

# **Design of some autopilot systems for the Lockheed Martin F-16 model**

*Exercise Automatic Flight Control System Design*

---

Saturday 11<sup>th</sup> February, 2023

Student Number	Name
5834104	Pedroso, Beatriz
5854326	Silva, Pedro
5709172	Yikilmaz, Cansu



Practical Assignment AE4-301P

# Contents

1	Introduction	1
2	Trim and Linearisation	2
2.1	Trim and linearisation procedures	2
2.2	Influence of the accelerometer position.	2
2.2.1	Normal acceleration as an additional output.	3
2.2.2	Analysis on the elevator to normal acceleration transfer function	3
3	Open Loop Analysis	5
3.1	Reduced state space model	5
3.2	Eigenvalues and eigenmotions	5
3.3	Computation of the inherent motion characteristics	6
3.4	Time responses of the eigenmotions	7
4	Design of a pitch rate command system satisfying CAP/Gibson Mil-specs	9
4.1	Short period reduced state space model	9
4.2	Time Responses of the pitch rate	9
4.3	Pitch Rate Command System	10
4.3.1	Gust conditions	10
4.3.2	Lead-lag prefilter	10
4.4	CAP criteria	11
4.5	Gibson criterion	12
4.5.1	Dropback criterion	12
4.6	Neighbouring Flight Condition	13
5	Design of an Automatic Glideslope Following and Flare Controller	15
5.1	Reduced LTI system and trimmed values determination	15
5.2	Aircraft model and control system	16
5.2.1	Actuator dynamics	16
5.2.2	Pitch attitude and airspeed control	17
5.3	Glideslope controller	18
5.3.1	Glideslope geometry	18
5.3.2	Simulink implementation	19
5.3.3	Simulation Results	19
5.4	Flare controller	20
5.4.1	Flare geometry	21
5.4.2	Simulink implementation	22
5.4.3	Simulation results	23
6	Conclusion	25
7	Supporting Material	27
7.1	Simulink Models used in Chapter 5	27
7.2	Simulink Models used in Chapter 8	27

# **Chapter 1**

## **Introduction**

Autopilots have become a crucial part of the aircraft, reducing the workload of the human operators and aiding them during the missions. Hence, as future leading engineers, it is of high importance to know how these work and develop them in order to ensure maximum efficiency and safety. Therefore, the goal of this assignment is to become more familiar with classic flight controllers, as well as to gain insight in handling qualities of open-loop and controlled aircraft. The report is structured as described below.

Firstly, in Section 2, the trimming and linearisation of the F-16 non-linear model for the chosen flight condition is performed. In fact, the resulting state space dynamics will serve as the foundation to design some flight control system components. Moreover, the effect of the position of the accelerometer in the aircraft is also discussed.

Secondly, in Section 3, the reduced state space model is computed and the open loop system dynamics is analysed, in order to investigate the inherent behavior of the aircraft.

Furthermore, in Section 4, a pitch rate command system is designed, which is used for tracking tasks at low velocities. This controller takes into consideration some requirements, namely the Control Anticipation Parameter (CAP) criterion and the Gibson criterion.

Lastly, in Section 5, is it discussed the implementation of an automatic glideslope and flare controller, that allows the aircraft to land securely.

# Chapter 2

## Trim and Linearisation

In this chapter, the F-16 model is trimmed and linearised in two flight conditions. Both high and low fidelity models were taken into account and the results are illustrated in Section 2.1. Furthermore, the influence of the accelerometer position will be analysed in Section 2.2. This will be performed by using the trimmed linear time invariant (LTI) model in the flight condition with an altitude of 15000 ft and a velocity of 500 ft/s.

### 2.1 Trim and linearisation procedures

In order to achieve stability, controllers for aircraft and other diverse systems are designed. Nonetheless, these are mostly based on linear models. Linearising a nonlinear model involves approximating its behaviour around an operating point, which is why the trimming procedure is fundamental to obtain meaningful results. In fact, the trimming procedure involves searching for operating values that are at steady-flight in specific conditions and flight modes. Such is done by minimizing the cost function and by varying the thrust and the other control surfaces until the derivatives are, ideally, null. Thus, the linearised model will consider as initial conditions these equilibrium points and uses the deviations from these set of values in the controller designed.

Resorting to the MATLAB file *FindF16Dynamics*, the F-16 model was trimmed and linearised, in the following flight conditions:

- (I) **Flight condition 1:** Altitude: 15000ft, Velocity: 500ft/s;
- (II) **Flight condition 2:** Altitude: 20000ft, Velocity: 600ft/s.

For these parameters, the results of the cost function for both high and low fidelity models are summarized in Table 2.1

Flight Condition	Fidelity	Cost
$h = 15000 \text{ ft}$	High	$7.1856\text{e-}06$
$V = 500 \text{ ft/s}$	Low	$5.5386\text{e-}29$
$h = 20000 \text{ ft}$	High	$4.4774\text{e-}06$
$V = 600 \text{ ft/s}$	Low	$4.9829\text{e-}29$

Table 2.1: Cost results for both high and low fidelity models, after one iteration.

For all situations, only one iteration was necessary. For both flight conditions, evaluating in steady wings-level flight, the cost function values of the high fidelity model are on the order of  $10^{-6}$ . For the low fidelity model, in the same conditions, the converged cost is on the order of  $10^{-29}$ . Hence, according to the F-16 Manual [1], the results are reliable.

### 2.2 Influence of the accelerometer position

An accelerometer has crucial and multiple utilities on board. In fact, not only is it important for the pilot to have precise control over his Zb-axis g-load during high-g maneuver, but it can also stabilize the short period

mode, since it contains a component proportional to the angle of attack, amongst other functionalities.

### 2.2.1 Normal acceleration as an additional output

Moreover, in case the accelerometer is positioned in the body Xb-axis, at a distance  $x_a$  forward of the aircraft center of gravity (cg) and considering the aircraft is not rolling and yawning, the transport acceleration on that point can be computed by the expression

$$a_n = n_z + \frac{\dot{q}x_a}{g_D}. \quad (2.1)$$

The values of the equation are in g units and  $n_z = -\frac{a_z}{g_D}$ , where  $g_D$  is the standard gravity  $9.80665 \text{ m/s}^2$ . [1]

Equation 2.1 was directly implemented in the *Simulink* file *lin\_f16block*, in order to add it as an additional output. The changes to the given file can be observed in Figures 7.1 and 7.2, which are present in the Appendices Section.

The model was trimmed and linearised in the flight condition (I). Adding this extra state resulted in the C matrix having an extra row. For a  $x_a = 0$ , the linearized output equation is  $y = Cx + Du$ , where

$$\mathbf{C} = [0 \ 0 \ 0.004 \ 9.9298 \ 0.9664 \ 0 \ 0.0208], \quad (2.2)$$

$$\mathbf{x} = [h \ \theta \ V_t \ \alpha \ q \ \delta_{th} \ \delta_{el}]^T, \text{ and } \mathbf{D} = [0].$$

Thus, it can be observed that  $a_n$  only depends on the third, fourth, fifth, and seventh state. These correspond to the states  $V_t$ ,  $\alpha$ ,  $q$ , and  $\delta_{el}$ , respectively.

### 2.2.2 Analysis on the elevator to normal acceleration transfer function

Furthermore, the state space matrices obtained in MATLAB were used to calculate the elevator-to-normal-acceleration transfer function, which is illustrated in Equation 2.3.

$$\frac{a_n}{\delta_e} = \frac{0.421s^5 - 1.84s^4 - 22.15s^3 + 0.077s^2 + 0.001231s}{s^6 + 21.73s^5 + 33s^4 + 41.48s^3 + 0.5546s^2 + 0.293s} \quad (2.3)$$

Given the aforementioned transfer function, the normal accelerations response to a negative step elevator command was plotted and it is illustrated in Figure 2.1.

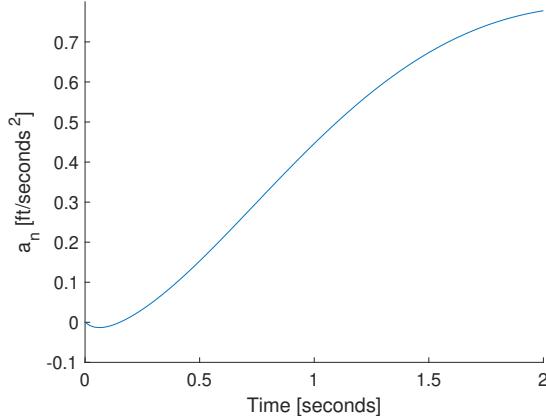


Figure 2.1: Normal acceleration response to a negative step elevator command at  $x_a = 0$  [ft].

The behaviour that can be witnessed in the first instances of the simulation is due to the zeros of the transfer function that are located in the open right hand side of the complex plane.

The zeros of the transfer function are the values  $s=9.76$ ,  $s=-5.39$ ,  $s=0.01$ ,  $s=-0.006$ , and  $s=0$ . As so, there are two non-minimum-phase zeros,  $s=9.76$  and  $s=0.01$ . These are responsible for the response being initially in the opposite direction of the reference signal.

Furthermore, non-minimum phase behaviours occur frequently when pitching up an aircraft to increase its altitude. If the pilot commands a negative deflection of the elevator (up), a downward force is produced on

the tail that pushes it down relative to the center of gravity. As a result, the angle of attack of the aircraft rises since the lift on the main wing causes it to pitch upwards and, thus, climb. Nonetheless, before the increase in altitude, the aircraft goes down due to the reduced net lift in the center of mass. This is the cause of the undershoot that can be observed in the first instances in Figure 2.1.

Additionally, it is interesting to analyse how the position of the accelerometer in the aircraft and the zeros of the resulting transfer function influence this behaviour. Hence, the zeros of the transfer function for different values of  $x_a$  are computed and, for these  $x_a$  values, the  $a_n$  response to a negative elevator command is also computed. The results are illustrated in Figures 2.2 and 2.3.

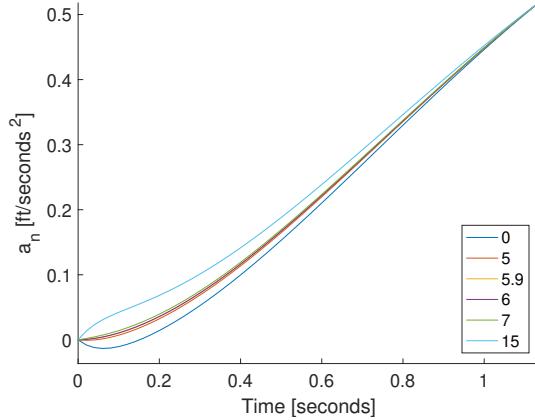


Figure 2.2: Normal acceleration response to a negative step elevator command for different values of  $x_a$ .

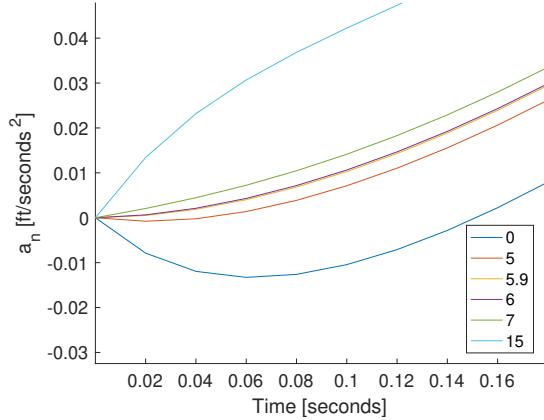


Figure 2.3: Normal acceleration response to a negative step elevator command for different values of  $x_a$ .

As discussed above, for  $x_a = 0$ , there is a zero in  $s=9.76$ , that is responsible for the undershoot. For increasing values of  $x_a$ , there are no significant zeros on the open right side plane. Therefore, when the accelerometer is placed in a position sufficiently forward from the center of gravity and by passing the instantaneous center of rotation (ICR), the undershoot is no longer occurring. This information is corroborated by the zoomed plot represented in Figure 2.3.

Hence, the ICR is located between 5ft and 5.9ft. This should be where the accelerometer is located, since it has more accuracy in the measurements of normal acceleration. Consequently, this point is also the most adequate choice to place the pilot's station, where the angular velocity is null. In fact, if it would be placed either after or before, rotational accelerations would be felt, as it was previously discussed.

Additionally, it needs to be highlighted that it is important to place the accelerometer close to a node of the most important fuselage bending mode in order to accurately measure the bending forces acting on the aircraft. As a result, the accelerometer can, with greater accuracy, measure the changes in acceleration, since the bending forces will be reduced near the node. Hence, more reliable information on the aircraft's condition and performance can be given to the pilot.

# Chapter 3

## Open Loop Analysis

The current chapter is concerned about the open loop analysis of the trimmed and linearised F-16 model. Flight condition (II), with altitude of 20000 ft and velocity of 600 ft/s, is the one considered. Firstly, so that this assessment is conducted, the LTI state space model is reduced to the familiar flight dynamics form of the fourth order. This procedures are presented in Section 3.1. Secondly, in Section 3.3, the motion characterists of the open loop model are computed, so that the inherent flight qualities of the aircraft can be determined.

### 3.1 Reduced state space model

The actuator dynamics have been neglected. As a result, there is no direct connection between inputs and states. Therefore, the ordinary reduction procedure, by selecting the adequate rows and columns of the state space matrices, would result in the input matrix  $\mathbf{B} = 0$ . As so, the alternative procedure recommended in the assignment instructions [3] was implemented.

The reduced models for both longitudinal and lateral directions are presented in Equations 3.1 and 3.2, respectively.

For longitudinal flight modes, the states of interest are the airspeed,  $V_t$ , the angle of attack,  $\alpha$ , the pitch angle,  $\theta$ , and the pitch rate,  $q$ . The resulting state space equations, of the form  $\dot{x} = Ax + Bu$ , are as follows:

$$\begin{bmatrix} \dot{\theta} \\ \dot{V}_t \\ \dot{\alpha} \\ \dot{q} \end{bmatrix} = \begin{bmatrix} 0 & 0 & 0 & 1 \\ -32.1700 & -0.0109 & -1.7611 & -0.8207 \\ 0 & -0.0002 & -0.6505 & 0.9482 \\ 0 & 0 & -1.9092 & -0.8893 \end{bmatrix} \cdot \begin{bmatrix} \theta \\ V_t \\ \alpha \\ q \end{bmatrix} + \begin{bmatrix} 0 & 0 \\ 0.0016 & 0.1093 \\ 0 & -0.0014 \\ 0 & -0.1389 \end{bmatrix} \cdot \begin{bmatrix} \delta_{th} \\ \delta_{el} \end{bmatrix} \quad (3.1)$$

For lateral flight modes, the states of interest are the sideslip angle,  $\beta$ , the roll angle,  $\phi$ , the roll rate,  $p$ , and the yaw rate,  $r$ . The resulting state space equations, of the form  $\dot{x} = Ax + Bu$ , are as follows:

$$\begin{bmatrix} \dot{\phi} \\ \dot{\beta} \\ \dot{p} \\ \dot{r} \end{bmatrix} = \begin{bmatrix} 0 & 0 & 1 & 0.0595 \\ 0.0535 & -0.2055 & 0.0594 & -0.9941 \\ 0 & -25.8584 & -2.3166 & 0.4924 \\ 0 & 7.2786 & -0.0294 & -0.3191 \end{bmatrix} \cdot \begin{bmatrix} \phi \\ \beta \\ p \\ r \end{bmatrix} + \begin{bmatrix} 0 & 0 \\ 0.0002 & 0.0005 \\ -0.5576 & 0.0755 \\ -0.0309 & -0.0571 \end{bmatrix} \cdot \begin{bmatrix} \delta_a \\ \delta_r \end{bmatrix} \quad (3.2)$$

### 3.2 Eigenvalues and eigenmotions

So that the inherent motion characteristics can be computed, the eigenvalues of the A matrices represented in Equation 3.1 have to be determined. Moreover, Figures 3.2 and 3.2 illustrate the longitudinal and lateral poles of the reduced model, respectively.

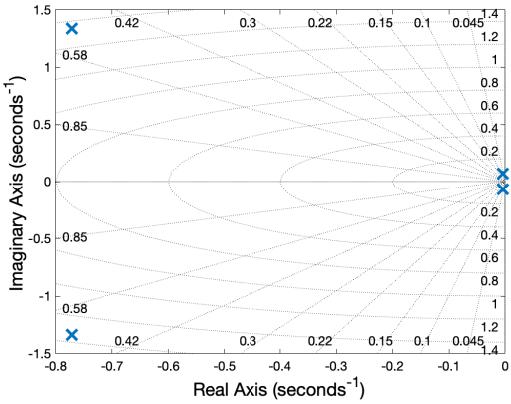


Figure 3.1: Longitudinal poles of the reduced model.

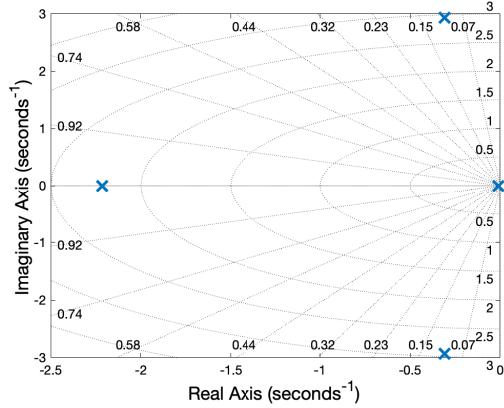


Figure 3.2: Lateral poles of the reduced model.

It can be observed that the system is dynamically stable, since all poles are located in the open left half plane.

On the one hand, the longitudinal eigenmotions are Phugoid and Short Period. Both are periodic, since the pole pairs are complex. As it can be observed in Figure 3.1, the Short Period is the most distanced from the origin, since it is more damped. On the contrary, Phugoid, related to the pair closer to the origin, is very little damped, which causes it to oscillate for a longer time. The poles and respective eigenmotions are present in Table 3.1.

Eigenmotion	Poles
Short Period	$-0.7713 \pm 1.3392i$
Phugoid	$-0.0040 \pm 0.0675i$

Table 3.1: Longitudinal eigenmotions and respective poles.

On the other hand, the lateral eigenmodes consist of the Spiral, the Aperiodic Roll, and the Dutch Roll. Only the latter is a complex pair. Thus the first two are aperiodic. As it can be seen in Figure 3.2, the pole more distanced from the origin belongs to the Aperiodic Roll, which is faster, when compared to the Spiral. The poles and respective eigenmotions are present in Table 3.2.

Eigenmotion	Poles
Spiral	$-0.0101$
Aperiodic Roll	$-2.2146$
Dutch Roll	$-0.3083 \pm 2.9290i$

Table 3.2: Lateral eigenmotions and respective poles.

### 3.3 Computation of the inherent motion characteristics

So that the inherent behaviour of the F-16 model can be analysed, some characteristics are computed. For the periodic eigenmotions, namely Short Period, Phugoid, and Dutch Roll, natural frequency ( $w_n$ ), damping ratio ( $\xi$ ), period ( $P$ ), and time to damp to half amplitude ( $T_{1/2}$ ) will be computed. For the aperiodic eigenmotions, namely Spiral and Aperiodic Roll, natural frequency ( $w_n$ ), time constant ( $\tau$ ), and time to damp to half amplitude ( $T_{1/2}$ ) will be computed.

Moreover,  $w_n$  and  $\xi$  were calculated resorting to the MATLAB function `damp()`. Equation 3.3 was used to calculate the Period, Equation 5.17 to compute  $\tau$  and Equation 3.5 to determine  $T_{1/2}$ .  $\text{Re}(p)$  and  $\text{Im}(p)$  are, respectively, the real and the imaginary part of the corresponding pole.

$$P = \frac{2\pi}{\text{Im}(p)}. \quad (3.3)$$

$$\tau = \frac{1}{w_n \xi}. \quad (3.4)$$

$$T_{1/2} = \frac{\ln(\frac{1}{2})}{Re(p)}. \quad (3.5)$$

These values for longitudinal and lateral eigenmotions are summarized in Tables 3.3 and 3.4.

Eigenmotion	$w_n[\text{rad/s}]$	$\xi$	$P[\text{s}]$	$\tau[\text{s}]$	$T_{1/2}[\text{s}]$
Short Period	1.5454	0.4991	4.6918	-	0.8986
Phugoid	0.0676	0.0588	93.1239	-	174.5140

Table 3.3: Longitudinal motion characteristics.

Eigenmotion	$w_n[\text{rad/s}]$	$\xi$	$P[\text{s}]$	$\tau[\text{s}]$	$T_{1/2}[\text{s}]$
Spiral	0.0101	-	-	99.1889	68.7525
Aperiodic Roll	2.2146	-	-	0.4515	0.3130
Dutch Roll	2.9452	0.1047	2.1452	-	2.2485

Table 3.4: Lateral motion characteristics.

## 3.4 Time responses of the eigenmotions

In order to support the values computed in Section 3.3, the eigenmotions time responses to a step input were plotted.

As it can be observed in Figures 4.5 and 4.6, both eigenmotions are periodic and damped. Clearly, the **short period** is more damped ( $\xi \approx 0.5$ ), while the **phugoid** has more oscillations and for a longer time ( $\xi \approx 0.06$ ), which is corroborated by the values in Table 3.3. Additionally, since the poles of the short period are located further away from the origin, this eigenmode has a faster response, which is also in accordance with the period. For this eigen mode, its value is of around 5 seconds, whilst for the phugoid, it is around 93 seconds.

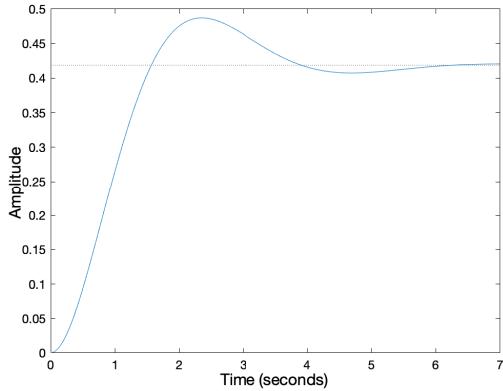


Figure 3.3: Short period time response to a step input.

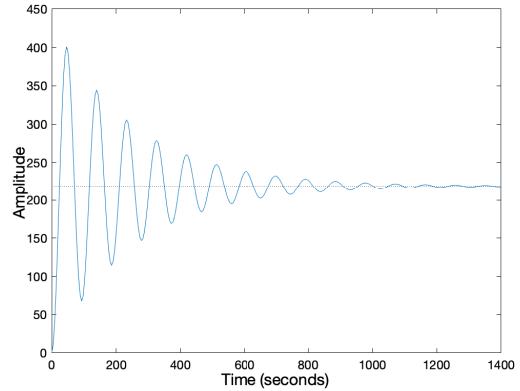


Figure 3.4: Phugoid time response to a step input.

Furthermore, Figures 3.5 and 3.6, illustrate the two aperiodic eigenmodes, namely **spiral** and **aperiodic roll**. As so, these do not have a period. Similarly, to what was described previously, the spiral has its poles near the origin, the opposite of the aperiodic roll. Therefore, the latter has a much faster response. This is also verified by the values of the time constant, which is around 99 seconds for the spiral and around 0.45 seconds for the roll. Hence, the time to damp to half amplitude follows the same behaviour.

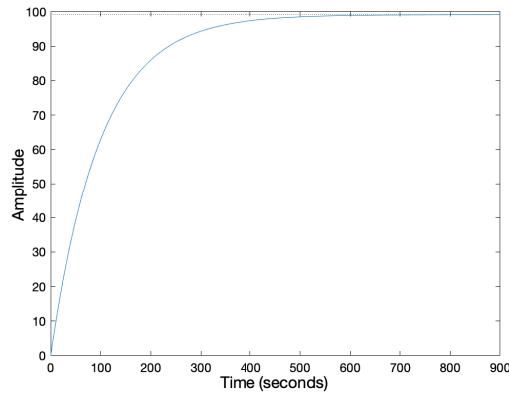


Figure 3.5: Spiral time response to a step input.

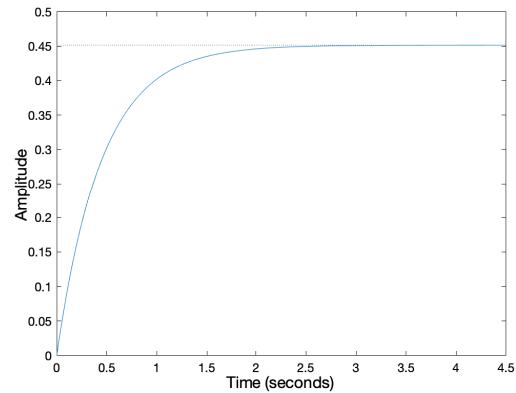


Figure 3.6: Aperiodic Roll time response to a step input.

Ultimately, the **dutch roll** is a periodic eigenmotion, with a period of around 2.25 seconds and it is slightly damped, with a damping ratio of 0.1047. These values, which are given in Table 3.4 are in accordance with Figure 3.7.

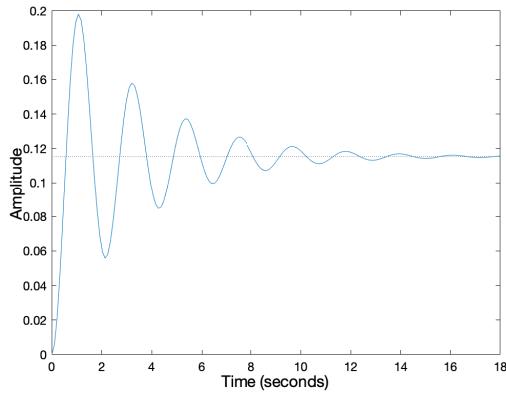


Figure 3.7: Dutch roll time response to a step input.

# Chapter 4

## Design of a pitch rate command system satisfying CAP/Gibson Mil-specs

### 4.1 Short period reduced state space model

The reduced state space model is constructed following the procedure presented in chapter 3. Reduction is made such that only two states' angle of attack  $\alpha$  and pitch rate  $q$  are given in the state space matrix. Note that the effect of actuator dynamics has been neglected.

The resulting state space equations, of the form  $\dot{x} = Ax + Bu$ , are as follows:

$$\begin{bmatrix} \dot{\alpha} \\ \dot{q} \end{bmatrix} = \begin{bmatrix} -0.6505 & 0.9482 \\ -1.9092 & -0.8893 \end{bmatrix} \cdot \begin{bmatrix} \alpha \\ q \end{bmatrix} + \begin{bmatrix} -0.0014 \\ -0.1389 \end{bmatrix} \cdot [\delta_{el}] \quad (4.1)$$

### 4.2 Time Responses of the pitch rate

In this section, time responses of the pitch rate to a step input for 2-state and 4-state models are given in the figures below. Figure 4.5 shows the short-term response whereas Figure 4.6 gives the long-term response of the pitch rate. While the pitch rate for the short reduced model is kept constant after 6 seconds, 4 state model shows an oscillatory behavior and then sits on the value of zero under the effects of phugoid mode. The reduced model ignores the contributions coming from the phugoid mode and keeps its non-zero value constant during simulations. Graphs indicate that the short-period mode is being dominant mode for a short period of time and it is a useful and logical way to analyze aircraft dynamics under these conditions. In the long run, phugoid mode will become the dominant mode and the current reduced model will not be useful anymore for design purposes.

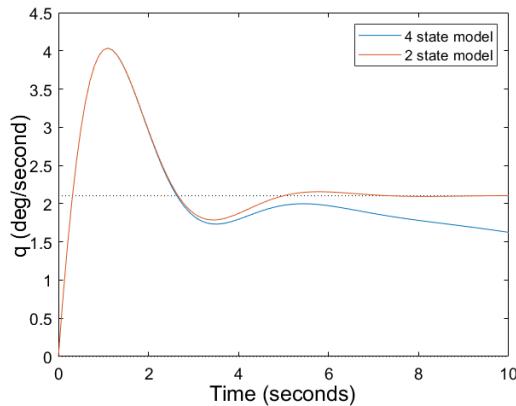


Figure 4.1: Time responses of the pitch rate to a step input (short-term response).

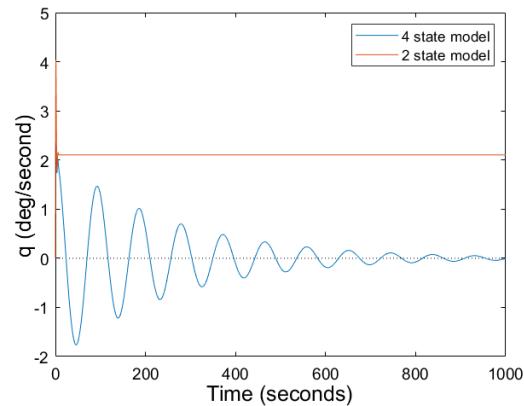


Figure 4.2: Time responses of the pitch rate to a step input (long-term response).

### 4.3 Pitch Rate Command System

To design a controller system that satisfies the requirements of the CAP and GAP criteria, following relations are computed.

$$\omega_{nsp}(V, h) = 0.03V(V, h) = 5.4864 \text{ [rad/s]} \quad (4.2)$$

$$1/T_{\theta_2}(V, h) = 0.75\omega_{nsp}(V, h) = 4.1148 \text{ [1/s]} \quad (4.3)$$

$$\zeta_{sp}(V, h) = 0.5 \quad (4.4)$$

To satisfy the values calculated above, the design of a pitch rate controller system is required. Since the desired short-period frequency and damping ratio of the closed-loop system are known, the poles of the system can be calculated by finding the zeros of the equation given below.

$$s^2 + 2\zeta_{sp}\omega_{nsp}s + \omega_{nsp}^2 \quad (4.5)$$

Poles of the closed-loop system are given as:

$$\begin{aligned} p_1 &= -2.7432 + 4.7514i \\ p_2 &= -2.7432 - 4.7514i \end{aligned} \quad (4.6)$$

To obtain the feedback gains  $K_\alpha$  and  $K_q$ , the pole placement method is used. Using the Matlab built-in function **pole**, gains are calculated as follows. Note that units for  $K_\alpha$  and  $K_q$  are  $\circ/rad$  and  $\circ/(rad/s)$ , respectively.

$$\begin{aligned} K_\alpha &= -190.9868 \text{ [\circ/rad]} \\ K_q &= -26.5239 \text{ [\circ/(rad/s)]} \end{aligned} \quad (4.7)$$

#### 4.3.1 Gust conditions

Considering severe gust (design vertical gust: 4.572m/s) according to MIL-F-8785C, the induced angle of attack and corresponding elevator deflection angle are calculated using the relations given below.

$$\begin{aligned} \alpha_{induced} &= \tan\left(\frac{v_{gust}}{V}\right) = 0.0250 \text{ [rad]} \\ \delta_c &= K_\alpha \alpha_{induced} = -4.7737^\circ \end{aligned} \quad (4.8)$$

It is important to keep the angle of the control surface deflection within allowable margins to obtain a good controller performance. Based on the calculated elevator deflection, it is seen that pre-determined values  $[-25^\circ, 25^\circ]$  are not exceeded indicating that the obtained levels of feedback gains are acceptable with reference to possible gust.

#### 4.3.2 Lead-lag prefilter

Time constant calculated in the Eq. 4.3 cannot be modified by any control loop structure, its modification must be done through pole-zero cancellation using a lead-lag prefilter.

Inside the loop, the system's poles should be preserved to get the desired behavior. That's why, no adjustment can be made that may result in the change of the zeros of the closed loop system which makes it necessary to put the lead-lag filter outside the loop.

Based on the gains calculated above, the new state matrix and the transfer function of the closed loop system are,

$$A_{cl} = A - B[K_\alpha, K_q] \quad (4.9)$$

$$\frac{q(s)}{\delta_{el}} = \frac{-0.1389s - 0.08771}{s^2 + 5.486s + 30.1} \quad (4.10)$$

of which the transfer function can also be represented as it is given in the Eq. 4.11 in terms of time constant and feedback gain  $K_q$ ,

$$\frac{kq(1 + T_{\theta_2} s)}{s^2 + 2\zeta_{sp}\omega_{nsp}s + \omega_{nsp}^2} \quad (4.11)$$

Substituting the values of the pitch rate feedback gain and time constant, the transfer function of the prefilter becomes:

$$\frac{-0.02132s - 0.08771}{s^2 + 5.486s + 30.1} \quad (4.12)$$

It is desired that the steady-state response of the system tracks the given input successfully and for this purpose a feed-forward gain is necessary. By including a feed-forward gain in the overall system, the closed-loop transfer function becomes approximately one minimizing the steady state error and providing a good tracking performance.

Feedforward gain is determined by first calculating the dc gain of the prefilter system. For this, Matlab built-in function **dcgain** is used.

$$K_{ff} = 1/\text{dcgain(filtered system)} = -343.1670 \quad (4.13)$$

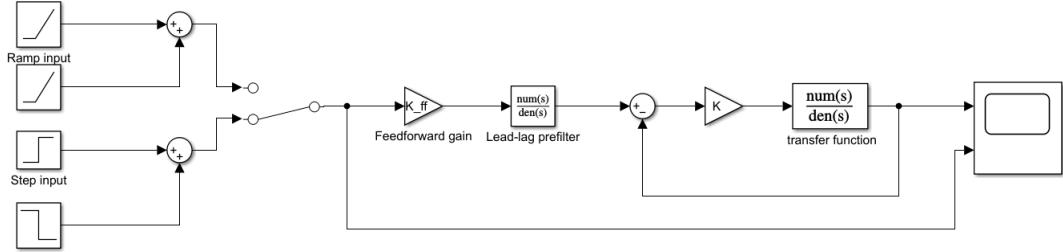


Figure 4.3: Pitch Rate Command System

## 4.4 CAP criteria

To check if the requirements set by the CAP criteria are satisfied, the following relations are computed. As shown in the table, CAP values are calculated for the system before and after the controller based on the given conditions. Figure 4.4 shows the current parameter value and design point corresponding to the damping values given in the table. From the figure, it is clear that the system lies in the region in which level 1 requirements are satisfied.

$$CAP = \frac{\omega_{nsp}^2}{\frac{V}{g} \frac{1}{T_{\theta 2}}} \quad (4.14)$$

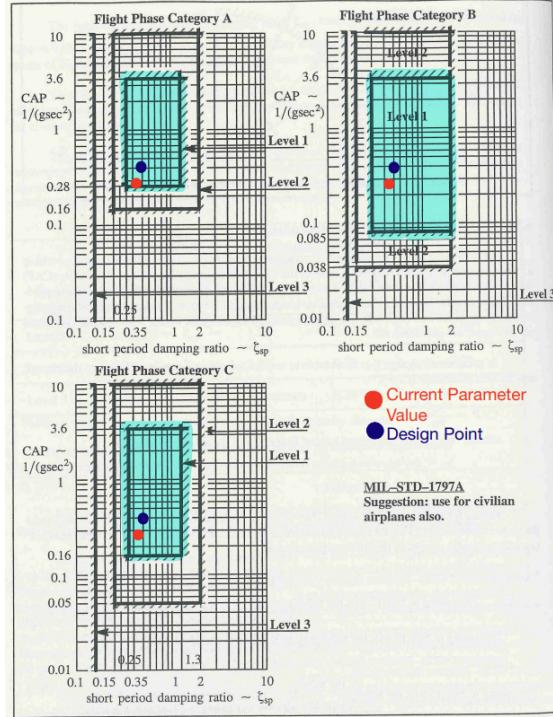


Figure 4.4: Control Anticipation Parameter and short period damping ratio requirements

## 4.5 Gibson criterion

The Gibson Criterion is to understand how to design command and stability augmentation systems with acceptable flying and handling characteristics. It consists of two separate criterion: the dropback criterion and the phase rate

### 4.5.1 Dropback criterion

The Dropback criterion analyzes the pitch rate response of the system by calculating the overshoot and drop-back ratio.

1.  $q_m$ : maximum pitch rate
2.  $q_s$ : steady-state value of pitch rate
3.  $\frac{q_m}{q_s}$  : pitch rate overshoot ratio
4. DB: dropback, amount of negative transition towards final value after the step input has been removed
5. OS: overshoot, amount of positive transition towards final value after the step input has been removed

$$\frac{DB}{q_{ss}} = T_{\theta 2} - \frac{2\zeta_{sp}}{w_{nsp}} \quad (4.15)$$

	$\omega_n$	$T_{\theta 2}$	CAP	$\frac{q_m}{q_s}$	$\frac{DB}{q_{ss}}$
Current Parameter Value	1.5456	1.5836	0.2029	1.9176	0.9390
Design Point	5.4864	0.2430	0.3923	1.4125	0.0608

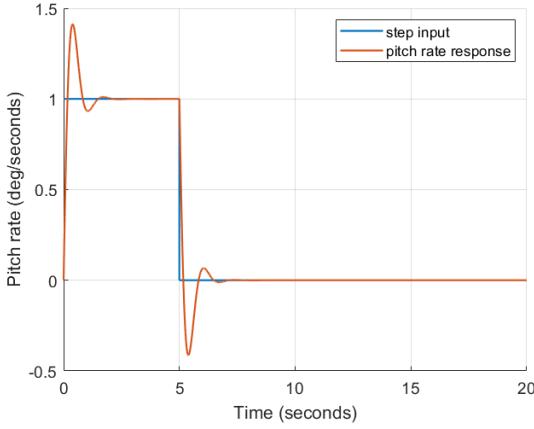


Figure 4.5: Pitch rate response to a step input

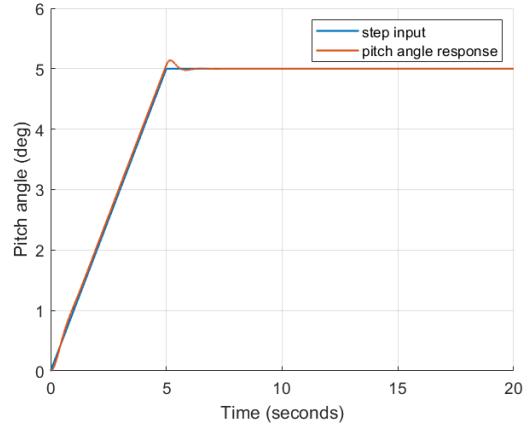


Figure 4.6: Pitch attitude response to a step input

Figure 4.7 shows that the design point lies in the satisfactory region with an acceptable overshoot and dropback ratio. However, the current parameter value is out of the satisfactory region indicating the system before the command system has been applied has an abrupt bobble tendency due to its high dropback ratio.

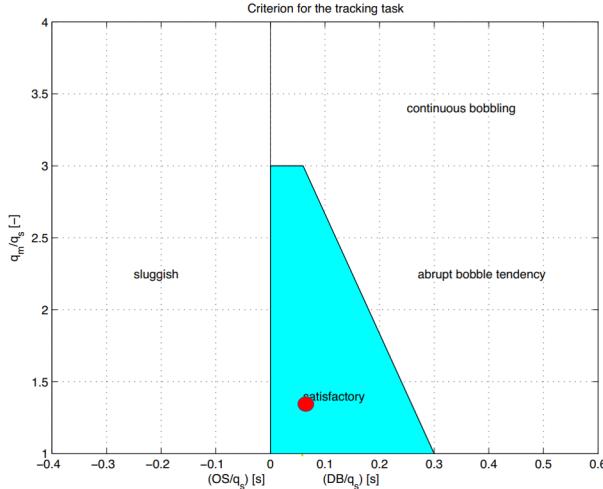


Figure 4.7: The criterion for the tracking task

## 4.6 Neighbouring Flight Condition

The design of a pitch rate command system is repeated for the neighboring flight condition corresponding to a velocity of  $V = 800$  ft/s. To calculate the feedback gains, gain scheduling is used by interpolating the values between the velocity of 600 ft/s and 900 ft/s, and for the rest same procedures used in the previous sections are followed. Results for the neighboring flight condition are given as follows.

$$K_\alpha = -190.1565 \text{ [o/rad]} \quad (4.16)$$

$$K_q = -20.6252 \text{ [o/(rad/s)]}$$

$$\alpha_{induced} = \text{atan}\left(\frac{v_{gust}}{V}\right) = 0.0057 \text{ [rad]} \quad (4.17)$$

$$\delta_c = K_\alpha \alpha_{induced} = -1.0867^\circ$$

Closed loop transfer function:

$$TF = \frac{-0.1389s - 0.08771}{s^2 + 4.666s + 29.47} \quad (4.18)$$

$$K_{ff} = -336.0101 \quad (4.19)$$

$$\begin{aligned} \frac{q_m}{q_s} &= 1.5084 \\ \frac{DB}{q_s} &= 0.0608 \end{aligned} \quad (4.20)$$

Figure 4.8 shows that Dropback criterion is also satisfied for the neighbouring flight condition.

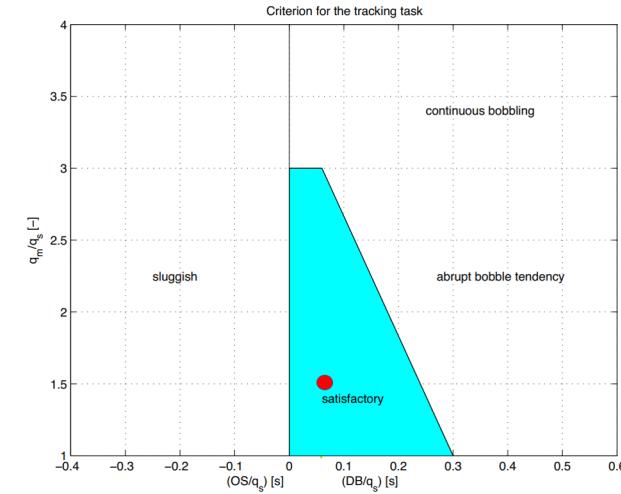


Figure 4.8: Dropback criterion for neighboring flight condition

# Chapter 5

## Design of an Automatic Glideslope Following and Flare Controller

In this chapter, an automatic glideslope following and flare control system for the F-16 is designed and a simulation of said system is carried out. This is done for a flight condition at which the aircraft is at an altitude of 5000 ft and with a velocity of 300 ft/s, with this velocity assumed fairly constant throughout all manoeuvres. The aircraft is considered to intercept a glideslope of  $3^\circ$  that originates from an airfield located at an altitude of 3000 ft, meaning that the interception occurs at 2000 ft above the runway. After this interception, which takes place approximately 10 seconds after the start of the simulation, the aircraft follows the glideslope, descending until a certain altitude is reached, after which a flare sequence is set in motion to ensure a smooth landing with safe vertical speed.

In section 5.1, a five-state reduced model for this flight condition is determined, and the respective trimmed values are obtained. In section 5.2, the aircraft model and its flight control system are presented. Finally, the glideslope and flare controllers are designed and simulated in sections 5.3 and 5.4, respectively.

### 5.1 Reduced LTI system and trimmed values determination

In section 3.1, the 4-state state space system was introduced. Now, it is extended with a fifth state, namely altitude,  $h$ , which is added as the first entry of the state space vector. Equation 5.1 shows how this state's derivative is expressed as a function of the previously defined states.

$$\dot{h} = V \sin(\theta - \alpha) \approx V(\theta - \alpha) \quad (5.1)$$

Therefore, the linearized reduced model considers longitudinal dynamics and is obtained using the methodology previously mentioned in section 3.1. In this case, the states of interest are altitude,  $h$ , pitch attitude angle,  $\theta$ , true velocity,  $V_t$ , angle of attack,  $\alpha$ , and pitch rate,  $q$ . Considering this as the new state vector, the state space for the desired flight condition is obtained and is given below in Equation 5.2.

$$\begin{cases} \begin{bmatrix} \dot{h} \\ \dot{\theta} \\ \dot{V}_t \\ \dot{\alpha} \\ \dot{q} \end{bmatrix} = \begin{bmatrix} 0 & 300 & 0 & -300 & 0 \\ 0 & 0 & 0 & 0 & 1 \\ 1.32 \times 10^{-4} & -32.17 & -0.03 & 2.13 & 2.89 \\ 3.15 \times 10^{-6} & 4.55 \times 10^{-13} & -6.97 \times 10^{-4} & -0.54 & 0.92 \\ -4.70 \times 10^{-21} & 0 & 1.04 \times 10^{-18} & 0.33 & -0.82 \end{bmatrix} \cdot \begin{bmatrix} h \\ \theta \\ V_t \\ \alpha \\ q \end{bmatrix} + \begin{bmatrix} 0 & 0 \\ 0 & 0 \\ 1.5 \times 10^{-3} & -4.5 \times 10^{-3} \\ -9.49 \times 10^{-7} & -1.10 \times 10^{-3} \\ 0 & -5.70 \times 10^{-2} \end{bmatrix} \cdot \begin{bmatrix} \delta_{th} \\ \delta_{el} \end{bmatrix} \\ \begin{bmatrix} h \\ \theta \\ V_t \\ \alpha \\ q \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 57.29 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 57.29 & 0 \\ 0 & 0 & 0 & 0 & 57.29 \end{bmatrix} \cdot \begin{bmatrix} h \\ \theta \\ V_t \\ \alpha \\ q \end{bmatrix} + \begin{bmatrix} 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \end{bmatrix} \cdot \begin{bmatrix} \delta_{th} \\ \delta_{el} \end{bmatrix} \end{cases} \quad (5.2)$$

An important thing to consider is the fact that the states of the system are only deviations from the trimmed values. This means that, in order to obtain the total values of the states, their trimmed values must be added. These are obtained using the low fidelity model for the specific flight condition and are presented below in Table 5.1.

Parameter	Value [Unit]
$h_{trim}$	5000 [ft]
$\theta_{trim}$	10.4511 [°]
$V_{t,trim}$	300 [ft/s]
$\alpha_{trim}$	10.4511 [°]
$q_{trim}$	0 [°/s]
$\delta_{th,trim}$	2826.8165 [lb]
$\delta_{el,trim}$	-4.1891 [°]

Table 5.1: Trimmed values obtained from the low fidelity model for  $h = 5000$  ft and  $V = 300$  ft/s.

## 5.2 Aircraft model and control system

Both the aircraft model and the aircraft flight control system (AFCS) are implemented in Simulink inside the "Aircraft + AFCS" subsystem seen in Figure 7.3. The aircraft model includes the actuator dynamics for engine and elevator, while the AFCS includes pitch attitude and velocity control. Again, it is important to note that the outputs of this subsystem are the deviations of the states around the trim condition. These values are relevant to the AFCS, but are insufficient for the glideslope and flare control. Therefore, as can also be seen in Figure 7.3, the trimmed values are added to these deviations. Finally, the airfield altitude is subtracted from the total altitude, allowing for the true altitude above the runway to be considered.

A full diagram of the "Aircraft + AFCS" subsystem can be consulted in Figure 7.6.

### 5.2.1 Actuator dynamics

As can be seen in Figure 7.6, the actuator dynamics consists of a subsystem whose inputs are the reference values for the deviations in engine thrust and elevator deflection. Its outputs are, then, the deviations in engine thrust and elevator deflection, which are fed to the reduced state space model.

Figure 7.7 shows the implementation of these dynamics in Simulink.

As can be seen, the transfer function representing the dynamics of the engine is given by Equation 5.3,

$$H_{engine} = \frac{1}{s+1}, \quad (5.3)$$

while Equation 5.4 is the transfer function used to represent the elevator actuator dynamics,

$$H_{elevator} = \frac{20.2}{s+20.2}. \quad (5.4)$$

These transfer function blocks are linked in series with saturation blocks, that model the saturation limits of the total values of engine thrust,  $\delta_{th,sat}$ , and elevator deflection,  $\delta_{el,sat}$ .

These limits are compiled below in Table 5.2.

Parameter	Lower Limit	Upper Limit	Unit
$\delta_{th,sat}$	1000	19000	lb
$\delta_{el,sat}$	-25	25	°

Table 5.2: Total Engine Thrust and Elevator Deflection Saturation Limits.

Because these are total values, and our subsystem deals with deviations, we must subtract the engine thrust and elevator deflection trimmed values from these limits, to obtain the saturation limits for the engine thrust and elevator deflection deviations. These limits are shown below in Table 5.3.

Parameter	Lower Limit	Upper Limit	Unit
$\delta_{th_{sat}}$	-1826.8165	16173.1835	lb
$\delta_{el_{sat}}$	-20.8109	29.1891	°

Table 5.3: Engine Thrust and Elevator Deflection Deviation Saturation Limits.

### 5.2.2 Pitch attitude and airspeed control

The pitch attitude and airspeed control systems are conceptually represented by the block diagrams of Figures 5.1 and 5.2, respectively.

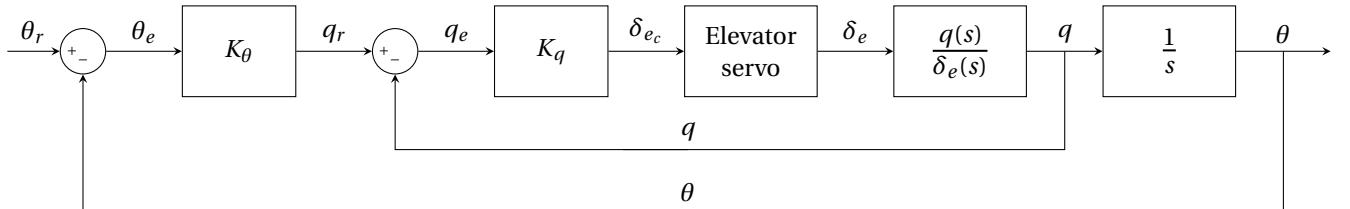


Figure 5.1: Pitch attitude control block diagram.

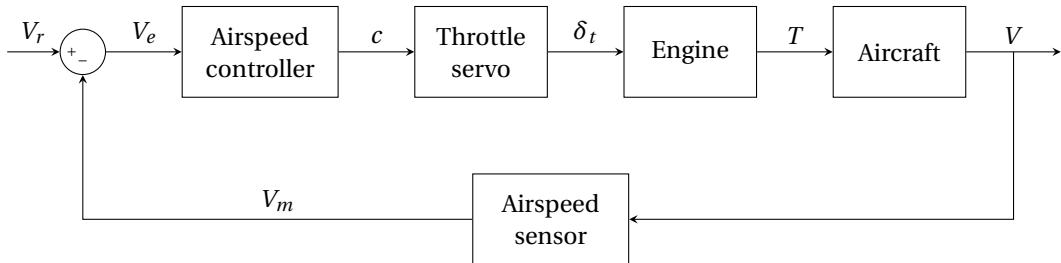


Figure 5.2: Velocity control system block diagram.

These systems are incorporated in Simulink and can be seen in Figure 7.6. The inputs are the reference values for the true velocity and pitch angle attitude deviations. The first is set to zero, since the goal of the controller is to achieve constant velocity of 300 ft/s (therefore null deviation is expected). The second is obtained from either the glideslope or flare control systems, depending on which flight phase is taking place.

The velocity controller was implemented as a PID controller tuned using the critical gain Ziegler–Nichols tuning method. The pitch attitude controller is divided in two parts, namely a pitch angle and pitch rate controllers. These are implemented as P controllers in Simulink and these were tuned using the auto-tuning tool. The Gains for the velocity and pitch attitude controllers are compiled below in Table 5.4.

Controller Type	P	I	D
Velocity	240	5	1.25
Pitch Angle	5	-	-
Pitch rate	-5	-	-

Table 5.4: Tuned gains for the velocity and pitch attitude controllers.

### 5.3 Glideslope controller

Conceptually, the glideslope controller is represented by the block diagram of Figure 5.3, where the total state values are used, and the glideslope error is defined as  $\Gamma$ . The goal, thus, is to follow a glideslope of  $3^\circ$ , while keeping the glideslope error angle at a reference value,  $\Gamma_r$ , of  $0^\circ$ .

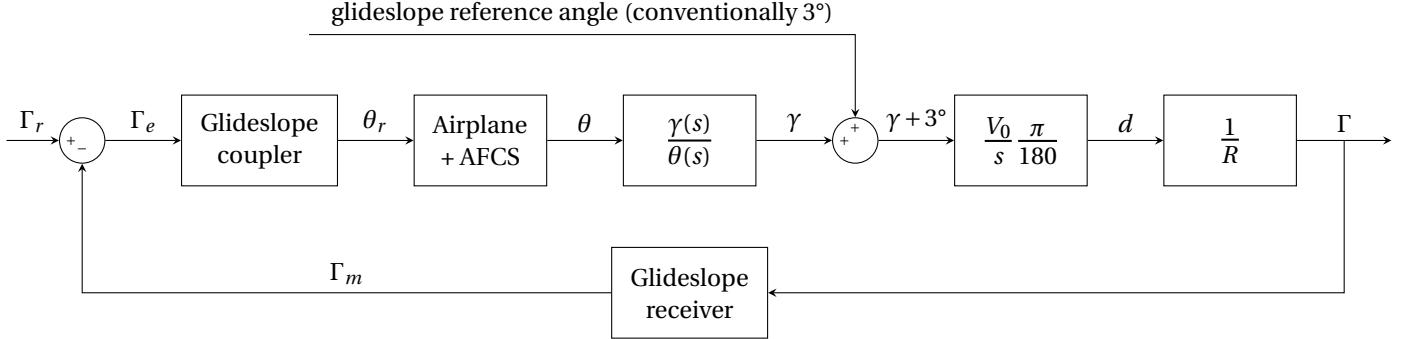


Figure 5.3: Glideslope controller block diagram.

#### 5.3.1 Glideslope geometry

The glideslope general geometry is visually shown in Figure 5.4.

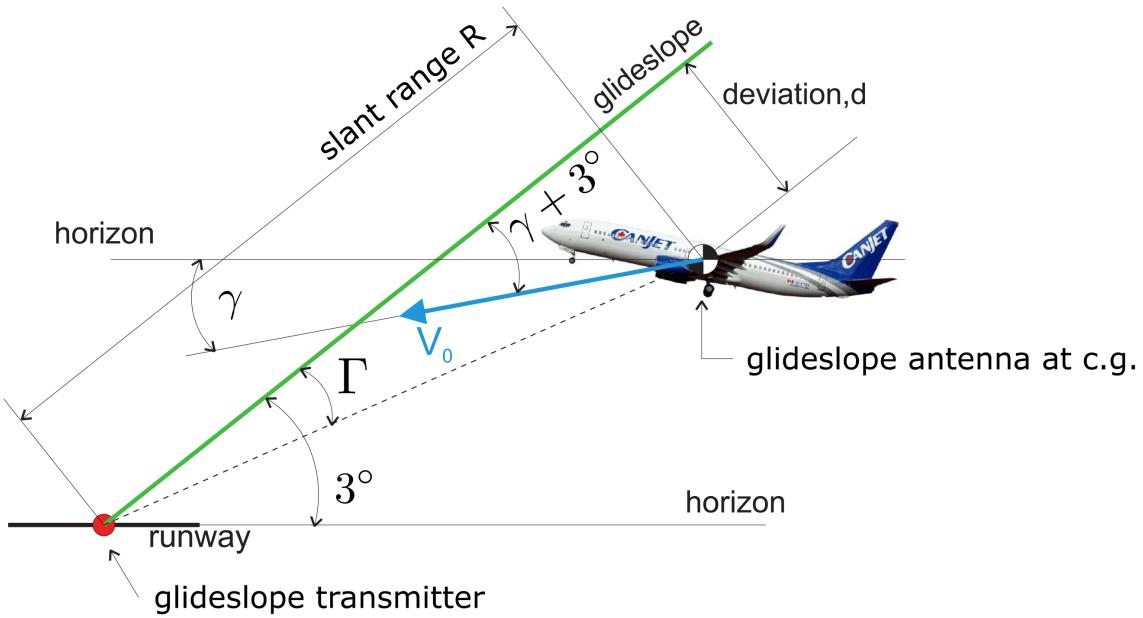


Figure 5.4: Glideslope geometry diagram.

From this geometry, some relations can be determined and are shown in Equations 5.5 to 5.8, which apply the small angle approximation.

$$\dot{d} = V_{t_{tot}} \sin\left(\gamma_{tot} + 3^\circ\right) \frac{\pi}{180^\circ} \approx V_{t_{tot}} (\gamma_{tot} + 3^\circ) \frac{\pi}{180^\circ} \quad (5.5)$$

$$d \approx \frac{V_{t_{tot}}}{s} (\gamma_{tot} + 3^\circ) \frac{\pi}{180^\circ} \quad (5.6)$$

$$R = \sqrt{h_{tot}^2 + x_{tot}^2} \quad (5.7)$$

$$\Gamma = \arctan\left(\frac{d}{R}\right) \approx \frac{d}{R} \quad (5.8)$$

In these equations,  $\dot{d}$  is the deviation range,  $\gamma_{tot} = \theta_{tot} - \alpha_{tot}$  is the flight path angle,  $h_{tot}$  is the altitude above the runway, and  $x_{tot} = x_0 - x$  is the total horizontal distance to the runway. While  $x_0$  is geometrically determined,  $x$  is calculated by integration of the total true velocity,  $V_{t_{tot}}$ . Initial values for the slant range,  $R_0$ , and deviation,  $d_0$ , can also be determined geometrically and are needed for the glideslope error angle calculations. These initial values are, thus, compiled below in Table 5.5.

Parameter	Value [ft]
$x_0$	41162.27
$R_0$	41210.83
$d_0$	157.0079

Table 5.5: Initial values relevant for glideslope error calculations

### 5.3.2 Simulink implementation

Using the information derived in subsection 5.3.1, the glideslope error angle calculation can be implemented using Simulink and corresponds to the "Glideslope Error Angle Calculation" subsystem shown in Figure 7.3. This subsystem, detailed in Figure 5.7, takes as inputs the total flight path angle, velocity, and altitude above the runway. The glideslope error angle is then fed back to the reference error angle, assumed to be zero, thus closing the loop.

As seen in Figure 7.3, there is a switch that ensures that the glideslope manoeuvre only begins after 10 seconds, at which point the loop is closed.

To ensure accurate tracking of the glideslope, a coupler is also implemented. The glideslope coupler is incorporated as a PI controller that follows the transfer function shown in Equation 5.9.

$$H_{coupler} = K_c \left(1 + \frac{W_1}{s}\right) \quad (5.9)$$

In this case,  $K_c$  is the coupler gain and  $W_1$  is a weighting constant used to mitigate steady-state errors. The implementation of this coupler is shown in Figure 7.4. The gain  $K_c$  was implemented as a P controller, allowing for the use of the Simulink auto-tuning tool. The final tuned glideslope coupler gain value was, then,  $K_c = 40$ .

### 5.3.3 Simulation Results

For this simulation to be considered successful, a few conditions must be met. Firstly, the aircraft should maintain a constant altitude above the highway of 2000ft for 10 seconds, before intercepting the glideslope. This also means that, at 10 seconds, the glideslope error angle should equal zero. Finally, the aircraft should also be able to track the 3° glideslope path and keep a fairly constant velocity in this process.

To this end, a few curves are plotted below. Figures 5.5 and 5.6 show the aircraft altitude above the highway as a function of time and x-position, respectively. Figure 5.7 plots the glideslope error angle in time, while Figure 5.5 shows the time variation of the aircraft x-position during the glideslope. In these figures, the curves were plotted until the point where the glideslope is replaced by the flare model, which occurs at an altitude above the runway of 45ft.

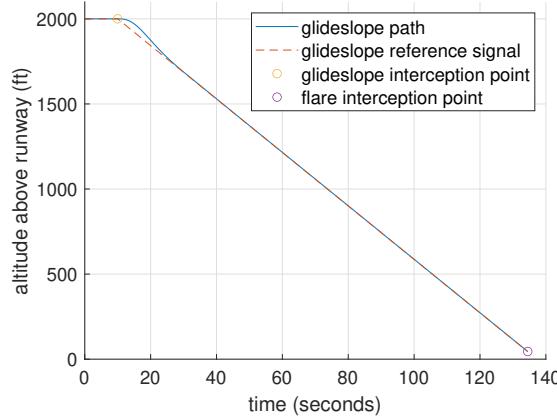


Figure 5.5: Aircraft altitude in time during glideslope.

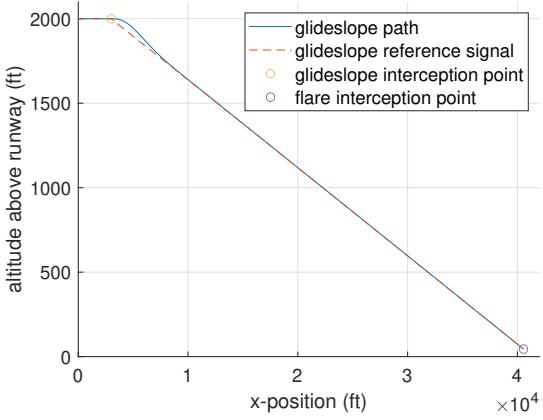
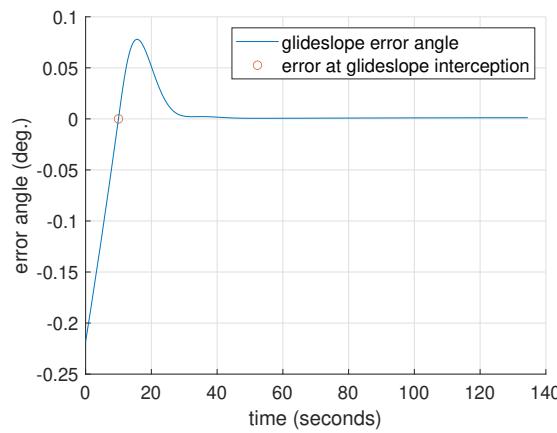
Figure 5.6: Aircraft  $h(x)$  path above runway during glideslope.

Figure 5.7: Glideslope error angle.

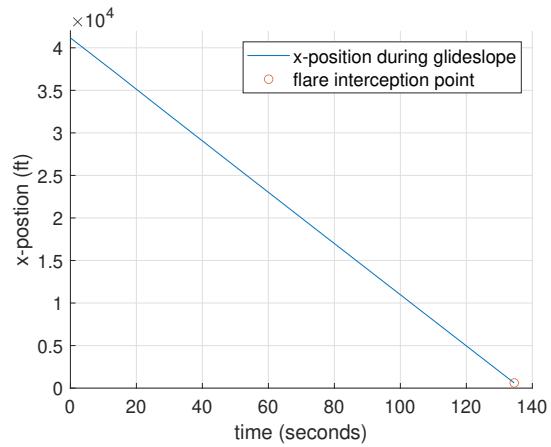


Figure 5.8: Aircraft x-position during glideslope.

Figure 5.13 clearly shows that the aircraft successfully maintains a 2000ft altitude above the highway for the first 10 seconds of the simulation. After this time, it descends and begins to track the 3° glideslope path. Figure 5.7 confirms this observation, by showing that, at 10 seconds, the glideslope error angle is exactly zero, meaning that the interception between the aircraft and glideslope path takes place. This figure also allows us to observe that there is a small transient period of around 15 seconds, after which the glideslope angle is successfully tracked to the 3° reference signal. These 15 seconds are an acceptable transient time period, considering that the whole manoeuvre takes close to 140 seconds to be completed. Finally, Figure 5.8 reveals that the x-position decreases linearly with time, indicating that the aircraft velocity does remain fairly constant throughout the glideslope tracking, as desired.

From this analysis, we are able to confirm that the glideslope controller successfully tracks the glideslope path, while meeting all the previously set requirements.

## 5.4 Flare controller

A flare manoeuvre is necessary after the glideslope in order to land the aircraft on the runway at a desired location and with a desired vertical touchdown velocity. This touchdown velocity is particularly critical for not only passenger and crew comfort, but also overall aircraft structural limits. Typically, a good range of absolute values for vertical touchdown velocity is between 2ft/s and 3ft/s.

Conceptually, the flare controller is represented by the block diagram of Figure 5.9, relating altitude,  $h$ , vertical velocity,  $\dot{h}$ , and pitch angle,  $\theta$ .

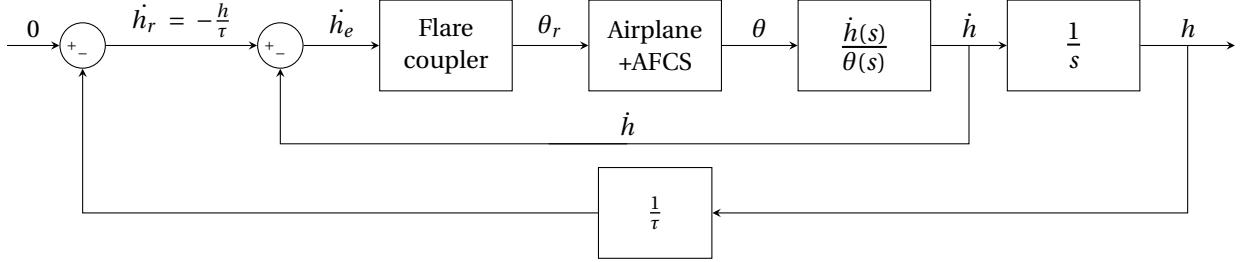


Figure 5.9: Automatic flare mode block diagram.

### 5.4.1 Flare geometry

The flare general geometry is visually shown in Figure 5.10.

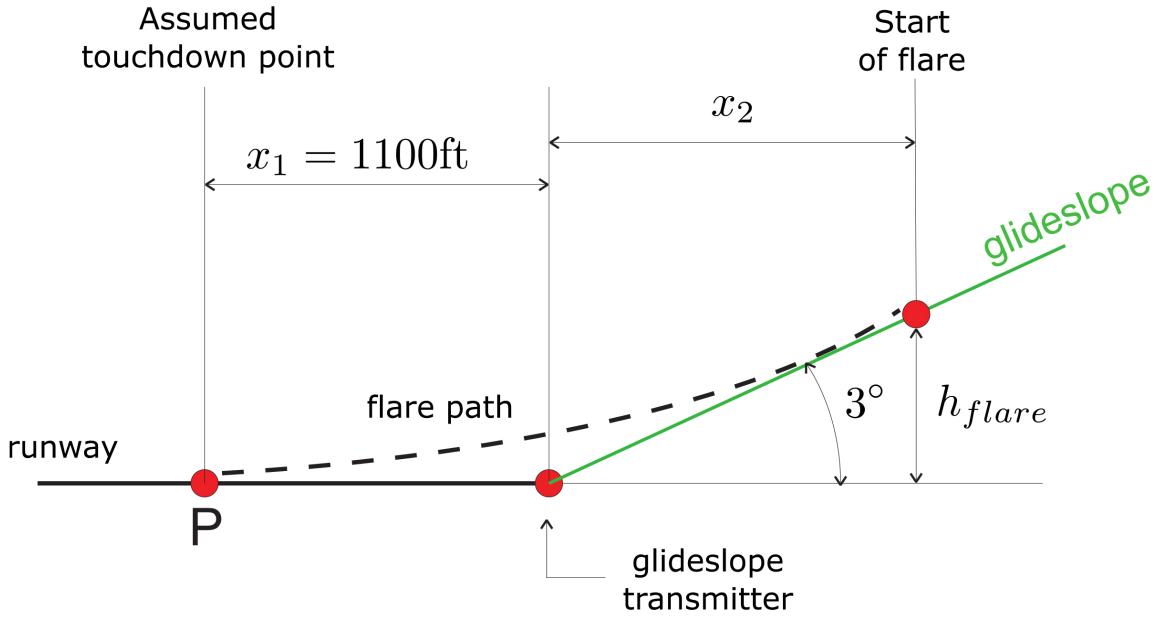


Figure 5.10: Flare geometry diagram.

The flare, assumed to start at a initial altitude  $h_{flare}$ , follows an exponentially decaying path and is expected to touchdown on the runway at a horinzontal distance  $x_1 = 1100\text{ft}$ , past the glideslope transmitter.

From this geometry and mathematical description, some relations can be determined and are shown in Equations 5.10 and 5.11.

$$h(t) = h_{flare} e^{-\frac{t}{\tau}} \quad (5.10)$$

$$\dot{h}(t) = -\frac{h_{flare}}{\tau} e^{-\frac{t}{\tau}} = -\frac{h(t)}{\tau} = V \sin(\gamma) \approx V\gamma \quad (5.11)$$

The unknowns in these equations are, thus, the flare initial altitude,  $h_{flare}$ , and the time constant,  $\tau$ . It is also important to determine the horizontal distance form the flare start to the glideslope transmitter,  $x_2$ .

Defining the boundary conditions with Equations 5.10 and 5.11 for  $t = 0\text{s}$  and  $t = T$  (touchdown time) yields the results shown from Equations 5.12 to 5.15.

$$h(0) = h_{flare} \quad (5.12)$$

$$\dot{h}(0) = -\frac{h_{flare}}{\tau} \quad (5.13)$$

$$h(T) = h_{flare} e^{-\frac{T}{\tau}} \quad (5.14)$$

$$\dot{h}(T) = -\frac{h_{flare}}{\tau} e^{-\frac{T}{\tau}} = \dot{h}_{td} \quad (5.15)$$

Because we assume constant airspeed ( $V = 300 \text{ ft/s}$ ), a simple relation can also be found for  $x_2$ , as shown in Equation 5.16

$$x_1 + x_2 = VT \quad (5.16)$$

With all these equations, we can express  $\tau$ ,  $h_{flare}$ , and  $x_2$  as functions of vertical touchdown velocity,  $\dot{h}_{td}$ .

$$\tau = \frac{x_1}{-V \left( \ln \left( \frac{-\dot{h}_{td}}{V \sin(\gamma)} \right) + \cos(\gamma) \right)} \quad (5.17)$$

$$h_{flare} = \tau V \sin(\gamma) \quad (5.18)$$

$$x_2 = \frac{h_{flare}}{\tan(\gamma)} \quad (5.19)$$

Therefore, making an initial choice of vertical touchdown speed  $\dot{h}_{td} = -3 \text{ ft/s}$ , the respective flare parameters can be computed and are compiled below in Table 5.6.

Parameter	Value	Units
$\tau$	5.585	s
$x_2$	1673.33	ft
$h_{flare}$	87.696	ft

Table 5.6: Initial flare mode parameters.

Finally, it should be noted that this mathematical model implies that the aircraft will only touchdown on the runway at infinity, because of the exponential decay behaviour of the flare path. This obviously will not occur in the simulation, since we want the aircraft to actually touchdown in a certain range of values. It is, thus, expected that the final results for  $\dot{h}_{td}$  will be slightly different from the theoretical assumption and that an iterative process will be needed in order to find a set of parameters that meets all requirements.

### 5.4.2 Simulink implementation

Using the information derived in subsection 5.4.1, the flare vertical speed and altitude calculations can be implemented using Simulink and correspond to the "Flare Calculations" subsystem shown in Figure 7.3. This subsystem, detailed in Figure 7.10, takes as inputs the total flight path angle, velocity, and altitude above the runway. The vertical speed and flare altitude are then fed back to the system, thus closing the two loops described in Figure 5.9.

Another switch is added to the system, to transition between the glideslope and flare modes once the initial flare altitude,  $h_{flare}$ , is reached.

Similarly to the glideslope, a coupler is also implemented. The flare coupler is incorporated as a PI controller that follows the transfer function shown in Equation 5.9.

### 5.4.3 Simulation results

As expected, the vertical touchdown velocity result obtained using the flare parameters from Table 5.6, although capable of making a smooth descent and transition from the glideslope, was not equal to the initially assumed value of  $\dot{h}_{td} = -3\text{ft/s}$ . Instead, this value was close to  $-5\text{ft/s}$ , clearly violating the velocity requirements. Even with tuning of the flare coupler, it was not possible to reach a value between  $-2\text{ft/s}$  and  $-3\text{ft/s}$ .

Therefore, a slightly different approach was used. Instead of defining  $\dot{h}_{td}$ , the value of  $h_{flare}$  was iterated until a satisfactory vertical touchdown velocity was reached. This was done because of ease of implementation, since two of our model's inputs were  $h_{flare}$  and  $\tau$ , and not  $\dot{h}_{td}$ . With  $h_{flare}$  chosen,  $\tau$  was easily computed using Equation 5.10.

After this iterative process, the design path was fully defined by the following set of theoretical parameters, compiled in Table 5.7.

Parameter	Value	Units
$h_{flare}$	45	ft
$\tau$	2.867	s
$x_2$	858.6511	ft
$\dot{h}_{td}$	-1.6099	ft/s

Table 5.7: Final flare mode parameters.

The system was tuned until a satisfactory flare coupler gain was reached,  $K_c = 0.06$ .

Figure 5.11 plots the aircraft vertical velocity in time, while Figure 5.12 shows the same parameters, but only during the flare manoeuvre. In these figures, the curves were plotted until aircraft touchdown.

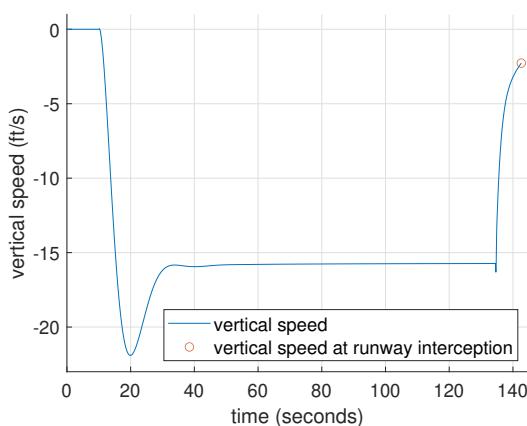


Figure 5.11: Aircraft vertical velocity during glideslope and flare.

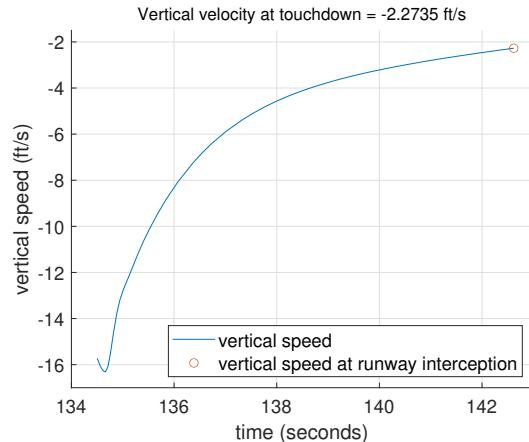


Figure 5.12: Aircraft vertical velocity during flare.

As can be seen in Figure 5.12, the obtained vertical touchdown velocity is  $\dot{h}_{td} = -2.2735\text{ft/s}$ , clearly inside the required range. This means the aircraft can successfully touchdown on the runway.

Figures 5.13 and 5.14 show the aircraft flare altitude above the highway as a function of time and x-position, respectively.

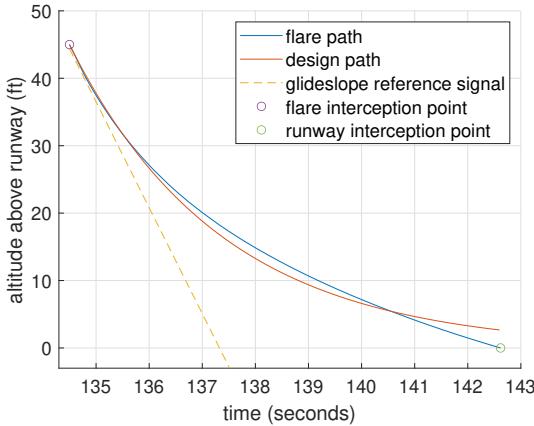


Figure 5.13: Aircraft altitude in time during flare. Comparison with glideslope path (without flare) and design path.

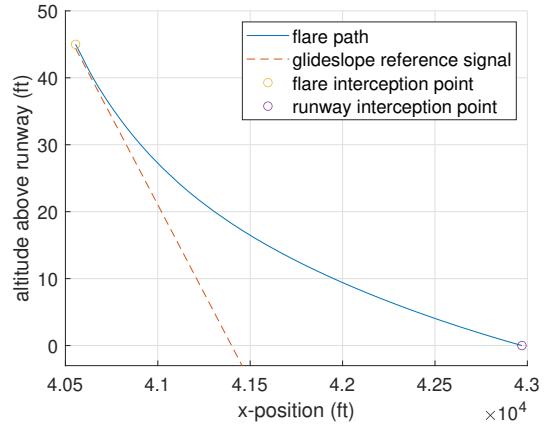


Figure 5.14: Aircraft  $h(x)$  path above runway during flare. Comparison with glideslope path (without flare).

Figure 5.13 contrasts the obtained flare path (in blue) with the theoretical design path (in red) and with the glideslope path (in yellow, dashed-line). As can be seen, the theoretical design path does not actually reach the runway (as explained at the end of subsection 5.4.1)

Finally, for completion purposes, the total flight altitude above the runway, including both glideslope and flare, is plotted as a function of time and x-position in Figures 5.15 and 5.16, respectively.

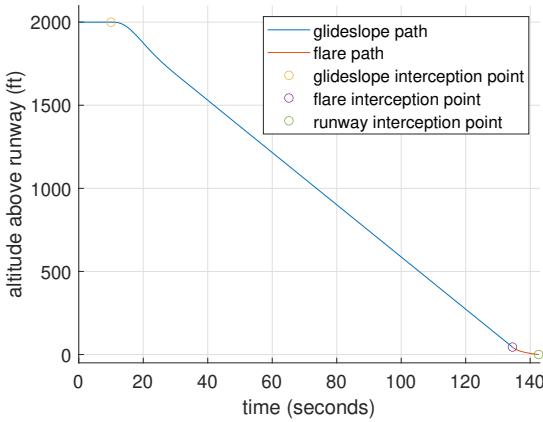


Figure 5.15: Aircraft altitude in time.

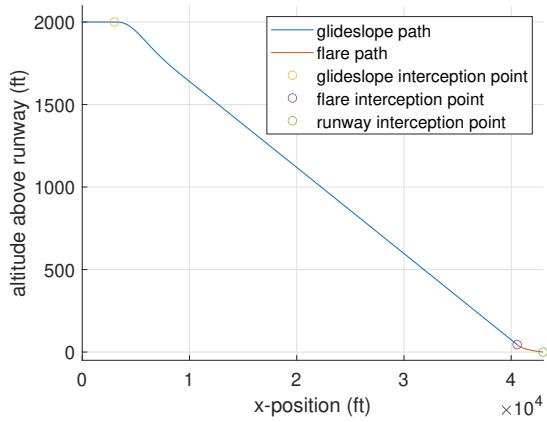


Figure 5.16: Aircraft  $h(x)$  path above runway.

It should be noted that the similarities between these two plots are not a coincidence. They appear to be the same because, in fact, the assumption of constant velocity throughout the whole simulation was met. Therefore, the x-position and time axis are linearly dependent, with the proportionality constant being the total aircraft velocity, *i.e.*,  $x \approx Vt$ .

# **Chapter 6**

## **Conclusion**

Ultimately, taking everything into consideration, a pitch rate control system and an automatic glideslope and flare controller were implemented successfully. Moreover, given the extent of this assignment, the F-16 model was linearised for a given flight condition and the low fidelity model was used in order to reduce the its complexity.

Furthermore, in Section 2, it was concluded that the best position to place the accelerometer in an aircraft is at the Instantaneous Center of Rotation (ICR). Section 3 focused on the analysis of the open-loop system. The results showed that all poles are in the open left half plane, which makes the aircraft dynamically stable. The eigenmotions were also plotted and discussed. Moreover, in Section 4, the pitch rate command system was designed and CAP and Gibson criterion were checked to see if the final system was suitable for the given flight conditions. Lastly, in Section 5, the results of the implementation of the glideslope and flare controller are presented. This was completed successfully, since the aircraft is intercepting a  $3^\circ$  glideslope path and the vertical velocity on touchdown is of  $-2.2735 \text{ ft/s}$ , which is in the acceptable range.

Concluding, all tasks were performed successfully and the goal to become more familiar with classic flight controllers, as well as to gain insight in handling qualities of open-loop and controlled aircraft was clearly met.

# Bibliography

- [1] Richard S Russell. "Non-linear F-16 simulation using Simulink and Matlab". In: *University of Minnesota, Tech. paper* (2003).
- [2] E. Smeur. "AE4301 Automatic Flight Control System Design: Lecture Notes". In: *Delft University of Technology* (2022).
- [3] E. Smeur. "Practical assignment AE4-301P: Exercise Automatic Flight Control System Design". In: *Delft University of Technology* (2022).

# Chapter 7

## Supporting Material

### 7.1 Simulink Models used in Chapter 5

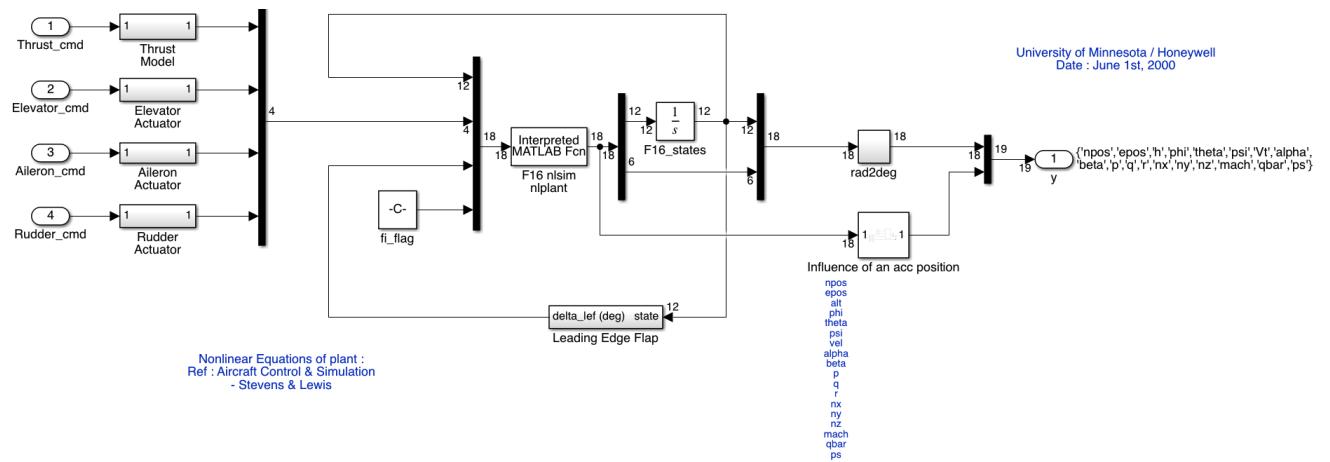


Figure 7.1: Simulink of the linearised F-16 model with  $an$  as an extra output.

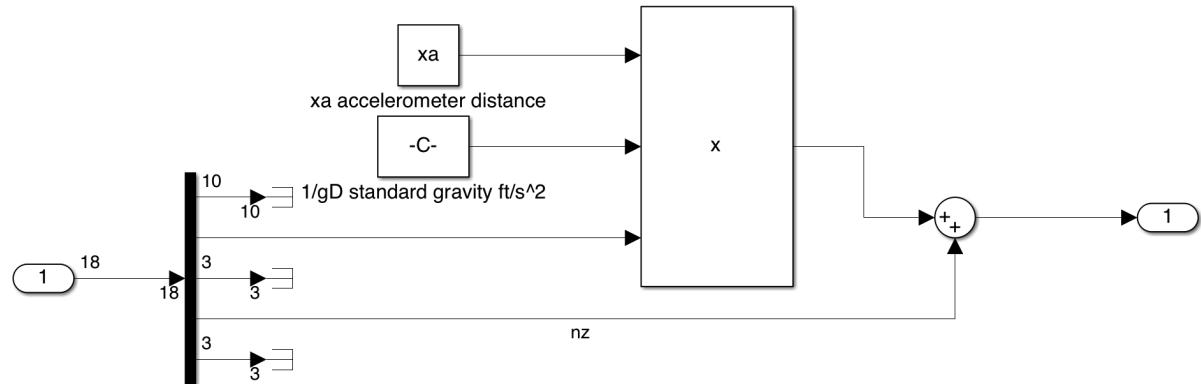


Figure 7.2: Simulink block "Influence of an acc position" from the model presented in Figure 7.1.

### 7.2 Simulink Models used in Chapter 8

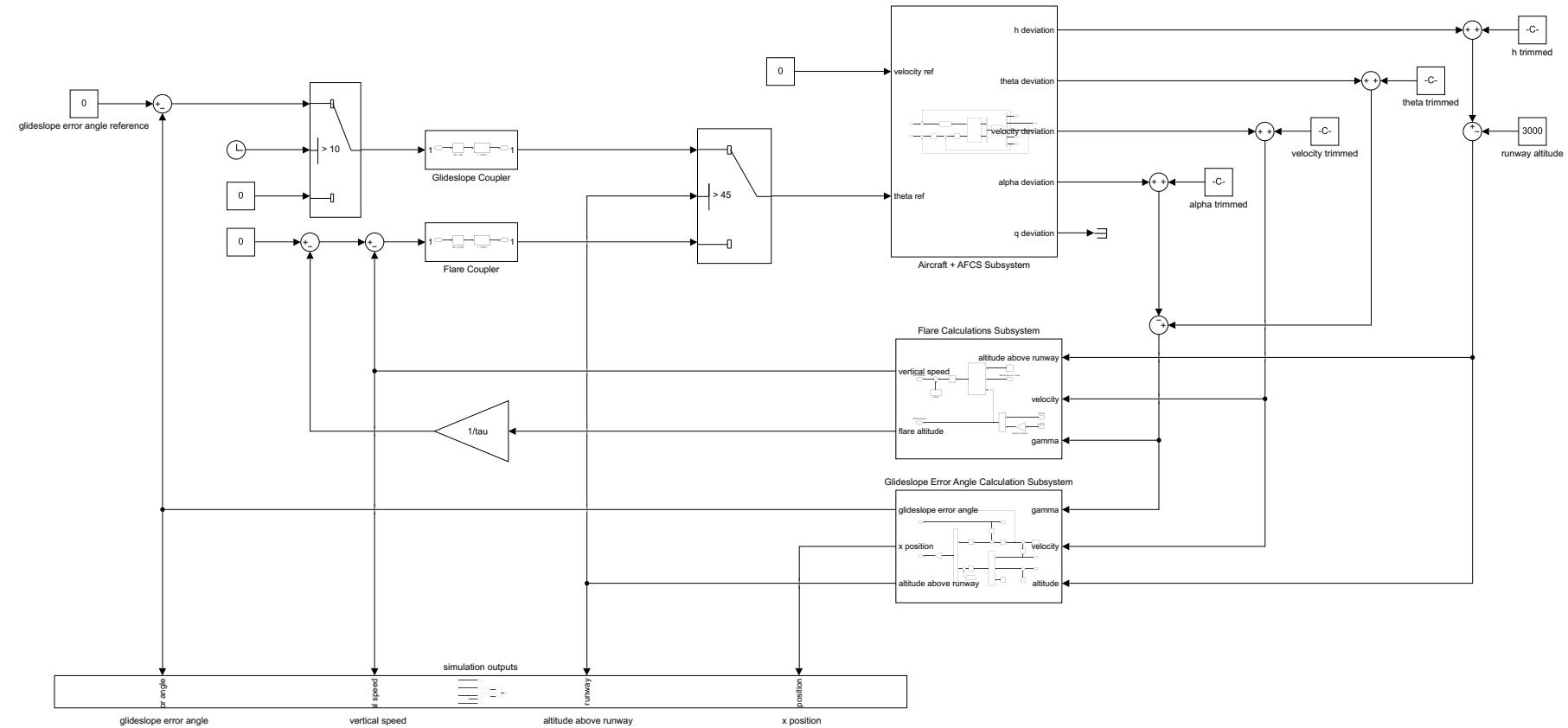


Figure 7.3: Complete Simulink Model used in chapter 5.

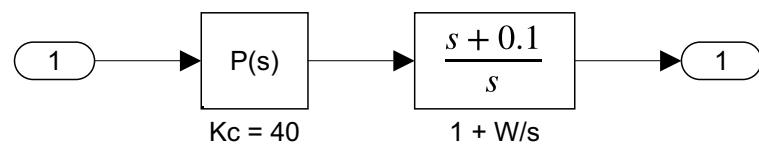


Figure 7.4: Glideslope Coupler.

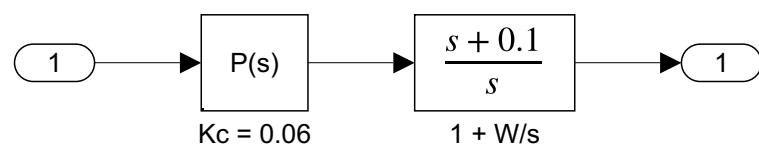


Figure 7.5: Flare Coupler.

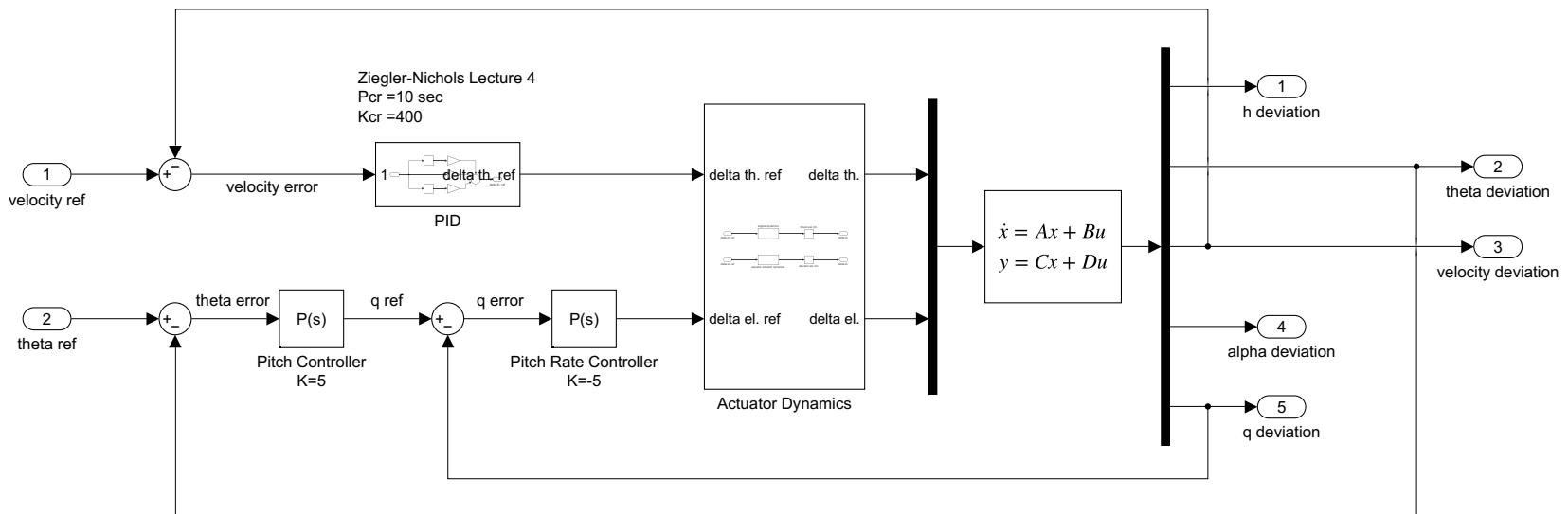


Figure 7.6: Aircraft + AFCS Subsystem.

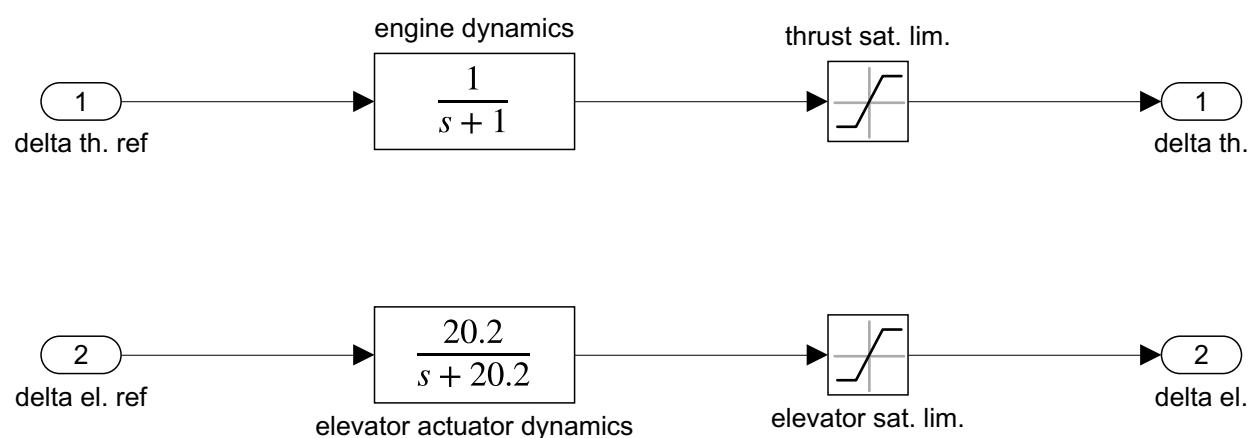


Figure 7.7: Aircraft Actuator Dynamics.

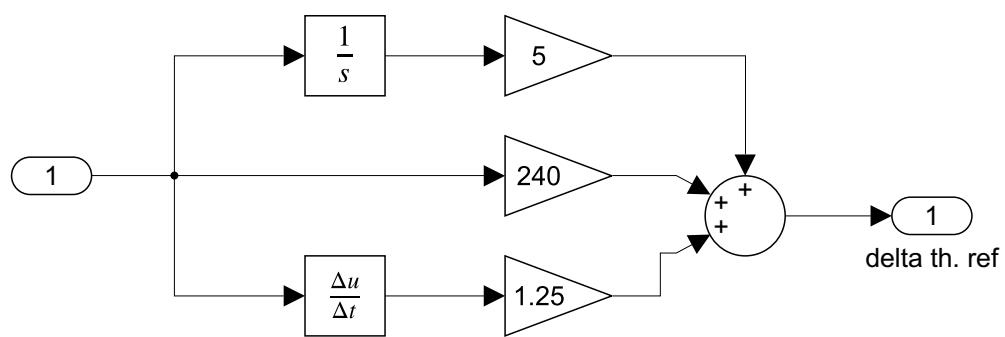


Figure 7.8: Velocity PID Controller.

34

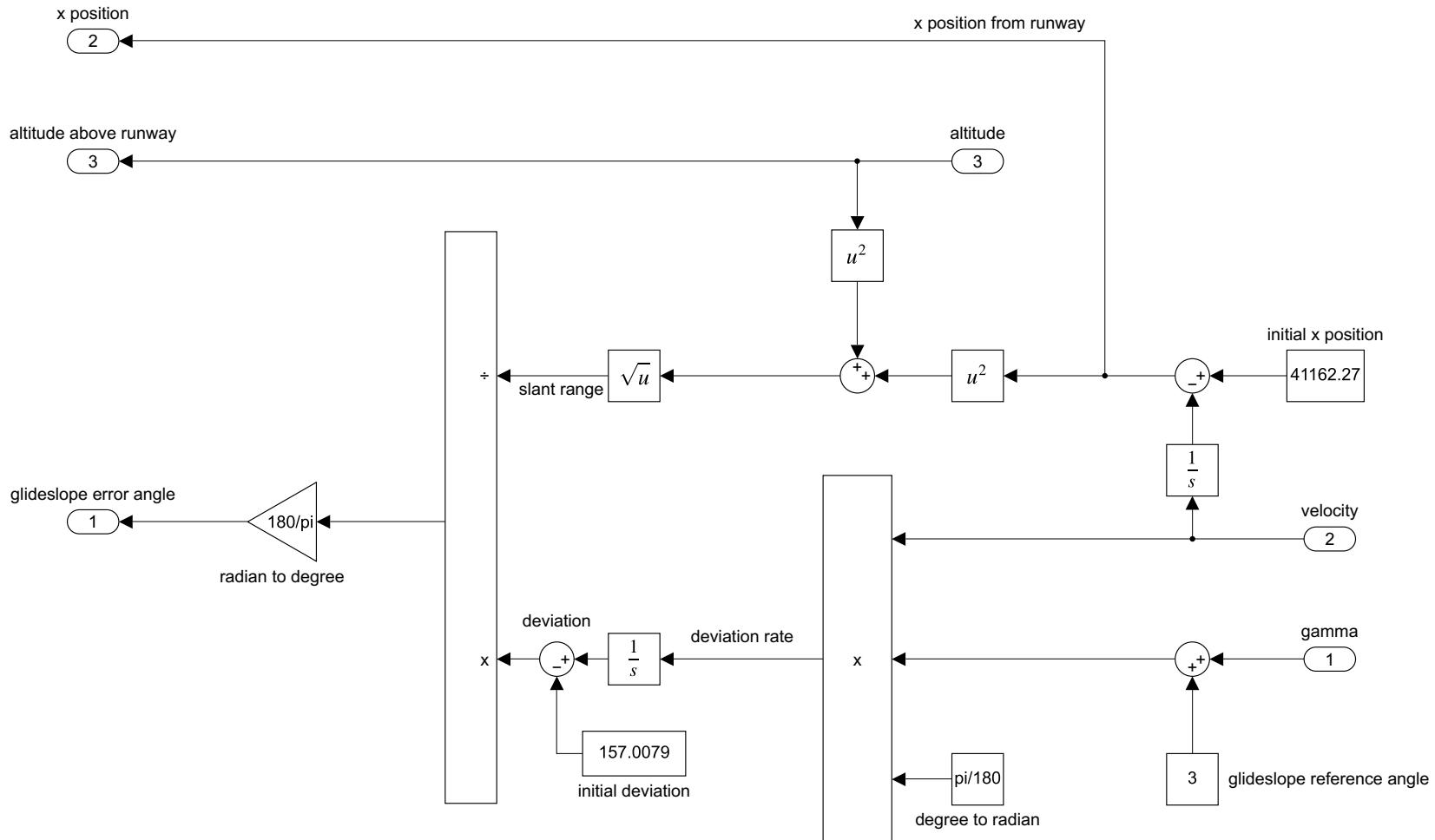


Figure 7.9: Glideslope Calculations Subsystem.

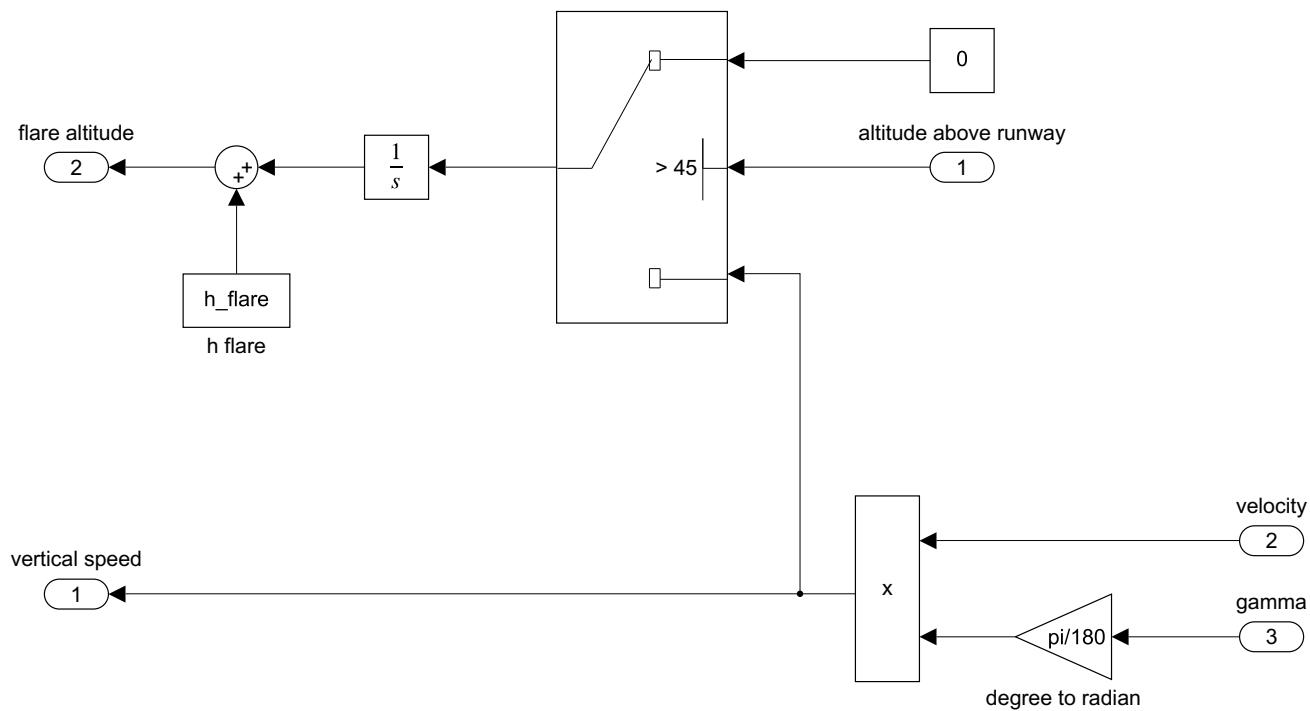


Figure 7.10: Flare Calculations Subsystem.

36

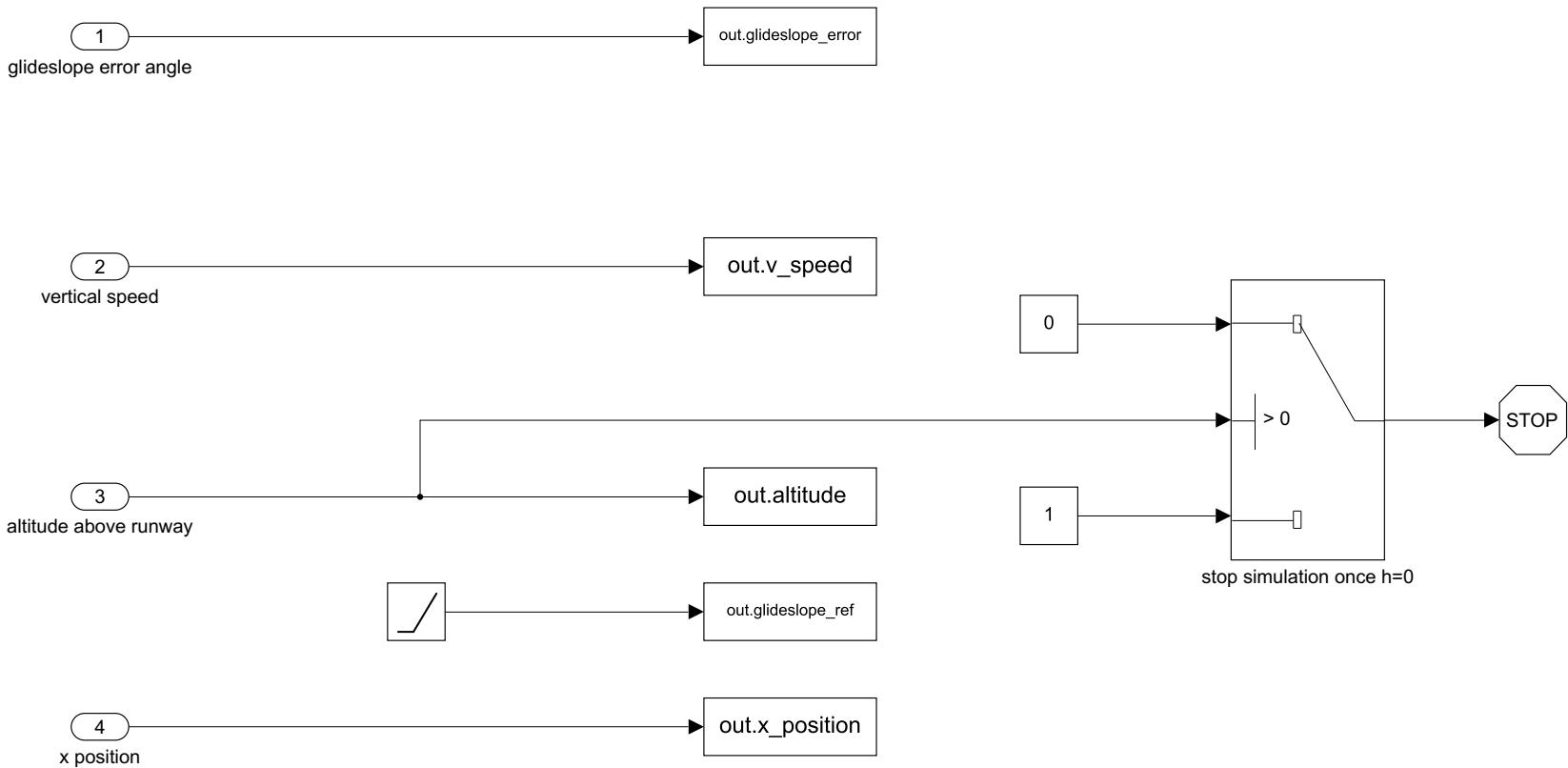


Figure 7.11: Output Subsystem.