Longitudinal MIMO Flight Control with PID controllers, Observers and Kalman Filter

Zhiren Chen

University of Toronto

Departmental of Mechanical and Industrial Engineering

2021-12-3

Contents

Nomenclature

Equilibrium Flight path angle	γ_e	Flight path angle	γ
Equilibrium AOA	α_e	Angle of attack, AOA	α
Equilibrium velocity	V_0	Velocity	V
Mass	m	Elevator angle	η
Pitch moment of inertia	I_y	Engine thrust	τ
Air density	ρ	Axial velocity	u
Wing area	S	Normal velocity	w
Frontal area	A_f	Pitch rate	q
Drag coefficient	C_d	Pitch angle	θ
Mean aerodynamic chord	$\bar{\bar{c}}$	Tailplane lift curve slope	a_1
Acceleration due to gravity	g	Elevator lift curve slope	a_2
Tail lift arm	l_T	Wing lift curve	a
Throttle lever angle	ε		

Table 1 Nomenclature

Abstract

Aircraft autopilot control is a crucial topic in aviation. This report focuses on controlling a linear MIMO aircraft longitudinal dynamic model with PID controllers using parameters of a McDonnell F-4C flying at 35000ft. Real world imperfect conditions such as noises and saturations are found to prevent the PID controlled system from entering steady state. The LQR observer is found to eliminate most noises effectively and stabilized the system. The Kalman filter is also found to effectively fuse outputs from two types of sensors with minimal error and greatly protects the stability of the system. Finally, this dynamic model is evaluated to be mostly reasonable, though there are unreasonable aspects probably due to deviating state conditions from equilibrium.

1. Introduction

This project studies the construction of the longitudinal aircraft state space model in *Chapter 4 The Equations of Motion* of the book *Flight Dynamic Principles*, implements this model on a McDonnell F-4C with certain flying configuration, validates this model in SISO and MIMO forms with PID controllers, realizes this model with simulated real-world conditions (including elevator angle restrictions, engine thrust restrictions and various noises in input and output), and utilizes LQR observer and Kalman Filter to optimize the system behaviors under different combinations of such non-ideal conditions. At each step this model and the controlling techniques will be evaluated based on their trueness and functionality. Also, the two important flight parameters, horizontal and vertical speeds, will be derived based on the state outputs and discussed.

2. Model Formulation

The coordinate system conversion, angles and velocities of an airplane longitudinal model is as shown in Figure 1:

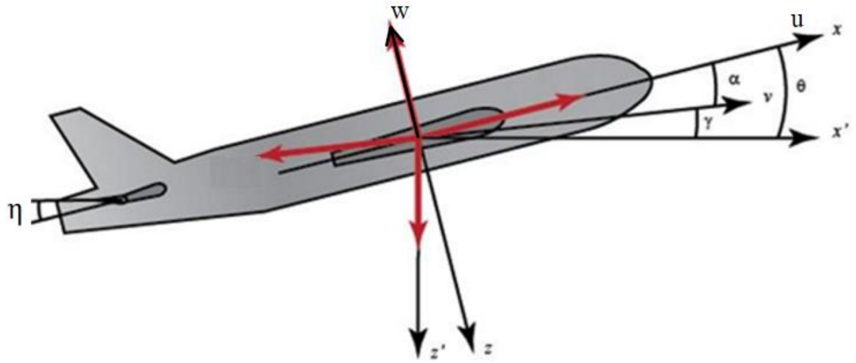


Figure 1 Aircraft angles, velocities and coordinate systems [6]

The free body diagram for forces can be drawn as Figure 2:

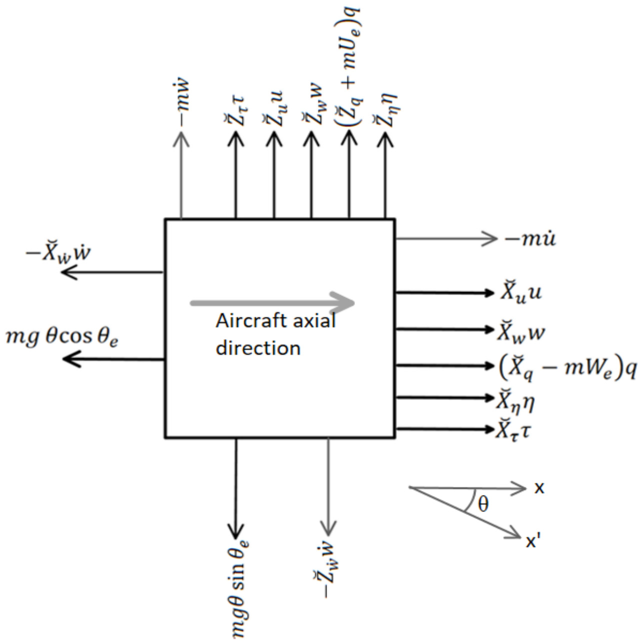


Figure 2 Free body diagram for forces

And the free body diagram for moments can be drawn as Figure 3:

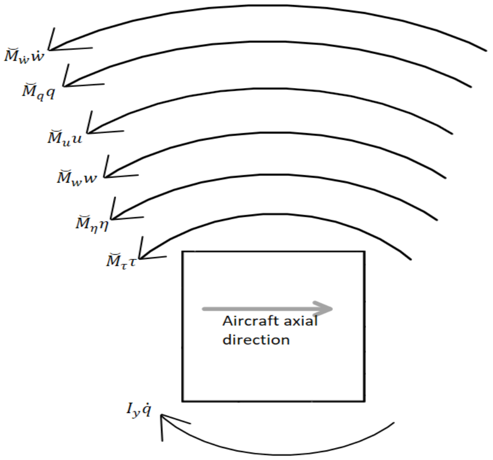


Figure 3 Free body diagram for moments

The intermediate model in the longitudinal direction of an aircraft in a stable flight with constant altitude and airspeed is: [1]

$$M\dot{x}(t) = A'x(t) + B'\mathbf{u}(t) \qquad (1)$$

From Figures 2 and 3, the state equations are: [1]

$$m\dot{u} - \breve{X}_{\dot{w}}\dot{w} = \breve{X}_u u + \breve{X}_w w + \left(\breve{X}_q - mW_e\right)q - mg\,\theta\cos\theta_e + \breve{X}_\eta\eta + \breve{X}_\tau\tau \qquad (2)$$

$$m\dot{w} - \breve{Z}_{\dot{w}}\dot{w} = \breve{Z}_u u + \breve{Z}_w w + \left(\breve{Z}_q + mU_e\right)q - mg\theta\sin\theta_e + \breve{Z}_\eta\eta + \breve{Z}_\tau\tau \qquad (3)$$

$$I_y\dot{q} - \breve{M}_{\dot{w}}\dot{w} = \breve{M}_u u + \breve{M}_w w + \breve{M}_q q + \breve{M}_\eta\eta + \breve{M}_\tau\tau \qquad (4)$$

$$\dot{\theta} = q \qquad (5)$$

Where the terms with ‘˘‘ are aerodynamic stability derivatives. Their derivations are shown in Appendix 2. The assumptions are constant m, rigid airplane body, constant ρ and constant airspeed. [1]

The intermediate state space matrices can thus be formed as Equations 6, 7 and 8: [1]

$$M = \begin{bmatrix} m & -\breve{X}_{\dot{w}} & 0 & 0 \\ 0 & m - \breve{Z}_{\dot{w}} & 0 & 0 \\ 0 & -\breve{M}_{\dot{w}} & I_y & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \qquad (6)$$

$$A' = \begin{bmatrix} \breve{X}_u & \breve{X}_w & (\breve{X}_q - mW_e) & -mg\cos\theta_e \\ \breve{Z}_u & \breve{Z}_w & (\breve{Z}_q + mU_e) & -mg\sin\theta_e \\ \breve{M}_u & \breve{M}_w & \breve{M}_q & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \qquad (7)$$

$$B' = \begin{bmatrix} \breve{X}_\eta & \breve{X}_\tau \\ \breve{Z}_\eta & \breve{Z}_\tau \\ \breve{M}_\eta & \breve{M}_\tau \\ 0 & 0 \end{bmatrix} \qquad (8)$$

And

$$x = \begin{bmatrix} u \\ w \\ q \\ \theta \end{bmatrix} \qquad (9)$$

$$\mathbf{u} = \begin{bmatrix} \eta \\ \tau \end{bmatrix} \qquad (10)$$

Equations 6,7 and 8 can be normalized to: [1]

$$M = \begin{bmatrix} m' & -\frac{X_{\dot{w}}\bar{\bar{c}}}{V_0} & 0 & 0 \\ 0 & m' - \frac{Z_{\dot{w}}\bar{\bar{c}}}{V_0} & 0 & 0 \\ 0 & -\frac{M_{\dot{w}}\bar{\bar{c}}}{V_0} & I'_y & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \qquad (11)$$

$$A' = \begin{bmatrix} X_u & X_w & (X_q\bar{\bar{c}} - m'W_e) & -m'g\cos\theta_e \\ Z_u & Z_w & (Z_q\bar{\bar{c}} + m'U_e) & -m'g\sin\theta_e \\ M_u & M_w & M_q\bar{\bar{c}} & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \qquad (12)$$

$$B' = \begin{bmatrix} V_0 X_\eta & V_0 X_\tau \\ V_0 Z_\eta & V_0 Z_\tau \\ V_0 M_\eta & V_0 M_\tau \\ 0 & 0 \end{bmatrix} \qquad (13)$$

With:

$$m' = \frac{m}{\frac{1}{2}\rho V_0 S} \qquad (14)$$

$$I_y' = \frac{I_y}{\frac{1}{2}\rho V_0 S \bar{\bar{c}}} \qquad (15)$$

The variables in Equations 11, 12 and 13 explained in Appendix 1 are dimensionless aerodynamic stability derivatives [1]. These variables can change when other parameters change and many of them are represented in Taylor series, but they are trunked to only the first-order term and evaluated at the start (with equilibrium conditions) to ease calculations. When the change in system states do not exceed a certain range, the accuracy by such approximation will be decent.

The quantities used for the later calculations are shown in Table 2: [1]

Equilibrium Flight path angle	γ_e	0°
Equilibrium AOA	α_e	9.4°
Equilibrium velocity	V_0	178m/s
Mass	m	17642kg
Pitch moment of inertia	I_y	165669kgm^2
Air density (35000ft)	ρ	0.3809kg/m^3
Wing area	S	49.239m^2
Frontal area	A_f	~12m^2
Drag coefficient	C_d	0.026
Mean aerodynamic chord	$\bar{\bar{c}}$	4.889m
Acceleration due to gravity	g	9.81m/s^2

Table 2 McDonnell F-4C at 35000ft model parameters

And the initial conditions are $u_0 = 178$m/s, $w_0 = 0$m/s, $q_0 = 0$rad/s, $\theta_0 = \gamma_e + \alpha_e = 0.164$rad.

The missing numbers for calculation are assumed to be 0 as they are probably not important.

After substituting in numbers, Equations 11, 12, 13 become: [1]

$$M = \begin{bmatrix} 10.569 & 0 & 0 & 0 \\ 0 & 10.580 & 0 & 0 \\ 0 & 0.0162 & 20.3 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \qquad (16)$$

$$A' = \begin{bmatrix} 0.0076 & 0.0483 & -307.26 & -102.29 \\ -0.7273 & -3.1245 & 1850.10 & -16.934 \\ 0.034 & -0.2169 & -6.2247 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \qquad (17)$$

$$B' = \begin{bmatrix} 11 & 5.333E-4 \\ -66.5898 & 0.00327 \\ -99.341 & 0 \\ 0 & 0 \end{bmatrix} \qquad (18)$$

To convert Equation 16, 17, 18 into $\dot{\mathbf{x}}(t) = A\,\mathbf{x}(t) + B\mathbf{u}(t)$, A and B are computed as: [1]

$$A = M^{-1}A' = \begin{bmatrix} 7.181E-4 & 0.00457 & -29.702 & -9.678 \\ -0.0687 & -0.2953 & 174.868 & -1.601 \\ 0.00173 & -0.0105 & -0.4462 & 0.001277 \\ 0 & 0 & 1 & 0 \end{bmatrix} \qquad (19)$$

$$B = M^{-1}B' = \begin{bmatrix} 1.041 & 5.046E-5 \\ -6.294 & 3.091E-4 \\ -4.888 & 0 \\ 0 & 0 \end{bmatrix} \qquad (20)$$

3. Implementing PID controller on SISO system

In this step the second row of B is disregarded and $B1$ is taken as

$$B_1 = \begin{bmatrix} 1.041 \\ -6.294 \\ -4.888 \\ 0 \end{bmatrix} \qquad (21)$$

By using the ‘ctrb()’ command in MATLAB and use rank() command to check the rank of the resulted controllability matrix, this system is controllable. By drawing the open-loop system in Simulink and testing the zero-input response, the following plots are generated:

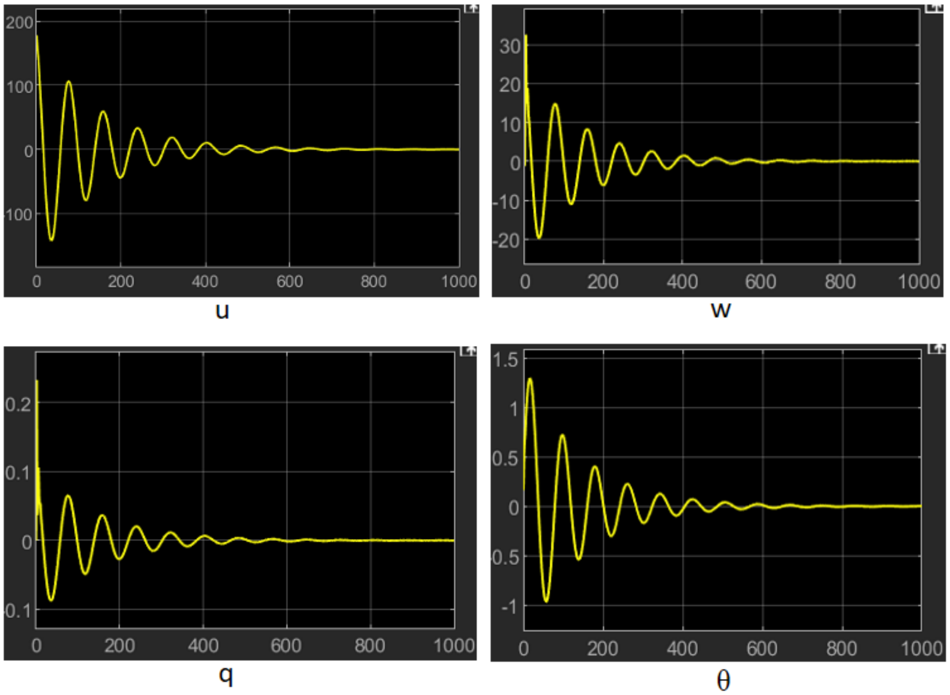


Figure 4 Zero input open loop responses

It is reasonable that u, w and q converge to 0 as time goes by due to damping of energy, but the oscillations to negative values are not as quite reasonable. However, this can be explained by the linearization done to the model, and the assumptions that the conditions for computing some figures (such as using a constant ρ and using equilibrium values for aerodynamic constants) are invariant. The original non-linear model could behave very differently when the conditions of the aircraft start to move away from the initial conditions. As for θ, it is reasonable that it is not diverging. Overall, this model seems valid with zero input.

The normal velocity w will then be tried for controlling with the input of η. A PID controller is tested for this use. The transfer function for a PID controller is: [7]

$$TF_{PID} = K_P + \frac{K_I}{s} + K_d s \qquad (22)$$

And the C matrix for controlling w in x is:

$$C_1 = [0\ 1\ 0\ 0] \qquad (23)$$

With the mentioned initial conditions, the controlled system can be constructed. Theoretically under the control of w, u will continue to decrease. In the reality the aircraft will stall at some point. But since this is an approximated linear model, u continuing to decrease will be reasonable. As for θ, it should be kept increasing, because the linearized model takes a linear relationship between AOA α and lift L, and a larger α leads to a larger θ ,and L is proportional to u^2. The Simulink model is as shown below:

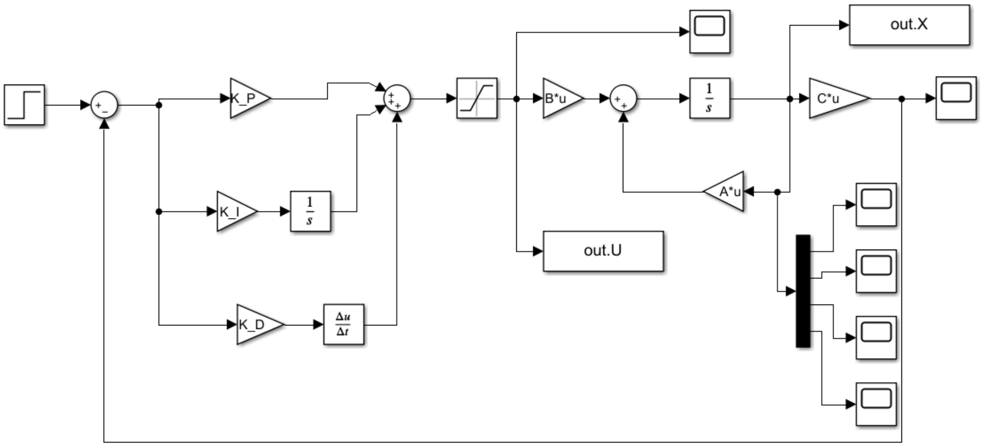


Figure 5 SISO PID close-loop control

The reference is a step input with zero at the start and $w_{ref} = 10m/s$ after 2 seconds. The K_P, K_I and K_D values are set to -0.08, -0.03 and -0.01 respectively. The open loop responses are:

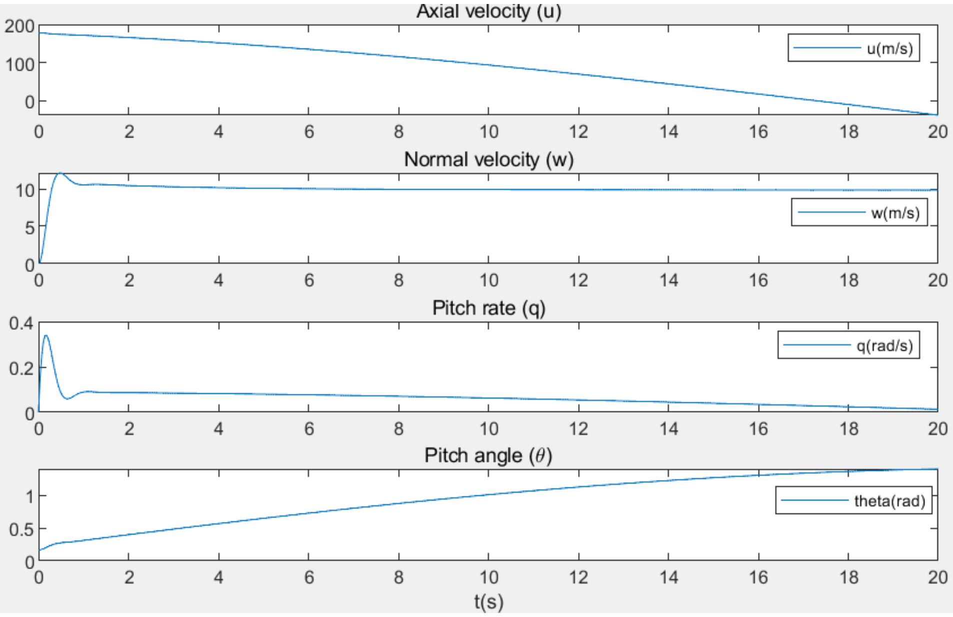


Figure 6 SISO PID controlled responses

As expected, u is decreasing and θ is increasing, while w is controlled fairly well. In the reality the aircraft will lose control at some point when u is small enough and θ is large enough for stalling, and u cannot hit negative like in Figure 6. Certainly, although this model becomes invalid when it goes too far from the equilibrium conditions, it is very reasonable at near the equilibrium conditions.

The control input η is shown in Figure 7:

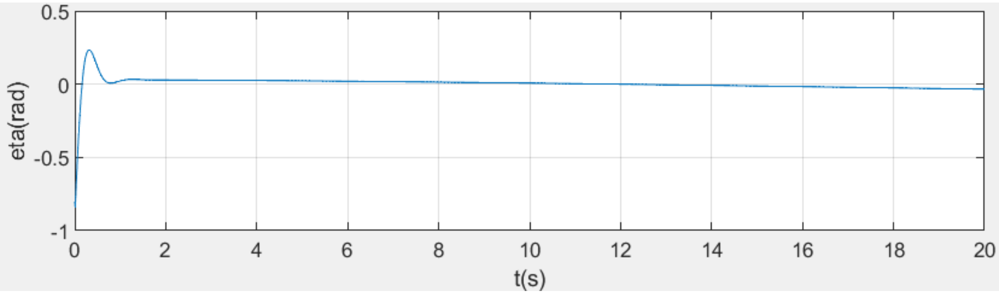


Figure 7 SISO PID control input

There is an overshoot to approximately 0.25 rads at about 0.04s, then it approaches somewhere slightly lower than 0. The overshoot makes sense since it takes some input to increase the AOA to a positive value for allowing the referenced normal velocity. The rest of the values at later times are not so informative as the model deviates from the equilibrium conditions.

4. Controlling Full MIMO system

Now the B matrix is fully 2*4 as shown in Equations 24 and 25. To control and observe u and θ, C is set as:

$$C = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \qquad (24)$$

$$D = \begin{bmatrix} 0 \\ 0 \end{bmatrix} \qquad (25)$$

Firstly, the model is tried without any restrictions (i.e. no restriction on maximum and minimum values of η and τ and their gradients). The model is as shown in Figure 8:

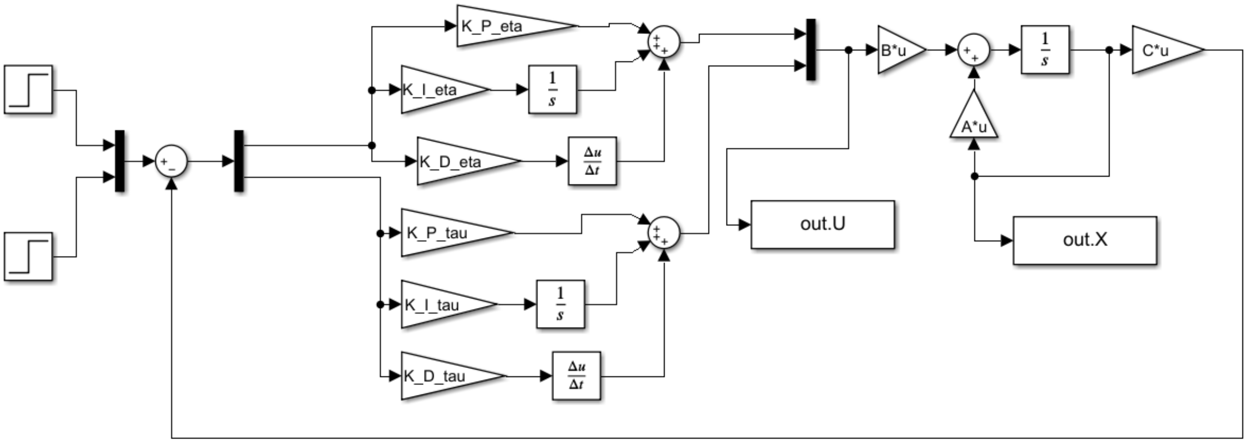


Figure 8 MIMO PID with direct feedback

The two PID controllers are tuned together by trail and error. The values of K_{P_η},K_{I_η}, K_{D_η}, K_{P_τ}, K_{I_τ} and K_{D_τ} are 50, 0.5, 1, 40000, 3000 and 2000 for all the rest simulations. The reference values are step inputs with zeros at the start and $\theta_{ref} = 0.2$rad and u_{ref}=300m/s after 2 seconds for all following simulations. All variables in x are plotted in Figure 9:

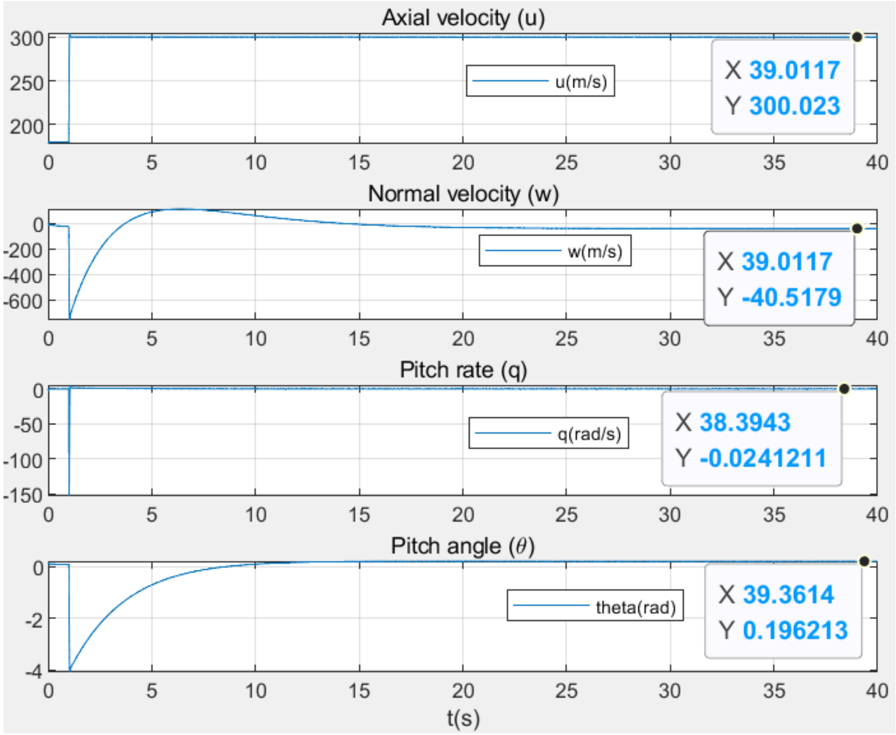


Figure 9 PID controlled model responses

The axial velocity u is almost 300m/s as expected at steady state and the pitch angle θ is quite close to 0.2 rad at steady state. It should be noted that tunning for a fast converging underdamped θ is really hard. The normal velocity approaches about -40m/s and the pitch rate approaches 0, which are reasonable. What are unrealistic are the extraordinary undershoot in θ, and the extremely high gradients in all variables. The accelerations in speed are as shown in Figure 10:

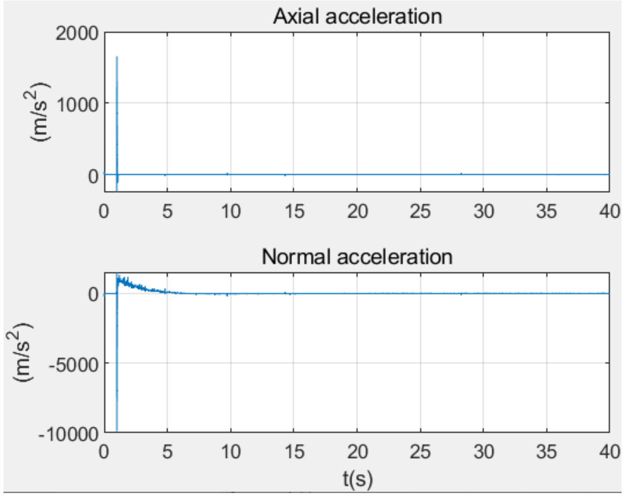


Figure 10 Accelerations in axial and normal directions with no input restriction

The values shown in Figure 10 cannot happen in the real world. But they could be due to the unrestricted controlling inputs, as shown in Figure 11:

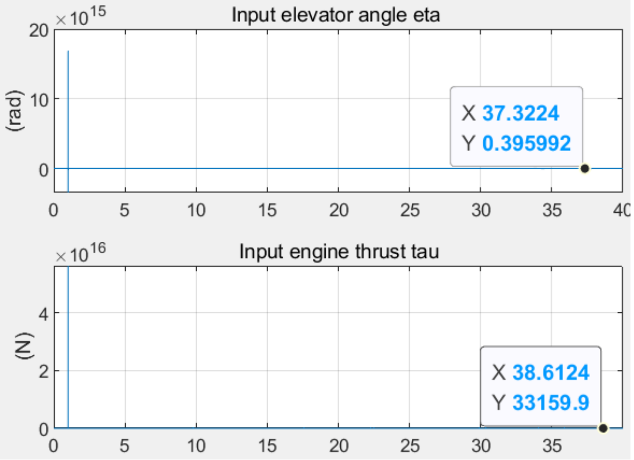


Figure 11 Inputs of elevators and engines under no restriction

These values are reaching extremely high, which are also unrealistic. The elevator angles should not go beyond π/4 anyway. At larger angles the linearized relationship between η and rolling moment will no longer be valid. Then restricting the inputs are tried. It is found that restricting η between ±π/4 rads and τ between 0 and 158760N (maximum engine thrust of F-4) with the

Simulink built-in saturation blocks will greatly affect the response profiles, as shown in Figure 12:

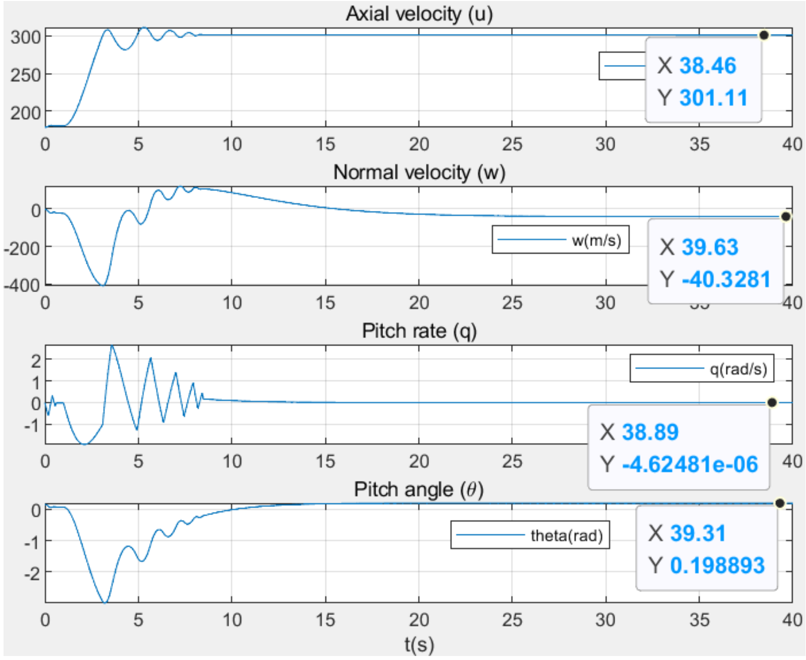


Figure 12 System responses with restricted inputs

As seen from Figure 12, the rise of u is smoother, but θ still has the -3 rad unrealistic undershoot. Probably this will not happen in the real world, as the aerodynamics will have already changed from the basics of this model. Another note is that K_{P_τ}, K_{I_τ} and K_{D_τ} are increased to 40000,3000 and 2000 respectively for θ to approach the reference value of 0.2 rads. The control inputs are shown in Figure 13:

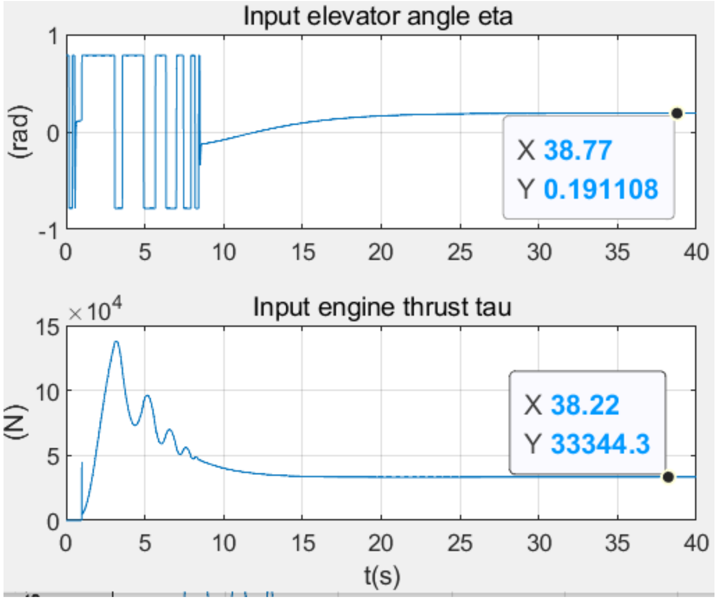


Figure 13 Restricted control inputs

As shown by Figure 13, the elevator angle experienced some very large fluctuations between its minimum and maximum values, then almost stabilized to a steady state of 0.19 rads. The engine thrust is now about 33kN at steady state. Then the horizontal and vertical speeds are as shown in Figure 14:

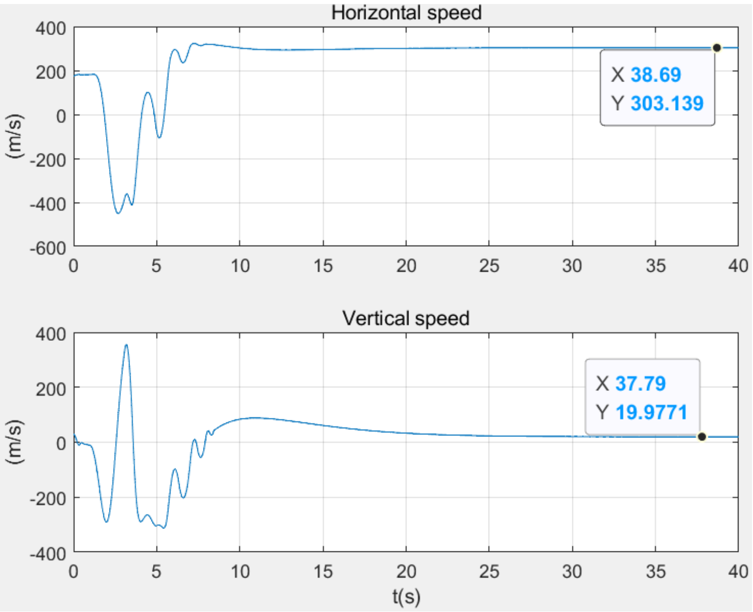


Figure 14 Derived horizontal and vertical speeds

Where the horizontal and vertical speeds $V_{x\prime}$ and $V_{z\prime}$ can be found as:

$$V_{x\prime} = sqrt(u^2 + w^2)\cos(\theta + atan2(w, u)) \qquad (26)$$

$$V_{z\prime} = sqrt(u^2 + w^2)\sin(\theta + atan2(w, u)) \qquad (27)$$

With $atan2(w, u)$ being the arctangent function able to figure out angles at any value between 0 and 2π.

Observer design

There can be errors in the inputs and output measurements. As such, observers can be utilized to reduce the errors in the state measurements. Observers can be designed conveniently by pole placement.

$$\dot{\hat{x}}(t) = A\hat{x}(t) + B_O u(t) + L\big(y(t) - C\hat{x}(t)\big) \qquad (28)$$

[3]

Where $\hat{x}$ is the state estimation. The matrix B_O is selected as an identity matrix with the same size as A.

The matrix L can be designed by pole placement. The larger the absolute values of the poles, the faster the observed state converges to the true state, but the ability to filter out errors also reduces. The observed state $\dot{\hat{x}}$ is used for feedback in stead of the output $\dot{x}$.

In the case in this project there are a total of 8 observation poles, 4 for observing u and 4 for observing θ. The Ackermann's formula can be used for finding proper observer gains with poles known. The approach is to find the polynomial with the desired poles, then solve det $(sI - (A - LC))$ equaling to that polynomial. [2] The pole placement method for finding observer gain L can easily be implemented in MATLAB as $acker(A^T, C^T, ObsPoles)$, assuming

$$ObsPoles = [p_1\ p_2\ p_3\ p_4] \qquad (29)$$

Note that Ackermann’s formula only works for row vectors of poles. For such MIMO system, the Ackermann’s formula must be used for 2 times for each row of C and each four poles to find the two columns of L for observing u and θ respectively.

Another way is to use LQR observer. The process to design the LQR observer is similar to designing the LQR controller, but substituting B with C^T. Firstly the Ricatti’s equation is solved: [3]

$$P_eA^T + AP_e - P_eC^TR^{-1}CP_e + Q = 0 \qquad (29)$$

Then use P_e to compute: [3]

$$L = P_eC^TR^{-1} \qquad (30)$$

Where Q and R are diagonal covariance matrices of the variances of input and output noises of the system, respectively (Q for input and R for output).

The LQR method can also be implemented in MATLAB conveniently, by using $[L_{transpose}, P_e] = LQR(A^T, C^T, Q, R)$. Then the observer gain $L = L_{transpose}^T$.

The Simulink diagram of the observer is shown in Figure 15:

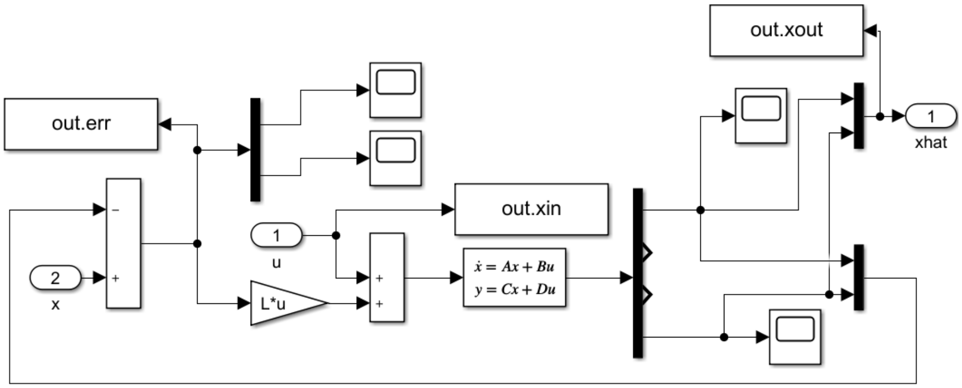


Figure 15 Simulink LQR observer

With the full system diagram in Figure 16:

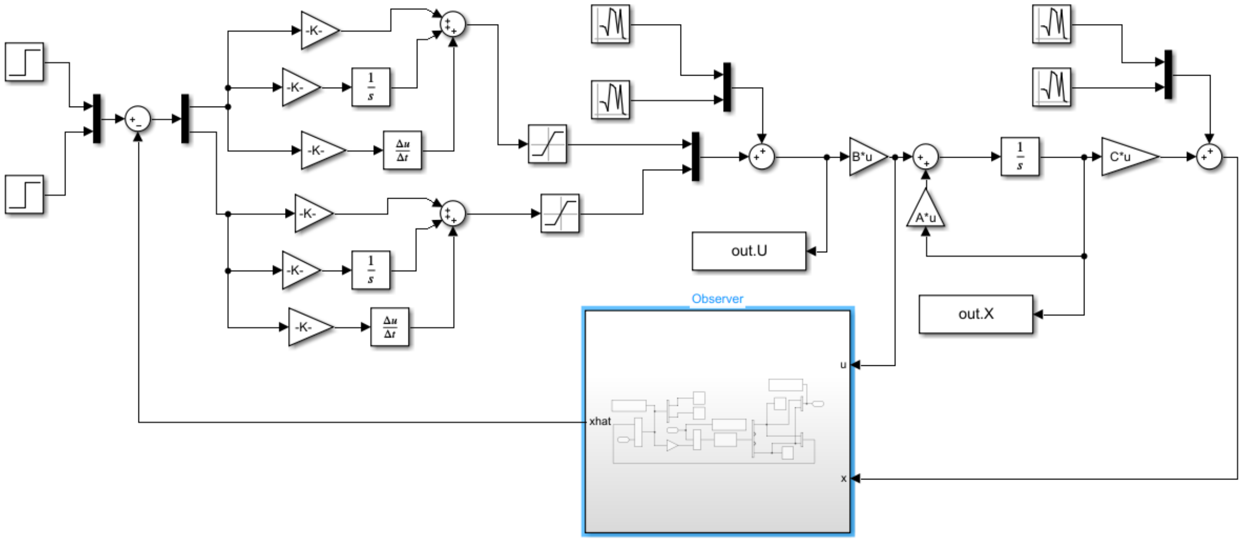


Figure 16 Whole system with observer

Where the B and C matrices in the state space block in Figure 16 are both 4*4 identity matrices.

The state plots are computed with Gaussian zero-mean noises:

$$Q = diag([(0.05rad)^2, 0,0, (5N)^2]) \qquad (31)$$

$$R = diag([(2m/s)\text{^}2, (0.02rad)\text{^}2]) \qquad (32)$$

The plots of all values in x with the use of an LQR observer, in comparison with a system with direct feedback noisy output and a clean system (Figure 12) with no noise are shown in Figure 17:

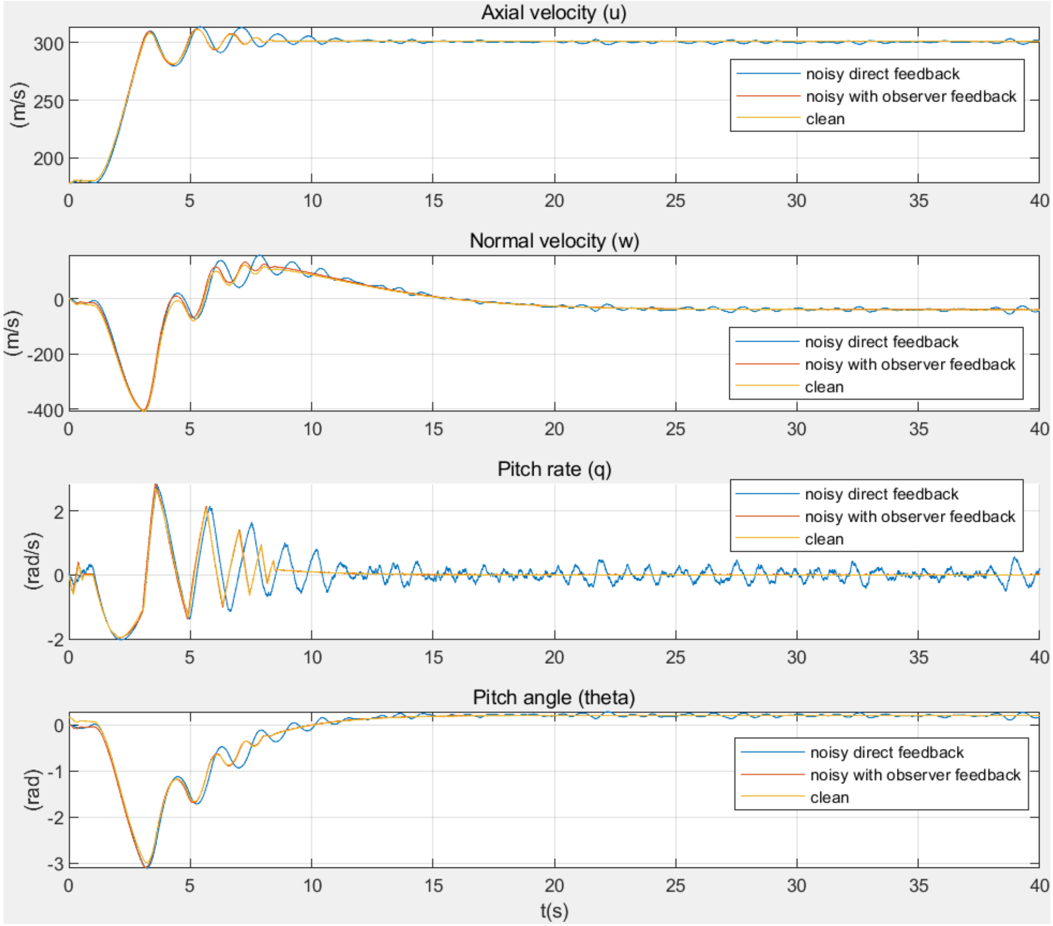


Figure 17 Comparison between three system configurations

Noticeably, the system responses are much smoother than directly feeding back the noisy output, and the noises are barely noticeable with the use of observers. Without using the observer, there are much more oscillations. The noises rejected by the observer are shown in Figure 18:

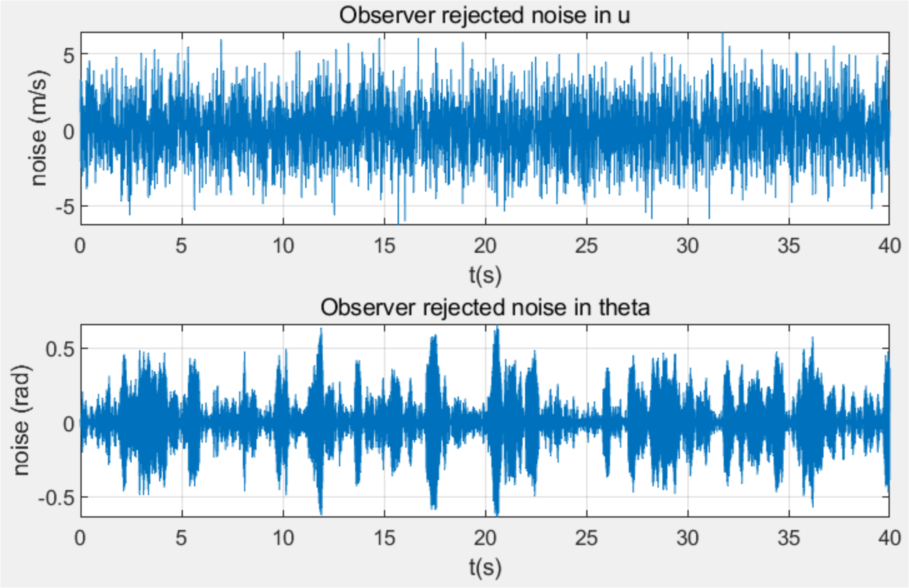


Figure 18 Observer rejected noise

This LQR observer did an excellent job in rejecting a great amount of noises. Then, the vertical and horizontal speeds for the three models are shown in Figure 19:

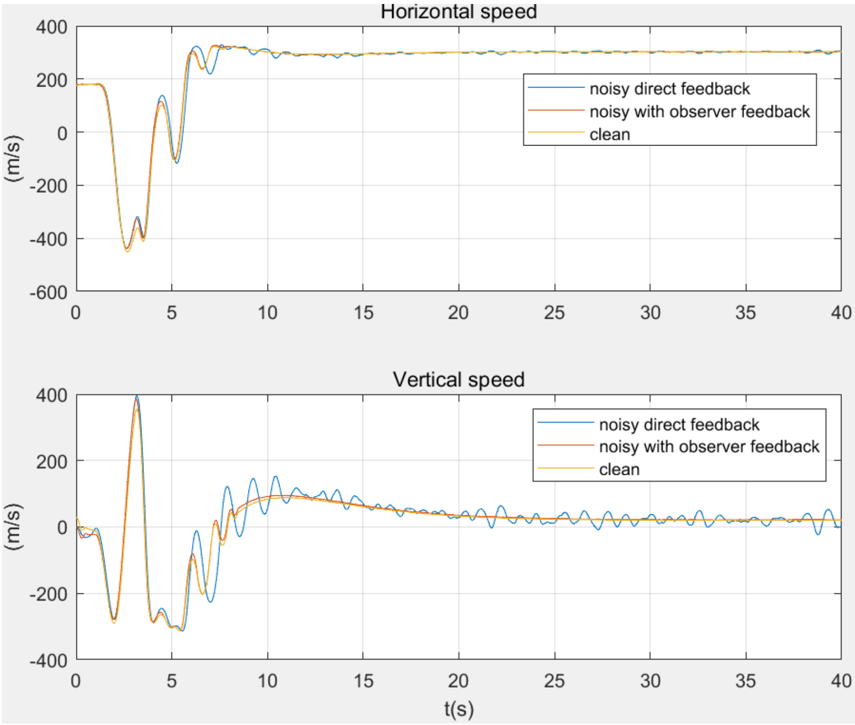


Figure 19 Horizontal and vertical speeds under three system configurations

The horizontal and vertical speeds are really smooth by using an LQR observer. Without the observer, the noise keeps the states from stabilizing. The vertical speed approaches about 20m/s in the steady state, indicating the aircraft is climbing at 20m/s under such conditions. Now have a look at the control inputs:

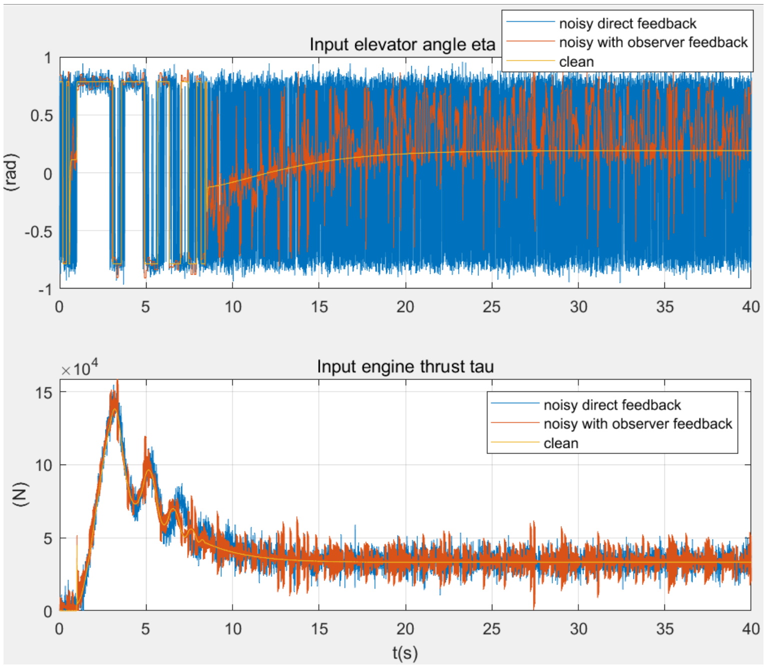


Figure 20 Noises in input under three system configurations

The observer effectively reduced the controlling efforts and fluctuations in the elevator. This is possible since a noisy feedback to the system will stimulate the control inputs and lead to accumulated noise in the next states, while the observer avoids the accumulation of noises. Note that the addition of controlling noise happens at this stage, so all the added controlling noise are recorded in Figure 20.

Kalman Filter

The LQR observer monitors and rejects the noise in the input and output of the plant. But there is another type of sensor combination, being accelerometer and speedometer. Typically airplanes have built-in inertia navigation systems containing accelerometers, angular accelerometers and more sensors, in addition with Pitot tubes for air speed measuring and AOA sensors combined with other sensors to output pitch angle. Kalman filters are excellent candidates for state estimating in such combinations with lower and higher order sensors (i.e. when a sensor measures in m/s and another measures in m/s^2, the first one is said to be higher-order). The requirements to the sensors are that they have zero-mean Gaussian noises.

The equations of Kalman filters are shown in Equations 33 to 37, recursively in k: [4]

$$\check{P}_k = A_{k-1}\hat{P}_{k-1}A_{k-1}^T + Q_k \qquad (33)$$

$$\check{x}_k = A_{k-1}\hat{x}_{k-1} + v_k \qquad (34)$$

$$K_k = \check{P}_k C_k^T \left(C_k \check{P}_k C_k^T + R_k\right)^{-1} \qquad (35)$$

$$\hat{P}_k = (I - K_k C_k)\check{P}_k \qquad (36)$$

$$\hat{x}_k = \check{x}_k + K_k(y_k - C_k\check{x}_k) \qquad (37)$$

Whereas the notation '^' denotes estimated states and 'ˇ' denotes expected states. The estimated states are derived from the expected states in one recursion. P is the covariance matrix and K is the Kalman gain. A is the relationship between x in the last recursion and x in this recursion, and v is the integration of the lower-order measurements for one timestep. v can be written as $B * v'$, B then represents the timestep matrix in this case and v' are the raw lower-order measurements. Note that the A, B and C matrices here in the Kalman filter differ from those in the analyzed system state space. Q is the diagonal variance matrix of the lower-order measurements and R is the diagonal variance matrix of the higher-order measurements. y is the raw higher-order measurements.

In this project, suppose the aircraft has lower-order measurements for axial acceleration a_o and angular pitch rate q_o, together with higher-order measurements axial velocity u_o and pitch θ_o. Then, in the Kalman filter, $x_k = \begin{bmatrix} u_k \\ \theta_k \end{bmatrix}$ are the states to be estimated for all k, $y_k = \begin{bmatrix} u_{o_k} \\ \theta_{o_k} \end{bmatrix}$ are the measurements for u and θ for each timestep k, and $v'_k = \begin{bmatrix} a_{o_k} \\ q_{o_k} \end{bmatrix}$ are the measurements for axial acceleration and pitch rate for each timestep k. Then $A = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$, $B = \begin{bmatrix} t_s & 0 \\ 0 & t_s \end{bmatrix}$ and $C = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$ can be defined to cover all the measurements, where t_s is the sampling time. Usually A, B and C are constant. Similar as mentioned in the LQR observer, Q and R are diagonal covariance matrices of the variances of lower-order and higher-order measurement noises, respectively (Q for lower-order and R for higher-order). To be clear, variance is computed as the square of standard deviation for zero-mean Gaussian noises. Usually, Q and R are taken as constants (i.e. assumed constant sensor noise variances).

Simulink has a built-in Kalman filter block, for which A, B, C, Q, R and more quantities can be specified, and it will work. But for better demonstrating, a customized Kalman filter is built in Simulink, as shown in Figure 21:

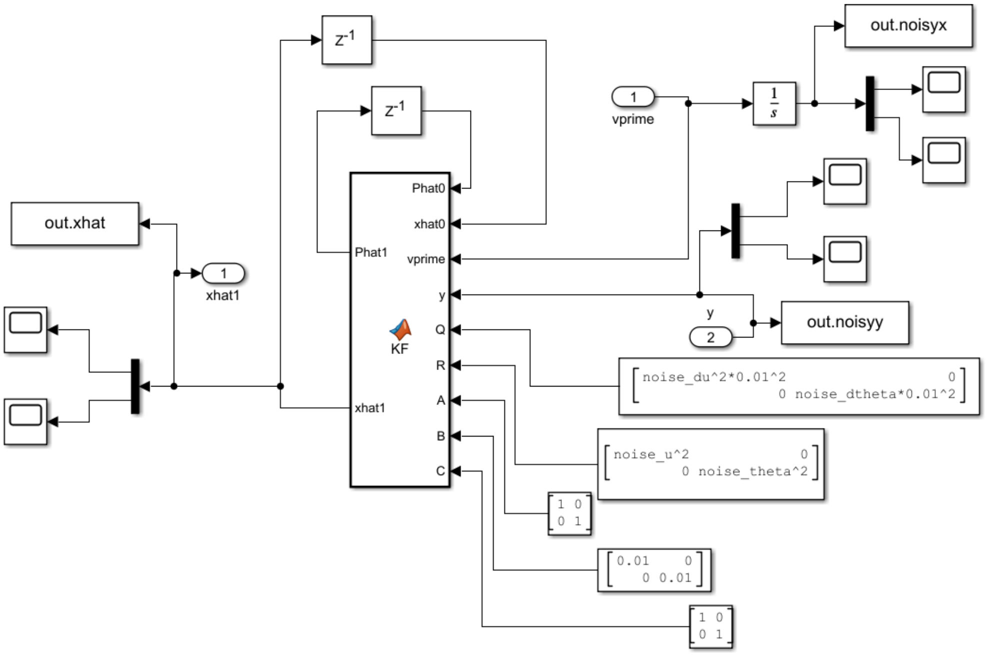


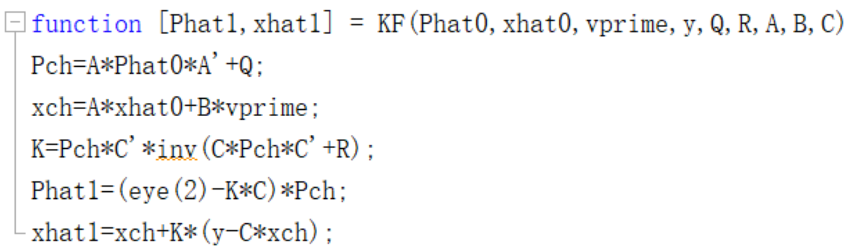
Figure 21 Kalman filter built in Simulink

In Figure 21, Q and R are, respectively:

$$Q = diag(\left[(0.05rad/s * 0.01s)^2, 0{,}0, \left(\frac{3m}{s^2} * 0.01s\right)^2\right]) \qquad (38)$$

$$R = diag([(0.05rad)^2, \left(\frac{3m}{s}\right)^2]) \qquad (39)$$

Note that the 0.01s terms in Q is the sampling time, for uniformizing its units with x. The zero order holds save the initial P_0 and $\hat{x}_0$ values. They also send the Kalman filter their values from the last run. P_0 is usually a guess, here being $\begin{bmatrix} 0.1 & 0 \\ 0 & 0.1 \end{bmatrix}$, and $\hat{x}_0$=$\begin{bmatrix} 0 \\ 0 \end{bmatrix}$. Inside the KF function block, the function in Figure 22 is written:



```
function [Phat1,xhat1] = KF(Phat0,xhat0,vprime,y,Q,R,A,B,C)
Pch=A*Phat0*A' +Q;
xch=A*xhat0+B*vprime;
K=Pch*C' *inv(C*Pch*C' +R);
Phat1=(eye(2)-K*C)*Pch;
xhat1=xch+K*(y-C*xch);
```

Figure 22 Simulink Kalman filter written function

which are exactly Equations 33~37.

The full diagram of the controlled system is then shown in Figure 23:

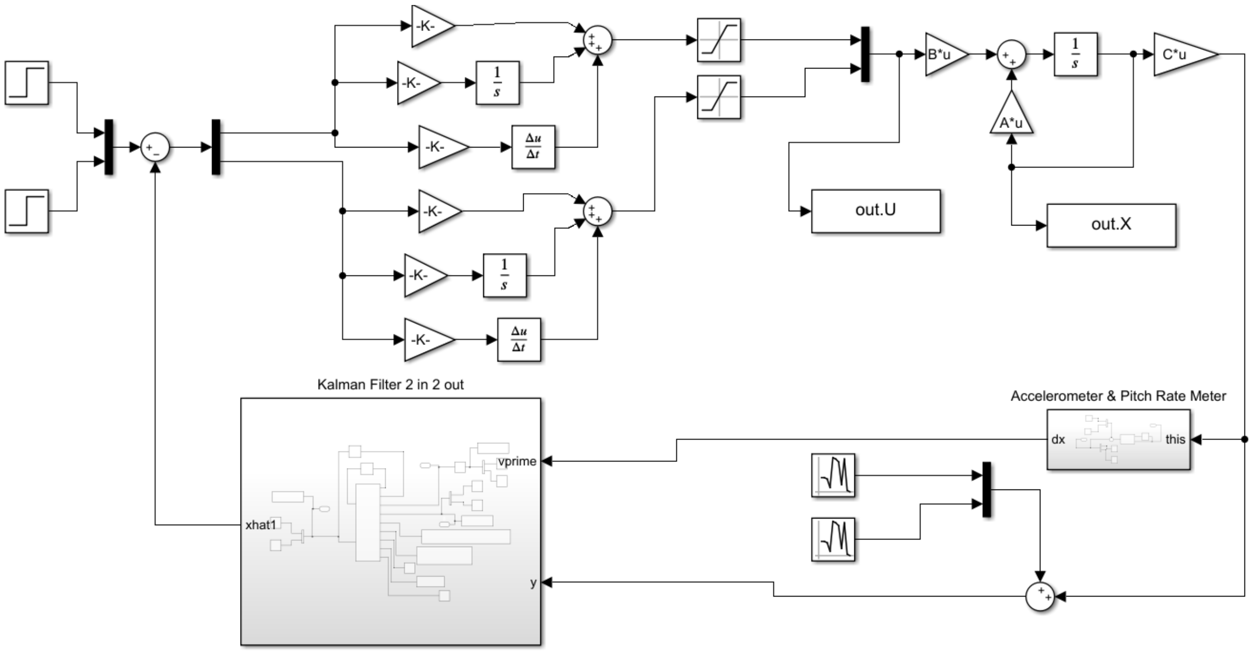


Figure 23 Whole controlled system with Kalman filter and emulated noisy sensors

The PID controllers in Figure 23 are not changed. The accelerometer & pitch rate meter block has noises in itself, being 3m/s^2 and 0.05rad/s respectively in standard deviation. The noises added to the u and θ measurements are 3m/s and 0.05rad respectively in standard deviation. This unit is a simulation of real word sensors and in this model it functions by differentiating the system output then add Gaussian noises. The reference of u is still a step input from 180m/s to

300m/s at 2s, and the reference of θ is still a step input from 0 to 0.2rad at 2s. The state plots are shown in Figure 24:

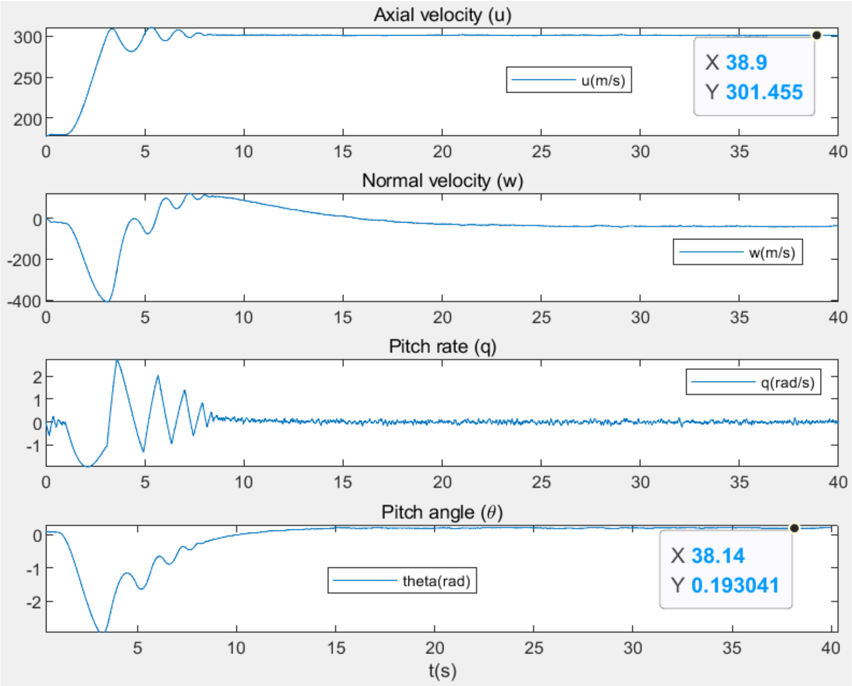


Figure 24 System responses with Kalman filter included

As seen from Figure 24, the state plots are barely noisy. Figure 25 compares the outputs of two orders of sensors with the Kalman filter outputs and the noises can be easily seen.

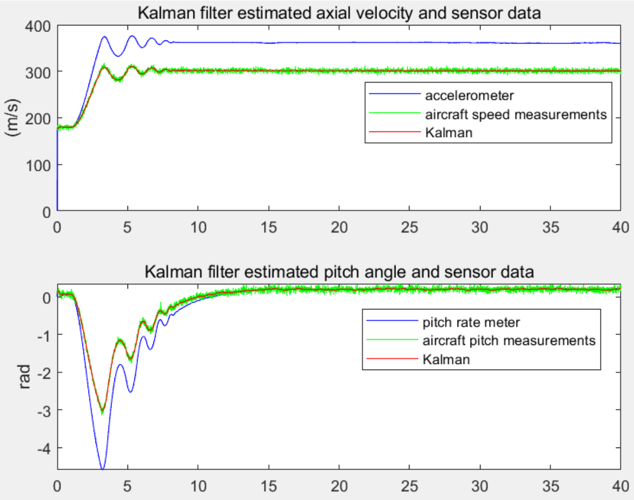


Figure 25 Kalman filter estimations

Zooming in for a part of the plots is shown in Figure 26:

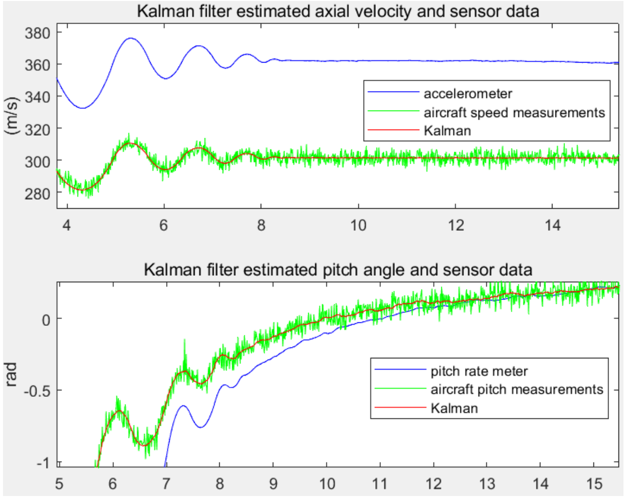


Figure 26 Kalman filter estimations magnified

The Kalman filter even worked well for an accelerometer with some bias error, as presented in Figure 26. It is able to stay almost unbiased while the high frequency noise from the higher-order measurements is also greatly reduced.

The horizontal and vertical speeds of this aircraft is shown in Figure 27:

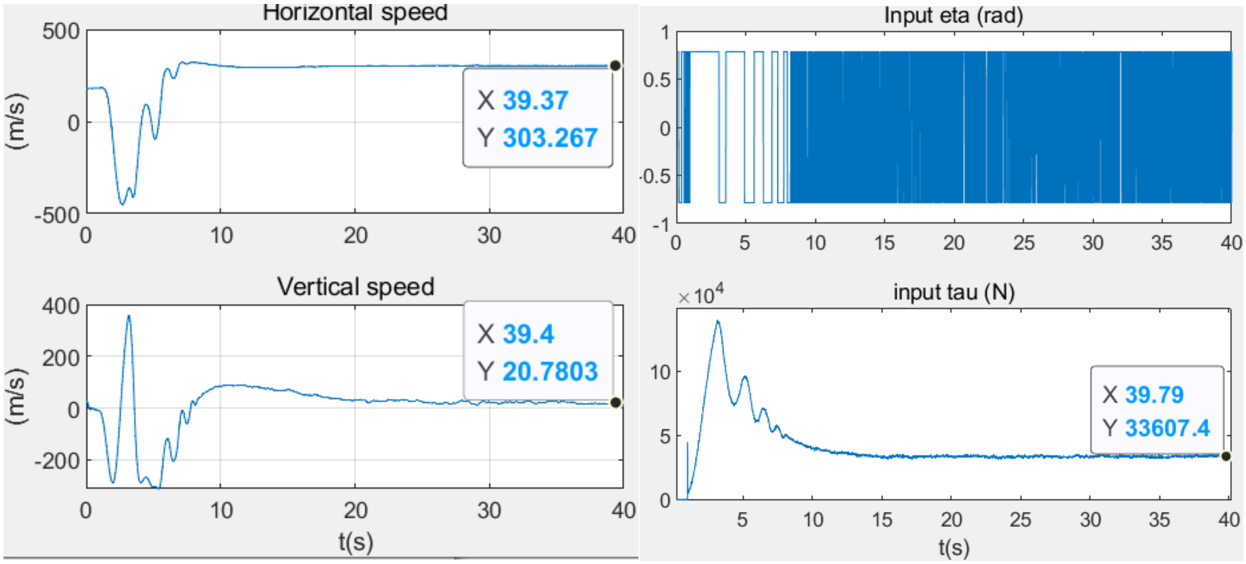


Figure 27 Speeds and inputs

Clearly, the aircraft is climbing with a vertical speed of approximately 20m/s. The engine thrust looks reasonable, being about 33kN. But the elevator angle is not stabilizing. This would not happen in the reality as they can not actuate at such a high frequency. Other things extraordinary are the horizontal speed undershooting to nearly -500m/s and the pitch angle undershooting to about -3 rads. These are not allowed in the reality, and could be caused by improper controlling gains or errors in the model.

5. Conclusion

The linearized model with aerodynamic parameters evaluated at equilibrium conditions discussed represents the actual aircraft fairly well when it is not deviated to far from the equilibrium. The zero-input responses are reasonable at the beginning but fails later when state variables have changed too much (i.e. u goes below 50m/s or supersonic, θ exceeds $\pi/2$). When the control inputs are not restricted, they will have extremely high spikes right after the step input, making the aircraft reaching reference speed almost immediately. It is found that no matter how the PID gains are tuned, θ will not converge to reference quickly and it always has an unrealistic undershoot of about 3 rads. When the control inputs are restricted to $-4.5\pi<\eta<4.5\pi$ and $0<\tau<158760$N, the behaviors are much smoother, but still with unusual aspects. Such aspects include negative horizontal speed at some time, and rapidly oscillating stabilizers for the beginning period of time. The LQR observer is found to attenuate noises in the inputs and state outputs effectively. It not only makes the outputs smoother, but also eases elevator controlling input. The Kalman filter works excellently to fuse measurement data from different types of sensors, with almost non-biased output and greatly reduced noise. Overall, the longitudinal aircraft model in this book meets the reality in most of the aspects, and will work reasonably when the controlling input is not large enough to drive it too far away from equilibrium. The

MIMO PID controllers are hard to tune but they work fairly well when tunned properly, except a slow converging of pitch angle. Also, LQR observers and Kalman filters are useful for state estimations and stabilizing the system controls.

6. References

[1] M. Cook. (2007). *Flight Dynamic Principles*. ScienceDirect. Retrieved from https://www-sciencedirect-com.myaccess.library.utoronto.ca/book/9780750669276/flight-dynamic-principles

[2] R. Buchi & M. Kottmann. (2011). *Advanced Control State Regulator with Observer*. MASTER OF SCIENCE IN ENGINEERING. Retrieved from https://www.researchgate.net/profile/Mohamed_Mourad_Lafifi/post/initial_condition_for_observer/attachment/5dd92817cfe4a777d4f3b1fa/AS%3A828380303851520%401574512663125/download/State+Regulator+with+Observer.pdf

[3] K. Passino & N. Quijano. (2002). *Linear Quadratic Regulator and Observer Design for a Flexible Joint*. The Ohio State University. Retrieved from http://eewww.eng.ohio-state.edu/~passino/lab4prelab.pdf

[4] T. Barfoot. (2017). *STATE ESTIMATION FOR ROBOTS*. Cambridge University Press. doi.org/10.1017/9781316671528

[5] M. Sadraey & V. Muller. (2009). *CHAPTER 3 Drag Force and Drag Coefficient*. Retrieved from https://ww2aircraft.net/forum/attachments/chapter-3-drag-force-and-its-coefficient-pdf.288335/ %F4?

[6] S. Anatolii, H. Htun, Z. Naing & H. Paing. (2009). *Designing, Simulation and Control of Autopilot using PID Controller*. IEEE. DOI: 10.1109/ElConRus51938.2021.9396112

[7] E. Itoh & S. Suzuki. (2013). *Evaluation on novel architecture for harmonizing manual and automatic flight controls under atmospheric turbulence*. ELSAVIER. doi.org/10.1016/j.ast.2011.11.012

Appendix 1 Dimensionless aerodynamic stability derivatives

$$X_u = \left(-2C_D - V_0\frac{\partial C_D}{\partial V} + \frac{1}{\frac{1}{2}\rho V_0 S}\frac{\partial \tau}{\partial V}\right)$$

$$X_w = (C_L - \frac{\partial C_D}{\partial \alpha})$$

$$X_q = \left(-\bar{V}_T\frac{\partial C_{D_T}}{\partial \alpha_T}\right)$$

$$X_\eta = \left(-\frac{2S_T}{S}k_T C_{L_T} a_2\right)$$

$$X_{\dot{w}} = \left(X_q\frac{d\varepsilon}{d\alpha}\right)$$

$$X_\tau = \frac{1}{0.5\rho C_d A_f V_0^2 cos\theta_0}$$

$$Z_\tau = \frac{1}{0.5\rho C_d A_f V_0^2 sin\theta_0}$$

$$M_u = \left(V_0\frac{\partial C_m}{\partial V}\right)$$

$$M_w = \left(\frac{dC_m}{d\alpha}\right)$$

$$M_\eta = (-\bar{V}_T a_2)$$

$$Z_q = (-\bar{V}_T a_1)$$

$$Z_\eta = -\frac{S_T}{S}a_2$$

$$M_{\dot{w}} = \left(-\frac{\bar{V}_T a_1 l_T}{\bar{\bar{c}}}\frac{d\varepsilon}{d\alpha}\right)$$

$$Z_{\dot{w}} = Z_q\frac{d\varepsilon}{d\alpha}$$

$$Z_w = -C_D - \frac{\partial C_L}{\partial \alpha}$$

$$Z_u = -2C_L - V_0\frac{\partial C_L}{\partial \alpha}$$

$$M_q = Z_q * \frac{l_T}{\bar{\bar{c}}}$$

Appendix 2 Dimensional aerodynamic stability derivatives

$$\breve{X}_u = \left(\frac{1}{2}\rho V_0 S\right)\left(-2C_D - V_0\frac{\partial C_D}{\partial V} + \frac{1}{\frac{1}{2}\rho V_0 S}\frac{\partial \tau}{\partial V}\right)$$

$$\breve{X}_w = \left(\frac{1}{2}\rho V_0 S\right)(C_L - \frac{\partial C_D}{\partial \alpha})$$

$$\breve{X}_q = \left(\frac{1}{2}\rho V_0 S\bar{\bar{c}}\right)\left(-\bar{V}_T\frac{\partial C_{D_T}}{\partial \alpha_T}\right)$$

$$\breve{X}_\eta = \left(\frac{1}{2}\rho V_0^2 S\right)\left(-\frac{2S_T}{S}k_T C_{L_T} a_2\right)$$

$$\breve{X}_{\dot{w}} = \left(\frac{1}{2}\rho S\bar{\bar{c}}\right)\left(X_q\frac{d\varepsilon}{d\alpha}\right)$$

$$\breve{X}_u = \left(\frac{1}{2}\rho V_0 S\right)\left(-2C_D - V_0\frac{\partial C_D}{\partial V} + \frac{1}{\frac{1}{2}\rho V_0 S}\frac{\partial \tau}{\partial V}\right)$$

$$\breve{X}_w = \left(\frac{1}{2}\rho V_0 S\right)(C_L - \frac{\partial C_D}{\partial \alpha})$$

$$\breve{X}_q = \left(\frac{1}{2}\rho V_0 S\bar{\bar{c}}\right)\left(-\bar{V}_T\frac{\partial C_{D_T}}{\partial \alpha_T}\right)$$

$$\breve{X}_\eta = \left(\frac{1}{2}\rho V_0^2 S\right)\left(-\frac{2S_T}{S}k_T C_{L_T} a_2\right)$$

$$\breve{X}_{\dot{w}} = \left(\frac{1}{2}\rho S\bar{\bar{c}}\right)\left(X_q\frac{d\varepsilon}{d\alpha}\right)$$

$$\breve{M}_u = \left(\frac{1}{2}\rho V_0 S\bar{\bar{c}}\right)\left(V_0\frac{\partial C_m}{\partial V}\right)$$

$$\breve{M}_w = \left(\frac{1}{2}\rho V_0 S\bar{\bar{c}}\right)\left(\frac{dC_m}{d\alpha}\right)$$

$$\breve{M}_\eta = \left(\frac{1}{2}\rho V_0^2 S\bar{\bar{c}}\right)(-\bar{V}_T a_2)$$

$$\breve{Z}_q = \left(\frac{1}{2}\rho V_0 S\bar{\bar{c}}\right)(-\bar{V}_T a_1)$$

$$\breve{M}_{\dot{w}} = \left(\frac{1}{2}\rho S\bar{\bar{c}}^2\right)(-\frac{\bar{V}_T a_1 l_T}{\bar{\bar{c}}}\frac{d\varepsilon}{d\alpha})$$