**Birzeit University**
**Faculty of Engineering & Techonology**
**Department of Electrical & Computer Engineering**
**ENEE4302-Control Systems**

# "UFSS Control"

**Student :Mohamad Emad Bornat**
**#1130842**

**Instructor:Dr.Jamal Siam**

**December,2017.**

# *Abstract*

This project aims to investigate the time response of the UFSS vehicle dynamics that relate the pitch angle output to the elevator deflection input.

# Contents

**Acronyms and Abbreviations** ........ **iv**

**List of Figures** ........ **v**

**Chapter 1 Introduction** ........ **1**

**Chapter 2 System Simulation** ........ **2**

2.1 Determining the Transfer Function of the system (k1=k2=10) ........ 2

2.2 Root Locus of the System & Stability ........ 3

2.3 Step Response & Transient Parameters of the system ........ 4

2.3.1 Stable Oscillatory ........ 4

2.3.2 Stable Non-Oscillatory ........ 5

2.3.3 Unstable ........ 6

2.3.4 Limit of Stability ........ 7

2.4 Steady State Error of the system ........ 8

2.5 Observer State Space Represenation of the feedback system ........ 8

**Conclusion** ........ **9**

**References** ........ **10**

**Appendices** ........ **11**

# Acronyms and Abbreviations

| | |
|---|---|
| UFSS | Unmaned Free Swimming Submarine |
| TF | Transfer Function |
| SSE | Steady State Error |

# List of Figures

*Fig.* 1.1: Block Diagram of UFSS using PD controller ........ 1

*Fig.* 2.1: UFSS Simulink Model using PD controller ........ 2

*Fig.* 2.2: UFSS Root Locus ........ 3

*Fig.* 2.3: Step Response for K=40 ........ 4

*Fig.* 2.4: Step Response for K=3 ........ 5

*Fig.* 2.5: Step Response for K=60 ........ 6

*Fig.* 2.6: Step Response for K=51.4 ........ 7

# Chapter 1

# Introduction

The pitch control contains the two controllers, which are gain (-K) and pitch rate sensor. Basically the sensor is a differentiator in which pitch rate is produced by pitch magnitudes. It is characterized by a transfer function. In this project, we analysed the system performance approach towards exception for the system response we used pitch rate sensor and here system properties can be changed by changing the values of K and Ks, and in this case we noticed that the system response is acceptable.

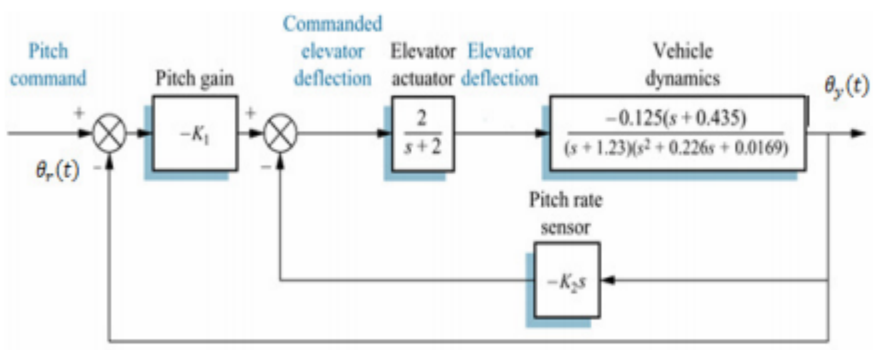


*Fig.* 1.1: Block Diagram of UFSS using PD controller

.

# Chapter 2

# System Simulation

## 2.1 Determining the Transfer Function of the system (k1=k2=10)

Using Simulink, transfer function can be extracted as follows:

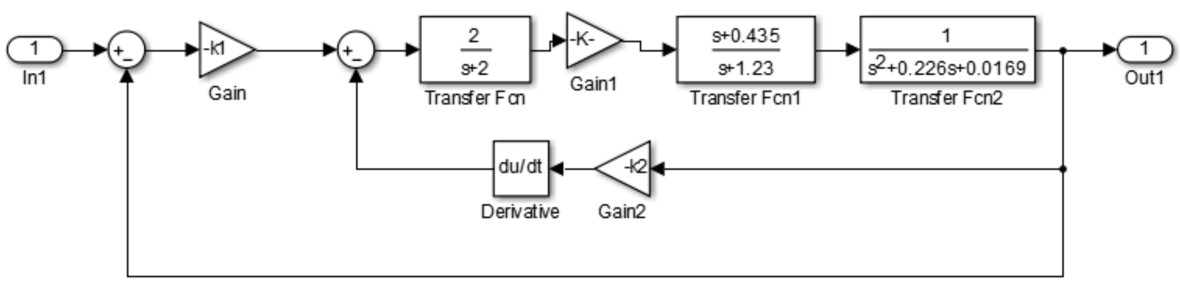


*Fig.* 2.1: UFSS Simulink Model using PD controller

We use the following Matlab Command to get the Closed loop TF :

```
[A,B,C,D]=linmod('control');
[num,den]=ss2tf(A,B,C,D) ; G=tf(num,den);
```

Then :

```
G =

          2.5 s + 1.088
  ---------------------------------------------
  s^4 + 3.456 s^3 + 3.207 s^2 + 3.111 s + 1.129
```

Using Math formulation , we use code (A.1) then TF is :

Ge =

$$\frac{2.5\ s + 1.087}{s^4 + 3.456\ s^3 + 5.707\ s^2 + 4.198\ s + 1.129}$$

## 2.2 Root Locus of the System & Stability

- Using rlocus command for the open-loop TF , the root locus of the system is(K1=1,K2=10) :

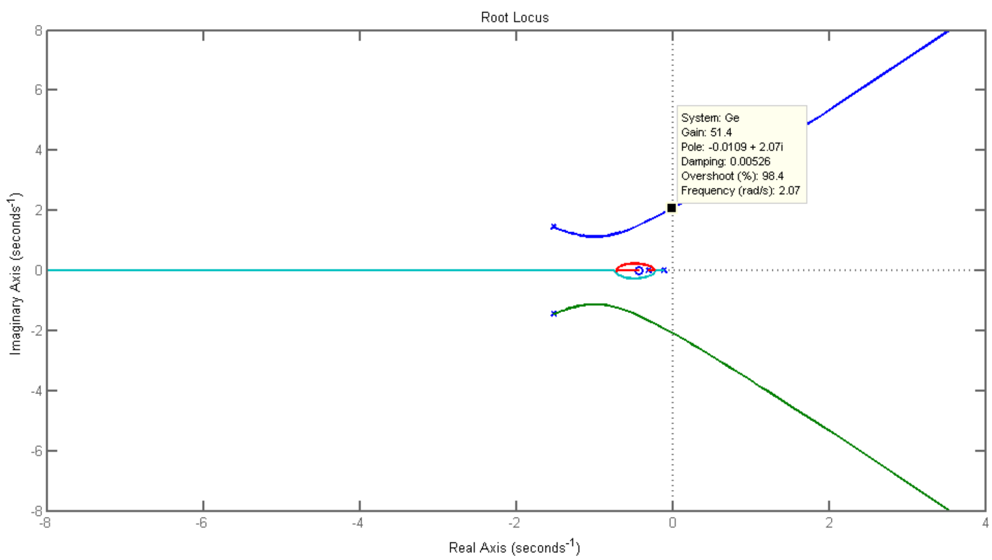


*Fig.* 2.2: UFSS Root Locus

- From the figure we notice that the system is stable if (0<K<51.4).

Then K1 become 51.4.

- For K1=K2=10, we use the MATLAB Command isstable(Ge)

ans =1 , then the system is stable.

## 2.3 Step Response & Transient Parameters of the system

We use Matlab Commands :

```
step(Ge)
s=stepinfo(Ge)
```

### 2.3.1 Stable Oscillatory

Let K1 =40

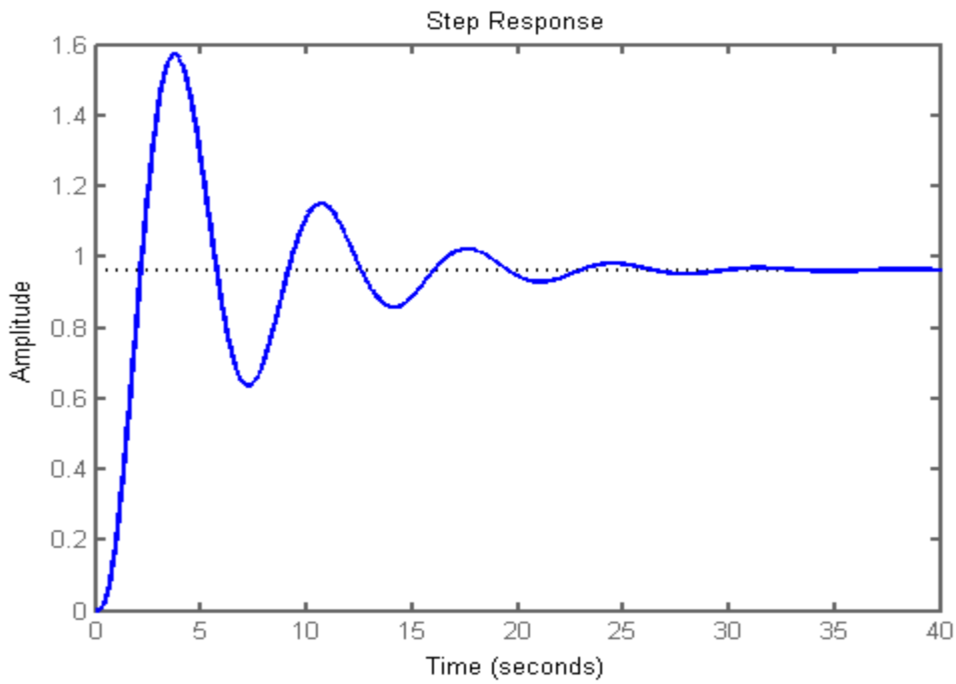


*Fig.* 2.3: Step Response for K=40

We notice the system is stable with oscillations.

RiseTime: 0.6885
SettlingTime: 25.6331
SettlingMin: 0.4796
SettlingMax: 1.6643
Overshoot: 68.0213
Undershoot: 0
Peak: 1.6643
PeakTime: 1.9973

### 2.3.2 Stable Non-Oscillatory

Let K1 =3

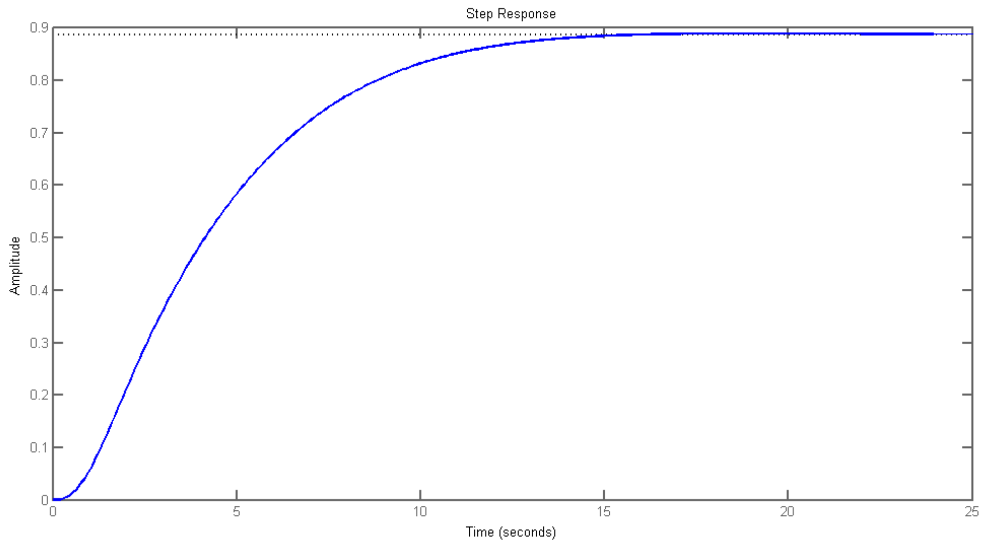


*Fig.* 2.4: Step Response for K=3

We notice the system is stable without oscillations.

RiseTime: 7.5077

SettlingTime: 12.3354

SettlingMin: 0.7997

SettlingMax: 0.8901

Overshoot: 0.3526

Undershoot: 0

Peak: 0.8901

PeakTime: 19.0154

### 2.3.3 Unstable

Let K1 =60

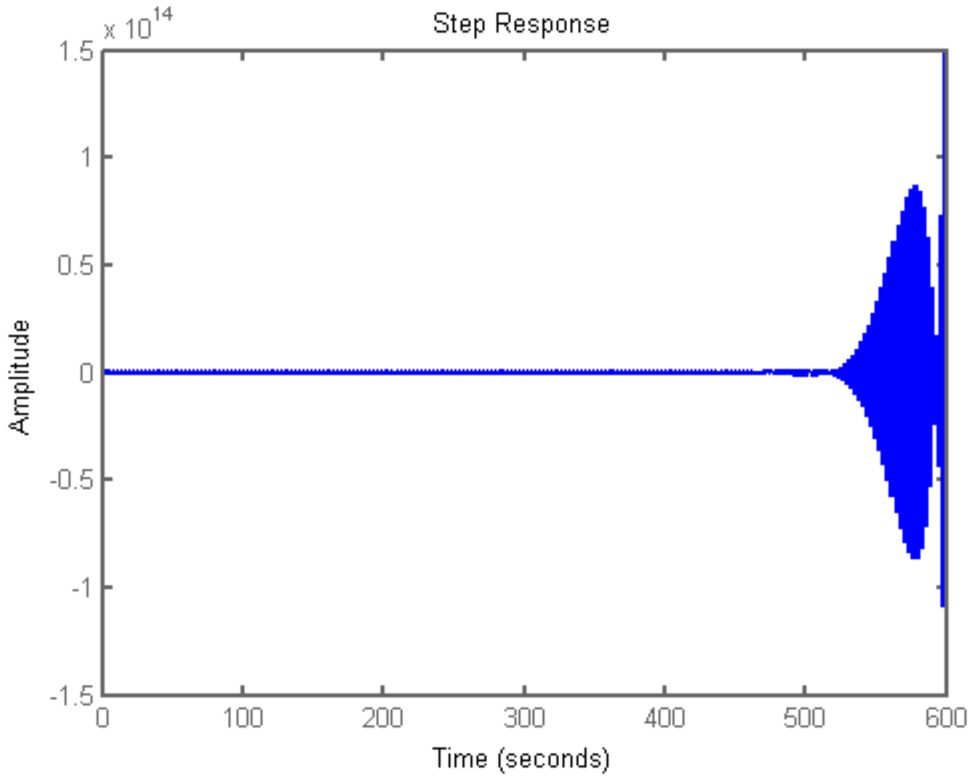


*Fig.* 2.5: Step Response for K=60

Transient Parameters : undefined Values,.
We notice the system is unstable.

### 2.3.4 Limit of Stability

Let K1 =51.4

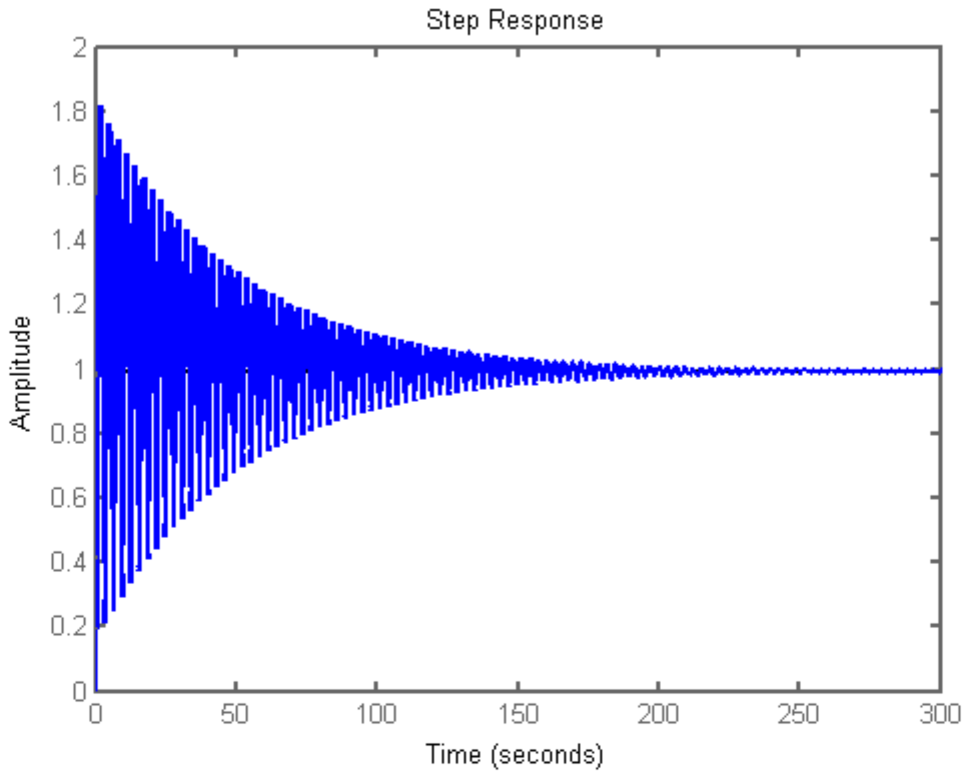


*Fig.* 2.6: Step Response for K=51.4

We notice the system is stable wit hight overshoot.

RiseTime: 0.6019
SettlingTime: 186.4673
SettlingMin: 0.2130
SettlingMax: 1.8154
Overshoot: 82.8940
Undershoot: 0
Peak: 1.8154
PeakTime: 1.8303

## 2.4 Steady State Error of the system

We find the static error constants that is used to find steady state error by using the following MATLAB Comand : dcgain(TF)

The following results were found:

kp = 26.1582 ,ep = 0.0368.
Kv = 0,ev =  Inf.
ka =   0,ea =Inf.

## 2.5 Observer State Space Represenation  of the feedback system

To find the observer state-space representation of the feedback system, code A.2 was used , then the following results were obtained :

```
Y1 =

   -3.4560    1.0000         0         0
   -5.7070         0    1.0000         0
   -4.1980         0         0    1.0000
   -1.1290         0         0         0


Y2 =

     1     0     0     0


Y3 =

         0
         0
    2.5000
    1.8700
```

# Conclusion

In this project we selected a closed loop (unmanned free-swimming submersible vehicle) system whose pitch is to be controlled by using Simulink. Initially, we used PD controller to control its pitch response. Therefore, adjusting the values of Kp and Kd, we approached to a fine step response of system.

# References

[1] Beards C.F., Vibrations and Control System. Ellis Horwood, 1988.

[2]Bennett S., A History of Control Engineering. 1800-1930. IET. 142−148, June 1986.

# Appendices

## A.1

```
numg1=[-10];
deng1=[1];
numg2=[0 2];
deng2=[1 2];
numg3=-0.125*[1 0.435];
deng3=conv([1 1.23],[1 0.226 0.0169]);

numh1=[-10 0];
denh1=[0 1];
G1=tf(numg1,deng1);

G2=tf(numg2,deng2);

G3=tf(numg3,deng3);

H1=tf(numh1,denh1);

G4=series(G2,G3);

G5=feedback(G4,H1);

G6=series(G1,G5);

Ge=feedback(G6,1);
```

## A.2

```
N=[2.5 1.87];
D=[1 3.456 5.707 4.198 1.129];
[A B C D]=tf2ss(N,D);
Y1=transpose(A)
Y2=transpose(B)
Y3=transpose(C)
```