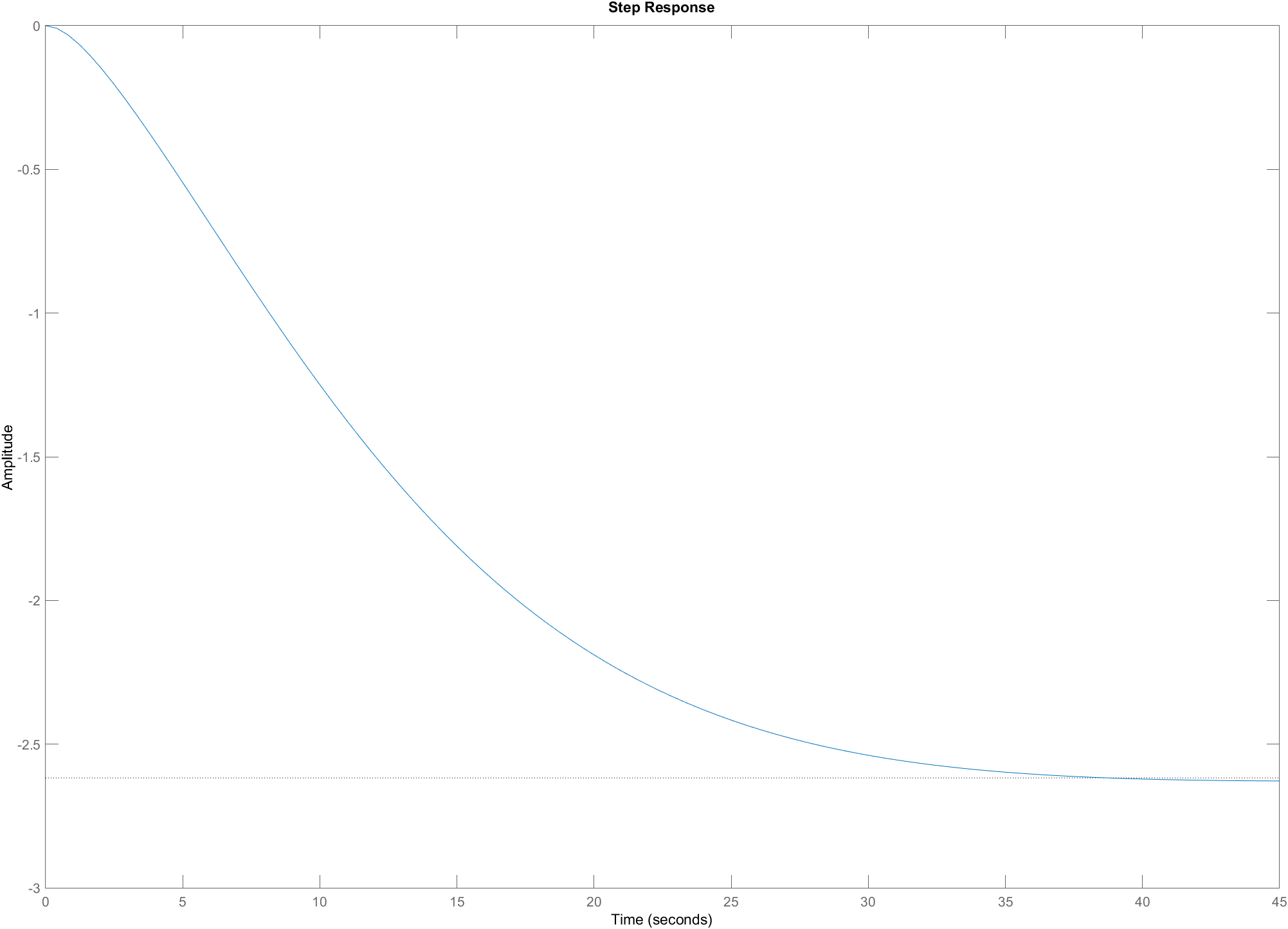
display(Tf\_s);

Inverse\_lap=ilaplace(Tf\_s);

tf = zpk(z1, p1, k1);

step (tf);

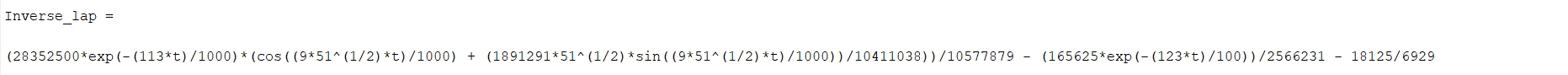
### Plot:



### Inverse Laplace for step response:

(28352500\*exp((113\*t)/1000)\*(cos((9\*51^(1/2)\*t)/1000)+(1891291\*51^(1/2)\*sin((9\*51^(1/2)\*t)/1000))/10411038))/10577879 - (165625\*exp(-(123\*t)/100))/2566231 - 18125/6929

### Screenshot:



## Part 3

* Plot the step response of the system having just two poles [Let’s say there is pole zero cancellation. Maintain the steady state gain of system].

### Code:

Zeros = []; Poles = [-0.113-1j\*0.0642 -0.113+1j\*0.0642]; gain = -0.0442;

Tf = zpk(Zeros, Poles, gain);

step(Tf)

### Plot:

Chart

Description automatically generated with low confidence

## Part 4

* Plot the step response of the system having just two poles and a zero [Let’s say additional pole is cancelled by 5 times rule of thumb but maintain the steady state gain of your system]

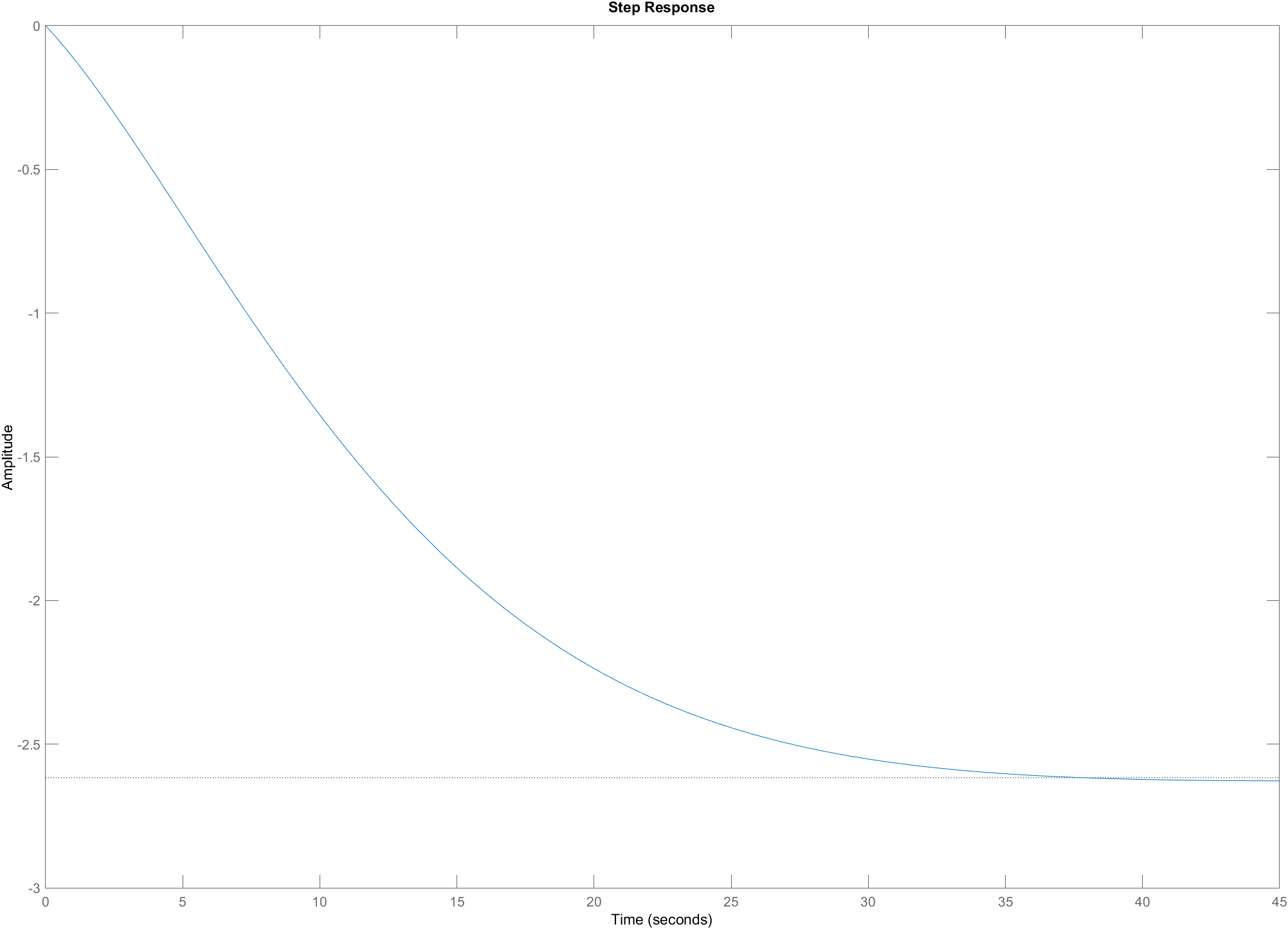
### Code:

Zeros = -0.435;; Poles = [-0.113-1j\*0.0642 -0.113+1j\*0.0642]; gain = -0.1016;

Tf = zpk(Zeros, Poles, gain);

step(Tf)

### Plot:



## Part 5

* Plot the step response of the complete system [having 3 poles and a zero].

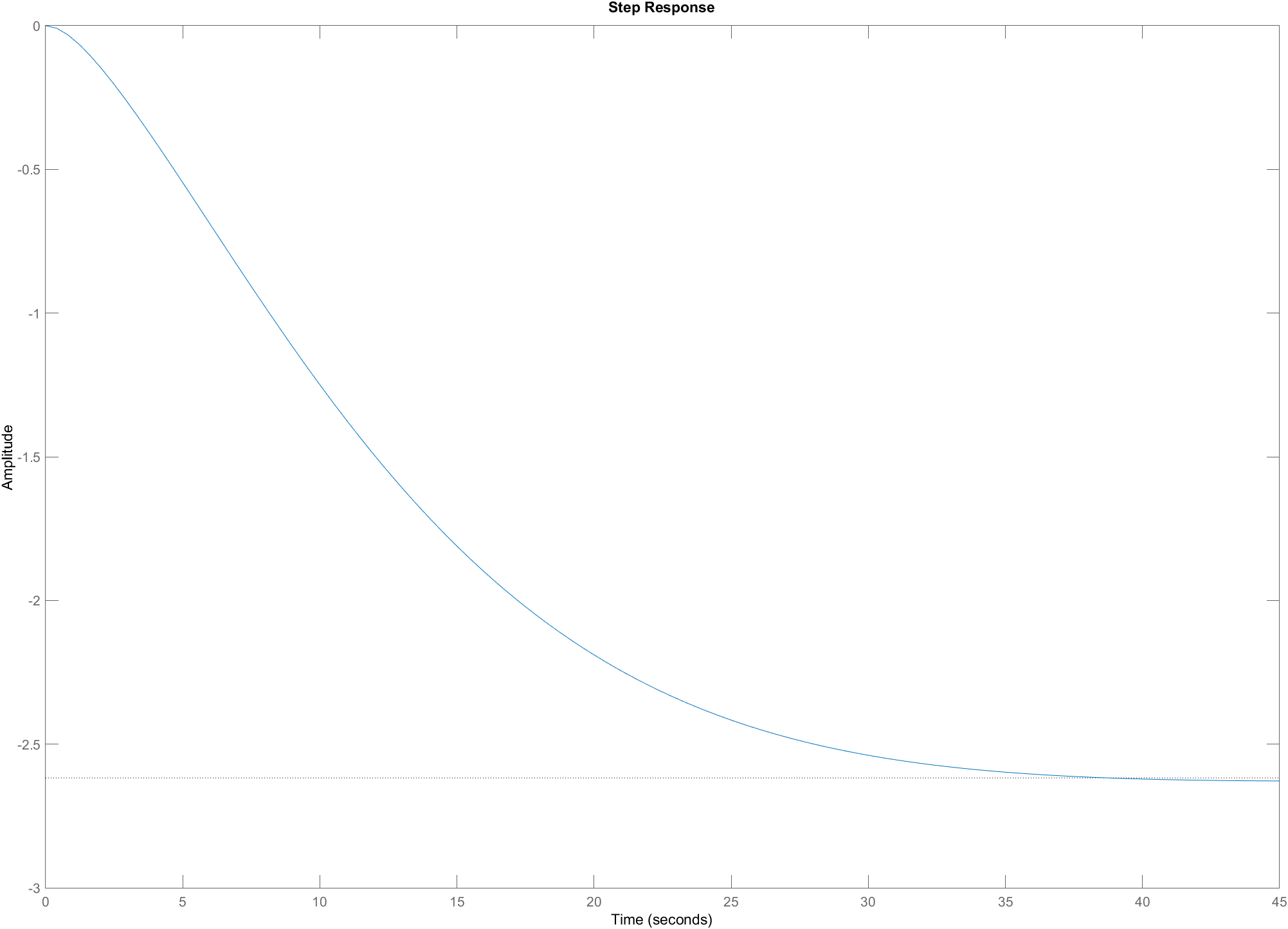
### Code:

Zeros = -0.435;; Poles = [-1.23 -0.113-1j\*0.0642 -0.113+1j\*0.0642]; gain = -0.125;

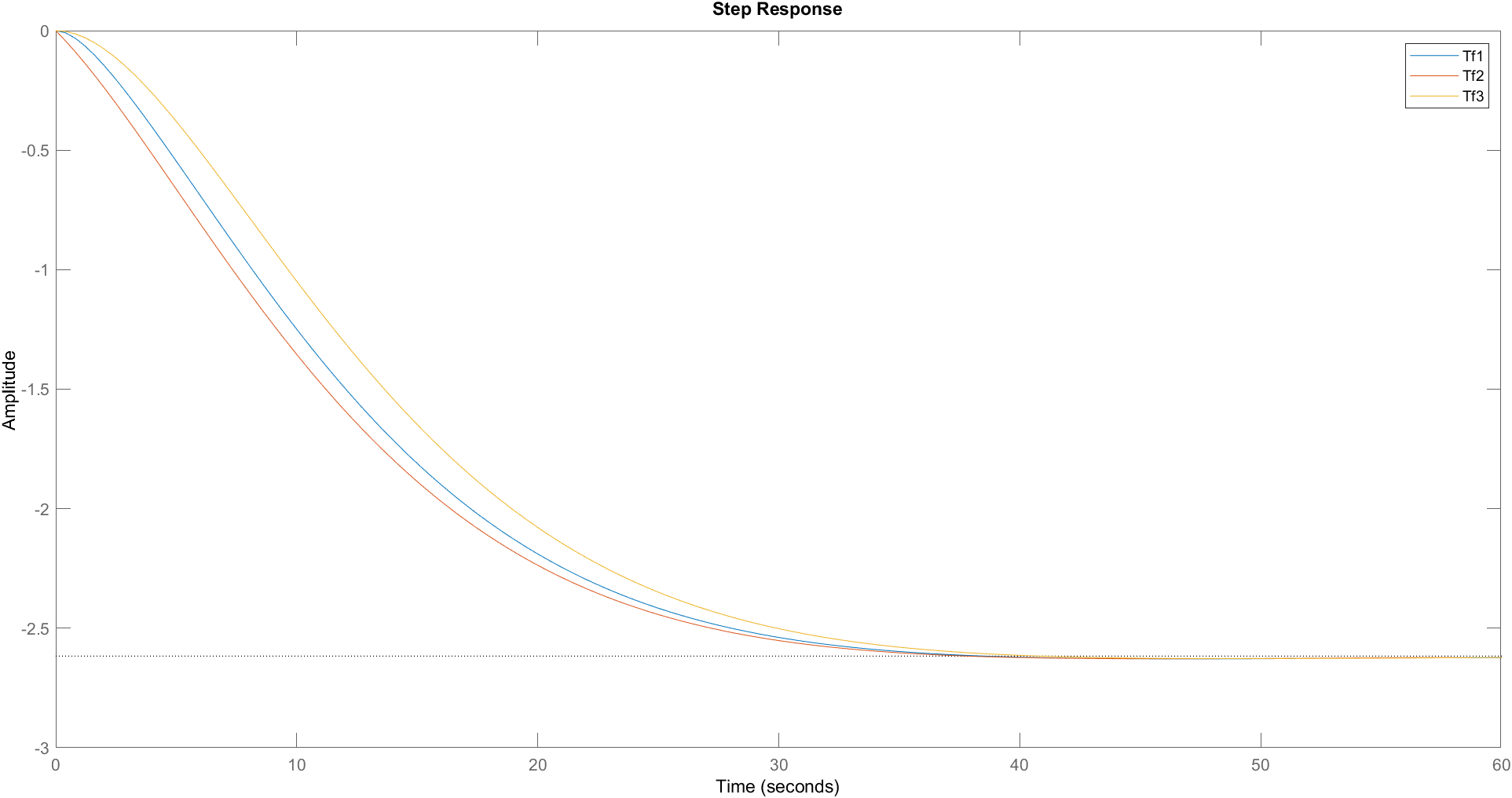
Tf = zpk(Zeros, Poles, gain);

step(Tf)

### Plot:



Plotting all of them on the same graph:



### Code:

Zer1 = [-0.435]; Poles1= [-1.23 -0.113-1j\*0.0642 -0.113+1j\*0.0642]; gain1 = -0.125;

Zer2 = [-0.435]; Poles2= [-0.113-1j\*0.0642 -0.113+1j\*0.0642]; gain2 = -0.1016;

Zer3 = []; Poles3= [-0.113-1j\*0.0642 -0.113+1j\*0.0642]; gain3 = -0.0442;

Tf1 = zpk(Zer1, Poles1, gain1);

Tf2 = zpk(Zer2, Poles2, gain2);

Tf3 = zpk(Zer3, Poles3, gain3);

figure

hold on;

step(Tf1);

step(Tf2);

step(Tf3);

legend('Tf1', 'Tf2', 'Tf3')

% for information regarding the responses:

display(stepinfo(Tf1));

display(stepinfo(Tf2));

display(stepinfo(Tf3));

## part 6

* Compare the responses in just 2 to 3 lines.

**From MATLAB we observe that**

We observe that the addition of pole far from imaginary axis slows down the response so the settling time of transfer function with two poles is less that settling time of transfer function with 3 poles. Moreover, addition of zero near the vicinity of imaginary axis increases the overshoot and decreases the response time thus transfer function 3 has more settling time that the above two. Addition of zero increased the overshoot and addition of pole increased the rise time of the systems.

Text

Description automatically generated with medium confidenceText

Description automatically generated with medium confidenceA picture containing table

Description automatically generated