

# Automatic Flight Control System Design

## Practical

4277856 Jeanette Derks

4922425 João Ferreira

4938054 Kevin-Christian Garzon Galindo

Lecturer: Dr. ir. E. van Kampen



# Contents

<b>1</b>	<b>Introduction</b>	<b>1</b>
<b>2</b>	<b>Trim and Linearization</b>	<b>2</b>
2.1	Trim and Linearization Results and Analysis . . . . .	2
2.1.1	Proposed Flight Conditions . . . . .	2
2.2	Influence of the accelerometer position . . . . .	3
2.2.1	Normal Acceleration Response when $x_a = 0ft$ . . . . .	3
2.2.2	Comparison of Normal Acceleration Response for different $x_a$ values . . . . .	6
<b>3</b>	<b>Open Loop Analysis</b>	<b>8</b>
3.1	State Space Model Reduction . . . . .	8
3.1.1	Reduced Matrices for Longitudinal Direction . . . . .	8
3.1.2	Reduced Matrices for Lateral Direction . . . . .	9
3.2	Calculation of the Inherent Motion Characteristics . . . . .	10
3.2.1	Periodic Eigenmotions . . . . .	10
3.2.2	Non-Periodic Eigenmotions . . . . .	10
3.2.3	Open Loop Time Response and Analysis . . . . .	10
<b>4</b>	<b>Design of a Pitch Rate Command System Satisfying CAP/Gibson Mil-Specs</b>	<b>14</b>
4.1	Reduced Model for the Short Period . . . . .	14
4.2	Design of the Pitch Rate Command System . . . . .	16
4.2.1	Frequency Domain Conversion for the CAP and Gibson criteria . . . . .	16
4.2.2	Method to obtain the Pitch Damper . . . . .	16
4.2.3	Validation of the Feedback Gain Matrix Considering Severe Gusts . . . . .	17
4.3	Time Constant Modification and Lead-Lag Prefilter Location . . . . .	17
4.4	CAP and Gibson Criteria Validation . . . . .	18
<b>5</b>	<b>Design of a Terrain Following System</b>	<b>20</b>
5.1	Trim and Linearization . . . . .	20
5.2	Reduced Model for the Given Flight Condition . . . . .	20
5.3	Definition of the Aircraft Model and Actuators Dynamics . . . . .	21
5.3.1	Actuators Dynamics in Simulink . . . . .	21
5.4	Design of an LQR Controller . . . . .	22
5.4.1	Complete Block Diagram for the Terrain Following System . . . . .	22
5.4.2	Method to Find Q and R Matrix . . . . .	22
5.5	LQR Controller Optimization for Pilot Comfort . . . . .	23
5.5.1	System Behaviour Analysis for varying values of matrix Q . . . . .	23
5.5.2	System Behaviour Analysis for varying values of matrix R . . . . .	26
5.6	Final Result for the LQR Controller . . . . .	27
<b>6</b>	<b>Last Conclusions</b>	<b>29</b>
	<b>Bibliography</b>	<b>30</b>

# 1

## Introduction

This report is the result of the additional practical of the course Automatic Flight Control System Design. The goal of this practical was to become familiar with classical flight controllers and to gain insight in handling qualities of open-loop and controlled aircraft.

The structure of this report follows the structure of the exercises in the assignment. In chapter 2 the trim and linearization of the Lockheed Martin F-16 is made for the proposed flight conditions. In chapter 3 the open loop analysis of the aircraft is presented and in chapter 4 a design of a pitch rate command system is made, which satisfies the CAP and Gibson criteria. Eventually, in the last chapter, chapter 5, a design of a terrain following control system is for the F-16, which is used by pilots when they are flying at low altitude.

The data used in this assignment is mainly obtained by running the *FindF16Dynamics.m* MATLAB script. The results for each different flight condition is then saved in MATLAB files, and loaded to the corresponding script, where it is used.

# 2

## Trim and Linearization

In this chapter, the importance of the trimming and linearization of a Lockheed Martin F-16 model around a certain flight condition is discussed. They are then implemented on that model for different flight conditions and the results are discussed. Finally, in the end of this chapter, the influence of the accelerometer position in the aircraft is studied. For the accelerometer part the Matlab file *accelerometer.m* is used.

### 2.1. Trim and Linearization Results and Analysis

The flight dynamics of any aircraft is represented by non-linear functions. Because it can be hard to analyse these kind of functions, a linearization is usually made around a certain point.

The choice of this point is very important. If a non steady-state flight condition is chosen, the model will naturally tend to go away from that flight condition, so it is not very accurate. On the other side, if one chooses an equilibrium (steady-state) flight condition, the model will stay close to this condition while the input is zero, making it more useful.

Because of this, the linearizations in this assignment are made around a steady-state point, which shall be determined first by means of a trimming to the model in a given altitude and velocity. Trim is achieved by changing the thrust value, the deflection of control devices - elevator, ailerons and rudder - and the angle of attack, starting from a certain initial guess.

To know when the values obtained are near enough to equilibrium, the minimization of a cost function is made. The cost function is given by the following equation:

$$J = \sum_{i=1}^n W_i \dot{x}_i^2 \quad (2.1)$$

Where  $x$  is the vector that includes all velocities [3]. That means that the cost function contains a weighted measure of the accelerations, so the smaller its value is, the closer the approximation will be from the real equilibrium state.

#### 2.1.1. Proposed Flight Conditions

Three different flight conditions are used in this assignment, and the file *FindF16Dynamics.m* is run to find the results of the trim and linearization for them. The results for each flight condition are saved in MATLAB **Data Files** and loaded to the respective script.

- **20000ft** altitude and **600 ft/s** velocity - used to **design** a pitch rate command system. Trimming results are shown on Table 2.1.
- **15000 ft** altitude and **500 ft/s** velocity - used to **discuss** the influence of the accelerometer position. Trimming results are shown on Table 2.2.
- **5000 ft** altitude and **300 ft/s** velocity - used to **design** a terrain following system. Trimming results for this condition are shown later, on Table 5.1 of Chapter 5.

	<b>Low Fidelity Model</b>	<b>High Fidelity Model</b>
<b>Cost Function</b>	$1.8443 \times 10^{-29}$	$4.4774 \times 10^{-6}$
<b>Thrust (lb.)</b>	2085.0722	2085.0498
<b>Elevator (deg.)</b>	-2.2057	-1.8735
<b>Aileron (deg.)</b>	$-6.6264 \times 10^{-16}$	-0.084144
<b>Rudder (deg.)</b>	$4.1323 \times 10^{-15}$	0.049982
<b>Alpha (deg.)</b>	3.4044	3.4212

Table 2.1: Results and comparison between the High Fidelity and Low Fidelity models for the 20000ft and 600ft/s flight condition.

	<b>Low Fidelity Model</b>	<b>High Fidelity Model</b>
<b>Cost Function</b>	$2.2906 \times 10^{-29}$	$7.1856 \times 10^{-6}$
<b>Thrust (lb.)</b>	2120.6214	2109.4129
<b>Elevator (deg.)</b>	-2.4607	-2.2441
<b>Aileron (deg.)</b>	$3.2114 \times 10^{-16}$	-0.093578
<b>Rudder (deg.)</b>	$2.7966 \times 10^{-15}$	0.094469
<b>Alpha (deg.)</b>	4.4655	4.5307

Table 2.2: Results and comparison between the High Fidelity and Low Fidelity models for the 15000ft and 500ft/s flight condition.

The trim is considered valid if the cost values for the low and high fidelity models are, respectively, on the order of  $10^{-29}$  and  $10^{-6}$  [1]. By looking at the results, it is possible to conclude that for the above two flight conditions, the cost functions are within the acceptable range. A trim is then achieved for each flight condition, from which Matlab automatically computes the linearized matrices for both the high fidelity and low fidelity models. However, only the low fidelity model is used throughout the assignment.

## 2.2. Influence of the accelerometer position

As mentioned before, the **15000 ft** and **500 ft/s** flight condition is used in this section. To test the influence of the accelerometer position, a new output is first added to the non-linearized system (*LIN\_F16BLOCK*). This output is the normal acceleration  $a_n$  which is given by (in g-units):

$$a_n = n_z + \frac{\dot{q}x_a}{g_D} \quad (2.2)$$

Where  $n_z$  and  $\dot{q}$  are state variables and  $g_D = 9.80665m/s^2$  is the standard gravity value.

The new block diagram is given in Figure 2.1. The new sub-system is called *Acceleration Block* and it is shown in detail in Figure 2.2.

### 2.2.1. Normal Acceleration Response when $x_a = 0ft$

The new F16 model is trimmed for the flight condition mentioned above and the output equation for the normal acceleration with  $x_a$  equals zero is given by:

$$a_n = [-3.2432 \times 10^{-5} \quad -9.6797 \times 10^{-6} \quad 0.004 \quad 9.9298 \quad 0.9664 \quad 0.0208] \begin{bmatrix} h \\ \theta \\ V_t \\ \alpha \\ n_y \\ \delta_e \end{bmatrix} \quad (2.3)$$

By looking at Equation 2.3 it is clear that the normal acceleration depends on the altitude, pitch angle, velocity, angle of attack, load factor along the y-axis and the elevator deflection. The D matrix is equal to zero. It should be noted that the contributions of  $h$  and  $\theta$  are very close to zero.  $\alpha$ , however, has a big influence in the normal acceleration, which makes sense, since a change in the angle of attack changes the flight path angle and, thus, may also change the normal acceleration.

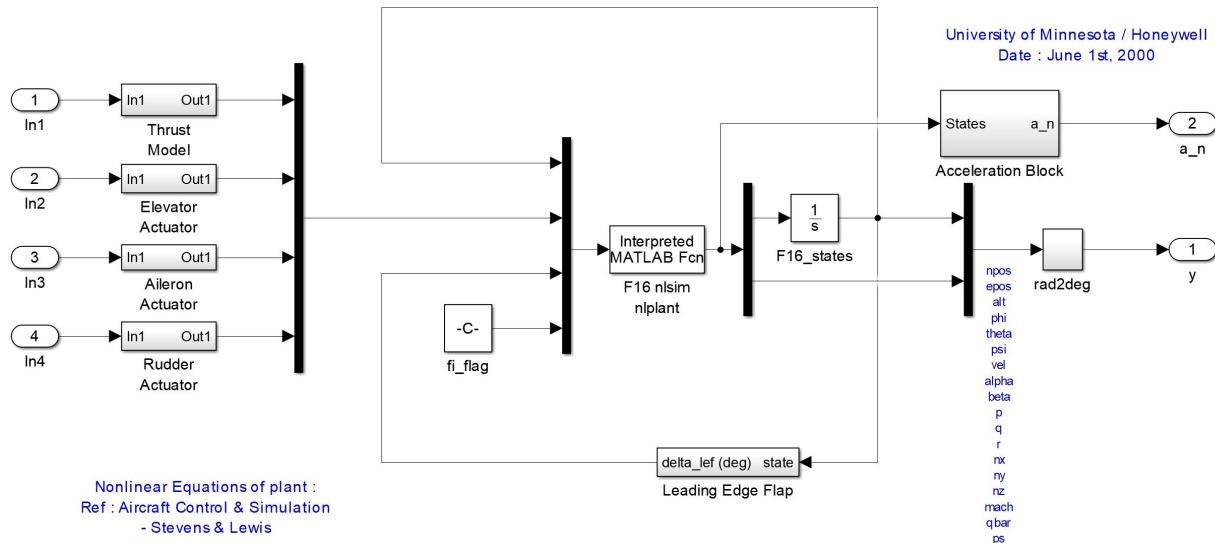


Figure 2.1: New block diagram, that includes the Acceleration Block on the upper-right side.

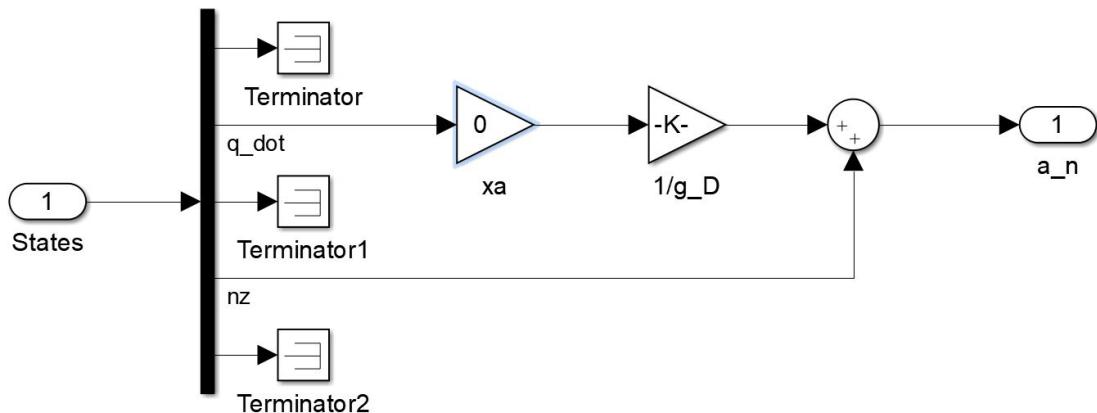


Figure 2.2: Detailed view of the Acceleration Block.

The transfer function of the elevator-to-normal-acceleration is given in Equation 2.4. To create these TF, only the variables that influence the normal acceleration were used. The terms in  $s^0$  for both the numerator and the denominator of the transfer function can also be neglected, which means that a pole-zero cancellation at the origin is possible. A simplified transfer function is given in Equation 2.5. However, the presented results were obtained with Equation 2.4 and not with the simplified version.

$$\frac{a_n}{\delta_e} = \frac{0.421s^5 - 1.84s^4 - 22.15s^3 + 0.077s^2 + 0.001231s + 3.953 \times 10^{-7}}{s^6 + 21.73s^5 + 33s^4 + 41.48s^3 + 0.5546s^2 + 0.293s - 2.844 \times 10^{-14}} \quad (2.4)$$

$$\frac{a_n}{\delta_e} = \frac{0.421s^4 - 1.84s^3 - 22.15s^2 + 0.077s + 0.001231}{s^5 + 21.73s^4 + 33s^3 + 41.48s^2 + 0.5546s + 0.293} \quad (2.5)$$

The normal acceleration response to a negative step elevation command is shown in Figure 2.3. Figure 2.4, which is a detailed view of that plot, shows that the system is a non-minimum

phase because the initial response goes the opposite direction of the reference signal. To explain why this happens, the transfer function is split into its zeros in Equation 2.6.

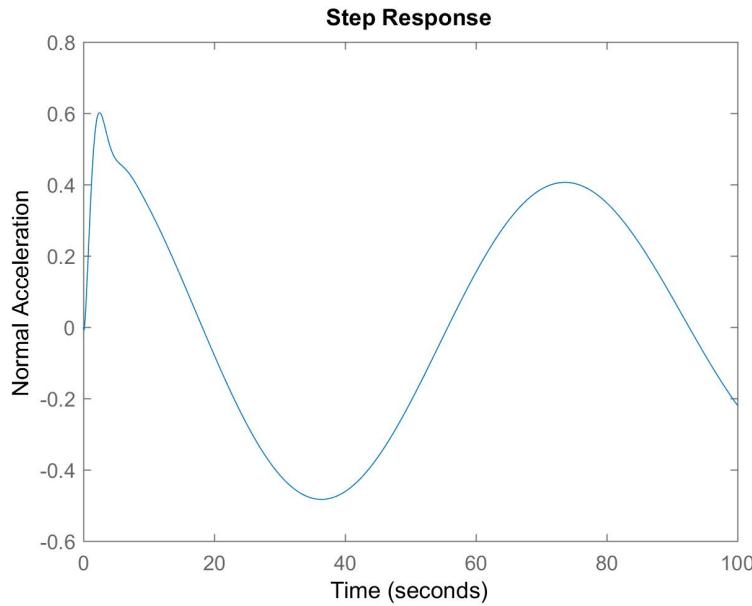


Figure 2.3: Normal Acceleration response to the negative deflection of the elevator.

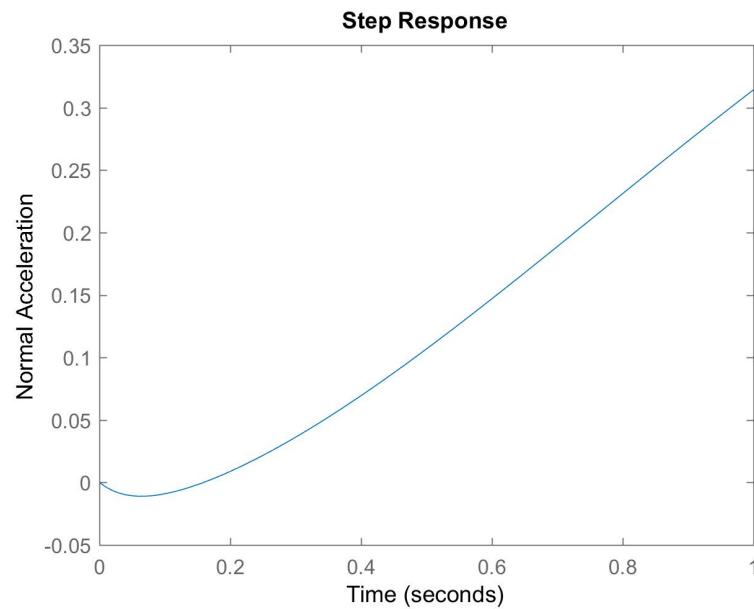


Figure 2.4: Detailed view of the time response of the Normal Acceleration response to a negative deflection of the elevator.

$$\frac{a_n}{\delta_e} = \frac{0.421s(s - 9.7597)(s + 5.3925)(s - 0.0094)(s + 0.0059)}{s^6 + 21.73s^5 + 33s^4 + 41.48s^3 + 0.5546s^2 + 0.293s - 2.844 \times 10^{-14}} \quad (2.6)$$

Looking at the previous equation, it is clear that two of the zeros are located on the open left-half plane. The terms  $(s - 9.7597)$  and  $(s - 0.0094)$  will, in theory, force the system to go the wrong

way on the first few seconds of the response. However one of them is very close to the origin, so this zero will not contribute to the non-minimum phase behaviour. Only the  $(s - 9.7597)$  term will actually contribute to this response.

The phenomenon can be physically explained by the fact that when the elevator is deflected upwards the aircraft will pitch up and its centre of gravity will accelerate backwards at first, because it is behind the point around which the aircraft is rotating - the instantaneous centre of rotation. This causes the negative normal acceleration that is seen in the time response.

### 2.2.2. Comparison of Normal Acceleration Response for different $x_a$ values

Figures 2.5 to 2.10 on the next page show the normal acceleration response when the distance to the aircraft centre of gravity is changed. The zeros of the transfer function for each location are shown below.

$x_a$	Zero	Zero	Zero	Zero	Zero
<b>0 ft</b>	9.7597	-5.3925	0.0094	-0.0059	0.0000
<b>5 ft</b>	40.6734	-8.4873	0.0094	-0.0059	0.0000
<b>5.9 ft</b>	-67141.5466	-10.5188	0.0094	-0.0059	0.0000
<b>6 ft</b>	-283.7181	-10.8977	0.0094	-0.0059	0.0000
<b>7 ft</b>	-13.7349 + 9.6727i	-13.7349 - 9.6727i	0.0094	-0.0059	0.0000
<b>15 ft</b>	-1.9412 + 5.5105i	-1.9412 - 5.5105i	0.0094	-0.0059	0.0000

Table 2.3: Pole Locations for all the distances to the aircraft centre of gravity.

A first analysis to the previous Table shows that only two of the zeros move when  $x_a$  is changed. Plus, one of the zeros eventually shifts to the open left-half plane and for  $x_a = 7\text{ft}$  and  $x_a = 15\text{ft}$  these two zeros are coupled and form a pair of complex poles.

The instantaneous centre of rotation (ICR) is, by definition, the point of a body that has zero velocity at a particular moment in time. That is, at that moment, the body is rotating around the ICR. In the case of a negative elevator deflection in an aircraft (nose up), this definition means that, when the aircraft starts pitching up, the normal acceleration is zero and will tend to remain with that value at the ICR (assuming an initial steady state flight).

Taking this into account, it is possible to find the approximate position of the instantaneous centre of rotation by examination of the time responses for different  $x_a$ . For  $x_a = 0\text{ft}$  and  $x_a = 5\text{ft}$ , the system is a non-minimum phase. As explained before, this means that both points start immediately rotating in the wrong direction, and only after a short time will accelerate in the desired one. So, these two points cannot be the ICR. For  $x_a = 7\text{ft}$  and  $x_a = 15\text{ft}$ , the acceleration immediately after  $t = 0\text{s}$  increases (its derivative is positive), so they are already accelerating in the desired direction. These two points cannot be the ICR as well. In fact, because their acceleration has the opposite sign of the first two points, it is possible to conclude that the ICR is somewhere in between these two pairs of points.

However, the plots for  $x_a = 5.9\text{ft}$  and  $x_a = 6\text{ft}$  show that the derivative of the normal acceleration when  $t = 0\text{s}$  is approximately zero, so for both locations no normal acceleration is initially being felt and thus the two points are barely rotating at first. Further analysis of the zeros location show that, for  $x_a = 5.9\text{ft}$  the zero that forces the system to be non-minimum phase shifts to the open left-half plane, so this point is the first in which no non-minimum phase effects are registered.

The two reasons presented on the last paragraph are enough evidence to conclude that the instantaneous centre of rotation should be somewhere near the locations  $x_a = 5.9\text{ft}$  and  $x_a = 6\text{ft}$ .

The pilot's station should be installed in a location that allows a reliable reading of the normal acceleration. Let's assume that the accelerometer is too far from the instantaneous centre of rotation. Then, if the pilot tries to pitch up, the sensor will measure a relatively big initial acceleration (could be in the desired direction or not, depending on which side of the ICR the accelerometer is). As a consequence, the pilot will try to correct for this "false acceleration" and, in the end, the aircraft will not exactly respond as it should. This fact increases the pilot

workload and degrades the handling qualities. So, the wrong installation of the pilot's station may ultimately lead to a poor control of the aircraft, which can be specially dangerous in some flight phases, like the landing, where a big precision and smooth control are needed. The best possible scenario would be to have the station installed at the ICR, as this would lead to a precise reading of the normal acceleration.

It is also important to place the accelerometer close to a node of the most important fuselage bending mode because that avoids the coupling between the structural oscillations and the rigid body control system.

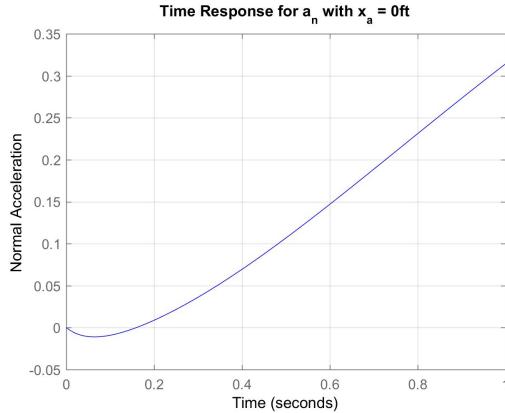


Figure 2.5: Time Response when  $x_a = 0\text{ft}$ .

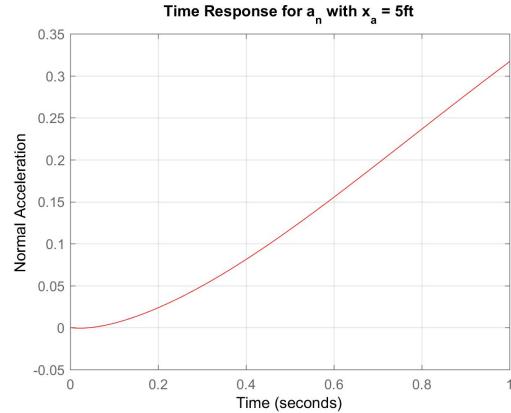


Figure 2.6: Time Response when  $x_a = 5\text{ft}$ .

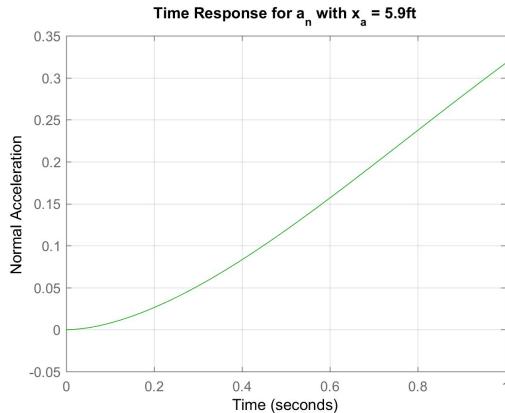


Figure 2.7: Time Response when  $x_a = 5.9\text{ft}$ .

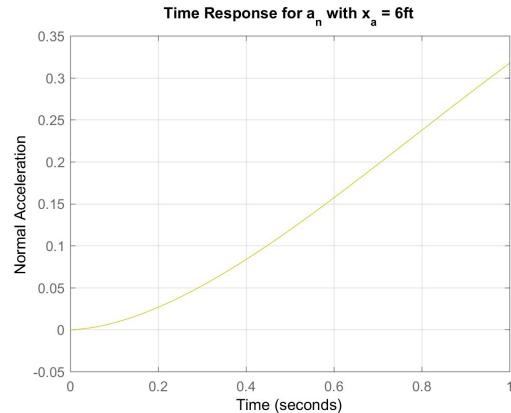


Figure 2.8: Time Response when  $x_a = 6\text{ft}$ .

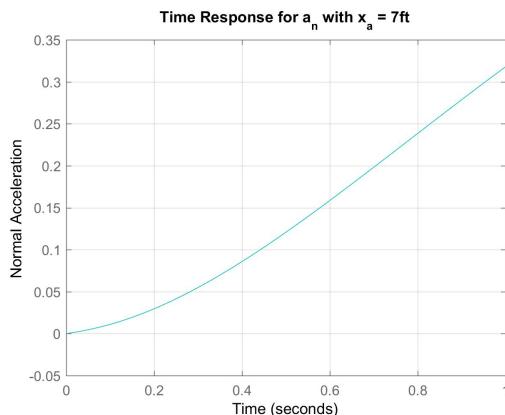


Figure 2.9: Time Response when  $x_a = 7\text{ft}$ .

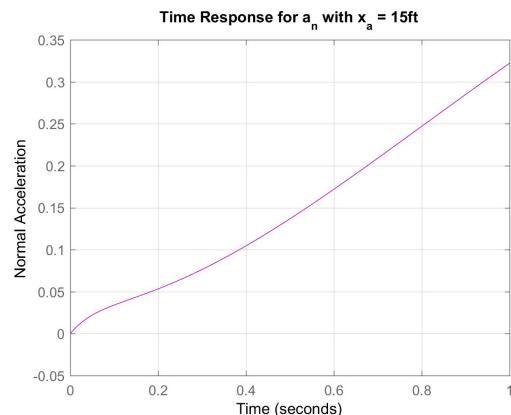


Figure 2.10: Time Response when  $x_a = 15\text{ft}$ .

# 3

## Open Loop Analysis

In this chapter, the LTI state space model, which is previously obtained, is reduced to two separate state space models - one longitudinal and one lateral - that only includes variables of interest. Using these new models, the inherent motion characteristics for **20000 ft** altitude and **600 ft/s** velocity are computed, and finally the corresponding open loop analysis of the aircraft is presented. The corresponding Matlab file used in Chapter 3 is *Linearization.m*.

### 3.1. State Space Model Reduction

After making the trimming and linearization around the equilibrium point, MATLAB retrieves matrices  $A_{model}$ ,  $B_{model}$ ,  $C_{model}$  and  $D_{model}$  separately for the longitudinal and lateral directions [1]. For each model, the state space system is given by:

$$\begin{cases} \dot{x}_{model} = A_{model}x_{model} + B_{model}u_{model} \\ y_{model} = C_{model}x_{model} + D_{model}u_{model} \end{cases} \quad (3.1)$$

As shown in the next two subsections, a few simplifications are made to each of the models.

#### 3.1.1. Reduced Matrices for Longitudinal Direction

For the longitudinal direction, the state and input vectors of interest are given by  $x_{long} = [V_t \alpha \theta q]^T$  and  $u_{long} = [\delta_{el} \delta_{th}]^T$  [2]. These variables are enough to study the longitudinal eigenmotions. The first system obtained from direct elimination of the variables that are not used is the following:

$$\begin{bmatrix} \dot{V}_t \\ \dot{\alpha} \\ \dot{\theta} \\ \dot{q} \\ \dot{\delta}_{el} \\ \dot{\delta}_{th} \end{bmatrix} = \begin{bmatrix} -0.0109 & -1.7611 & -32.1700 & -0.8207 & 0.1093 & 0.0016 \\ -0.0002 & -0.6505 & 0 & 0.9482 & -0.0014 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & -1.9092 & 0 & -0.8893 & -0.1389 & 0 \\ 0 & 0 & 0 & 0 & -20.2 & 0 \\ 0 & 0 & 0 & 0 & 0 & -1 \end{bmatrix} \begin{bmatrix} V_t \\ \alpha \\ \theta \\ q \\ \delta_{el} \\ \delta_{th} \end{bmatrix} + \begin{bmatrix} 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ 20.2 & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} u_{el} \\ u_{th} \end{bmatrix}$$

$$\begin{bmatrix} V_t \\ \alpha \\ \theta \\ q \end{bmatrix} = \begin{bmatrix} 1.0000 & 0 & 0 & 0 & 0 & 0 \\ 0 & 57.2958 & 0 & 0 & 0 & 0 \\ 0 & 0 & 57.2958 & 0 & 0 & 0 \\ 0 & 0 & 0 & 57.2958 & 0 & 0 \end{bmatrix} \begin{bmatrix} V_t \\ \alpha \\ \theta \\ q \\ \delta_{el} \\ \delta_{th} \end{bmatrix} + \begin{bmatrix} 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} u_{el} \\ u_{th} \end{bmatrix}$$

This first system includes  $\delta_{el}$  and  $\delta_{th}$  as state variables, because the actuator and engine dynamics are represented in the model. This means that, in practice, the real inputs considered are electrical signals for the elevator actuator ( $u_{el}$ ) and for the engine ( $u_{th}$ ) [2].

However, to remove the actuator dynamics and to make  $\delta_{el}$  and  $\delta_{th}$  the inputs in the final reduced system, another change has to be made. For this, it should be noticed that the relationship between the state variables  $V_t$ ,  $\alpha$ ,  $\theta$ ,  $q$  and the inputs  $\delta_{el}$  and  $\delta_{th}$  is contained inside matrix A of the system obtained above [2]. The final system for the longitudinal direction is therefore:

$$\begin{bmatrix} \dot{V}_t \\ \dot{\alpha} \\ \dot{\theta} \\ \dot{q} \end{bmatrix} = \begin{bmatrix} -0.0109 & -1.7611 & -32.1700 & -0.8207 \\ -0.0002 & -0.6505 & 0 & 0.9482 \\ 0 & 0 & 0 & 1 \\ 0 & -1.9092 & 0 & -0.8893 \end{bmatrix} \begin{bmatrix} V_t \\ \alpha \\ \theta \\ q \end{bmatrix} + \begin{bmatrix} 0.1093 & 0.0016 \\ -0.0014 & 0 \\ 0 & 0 \\ -0.1389 & 0 \end{bmatrix} \begin{bmatrix} \delta_{el} \\ \delta_{th} \end{bmatrix}$$

$$\begin{bmatrix} V_t \\ \alpha \\ \theta \\ q \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 57.2958 & 0 & 0 \\ 0 & 0 & 57.2958 & 0 \\ 0 & 0 & 0 & 57.2958 \end{bmatrix} \begin{bmatrix} V_t \\ \alpha \\ \theta \\ q \end{bmatrix} + \begin{bmatrix} 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} \delta_{el} \\ \delta_{th} \end{bmatrix}$$

### 3.1.2. Reduced Matrices for Lateral Direction

For the lateral direction, the state and input vectors of interest are  $x_{lat} = [\beta \phi p r]^T$  and  $u_{lat} = [\delta_a \delta_r]^T$ . Similarly to the longitudinal model, these variables are enough to study the lateral eigenmotions. The same procedure used on the previous section is followed here. Thus, the first systems obtained from direct elimination of the unused variables are:

$$\begin{bmatrix} \dot{\beta} \\ \dot{\phi} \\ \dot{p} \\ \dot{r} \\ \dot{\delta}_a \\ \dot{\delta}_r \end{bmatrix} = \begin{bmatrix} -0.2055 & 0.0535 & 0.0594 & -0.9941 & 0.0002 & 0.0005 \\ 0 & 0 & 1 & 0.0595 & 0 & 0 \\ -25.8584 & 0 & -2.3166 & 0.4924 & -0.5576 & 0.0755 \\ 7.2786 & 0 & -0.0294 & -0.3191 & -0.0309 & -0.0571 \\ 0 & 0 & 0 & 0 & -20.2 & 0 \\ 0 & 0 & 0 & 0 & 0 & -20.2 \end{bmatrix} \begin{bmatrix} \beta \\ \phi \\ p \\ r \\ \delta_a \\ \delta_r \end{bmatrix} + \begin{bmatrix} 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ 20.2 & 0 \\ 0 & 20.2 \end{bmatrix} \begin{bmatrix} u_a \\ u_r \end{bmatrix}$$

$$\begin{bmatrix} \beta \\ \phi \\ p \\ r \end{bmatrix} = \begin{bmatrix} 57.2958 & 0 & 0 & 0 \\ 0 & 57.2958 & 0 & 0 \\ 0 & 0 & 57.2958 & 0 \\ 0 & 0 & 0 & 57.2958 \end{bmatrix} \begin{bmatrix} \beta \\ \phi \\ p \\ r \end{bmatrix} + \begin{bmatrix} 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} u_a \\ u_r \end{bmatrix}$$

Once again, it is possible to conclude that the actual inputs of the system are electrical signals for the aileron ( $u_a$ ) and for the rudder actuators ( $u_r$ ). The values concerning the throttle were also eliminated, just like before, because its influence is minimal.

To transform the variables  $\delta_a$ ,  $\delta_r$  into inputs, it is again noted that the relationships between these two variables and the state variables  $\beta$ ,  $\phi$ ,  $p$ ,  $r$  are contained inside matrix A of the above system. Therefore, the final system is given by:

$$\begin{bmatrix} \dot{\beta} \\ \dot{\phi} \\ \dot{p} \\ \dot{r} \end{bmatrix} = \begin{bmatrix} -0.2055 & 0.0535 & 0.0594 & -0.9941 \\ 0 & 0 & 1 & 0.0595 \\ -25.8584 & 0 & -2.3166 & 0.4924 \\ 7.2786 & 0 & -0.0294 & -0.3191 \end{bmatrix} \begin{bmatrix} \beta \\ \phi \\ p \\ r \end{bmatrix} + \begin{bmatrix} 0.0002 & 0.0005 \\ 0 & 0 \\ -0.5576 & 0.0755 \\ -0.0309 & -0.0571 \end{bmatrix} \begin{bmatrix} \delta_a \\ \delta_r \end{bmatrix}$$

$$\begin{bmatrix} \beta \\ \phi \\ p \\ r \end{bmatrix} = \begin{bmatrix} 57.2958 & 0 & 0 & 0 \\ 0 & 57.2958 & 0 & 0 \\ 0 & 0 & 57.2958 & 0 \\ 0 & 0 & 0 & 57.2958 \end{bmatrix} \begin{bmatrix} \beta \\ \phi \\ p \\ r \end{bmatrix} + \begin{bmatrix} 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} \delta_a \\ \delta_r \end{bmatrix}$$

## 3.2. Calculation of the Inherent Motion Characteristics

After computing the fourth-order systems for both the lateral and longitudinal directions, the motion characteristics can now be derived. For the next two sections, the periodic eigenmotions are separated from the non-periodic eigenmotions.

### 3.2.1. Periodic Eigenmotions

There are three periodic eigenmotions: the **Short Period** and the **Phugoid** for the longitudinal direction and the **Dutch Roll** for the lateral direction. The poles locations for each eigenmotion, as well as their characteristics, are shown on Table 3.1.

Eigenmotion	Poles Location	$\omega_n(\text{rad/s})$	$\zeta$	$P(s)$	$T_{1/2}(s)$
<b>Short Period</b>	$-0.7713 \pm 1.3392i$	1.5454	0.4991	4.6918	0.8986
<b>Phugoid</b>	$-0.0040 \pm 0.0675i$	0.0676	0.0588	93.1239	174.5140
<b>Dutch Roll</b>	$-0.3083 \pm 2.9290i$	2.9452	0.1047	2.1452	2.2485

Table 3.1: Periodic Eigenmotions and corresponding characteristics.

In this Table,  $\omega_n$  is the natural frequency,  $\zeta$  is the damping ratio,  $P$  is the period and  $T_{1/2}$  is the time to damp to half amplitude. The expression for periodic eigenvalues usually is  $\lambda = n \pm i\omega$  and if  $Re\{p\}$  and  $Im\{p\}$  denote the real and imaginary part respectively of a pole  $p$ , then the value for each characteristic is obtained by applying the following expressions [3]:

$$\omega_n = \sqrt{(Re\{p\})^2 + (Im\{p\})^2} \quad \zeta = \frac{-Re\{p\}}{\omega_n} \quad P = \frac{2\pi}{Im\{p\}} \quad T_{1/2} = \frac{\ln 2}{|Re\{p\}|} \quad (3.2)$$

### 3.2.2. Non-Periodic Eigenmotions

There are two non-periodic eigenmotions, both in the lateral direction: the **Roll** and the **Spiral**. Their poles locations and the associated characteristics are presented on Table 3.2.

Eigenmotion	Poles Location	$\omega_n(\text{rad/s})$	$\tau(s)$	$T_{1/2}(s)$
<b>Roll</b>	-2.2146	2.2146	0.4515	0.3130
<b>Spiral</b>	-0.0101	0.0101	99.1889	68.7525

Table 3.2: Anti-Periodic Eigenmotions and corresponding characteristics.

The parameters presented in this Table are the natural frequency  $\omega_n$ , the time constant  $\tau$  and the time to damp to half amplitude  $T_{1/2}$ . Because the imaginary part of these poles are zero, the natural frequency is simply the absolute value of their real part. The other two parameters are obtained using the expressions:

$$\tau = -\frac{1}{Re\{p\}} \quad T_{1/2} = \frac{\ln 2}{|Re\{p\}|} \quad (3.3)$$

### 3.2.3. Open Loop Time Response and Analysis

In order to confirm the previous results, the time responses of the state variables are plotted for each eigenmotion. To see the response for each eigenmotion separately, the MATLAB command *initial* is used, because it allows to set the eigenvalue of a single eigenmotion as the initial condition. This command, however, doesn't accept complex initial conditions. Because some eigenvectors are indeed complex, a linear combination of an eigenvector and its conjugate is used. The contribution of each one is  $\frac{1}{2}$ , so the result is the real part of that particular eigenvector:  $\frac{1}{2}X + \frac{1}{2}X^* = Re\{X\}$ , where  $X$  is the eigenvector and  $X^*$  its conjugate.

- **Short Period Time Response:** The time response for the Short Period is shown in Figure 3.1. Its damping is relatively large - 0.4991 from Table 3.1 - which is confirmed by looking at the plot, because the response quickly converges to the final value. The period is harder to calculate, but it is possible to say that if one takes  $t_i = 0s$  and  $t_f = 4.65s$ , a period has passed ( $P = 4.65s$ ). This value is not far from the one in the Table.

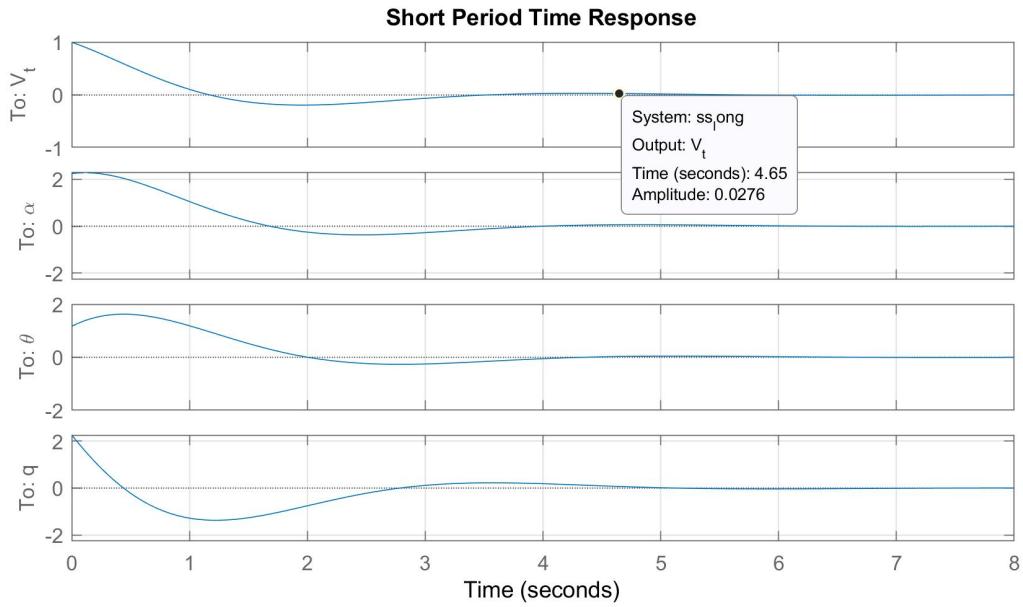


Figure 3.1: Time response for the Short Period Eigenmotion.

- **Phugoid Time Response:** The time response for this eigenmotion is plotted in Figure 3.2. The Phugoid damping ratio is much smaller compared to the Short Period's, so the response takes longer to converge to the final value (more than 600s) and is much more oscillatory. The time instants  $t_i = 93$ s and  $t_f = 186$ s correspond approximately to two consecutive peaks and thus to a period ( $P = 93$ s). This value for the period  $P$  is really close to the one computed in Table 3.1.

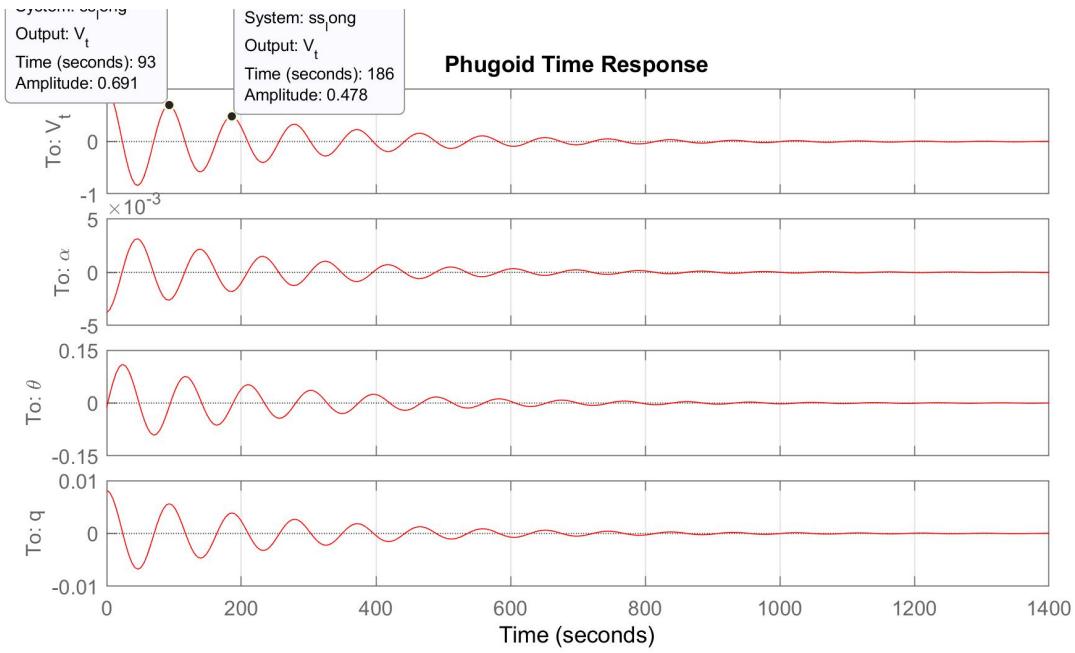


Figure 3.2: Time response for the Phugoid Eigenmotion.

- **Dutch Roll Time Response:** The Dutch Roll is the only oscillatory lateral eigenmotion. According to Table 3.1 its damping ratio is somewhere in between the ones from the two

previous eigenmotions. As a consequence, the response converges to the final value faster than the Phugoid but slower than the short period. This fact is confirmed by Figure 3.3. Again, by taking  $t_i = 1.81\text{s}$  and  $t_f = 3.95\text{s}$  (time instants of two consecutive peaks) the period can be calculated and its value is  $P = 2.14\text{s}$ . Again, this value is very close to the one computed with the formulas.

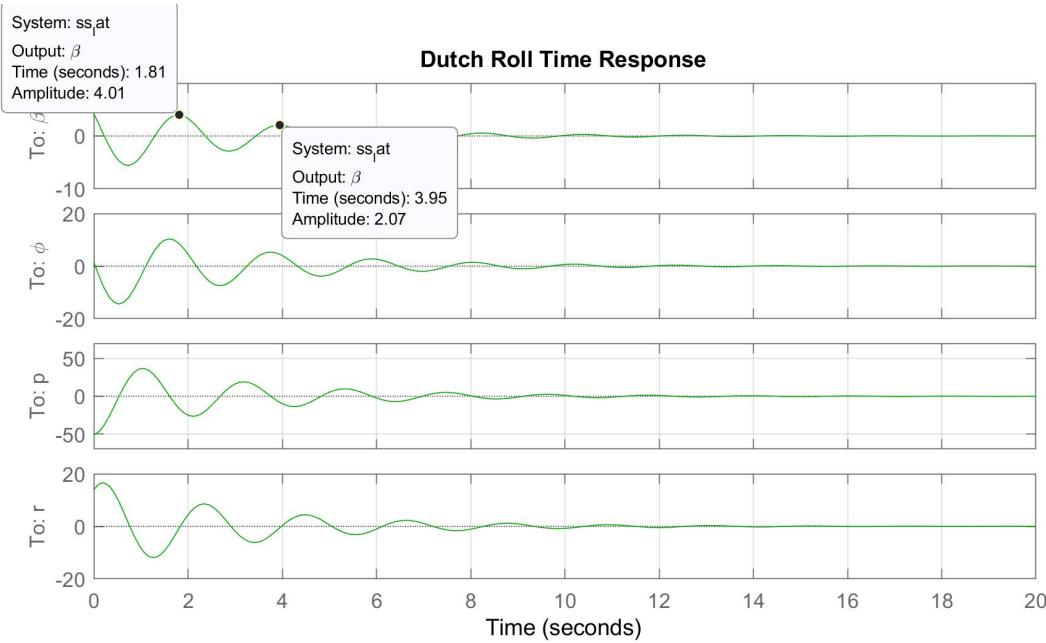


Figure 3.3: Time response for the Dutch Roll Eigenmotion.

- **Roll Time Response:** Looking at Figure 3.4, the response of the variable  $\beta$  starts at an amplitude of 0.177 and reaches the value 0.0645, that is approximately 36.8% (that is  $e^{-1}$ ) of the initial value after 0.457s. For this kind of response, this instant gives the value of the time constant. Indeed, the computed value in Table 3.2 is very close to it.

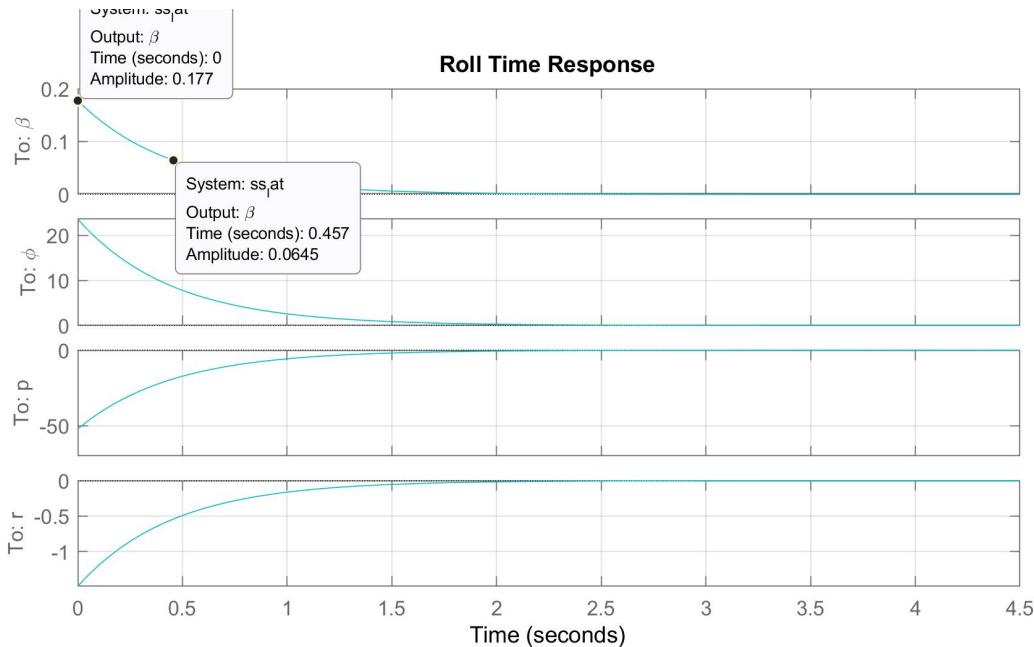


Figure 3.4: Time response for the Roll Eigenmotion.

- **Spiral Time Response:** For this eigenmotion let's, for convenience, look at the response of the variable  $p$ . It starts at an amplitude of 0.756 and reaches the value 0.278, that is, 36.8% of the initial value after 99.2s. Once again, this value corresponds to the time constant. It is also very similar to the one that can be found on Table 3.2.

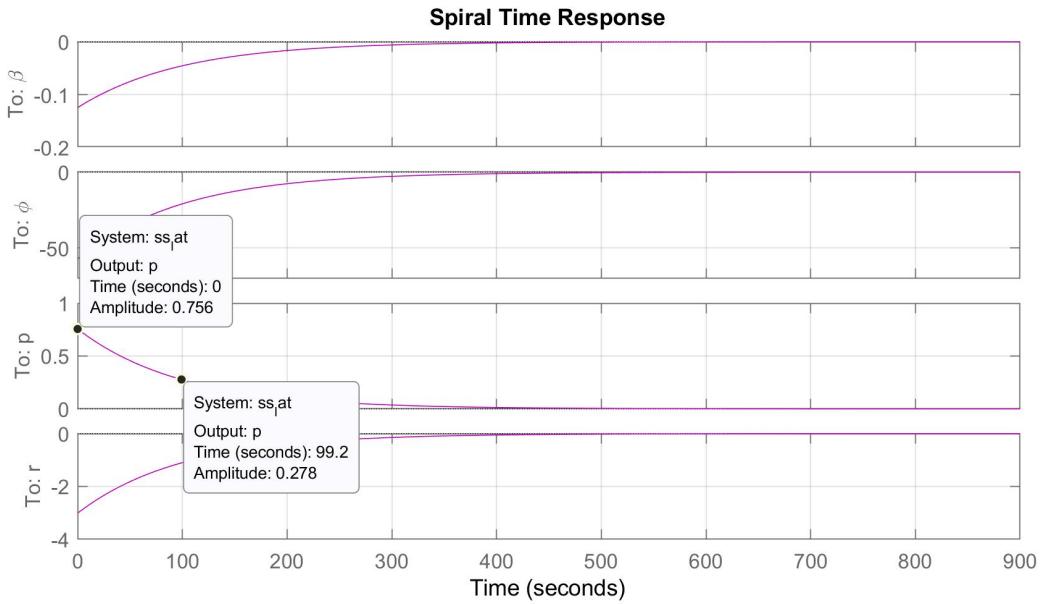


Figure 3.5: Time response for the Spiral Eigenmotion.

Overall, the theoretical values obtained using the formulas 3.2 and 3.3 are very close to the values observed and derived from the above plots, thus validating their use.

# 4

## Design of a Pitch Rate Command System Satisfying CAP/Gibson Mil-Specs

In this chapter, a pitch rate controller for the F16 model is designed. This controller has to fulfill two criteria, namely:

- the CAP (Control Anticipation Parameter) criterion
- the Gibson Dropback criterion

Firstly, the short period reduced model is obtained. Then, the pitch rate command system is designed and validated. To obtain the desired pitch controller, a prefilter is also added to the system and finally, the CAP and Dropabck criterion are validated and some conclusions are made. For Chapter 4 the matlab file *Pitch\_damper.m* is used.

### 4.1. Reduced Model for the Short Period

From the Longitudinal simplified model obtained in the end of subsection 3.1.1, it is possible to obtain the reduced model for the Short Period, by only taking into account the variables  $\alpha$  and  $q$  [2]. The Longitudinal simplified model is shown again below:

$$\begin{bmatrix} \dot{V}_t \\ \dot{\alpha} \\ \dot{\theta} \\ \dot{q} \end{bmatrix} = \begin{bmatrix} -0.0109 & -1.7611 & -32.1700 & -0.8207 \\ -0.0002 & -0.6505 & 0 & 0.9482 \\ 0 & 0 & 0 & 1 \\ 0 & -1.9092 & 0 & -0.8893 \end{bmatrix} \begin{bmatrix} V_t \\ \alpha \\ \theta \\ q \end{bmatrix} + \begin{bmatrix} 0.1093 \\ -0.0014 \\ 0 \\ -0.1389 \end{bmatrix} \delta_{el}$$

$$\begin{bmatrix} V_t \\ \alpha \\ \theta \\ q \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 57.2958 & 0 & 0 \\ 0 & 0 & 57.2958 & 0 \\ 0 & 0 & 0 & 57.2958 \end{bmatrix} \begin{bmatrix} V_t \\ \alpha \\ \theta \\ q \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \end{bmatrix} \delta_{el}$$

which then reduces to:

$$\begin{bmatrix} \dot{\alpha} \\ \dot{q} \end{bmatrix} = \begin{bmatrix} -0.6505 & 0.9482 \\ -1.9092 & -0.8893 \end{bmatrix} \begin{bmatrix} \alpha \\ q \end{bmatrix} + \begin{bmatrix} -0.0014 \\ -0.1389 \end{bmatrix} \delta_{el}$$

$$\begin{bmatrix} \alpha \\ q \end{bmatrix} = \begin{bmatrix} 57.2958 & 0 \\ 0 & 57.2958 \end{bmatrix} \begin{bmatrix} \alpha \\ q \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \end{bmatrix} \delta_{el}$$

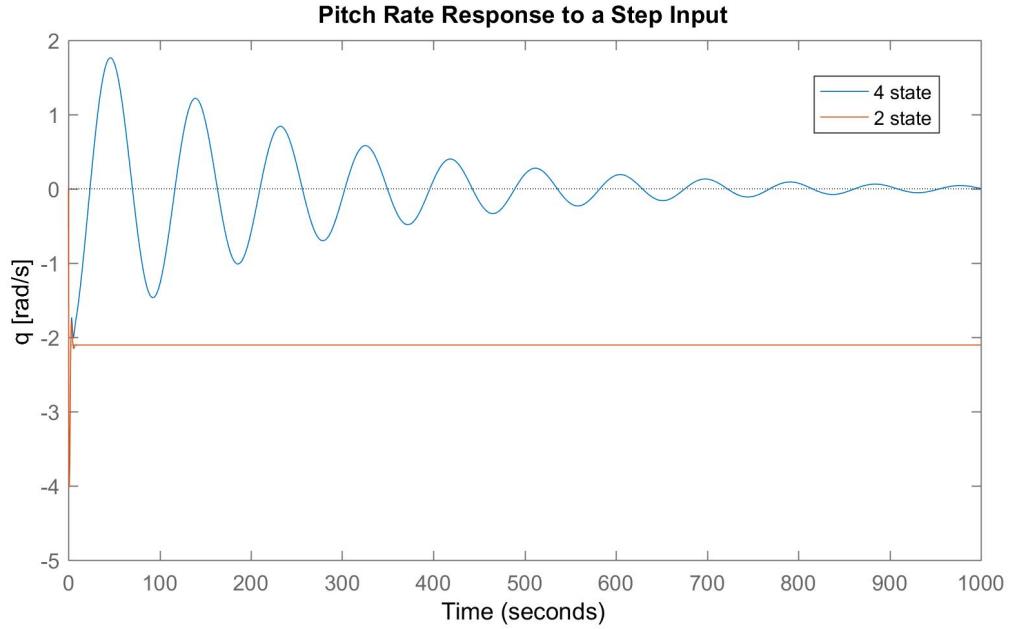


Figure 4.1: Time response for a step input for the pitch rate.

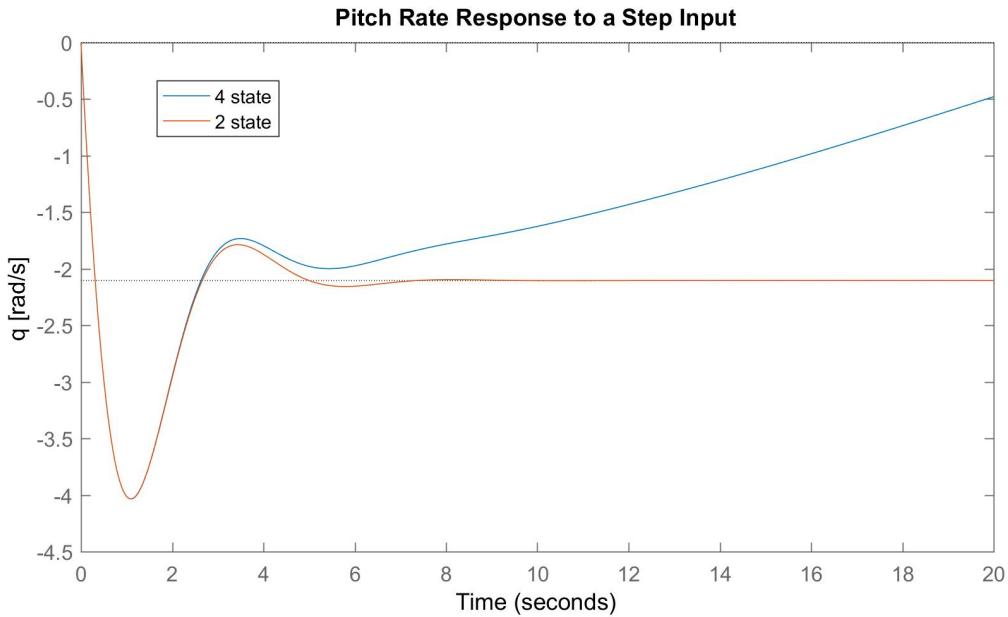


Figure 4.2: Time response for a step input for the pitch rate (detail view).

The responses of the pitch rate  $q$  for a step input for the 2 and 4 state model without actuator dynamics can be found above, in Figure 4.1. A detailed view of that time response is shown in Figure 4.2.

The transfer functions for the 4 state and 2 state models are shown in equations 4.1 and 4.2, respectively.

$$\frac{q(s)}{\delta_e(s)} = \frac{-7.958s^3 - 5.112s^2 - 0.05013s - 1.41 \times 10^{-14}}{s^4 + 1.551s^3 + 2.405s^2 + 0.026s + 0.0109} \quad (4.1)$$

$$\frac{q(s)}{\delta_e(s)} = \frac{-7.958s - 5.026}{s^2 + 1.54s + 2.389} \quad (4.2)$$

By looking at Figure 4.1, one can conclude that both the 2 state and 4 state models stabilize at some point. The 4 state model takes some time to stabilize, and it converges to 0. The 2 state model is much faster, but it converges to approximately  $-2.2\text{rad/s}$ . Figure 4.2 shows that for the first 6 seconds, the two models behave the same way, so the 2 state model is a good approximation for this time interval. This goes well with the fact that the Short Period is the dominant mode for the first few seconds. After that, this eignemotion converges and the Phugoid starts to be more dominant. This is why the 4-state model diverges after some seconds, while the 2-state model (onlye representing the Short Period) converges quickly. And because only short term movements are being considered for this assignment, it is enough to only take into account the first few seconds of the time response, which means that it is accurate enough to use the reduced 2 state model.

## 4.2. Design of the Pitch Rate Command System

### 4.2.1. Frequency Domain Conversion for the CAP and Gibson criteria

Before starting the design of the pitch rate command system, some new values need to be obtained. Firstly, the CAP and Gibson criteria are converted to the frequency domain and the results are written below [2], where the speed  $V = 600 \times 0.3048 = 182.88\text{m/s}$ :

- $\omega_{n_{sp}}(V, h) = 0.03V(V, h) = 0.03 \times 182.88 = 5.49\text{rad/s}$
- $\frac{1}{T_{\theta_2}}(V, h) = 0.75\omega_{n_{sp}} = 0.75 \times 5.49 = 4.12\text{Hz} \Leftrightarrow T_{\theta_2} = 0.243\text{s}$
- $\zeta_{sp} = 0.5$

The denominator of Equation 4.2 can be used to calculate the values of  $\omega_n$  and  $\zeta$  for the 2 state model:

- $\omega_n = \sqrt{2.389} = 1.546\text{rad/s}$
- $\zeta = \frac{1.54}{2\omega_n} = 0.498$

It should be noted that these values are very close to the values obtained from the 4 state model, shown on Section 3.2.1 for the Short Period, thus proving once more that this simplification is very accurate.

### 4.2.2. Method to obtain the Pitch Damper

The values for the natural frequency and the damping ratio of the 2 state system do not meet the CAP and Gibson requirements. However, because the required values are known, it is possible to create a pitch damper by using the pole placement technique. The characteristic equation of the desired system is given by:

$$s^2 + 2\omega_{n_{sp}}\zeta_{sp}s + \omega_{n_{sp}}^2 = s^2 + 5.49s + 30.14 \quad (4.3)$$

from which the poles are obtained:

$$s_1 = -2.745 + 2.377i, \quad s_2 = -2.745 - 2.377i \quad (4.4)$$

With the desired poles calculated, and using the 2 state matrices, it is possible to obtain the state feedback matrix K using MATLAB function *place*. The result is the following:

$$K = [-62.877 - 27.819] \quad (4.5)$$

where  $K_\alpha = -62.877 [\text{°}/\text{rad}]$  and  $K_q = -27.819 [\text{°}/(\text{rad/s})]$ .

### 4.2.3. Validation of the Feedback Gain Matrix Considering Severe Gusts

To check if the feedback gains are acceptable in the presence of possible gusts, a severe vertical gust with a value of  $W_g = 4.572 \text{ m/s}$  is considered. The angle of attack induced from this wind gust is:

$$\alpha_{ind} = \arctg\left(\frac{W_g}{V}\right) = 1.432^\circ = 0.0249 \text{ rad} \quad (4.6)$$

The subsequent elevator deflection  $\delta_e$  is given by:

$$\delta_e = K_\alpha \times \alpha_{ind} = -1.571^\circ \quad (4.7)$$

The deflection of the elevator is within the allowable range (from  $-25^\circ$  to  $25^\circ$ ) so the pitch damper can cope with a vertical wind gust.

## 4.3. Time Constant Modification and Lead-Lag Prefilter Location

In the previous section, it was showed that the pitch damper works efficiently under severe gusts. But the time constant  $T_{\theta_2}$  still needs to be changed in order to obtain the desired pitch rate command, so that it meets the CAP and Gibson requirements. To change the time constant, a Lead-Lag compensator (prefilter) needs to be added, because that time constant cannot be modified by pole placement. The general Transfer Function for a Lead-Lag compensator is given in Equation 4.8.

$$Filter(s) = K \frac{s - z}{s - p} \quad (4.8)$$

If  $|z| > |p|$ , one can obtain a lag compensator, and if  $|z| < |p|$ , a lead compensator is obtained. The pole of this filter can cancel out the zero of the original system, while its zero will become the new zero for the whole system. So, using a prefilter, it is indeed possible to modify the parameter  $T_{\theta_2}$ . The prefilter should thus be located outside the loop, because this way it will be multiplied with the already obtained transfer function, so the zero of the original system will really be cancelled out. Plus, if it is outside the loop, the closed loop system poles (which were already obtained in the previous section using the pole placement technique) will not change.

The complete Transfer Function from  $\delta_e$  to  $q$  is then:

$$\frac{q(s)}{\delta_e(s)} = K \frac{s - z}{s - p} \times \frac{k_q(1 + T_{\theta_2}s)}{s^2 + 2\omega_{n_{sp}}\zeta_{sp}s + \omega_{n_{sp}}^2} \quad (4.9)$$

The closed loop system transfer function after the pole placement but **before** adding the compensator is:

$$\frac{q(s)}{\delta_e(s)} = \frac{-7.958s - 5.026}{s^2 + 5.49s + 30.14} = \frac{-5.026(1 + 1.583s)}{s^2 + 5.49s + 30.14} = \frac{-7.958(s + 0.632)}{s^2 + 5.49s + 30.14} \quad (4.10)$$

Thus, by comparison with Equation 4.9,  $T_{\theta_2} = 1.583$  and  $k_q = -5.026$ . However, the new value for  $T_{\theta_2}$  should be 0.243, which means that the **final** system transfer function should be:

$$\frac{q(s)}{\delta_e(s)} = \frac{-5.026(1 + 0.243s)}{s^2 + 5.49s + 30.14} = \frac{-20.683(s + 4.115)}{s^2 + 5.49s + 30.14} \quad (4.11)$$

By looking at Equations 4.10 and 4.11 it is clear that the zero  $(s + 0.632)$  needs to be cancelled, and replaced by the term  $(s + 4.115)$ . This means that  $(s + 0.632)$  should be the pole of the filter and  $(s + 4.115)$  should be its zero.

Finally, to obtain the gain  $K$  of the filter, it should be noted that the steady state error of the final transfer function (closed loop system + compensator) shoul be 0 for a step input. That happens when the transfer function for  $s=0$  is 1:

$$\frac{q(0)}{\delta_e(0)} = \frac{-20.683K(4.115 + 0)}{0^2 + 5.49 \times 0 + 30.14} = \frac{-85.111K}{30.14} = 1 \quad (4.12)$$

Using the last equation, it comes that  $K = -0.354$ .

Because  $|z| > |p|$ , a lag compensator is obtained, and its transfer function equals:

$$\text{Filter}(s) = -0.354 \frac{s + 4.115}{s + 0.632} \quad (4.13)$$

The final transfer function for the whole system is:

$$\frac{q(s)}{\delta_e(s)} = \frac{7.322(4.115 + s)}{s^2 + 5.49s + 30.14} \quad (4.14)$$

## 4.4. CAP and Gibson Criteria Validation

The positions of the design points in the CAP and Gibson criteria can be calculated using the following equations [2]:

$$\text{CAP} = \frac{\omega_{n_{sp}}^2}{\frac{V}{g T_{\theta_2}}} = \frac{5.49^2}{\frac{182.88}{9.80665} \cdot 4.12} = 0.393 \quad (4.15)$$

$$\frac{DB}{q_s} = T_{\theta_2} - \frac{2\zeta_{sp}}{\omega_{n_{sp}}} = 0.243 - \frac{2 \times 0.5}{5.49} = 0.061 \quad (4.16)$$

To obtain the design point in the Gibson criterion, the value for  $q_m/q_s$  is needed as well. To obtain these values, the time response of the pitch rate to a step input is plotted in Figure 4.3.  $q_m$  is the maximum value of the pitch rate response to a step input and  $q_s$  is its final value:

$$q_m = 1.412, \quad q_s = 1, \quad \frac{q_m}{q_s} = 1.412 \quad (4.17)$$

The time response of the pitch angle to a ramp input is plotted in Figure 4.4. From this plot, it is possible to obtain the value for  $DB$ : It is the difference between the maximum value of the pitch angle response to the ramp input,  $DB_{max}$ , and its final value,  $DB_{final}$ :

$$DB = DB_{max} - DB_{final} = 5.135 - 5 = 0.135 \quad (4.18)$$

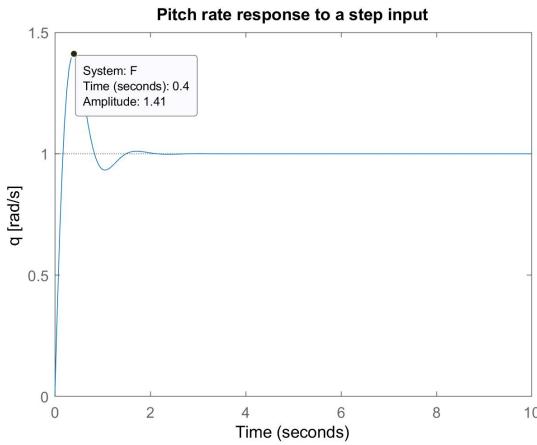


Figure 4.3: Pitch rate plot.

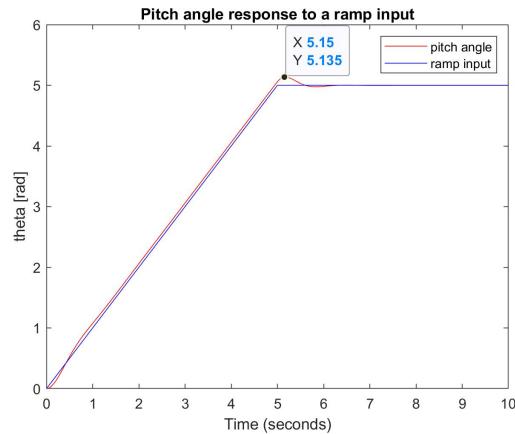


Figure 4.4: Pitch angle plot.

With these results, the Dropback criterion and the CAP and short period damping requirements are shown in Figures 4.5 and 4.6.

It can be said that the plane is flying in Category B, because it is a non-terminal flight phase that requires gradual manoeuvring. In Figure 4.5, it is shown that the aircraft verifies the drop-back criterion ( $\frac{q_m}{q_s} = 1.412$  and  $\frac{DB}{q_s} = 0.061$ ), because it is inside the satisfactory region (blue region). In figure 4.6 it is clear that the design point for the CAP criterion ( $\zeta_{sp} = 0.5$  and  $CAP = 0.393$ ) is located within Level 1, which corresponds to a clearly adequate flight quality. To conclude, both requirements are met so the pitch rate command system was successfully designed.

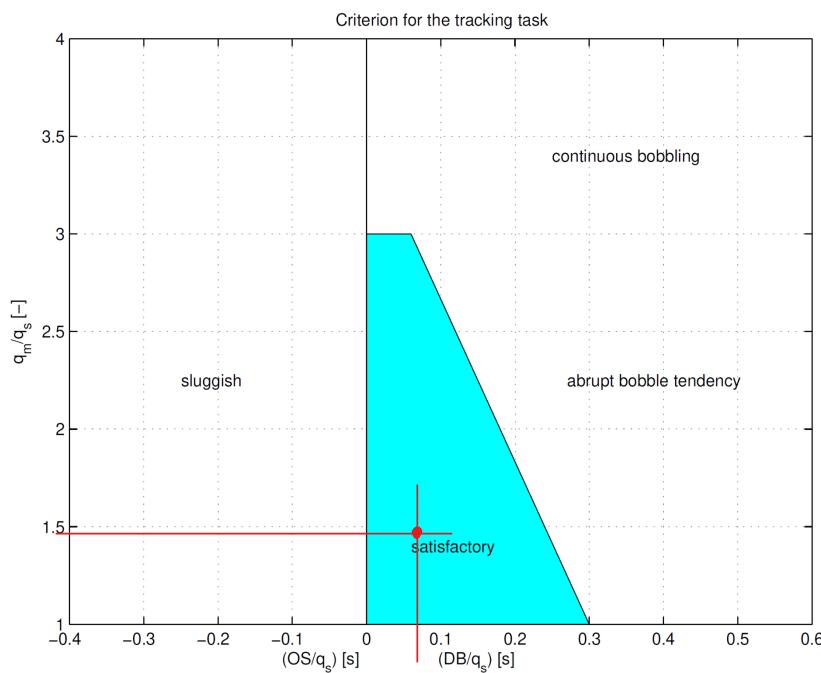


Figure 4.5: Verification of the Gibson dropback criterion.

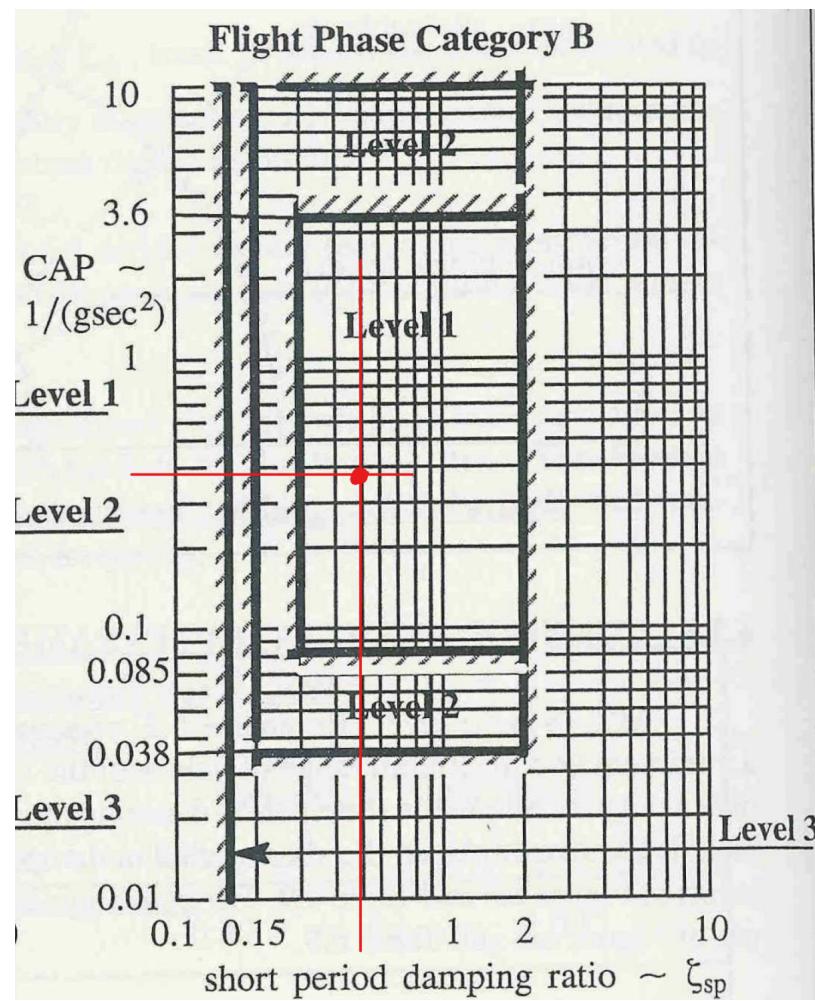


Figure 4.6: Verification of the CAP criterion.

# 5

## Design of a Terrain Following System

In this Chapter, the design of a terrain following control system for the F-16 is made. Firstly, the model is linearized for the new flight conditions. Then, the aircraft model is defined and the actuators dynamics are added. Finally, the terrain following system itself is designed, by a try and error technique. Here, the MATLAB script *Terrain\_following.m* is used.

### 5.1. Trim and Linearization

As said before, the flight dynamics of any aircraft is represented by non-linear functions. Thus, to analyse the behaviour of the F-16 for the flight condition of  **$h = 5000\text{ft}$**  altitude and  **$\mathbf{V} = 300\text{ft/s}$**  velocity, mentioned on Chapter 3, another trimming and linearization are made. The trimming results can be found in Table 5.1.

Low Fidelity Model	
<b>Thrust (lb.)</b>	2826.8165
<b>Elevator (deg.)</b>	-4.1891
<b>Aileron (deg.)</b>	$-1.9926 \times 10^{-15}$
<b>Rudder (deg.)</b>	$1.2406 \times 10^{-14}$
<b>Alpha (deg.)</b>	10.4511

Table 5.1: Results for the trim and linearization for the 5000ft and 300ft/s flight condition.

### 5.2. Reduced Model for the Given Flight Condition

The reduced model for the given flight condition, with a state vector given by  $x = [h, V_t, \alpha, \theta, q]^T$  is obtained from MATLAB and is presented below. The same procedure as the one used in Section 3.1 was applied to these results in order to eliminate the engine and elevator actuator dynamics from the state space model.

$$\begin{bmatrix} \dot{h} \\ \dot{V}_t \\ \dot{\alpha} \\ \dot{\theta} \\ \dot{q} \end{bmatrix} = \begin{bmatrix} 0 & 0 & -300 & 300 & 0 \\ 0.0001 & -0.0291 & 2.13 & -32.17 & -2.8952 \\ 0 & -0.0007 & -0.5447 & 0 & 0.9152 \\ 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0.3303 & 0 & -0.8169 \end{bmatrix} \begin{bmatrix} h \\ V_t \\ \alpha \\ \theta \\ q \end{bmatrix} + \begin{bmatrix} 0 & 0 \\ 0.0015 & -0.0045 \\ 0 & -0.0011 \\ 0 & 0 \\ 0 & -0.057 \end{bmatrix} \begin{bmatrix} \delta_{th} \\ \delta_{el} \end{bmatrix}$$

$$\begin{bmatrix} h \\ V_t \\ \alpha \\ \theta \\ q \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 57.2958 & 0 & 0 \\ 0 & 0 & 0 & 57.2958 & 0 \\ 0 & 0 & 0 & 0 & 57.2958 \end{bmatrix} \begin{bmatrix} h \\ V_t \\ \alpha \\ \theta \\ q \end{bmatrix} + \begin{bmatrix} 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} \delta_{th} \\ \delta_{el} \end{bmatrix}$$

### 5.3. Definition of the Aircraft Model and Actuators Dynamics

Before starting the construction of the Simulink model, it should be noted that it is requested to initialize the simulation at an altitude of 1500m, which corresponds to 4921ft. Although this is not the exact altitude value for which the aircraft was trimmed (5000ft) it is really close, so the trimming is still valid.

Apart from the block that defines the aircraft model, a series connection is made to another block containing the actuator and engine dynamics, because they are not included in the aircraft model itself. The block with these dynamics contains the transfer functions showed below, but also saturation blocks which don't allow the elevator and the engine thrust to go beyond the acceptable range. However, as a consequence of the trimming, the elevator is already deflected by  $-4.1891$  and the thrust is producing a force of  $2826.8165$ , which means that the limits of the saturation blocks have to take these values into account. Hence, the actual limits of the elevator deflection and engine thrust:

$$\delta_{el} \in [-25^\circ; 25^\circ] \quad \delta_{th} \in [1000lb; 19000lb] \quad (5.1)$$

become the relative limits for the saturation blocks:

$$\delta_{el} \in [-25 - (-4.1891); 25 - (-4.1891)] = [-20.8109^\circ; 29.1891^\circ] \quad (5.2)$$

$$\delta_{th} \in [1000 - 2826.8165; 19000 - 2826.8165] = [-1826.8165; 16173.1835] \quad (5.3)$$

The Transfer Functions for the engine and thrust actuator dynamics are:

$$H_{engine} = \frac{1}{s + 1} \quad (5.4)$$

$$H_{elevator} = \frac{20.2}{s + 20.2} \quad (5.5)$$

#### 5.3.1. Actuators Dynamics in Simulink

The aircraft model is defined in Simulink by using a series connection at the input with a subsystem block that contains the actuator dynamics and saturation limits for the elevator and engine. The actuator dynamics is found in the *Actuator and Thrust Dynamics* block. After the elevator and engine transfer function blocks the trim values are added to both links in order to calculate the real saturations before going in the saturation block. After this the trim values are removed again before entering the simplified model.

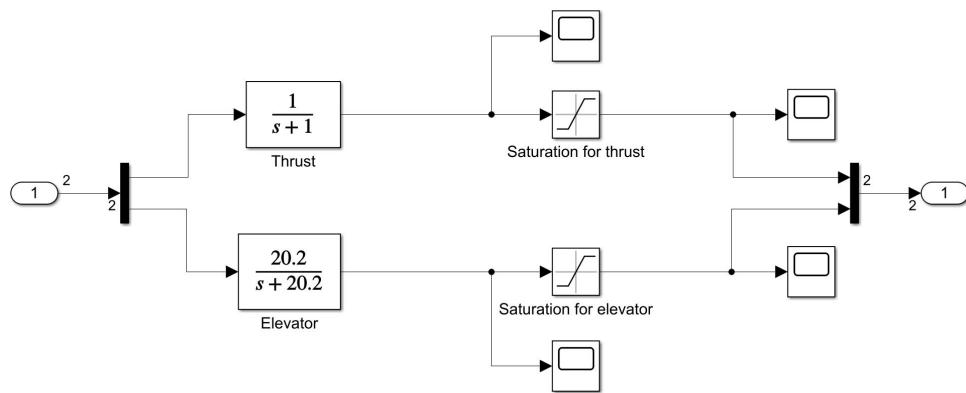


Figure 5.1: Actuator and Thrust dynamics in Simulink.

## 5.4. Design of an LQR Controller

In this section, the Optimal Control is used. Optimal Control tries to find a control law for a system such that a certain optimality criterion is achieved. It includes a cost function that is a function of state and control variables. Particularly in this assignment, the Linear Quadratic Control (LQR) is applied, and its cost function is:

$$J = \int_0^\infty (x^T(t)Qx(t) + u^T(t)Ru(t))dt \quad (5.6)$$

where the matrices Q and R are positive-semidefinite and positive-definite, respectively. They are both chosen by the designer of the lqr controller.

The solution of the LQR approach is a feedback gain matrix K so that  $u = -Kx$ , so it is a feedback of all states. To obtain this matrix, the *Riccati Equation* is solved:

$$A^T S + S A - S B R^{-1} B^T S + Q = 0 \quad (5.7)$$

and K is the solution of  $K = R^{-1}B^T S$ .

### 5.4.1. Complete Block Diagram for the Terrain Following System

To design the LQR controller, the MATLAB function *lqr* is used. It computes the optimal control and creates the feedback matrix K. This K matrix has a size of 2x5, each row representing the feedback gain of both inputs  $\delta_{th}$  and  $\delta_{el}$  for a single state variable.

However, because the objective of this Chapter is to create a terrain following system, the idea is to minimize the error between the real altitude and the reference altitude. Thus, the gains that correspond to the altitude should be outside the inner feedback loop. Instead, those gains are contained in the outer loop, and are applied to the error mentioned above.

The final block diagram is shown in Figure 5.2. The C matrix in the Reduced Model was replaced by the identity matrix, so that the outputs are exactly equal to the state variables.

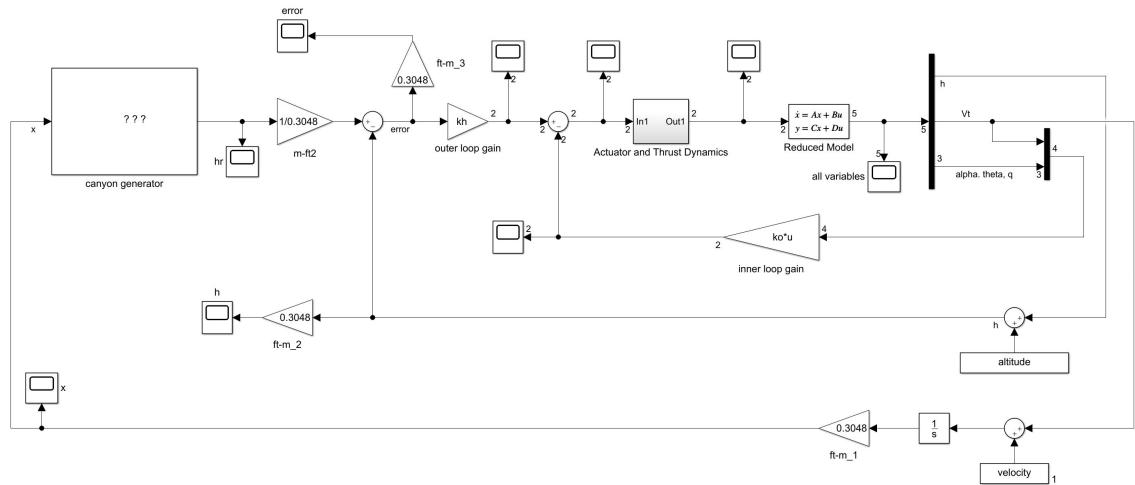


Figure 5.2: Complete Block Diagram for the terrain following system.

### 5.4.2. Method to Find Q and R Matrix

Matrix Q and R should be chosen so that the altitude overshoot for constant reference does not exceed 1m and the vertical altitude above the ground is always larger than 20m. A good option to find both Matrix Q and R is to force only the diagonal values to be non-zero [2].

$$Q = \begin{bmatrix} q_{11} & 0 & 0 & 0 & 0 \\ 0 & q_{22} & 0 & 0 & 0 \\ 0 & 0 & q_{33} & 0 & 0 \\ 0 & 0 & 0 & q_{44} & 0 \\ 0 & 0 & 0 & 0 & q_{55} \end{bmatrix} \quad R = \begin{bmatrix} r_{11} & 0 \\ 0 & r_{22} \end{bmatrix} \quad (5.8)$$

## 5.5. LQR Controller Optimization for Pilot Comfort

To optimize the LQR controller for pilot comfort the appropriate values on the diagonal entries of the Q and R matrix have to be chosen such that the overshoot is smaller than 1m and the aircraft altitude above ground is always larger than 20m. Because the reference altitude is always 40m above the ground, stating that the aircraft altitude above ground is larger than 20m is equivalent to stating that the error is smaller than  $40 - 20 = 20m$ .

One should note that increasing a diagonal element results in an increased effort to minimize the corresponding variable or input. Since the altitude is the most relevant variable, the corresponding coefficient in matrix Q, that is,  $q_{11}$  will probably have the highest value and its variations will have the biggest effects on the behaviour of the aircraft when it comes to follow the reference terrain.

To find the right values for matrices Q and R, each diagonal entry will be tested at a time. Starting with the identity matrix for both Q and R, the individual diagonal entries are changed by factors of ten, while the others are kept with the value 1. The values used in this assignment are {10, 1, 0.1, 0.01, 0.001}.

Each of the plots presented in the next two sections contain two reference graphs: the one of the terrain itself, that should ideally be followed, and the one resulting from the identity matrices for Q and R.

### 5.5.1. System Behaviour Analysis for varying values of matrix Q

- Changes in  $q_{11}$

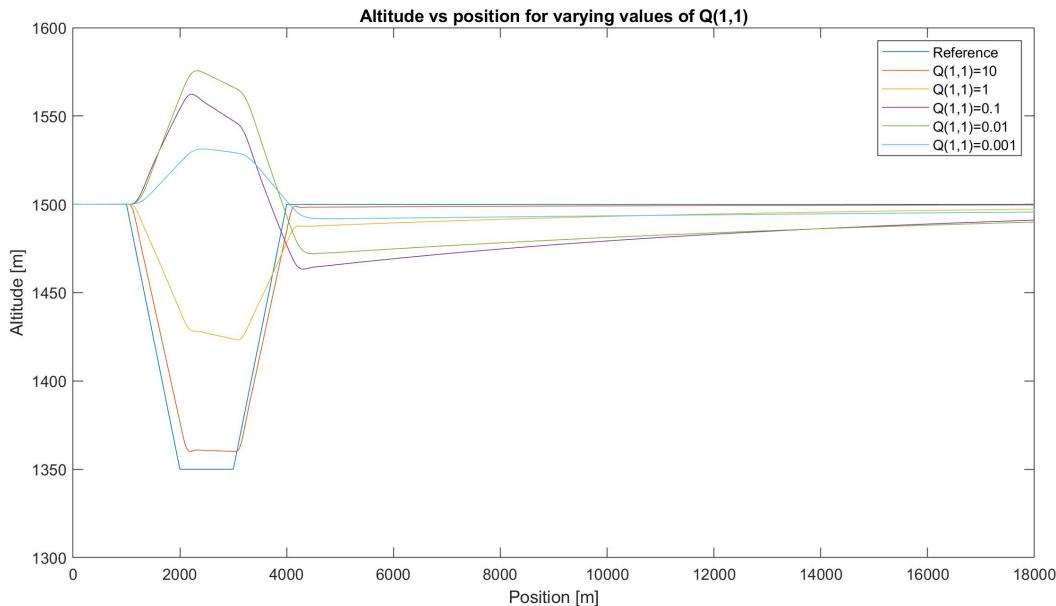


Figure 5.3: Comparison between aircraft altitude and reference altitude for different values of  $q_{11}$ .

By looking at this first results (Figure 5.3), it is now clear that the element  $q_{11}$  has indeed a large influence on the aircraft behaviour, because this element directly influences the altitude variable. For  $q_{11} = 10$ , the aircraft follows the reference altitude with the lowest error. On the other hand, for the values  $q_{11} = 0.1$ ,  $q_{11} = 0.01$  and  $q_{11} = 0.001$ , the aircraft goes up when the canyon reference altitude decreases, so these values are not a good option for the controller. Thus,  $q_{11}$  should be somewhere in between 1 and 10.

- **Changes in  $q_{22}$**

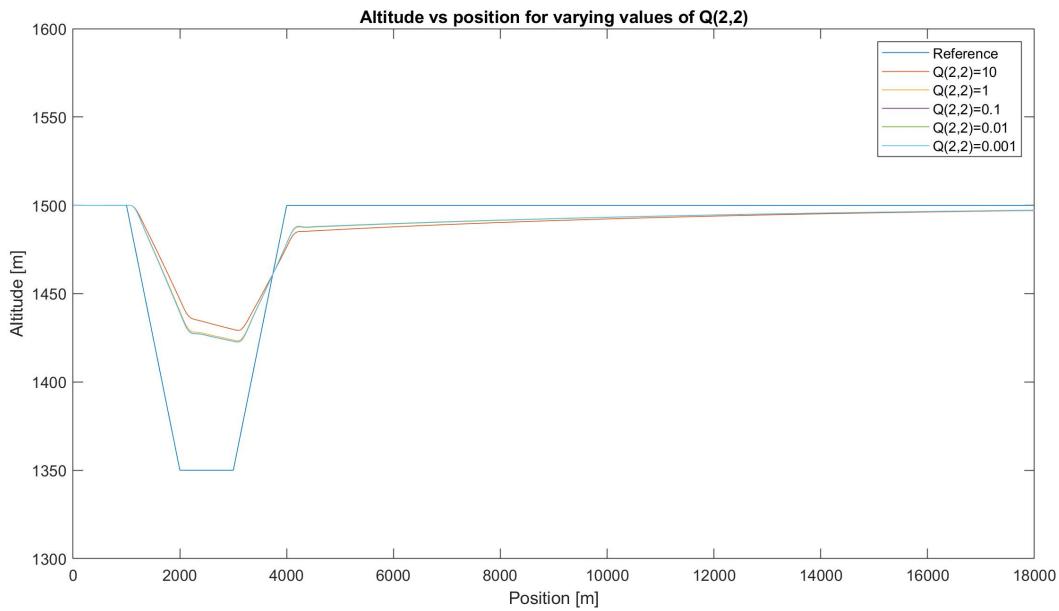


Figure 5.4: Comparison between aircraft altitude and reference altitude for different values of  $q_{22}$ .

Figure 5.4 shows that the element  $q_{22}$  doesn't affect the aircraft behaviour as much as in the previous case. The value  $q_{22} = 10$  presents the highest error. All the other values lead to approximately the same behaviour, and a closer look to the plots reveal that any choice between  $q_{22} = 0.1$ ,  $q_{22} = 0.01$  and  $q_{22} = 0.001$  should result in almost the same control system because the three graphs overlap.

- **Changes in  $q_{33}$**

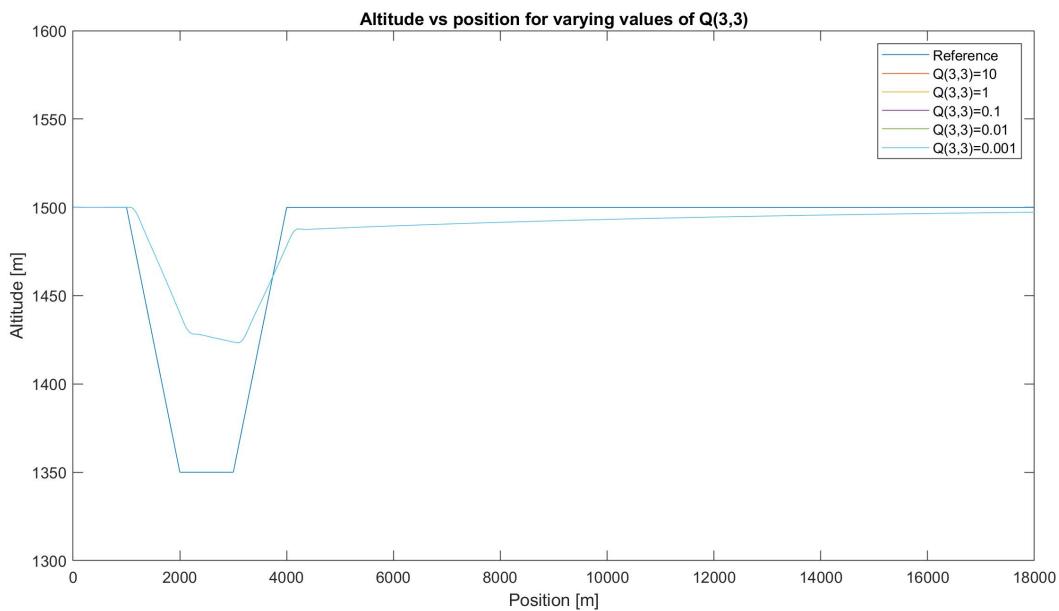


Figure 5.5: Comparison between aircraft altitude and reference altitude for different values of  $q_{33}$ .

- **Changes in  $q_{44}$**

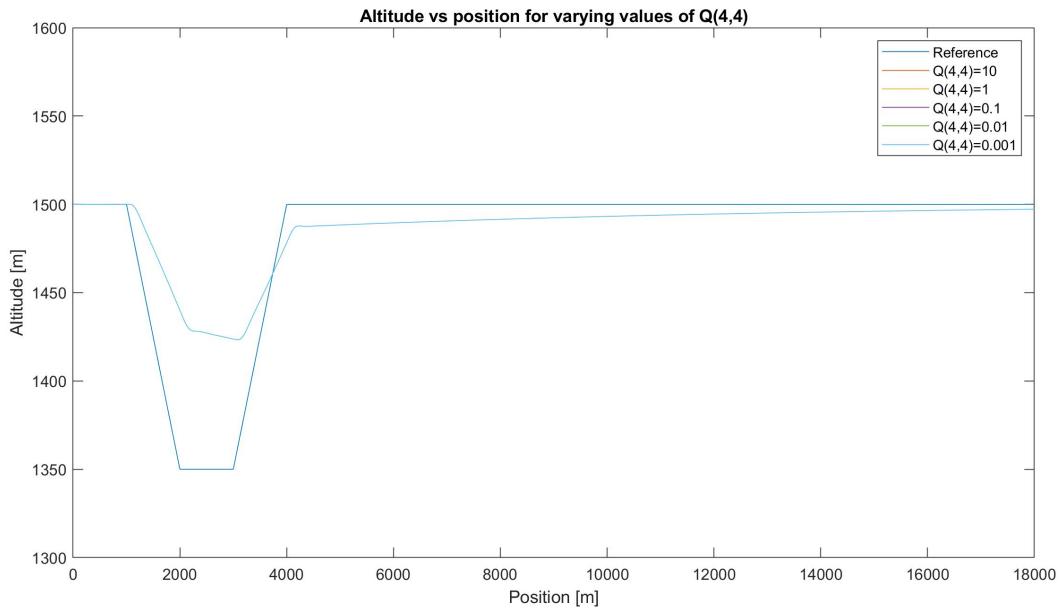


Figure 5.6: Comparison between aircraft altitude and reference altitude for different values of  $q_{44}$ .

- **Changes in  $q_{55}$**

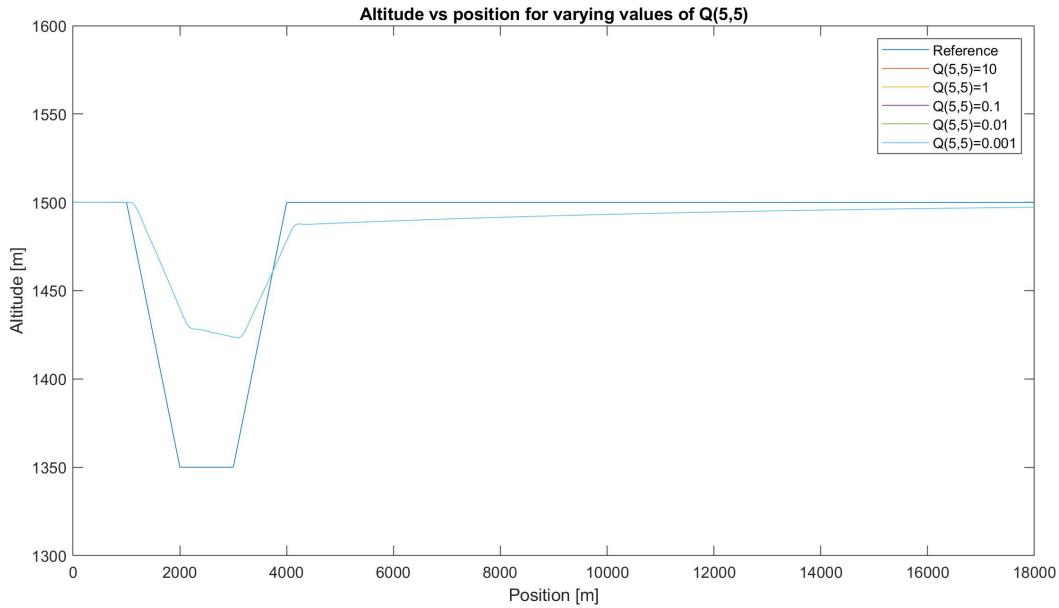


Figure 5.7: Comparison between aircraft altitude and reference altitude for different values of  $q_{55}$ .

Figures 5.5, 5.6 and 5.7 show that the values  $q_{33}$ ,  $q_{44}$  and  $q_{55}$  have almost no influence on the behaviour of the aircraft when it tries to follow the terrain. Although it looks like only the reference altitude (in dark blue) and the result for the value 0.001 (in light blue) are plotted in these figures, in fact the graphs overlap for all values of the set {10, 1, 0.1, 0.01, 0.001}. For that reason, it is reasonable to keep the original value 1 for these diagonal entries.

### 5.5.2. System Behaviour Analysis for varying values of matrix R

- Changes in  $r_{11}$

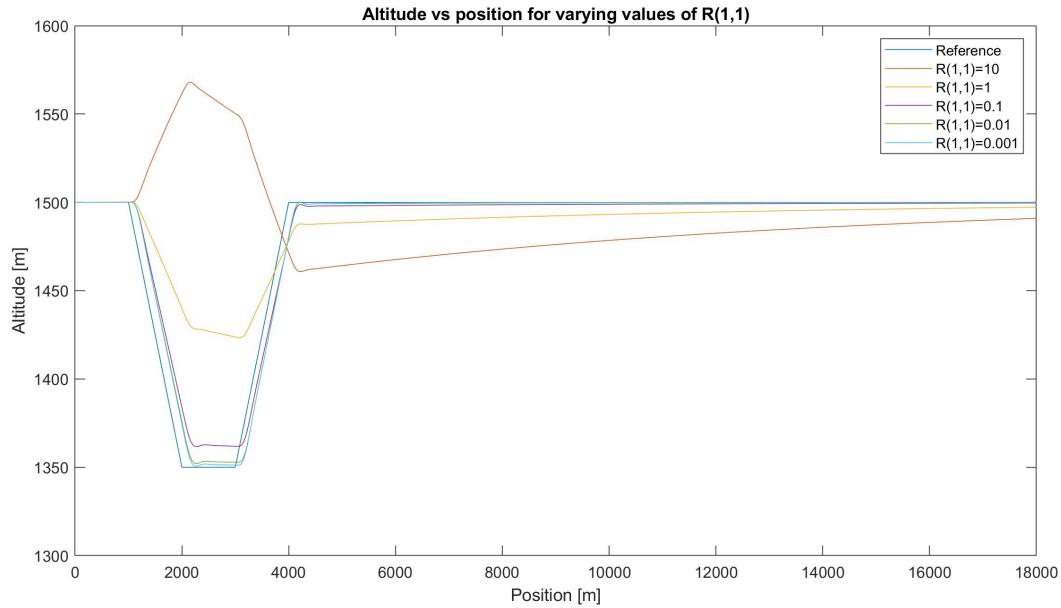


Figure 5.8: Comparison between aircraft altitude and reference altitude for different values of  $r_{11}$ .

Figure 5.8 shows that increasing the value of  $r_{11}$  only leads to greater errors. On the other, the smaller its value is, the better the aircraft follows the reference. For really small values though ( $< 0.01$ ) the improvement is negligible. A good choice for this entry should be close to 0.001.

- Changes in  $r_{22}$

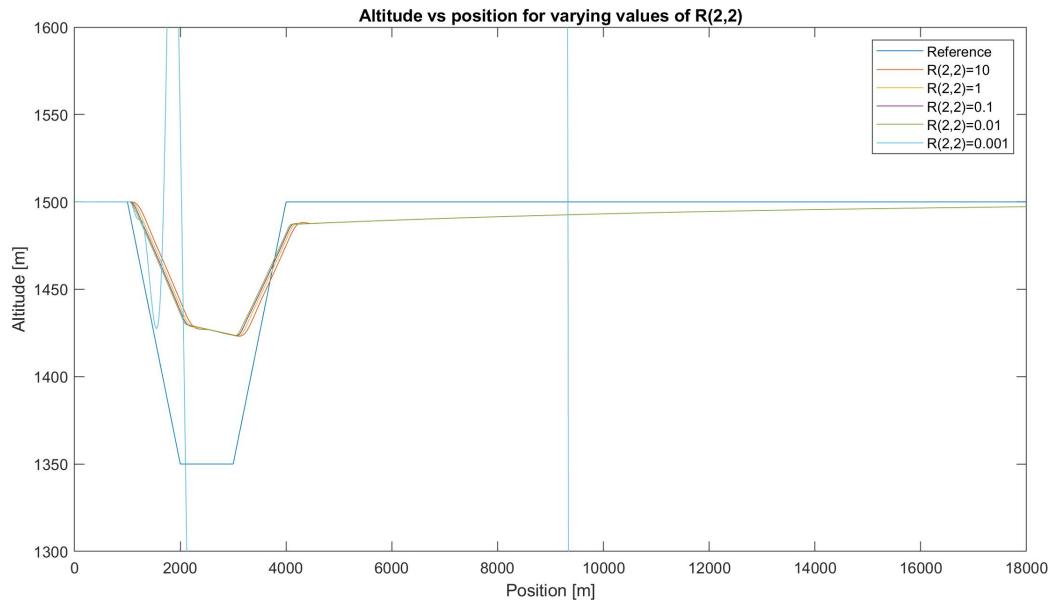


Figure 5.9: Comparison between aircraft altitude and reference altitude for different values of  $r_{22}$ .

From Figure 5.9 it is clear that the system is not stable when  $r_{22} = 0.001$ . After approximately 2000m, the aircraft starts going up really fast and from this point onward the plot loses its meaning and completely diverges. For the remaining values, the differences are relatively small.

## 5.6. Final Result for the LQR Controller

After testing the behaviour of the aircraft for each individual diagonal entry and identifying what could be the best fitting value(s) for them, it is possible to obtain the final result for the LQR controller. One should note that, although the result in Figure 5.9 for the diagonal entry  $r_{22}$  may suggest that for almost all values the difference in the behaviour is relatively small, it turns out that for all values except  $r_{22} = 1$  the response is very sensitive when combined with other values for the remaining diagonal entries and diverges very easily. Thus, the value 1 is to be kept in this position.

The final Q and R matrices are given by:

$$Q = \begin{bmatrix} 10 & 0 & 0 & 0 & 0 \\ 0 & 0.01 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 \end{bmatrix} \quad R = \begin{bmatrix} 0.001 & 0 \\ 0 & 1 \end{bmatrix} \quad (5.9)$$

The result can be seen in Figure 5.10 and the corresponding altitude error is shown in Figure 5.11. The result allows to conclude that this attempt to get a LQR controller is very accurate. The analysis to the error plot confirms that the deviation of the aircraft altitude is always smaller than  $\pm 20$ , just like it was asked. The aircraft will then always be more than 20m from the ground. To check whether the overshoot is smaller than 1m, a more detailed plot of the error is shown in Figure 5.12. The plot shows that in the moments when there is a change in the altitude reference, the aircraft altitude overshoots - visible in the two small peaks. However, both of them are smaller than 1m, so this requirement is also verified. All the requirements are satisfied, so the terrain following system was designed with success.

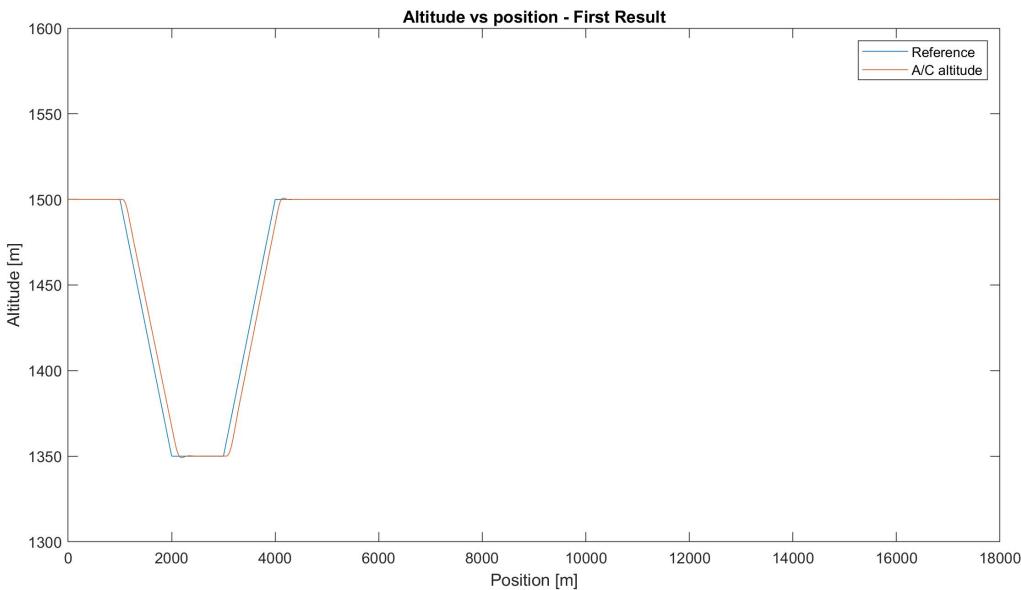


Figure 5.10: Altitude of the aircraft for the final obtained result.

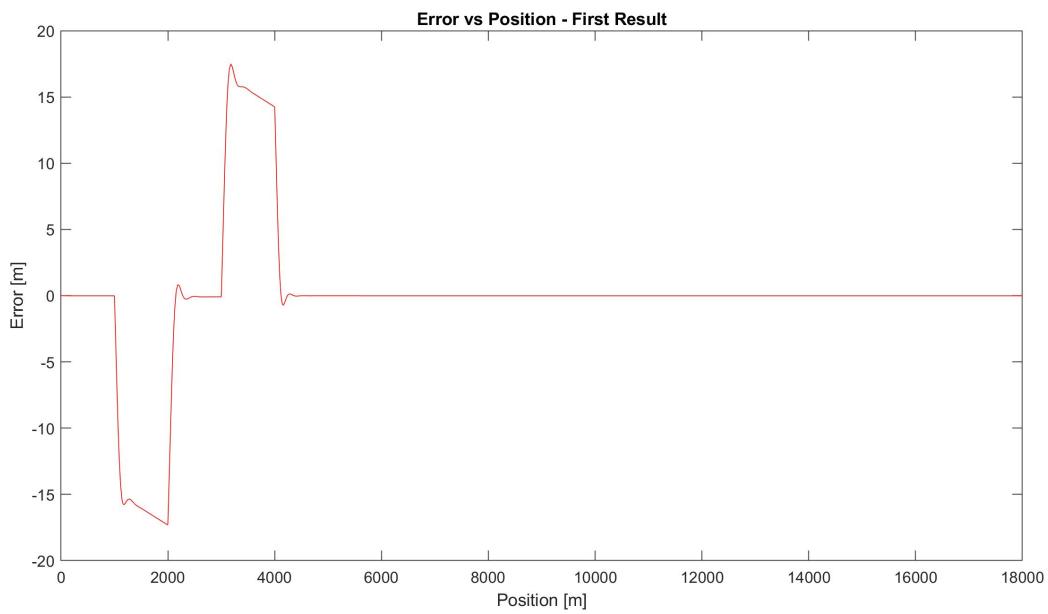


Figure 5.11: Altitude error obtained in the final result.

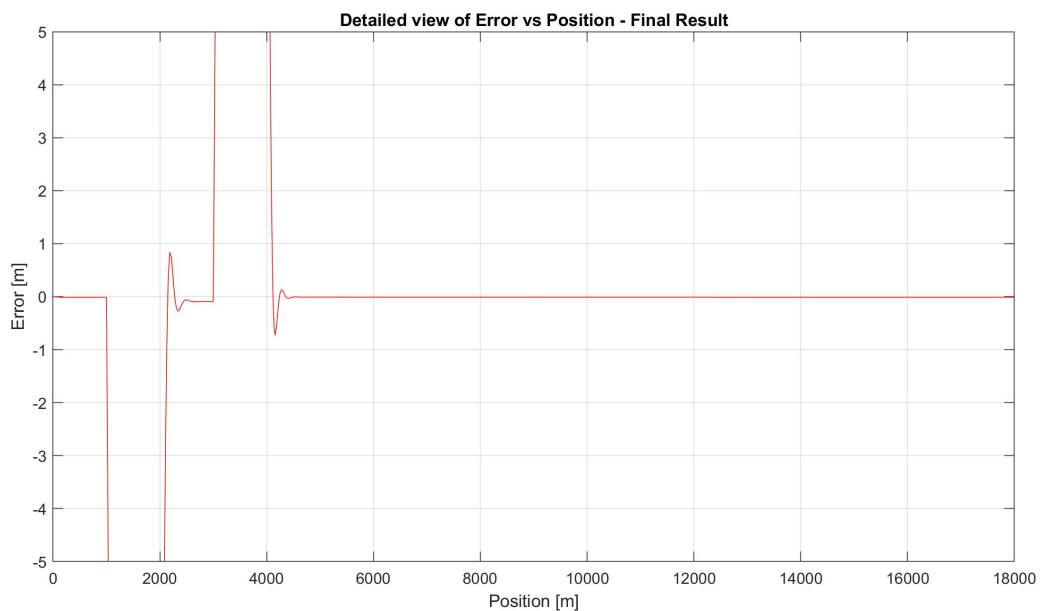


Figure 5.12: Detailed view of the altitude error obtained in the final result.

# 6

## Last Conclusions

The goal of this exercise was to get familiar with classical flight controllers and to gain insight in handling qualities of open-loop and controlled aircraft. By discussing and finishing the exercises successfully, in a group of three students, each of us learned how to apply our control system knowledge in practice. Besides this we got more experiences in making simulations in Simulink.

At last, it should be noted that the calculations done in this report are approximations of the real dynamics of a F16 aircraft, but they are still valid.

# Bibliography

- [1] Richard S. Russell. Non-linear f-16 simulation using simulink and matlab. *University of Minnesota*, Version 1.0, 2003.
- [2] E.-J. van Kampen. Ae4-301p - exercise automatic flight control system design. *Technische Universiteit Delft*, 2018.
- [3] E.-J. van Kampen. Ae4-301 - automatic flight control system design. *Technische Universiteit Delft*, 2018.