Aerosonde Modeling and Control, Checkpoint2

Joseph Kennedy
josephpk@umich.edu

I. Introduction

Autonomous drones are used in many different fields to achieve many different goals, and today their operations include reconnaissance, transportation, and data collection. An early pioneer in autonomous drones was Aerosonde Robotic Aircraft Ltd of Australia (now part of Textron Systems), which devopled the Aerosonde fixed wing drone in the 1990s, as shown in Figure 1. In 1998 the Arosonde "Laima" made the first autonomous flight across the Atlantic, crossing in 26 hours and 45 minutes [1]. The Arosonde is designed as a remote sensing platform with GPS navigation that can measure temperature, humidity, pressure, and wind [2]. The Aerosonde's specifications and modeling parameters are well documented and publicly available. The specifications were provided in the AeroSim Blockset for Simulink. These values have also been used to validate the modeling parameters derived for other similar platforms [3].

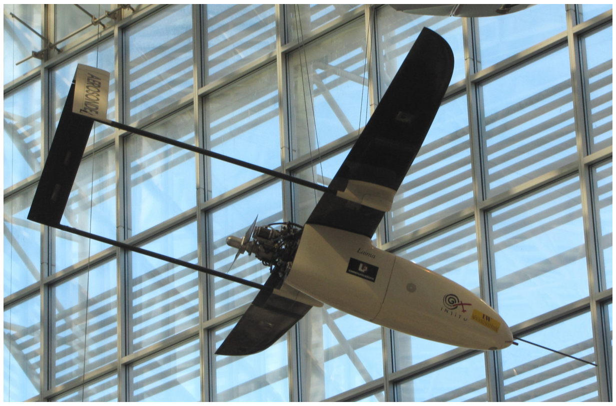

Fig. 1. The Aerosonde Laima.

In this paper, the equations of motion for a general fixed wing drone will be described and then applied specifically to model the Aerosonde. The equations will be lineraized so that the control characteristcs of the system can be determined. Next, the limits of the control inputs will be estimated so that the system can be modelled in a realistic way. Then, an LQR controller will be implemented. Finally, the controller will be tested on a series of waypoints.

II. Assumptions

The drone will be modeled as a rigid body with a plane of mass symmetry. Looking from above the drone, a line can be drawn from the nose to the tail which will divide the drone symmetrically.

Three separate reference frames will be used to describe the motion of the drone. The earth frame will have its origin centered at ground level, with the x axis pointing north, the y axis pointing east, and the z axis pointing into the earth. This axis will remain fixed in place, so the position of the drone with respect to this frame will help determine if it has reached set waypoints. The earth frame is assumed to be an inertial frame, so it is assumed that the earth does not rotate or accelerate during flight. A flat earth is also assumed.

The body frame will have its origin at the center of gravity of the drone. The x axis points from the center to the nose of the drone along the plane of mass symmetry, the z axis points down along the plane of mass symmetry, and the y axis points out along the wing to complete the right handed coordinate frame. The body frame is useful for describing angular velocity since is is the only frame where the moments of inertia are constant.

Finally, the wind frame also has its origin at the center of gravity, but the x axis points along the velocity vector for the drone, the z axis normal to the x axis in the plane of mass symmetry, and the y axis completing the right hand coordinate frame. The wind frame is useful for describing the aerodynamic forces.

Twelve state variables will be used to describe the system. (x, y, z) will describe the location of the drone in the earth frame. (ϕ, θ, ϕ) are the Euler angles that represent the rotation from the earth frame to the body frame. (u, v, w) are the velocities of the drone with respect to the body frame. (p, q, r) are the angular velocities with respect to the body frame.

Four inputs will be used to control the system. $Thrust$ will represent the force applied by the motor. $\delta elevator$ will represent the angle of the elevator surface used to control pitch. $\delta aileron$ will represent the angle of the ailerons used to control roll. Finally, $\delta rudder$ will represent the angle of the rudder used to control yaw.

The aerodynamic forces of Lift (L) and Drag (D) will be applied in the wind frame. The drag force acts opposite to the velocity vector. The lift force acts normal to the velocity vector in the plane of mass symmetry. The force of gravity, g will act in the earth frame, and gravity is constant. F_y is a force that acts in the y axis of the body frame, which is influenced by p, r, $\delta aileron$, and $\delta rudder$.

The torques about the body x axis, y axis, and z axis are represented as (l, m, n), respectively.

It is assumed that the atmosphere is stationary with respect to the earth frame.

It is assumed that the thrust is a force produced along the body x axis. It is assumed that the thrust vector originates at the center of mass, and thus the thrust does not generate a pitching torque m. This is assumed because there is no Aerosonde parameter given to relate m as a function of $Thrust$. Just from pictures of the Aerosonde, this appears to be a reasonable assumption, since the motor seems very near to the center of mass.

V is the magnitude of the velocity vector. α is the angle of attack, or the angle from the z axis of the wind frame to the z axis of the body frame. β is the sideslip angle, or the angle from the the y axis of the body frame to the y axis of the wind frame.

$$V = \sqrt{u^2 + v^2 + w^2} \tag{1}$$

$$\alpha = \arctan w/u \tag{2}$$

$$\beta = \arcsin v/V \tag{3}$$

III. Equations of Motion

The translational kinematics of the drone in Equation 4 relate the velocities in the body frame (u, v, w) to the velocities in the earth frame $(\dot{x}, \dot{y}, \dot{z})$. s is $sine$ and c is $cosine$.

$$\begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{z} \end{bmatrix} = \begin{bmatrix} c\theta c\psi & -s\psi c\phi + c\psi s\theta s\phi & s\theta s\psi + c\phi s\theta c\psi \\ c\theta s\psi & c\phi c\psi + s\phi s\theta s\psi & -c\psi s\phi + s\psi s\theta c\phi \\ -s\theta & s\phi c\theta & c\phi c\theta \end{bmatrix} \begin{bmatrix} u \\ v \\ w \end{bmatrix} \tag{4}$$

The rotational kinematics of the drone in Equation 5 relate the angular velocities in the body frame (p, q, r) to the angular velocities in the earth frame $(\dot{\phi}, \dot{\theta}, \dot{\psi})$

$$\begin{bmatrix} \dot{\phi} \\ \dot{\theta} \\ \dot{\psi} \end{bmatrix} = \begin{bmatrix} 1 & s\phi tan\theta & c\phi tan\theta \\ 0 & c\phi & -s\phi \\ 0 & s\phi sec\theta & c\phi sec\theta \end{bmatrix} \begin{bmatrix} p \\ q \\ r \end{bmatrix} \tag{5}$$

The translational dynamics of the drone in Equation 6 are obtained by using the transport theorem to obtain Newton's second law in a non inertial frame (the body frame). Then all the forces are represented in the body frame and summed together. M is the mass of the drone.

$$\begin{bmatrix} \dot{u} \\ \dot{v} \\ \dot{w} \end{bmatrix} = \begin{bmatrix} -gs\theta - \frac{Dc\alpha c\beta}{M} + \frac{Ls\alpha}{M} + \frac{Thrust}{M} - qw + rv \\ gc\theta s\phi - \frac{Ds\beta}{M} + F_y - ru + pw \\ gc\theta c\phi - \frac{Ds\alpha c\beta}{M} - \frac{Lc\alpha}{M} - pv + qu \end{bmatrix} \tag{6}$$

The rotational dynamics of the drone in Equation 7 are obtained by using the transport theorem for Newton's second law for angular accelerations. I_{xx} is the moment of inertia about the x axis, I_{yy} is the moment of inertia about the y axis, and I_{zz} is the moment of inertia about the z axis. I_{xz} is the product of inertia, which is nonzero because the drone does not have mass symmetry about all axes.

$$\begin{bmatrix} \dot{p} \\ \dot{q} \\ \dot{r} \end{bmatrix} = \begin{bmatrix} \tau_1 pq - \tau_2 qr + \tau_3 l + \tau_4 n \\ \tau_5 pr - \tau_6 (p^2 - r^2) + \frac{m}{I_{yy}} \\ \tau_7 pq - \tau_1 qr + \tau_4 l + \tau_8 n \end{bmatrix} \tag{7}$$

where

$$\tau = I_{xx} I_{zz} - I_{xz}^2 \tag{8}$$

$$\tau_1 = I_{xz}(I_{xx} - I_{yy} + I_{zz})/\tau \tag{9}$$

$$\tau_2 = (I_{zz}(I_{zz} - I_{yy}) + Ixz^2)/\tau \tag{10}$$

$$\tau_3 = I_{zz}/\tau \tag{11}$$

$$\tau_4 = I_{xz}/\tau \tag{12}$$

$$\tau_5 = (I_{zz} - I_{xx})/I_{yy} \tag{13}$$

$$\tau_6 = I_{xz}/I_{yy} \tag{14}$$

$$\tau_7 = ((I_{xx} - I_{yy})I_{xx} + I_{xz}^2)/\tau \tag{15}$$

$$\tau_8 = I_{xx}/\tau \tag{16}$$

S is the surface area of the wings. b is the wingspan. c is the wing chord. ρ is the air density.

L, D, and m are primarily influenced by α, q, and $\delta elevator$.

$$\begin{aligned} L = \frac{1}{2}\rho V^2 S(C_{L0} + C_{L\alpha}\alpha + C_{Lq}\frac{c}{2V}q \\ + C_{L\delta elevator}\delta elevator \end{aligned} \tag{17}$$

$$\begin{aligned} D = \frac{1}{2}\rho V^2 S(C_{D0} + C_{D\alpha}\alpha + C_{Dq}\frac{c}{2V}q \\ + C_{D\delta elevator}\delta elevator \end{aligned} \tag{18}$$

$$\begin{aligned} m = \frac{1}{2}\rho V^2 Sc(C_{m0} + C_{m\alpha}\alpha + C_{mq}\frac{c}{2V}q \\ + C_{m\delta elevator}\delta elevator \end{aligned} \tag{19}$$

F_y, l, and n are primarily influenced by β, p, r, $\delta aileron$, and $\delta rudder$.

$$F_y = \frac{1}{2}\rho V^2 S(C_{y0} + C_{y\beta}\beta + C_{yp}\frac{b}{2V}p + C_{yr}\frac{b}{2V}r$$
$$+C_{y\delta aileron}\delta aileron + C_{y\delta rudder}\delta rudder) \quad (20)$$

$$l = \frac{1}{2}\rho V^2 Sb(C_{l0} + C_{l\beta}\beta + C_{lp}\frac{b}{2V}p + C_{lr}\frac{b}{2V}r$$
$$+C_{l\delta aileron}\delta aileron + C_{l\delta rudder}\delta rudder) \quad (21)$$

$$n = \frac{1}{2}\rho V^2 Sb(C_{n0} + C_{n\beta}\beta + C_{np}\frac{b}{2V}p + C_{nr}\frac{b}{2V}r$$
$$+C_{n\delta aileron}\delta aileron + C_{n\delta rudder}\delta rudder) \quad (22)$$

The coefficients are dimensionless quantities that represent how much one variable affects another variable. The coefficients for the Aerosonde are publicly available [4]. Because the coefficients are dimensionless quantities, p, q, and r must be nondimensionalized, which is done using either the expression $\frac{b}{2V}$ or $\frac{c}{2V}$.

IV. Linearizing the Model

The equations of motion can be linearized about a trajectory. Steady straight flight is the trajectory chosen here. In steady state flight, $u = u_0$, $w = w_0$, $\theta = \theta_0$, and $\psi = \psi_0$ where $(w_0, u_0, \theta_0, \psi_0)$ are constant. The drone will move in a straight line without accelerating, but the yaw angle and pitch angle do not have to be 0.

When the equations of motion are linearized, they will naturally separate into equations for lateral movement and equations for longitudinal movement. The lateral variable are (v, p, r, ϕ, ψ, y) with inputs $(\delta aileron, \delta rudder)$. The longitudinal variable are (u, w, q, θ, x, z) with inputs $(Thrust, \delta elevator)$.

The linearized longitudinal equation of motion is Equation 23.

$$\begin{bmatrix} \Delta\dot{u} \\ \Delta\dot{w} \\ \Delta\dot{q} \\ \Delta\dot{\theta} \\ \Delta\dot{x} \\ \Delta\dot{z} \end{bmatrix} = \begin{bmatrix} A_{uu} & A_{uw} & A_{uq} & A_{u\theta} & 0 & 0 \\ A_{wu} & A_{ww} & A_{wq} & A_{w\theta} & 0 & 0 \\ A_{qu} & A_{qw} & A_{qq} & A_{q\theta} & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ A_{xu} & A_{xw} & 0 & A_{x\theta} & 0 & 0 \\ A_{zu} & A_{zw} & 0 & A_{z\theta} & 0 & 0 \end{bmatrix} \begin{bmatrix} \Delta u \\ \Delta w \\ \Delta q \\ \Delta\theta \\ \Delta x \\ \Delta z \end{bmatrix}$$
$$\begin{bmatrix} \frac{1}{M} & B_{u\delta elevator} \\ 0 & B_{u\delta elevator} \\ 0 & B_{u\delta elevator} \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} \Delta Thrust \\ \Delta\delta elevator \end{bmatrix} \quad (23)$$

The linearized lateral equation of motion is Equation 24.

$$\begin{bmatrix} \Delta\dot{v} \\ \Delta\dot{p} \\ \Delta\dot{r} \\ \Delta\dot{\phi} \\ \Delta\dot{\psi} \\ \Delta\dot{y} \end{bmatrix} = \begin{bmatrix} A_{vv} & A_{vp} & A_{vr} & A_{v\phi} & 0 & 0 \\ A_{pv} & A_{pp} & A_{pr} & 0 & 0 & 0 \\ A_{rv} & A_{rp} & A_{rr} & 0 & 0 & 0 \\ 0 & 1 & A_{\phi r} & A_{\phi\phi} & 0 & 0 \\ 0 & 0 & A_{\psi r} & A_{\psi\phi} & 0 & 0 \\ A_{yv} & 0 & 0 & 0 & A_{t\psi} & 0 \end{bmatrix} \begin{bmatrix} \Delta v \\ \Delta p \\ \Delta r \\ \Delta\phi \\ \Delta\psi \\ \Delta y \end{bmatrix}$$
$$\begin{bmatrix} B_{v\delta aileron} & B_{v\delta rudder} \\ B_{p\delta aileron} & B_{p\delta rudder} \\ B_{r\delta aileron} & B_{r\delta rudder} \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} \Delta\delta aileron \\ \Delta\delta rudder \end{bmatrix} \quad (24)$$

The elements of the matrices for the longitudinal equations can be found in the appendix as an example. The rest of the elements can be found in Chapter 5 of *Small Unmanned Aircraft* [4].

V. Simulation

The simulation results can be seen in the included video. First, the equilibrium trajectory is shown. Next, the system response to longitudinal disturbances when θ is small is shown. Then, the system response to longitudinal disturbances when θ is large is shown. Finally, the system response to lateral disturbances is shown.

VI. Control Metrics

If x and z are removed from the linearized model and the model is linearized about a straight flight trajectory where θ is relatively small, the system has 2 pairs of complex eigenvalues that are stable, as seen in Figure 2 (If x and z are included, it only adds two eigenvalues that are 0). The first pair has a real part of -3.92, so the oscillations induced by these modes are relatively small. The second pair of eigenvalues has a real part of -0.05, so the oscillations induced by these modes are relatively large. These mean that u, w, q, and θ all oscillate and approach the equilibrium value after a disturbance, as seen in Figure 3

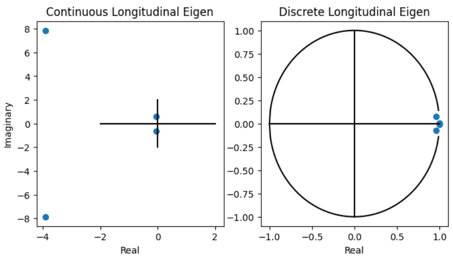

Fig. 2. The longitudinal eigenvalues when θ is small.

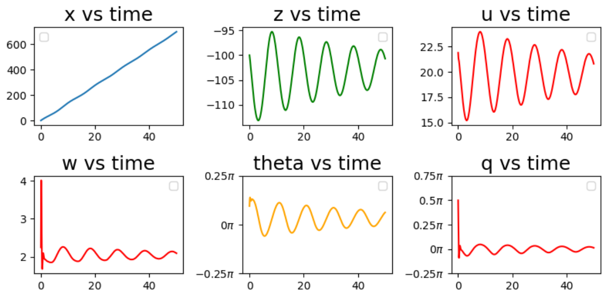

Fig. 3. The response to a longitudinal disturbance when θ is small.

When θ is large, one pair of eigenvalues for the longitudinal dynamics has a positive real part, meaning the pair is unstable, as seen in Figure 4. This means that the u, w, q, and θ will not return to the equilibrium trajectory, as shown in Figure 5

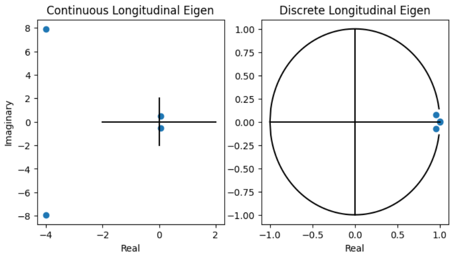

Fig. 4. The longitudinal eigenvalues when θ is large.

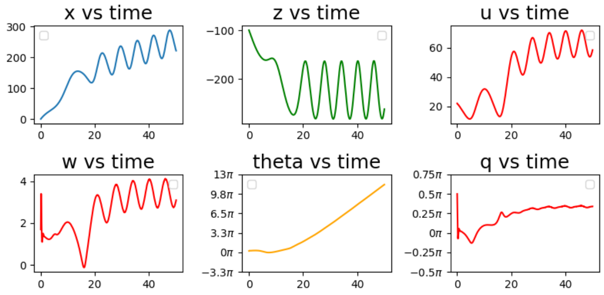

Fig. 5. The response to a longitudinal disturbance when θ is large.

If psi and y are removed from the lateral linearized model and the model is linearized about a straight flight trajectory, the system has a pair of complex eigenvalues that are stable, an additional eigenvalue that is stable, and a final eigenvalue that is unstable, as seen in Figure 6. When the system experiences lateral disturbances, it appears that v, p, and r die down to the equilibrium initially, but ϕ moves away from the equilibrium trajectory, as seen in Figure 7.

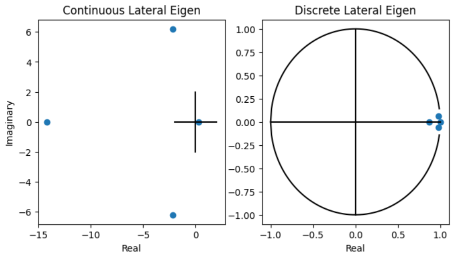

Fig. 6. The lateral eigenvalues.

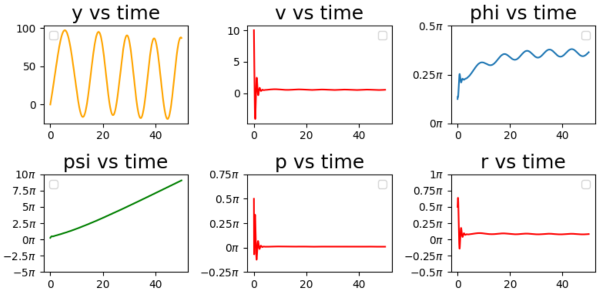

Fig. 7. The response to a lateral disturbance.

VII. Control Input Limits

The limits of the thrust and control surfaces must be determined so that the simulation does not allow the drone to do infeasible maneuvers. It is assumed that the control surfaces can operate between ±22.5 degrees, or $\pm\pi/8$ radians. Beyond these values, the equations of motion and the equations of aerodynamic forces may not hold true anymore, and the physical aircraft may not be able to handle anything above this range either. It is also assumed that a servo motor is used to change the angle of the control surfaces. Servos can typically sweep 60 degrees in 0.15 to 0.2 seconds, which is true for this 20kg-cm servo [5]. If the slower value is assumed, then the speed in which the servo can change the control surfaces is found in Equation 25.

$$\frac{0.2second}{60^\circ}, \frac{60^\circ}{0.2second} * \frac{\pi}{180} = 5.2360rad/sec \quad (25)$$

The thrust of the drone will be limited by the powerplant. The Aerosonde uses a 1.2 kw[6] four-stroke engine[4]. However, Randal W. Beard and Timothy W. McLain proposed a suitable electric motor[7] and propeller[8] pair to power the Aerosonde instead [4]. The characteristics of this electric motor can be modeled, and the voltage input to the motor and the motor speed become the limiting factor on the thrust that can be produced.

A standard model of how thrust and torque produced by a propeller varies with motor speed can be seen in Equation 26 and 27, where $thrust_{prop}$ and τ_{prop} are the thrust and torque generated by the propeller, ρ is air density, d is propeller diameter, ω is the angular velocity of the propeller, V is the magnitude of the velocity vector, and $C_{T0}, C_{T1}, C_{T2}, C_{\tau 0}, C_{\tau 1}, C_{\tau 2}$ are thrust and torque coefficients.

$$thrust_{prop} = \frac{\rho d^4 C_{T0}}{4\pi^2}\omega^2 + \frac{\rho d^3 C_{T1} V}{2\pi}\omega + \rho d^2 C_{T2} V^2 \tag{26}$$

$$\tau_{prop} = \frac{\rho d^5 C_{\tau 0}}{4\pi^2}\omega^2 + \frac{\rho d^4 C_{\tau 1} V}{2\pi}\omega + \rho d^3 C_{\tau 2} V^2 \tag{27}$$

For a DC motor, the steady state torque is described by Equation 28, where τ_{motor} is the torque generated by the motor, K_τ is the motor torque constant, $Volt_{in}$ is the input voltage, R is the resistance of the motor windings, K_V is the back-emf voltage constant, i_0 is the no-load current, and ω is the motor speed.

$$\tau_{motor} = K_\tau\left(\frac{1}{R}(Volt_{in} - K_V\omega) - i_0\right) \tag{28}$$

When the motor is driving the propeller, the speed of the propeller and the speed of the motor must be the same, and the torque of the motor (τ_{motor}) and the torque of the propeller (τ_{prop}) must be the same. Setting Equation 28 and 27 equal to each other, the result is Equation 29.

$$\begin{aligned} 0 = &\tfrac{\rho d^5 C_{\tau 0}}{4\pi^2}\omega^2 + (\tfrac{\rho d^4 C_{\tau 1} V}{2\pi} + \tfrac{K_\tau K_V}{R})\omega + \\ &\rho d^3 C_{\tau 2} V^2 - \tfrac{K_\tau Volt_{in}}{R} + K_\tau i_0 \end{aligned} \tag{29}$$

Note that Equation 26 can also be rearranged into a similar form, as shown in Equation 30.

$$0 = \frac{\rho d^4 C_{T0}}{4\pi^2}\omega^2 + \frac{\rho d^3 C_{T1} V}{2\pi}\omega + \rho d^2 C_{T2} V^2 - thrust_{prop} \tag{30}$$

Both Equation 29 and 30 can be used to solve for ω by finding the positive root of the quadratic equation.

Additionally, Equation 29 can be rearranged to solve for $Volt_{in}$, as shown in Equation 31.

$$\begin{aligned} Volt_{in} = &(\tfrac{\rho d^5 C_{\tau 0}}{4\pi^2}\omega^2 + (\tfrac{\rho d^4 C_{\tau 1} V}{2\pi} + \tfrac{K_\tau K_V}{R})\omega + \\ &\rho d^3 C_{\tau 2} V^2 + K_\tau i_0)\tfrac{R}{K_\tau} \end{aligned} \tag{31}$$

Thus, given V and $Volt_{in}$, Equation 29 can be used to find ω and then Equation 26 can be used to find $thrust_{prop}$. Given V and $thrust_{prop}$, Equation 30 can be used to find ω and then Equation 31 can be used to find $Volt_{in}$. In this manner, it is possible to convert between these three parameters.

It is unintuitive to define a maximum thrust or a max change in thrust. However, if it is assumed that the motor is using a 12 cell LiPo battery, which has a nominal voltage of 3.7 volts per cell, then the maximum $Volt_{in}$ value is 44.4 volts. To determine the maximum change in thrust, the maximum torque of the electric motor is determined using Equation 32, where i_{max} is the maximum current of the motor and KV_{rating} is the rate at which the motor will spin given a voltage (units are in $\frac{radpersecond}{Volts}$.

$$\tau_{motor} = \frac{i_{max}}{KV_{rating}} \tag{32}$$

Using the motor datasheet [7], the maximum efficiency current the motor can operate at is 63 A, and the KV_{rating} is 145 RPM/Volt = 15.18 rad/sec/Volt. Thus, the maximum torque the motor can constantly generate is $4.15\frac{kgm^2}{sec^2}$.

The moment of inertia of the propeller can be estimated by determining the inertia of a rotating bar. Equation 33 is the equation for the moment of inertia of a rotating bar I_{bar} with mass m_{bar} and length l_{bar}.

$$I_{bar} = \frac{1}{3} m_{bar} l_{bar}^2 \tag{33}$$

Using the datasheet for the propeller [8], the propeller weight is 3.38 oz or 0.0961 kg and the diameter is 20 in. or 0.508 m. The propeller is modeled as two rotating bars, each with half the mass and length of the full propeller. Thus, $I_{bar} = 0.00103 kgm^2$, and $I_{prop} = 2I_{bar} = 0.00207 kgm^2$.

The change in thrust is limited by the maximum acceleration of the motor and the maximum change in input voltage. It is assumed that the voltage input can change near instantaneously. The rotational acceleration of the motor is limited by Equation 34, where $a_{motormax}$ is the maximum rotational acceleration, $\tau_{motormax}$ is the maximum torque the motor can generate calculated in Equation 32, and I_{prop} is the moment of inertia of the propeller calculated above.

$$a_{motormax} = \frac{\tau_{motormax}}{I_{prop}} \tag{34}$$

Since $\tau_{motormax} = 4.15\frac{kgm^2}{sec^2}$ and $I_{prop} = 0.00207 kgm^2$, $a_{motormax} = 2004.8\frac{rad}{sec^2}$.

The cruising velocity of the Aerosonde is 20-30 m/s [6]. In the project, it is assumed that the desired cruising velocity is 20 m/s. Using this cruising velocity and the maximum voltage the proposed DC motor can handle (44.4V), Equation 29 and Equation 26 can be used to find the maximum thrust possible, which is 50 N. Using these two equations to calculate the motor angular speed and thrust for the range of $Volt_{in} = [0, ..., 44.4]$, generated thrust vs motor speed can be plotted, as shown in Figure 8. Using linear least squares, a line of best fit can be generated, giving a slope that expresses how thrust changes with respect the a change in motor speed. This slope is 0.17103389, so a change in motor speed of 2004.8 rad/sec is equivalent to a change in thrust of $2004.8 * 0.17103389 = 342.9 N/sec$. Thus, $342.9 N/sec$ is the maximum change in thrust possible.

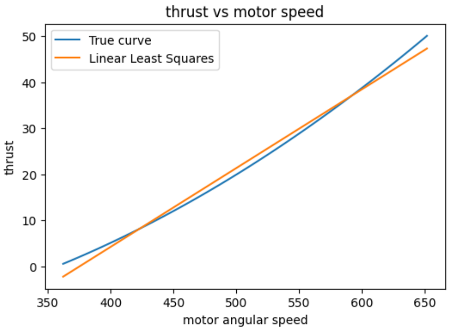

Fig. 8. The thrust vs speed curve and line of best fit.

The maximum and minimum values as well as the maximum change in value for each input is listed in Table I.

TABLE I
CONTROL INPUT CONSTRAINTS, FOR VELOCITY MAG = 20 M/S

Input	Minimum	Maximum	Max change
thrust	0.0 N	50 N	342.5 N/s
elevator	-0.3927 rad	0.3927 rad	5.2360 rad/s
aileron	-0.3927 rad	0.3927 rad.2	5.2360 rad/s
rudder	-0.3927 rad	0.3927 rad	5.2360 rad/s

VIII. LQR CONTROLLER

A. Linearize about Trajectory

Given the full state and the initial inputs of the system, the system can be linearized about that point. The system should be linearized about equilibrium points or equilibrium trajectories. One equilibrium trajectory is steady straight flight, which occurs when none of the states or inputs are changing and the aircraft is moving in a straight line. More specifically, this occurs when $u = u_{init}, v = 0, w = w_{init}, p = 0, q = 0, r = 0, \phi = 0,$ and $\theta = \theta_{init}$.

If the drone is at some initial point $(x_{init}, y_{init}, z_{init})$, and it is desired that the drone be at waypoint $(x_{des}, y_{des}, z_{des})$, then the desired heading and the angle between the xy plane and the desired waypoint can be calculated, as seen in Figure 9. $x_{des} - x_{init}$ and $y_{des} - y_{init}$ are the sides of a right triangle with hypotenuse $\sqrt{(x_{des} - x_{init})^2 + (y_{des} - y_{init})^2}$. The desired heading is $\psi_{des} = atan2((y_{des} - y_{init}), (x_{des} - x_{init}))$. Now the angle between the xy plane and the desired waypoint can be calculated using $atan2(-(z_{des} - z_{init}), \sqrt{(x_{des} - x_{init})^2 + (y_{des} - y_{init})^2})$.

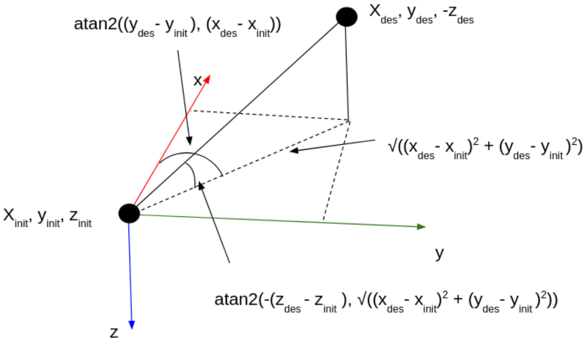
atan2((y_des - y_init), (x_des - x_init))
X_des, y_des, -Z_des
x
√((x_des - x_init)^2 + (y_des - y_init)^2)
X_init, y_init, z_init
y
atan2(-(z_des - z_init), √((x_des - x_init)^2 + (y_des - y_init)^2))
z

Fig. 9. Calculating the angle between the $x - y$ plane and the desired waypoint.

Given the desired angle from the xy plane to the waypoint and the cruising speed of the drone (which was set at 20 m/s), the $u_{des}, w_{des}, \theta_{des}, Thrust_{des}$, and $elevator_{des}$ can be calculated. v, p, q, r, and ϕ are set to 0. β is set to 0 so $\dot{v}$ is 0 (from Equation 6). $aileron$ and $rudder$ are set to 0 so that $\dot{p}$ and $\dot{r}$ will be 0 (from Equation 7). If m is 0, then $\dot{q}$ will be 0 (from Equation 7), and thus $\dot{\phi}$, $\dot{\theta}$, and $\dot{\psi}$ will be 0 (from Equation 5). Thus, it is desired that $m = 0$, $\dot{u} = 0$, $\dot{w} = 0$, $V = 20m/s = \sqrt{u_{des}^2 + w_{des}^2}$ (from Equation 1), and xy plane to waypoint angle $= \theta_{des} - \alpha = \theta_{des} - arctan(w_{des}/u_{des})$ (from Equation 2). These five equations are functions of $u_{des}, w_{des}, \theta_{des}, Thrust_{des}$, and $elevator_{des}$. Thus, this is a system of equations that can be solved using a function like fsolve in MATLAB or Python.

B. LQR Control Loop

It was shown before that if the drone equations of motion are linearized, the A matrices for both the longitudinal and lateral dynamics can have unstable poles, meaning that some of the states will not converge to 0. To fix this, inputs must be calculated to stabilize the A matrix. Let $u = Kx$, then $\dot{x} = Ax + Bu = Ax + BKx = (A + BK)x)$. Thus, $(A + BK)$ becomes the A matrix for the closed loop system, and it is desired to design the K matrix such that $(A + BK)$ has stable poles. This method is called pole placement.

However, sometimes it is difficult to determine where the poles of the system should be placed. Linear Quadratic Regulator (LQR) is a method of pole placement that minimizes the cost function in Equation 35, where Q is a symmetric matrix the size of the state that places a cost on the state and R is a symmetric matrix the size of the input that places a cost on the input. Typically, Q and R are diagonal matrices where the diagonal element that matches up with a specific state or input is the cost on that state or input, or how much it is desired that the state or input is minimized. For LQR to work properly, Q $\succeq$ 0 and R $\succ$ 0.

$$J = \frac{1}{2}\int_0^\infty x^T Q x + u^T R u \tag{35}$$

Using the solution to the algebraic Riccati equation, a gain matrix K can be calculated that minimizes the cost function J. This can be done using the lqr function in MATLAB or the lqr function in the Python Control Systems Library. Note that since the aircraft dynamics can be separated into the longitudinal and lateral dynamics, the longitudinal and lateral dynamics will each have their own Q, R, and K matrices, so lqr will be run twice.

The control loop that is executed at every timestep is shown below.

1) Determine if the distance between the drone and the current waypoint is less than some threshold. In these experiments, the threshold was set to 5 meters. If the drone is close enough to the waypoint, remove the waypoint from the list and move on to the next waypoint.
2) If this is a new waypoint, linearize the system about its current state. Using the A and B matrices from this linearization as well as Q and R matrices for both the lateral and longitudinal dynamics, run the lqr function to determine the gain matrices K_{long} and K_{lat}.
3) The current waypoint determines $(x_{des}, y_{des}, z_{des})$. Calculate ψ_{des} and the angle between the xy plane and the waypoint, then determine u_{des}, w_{des}, and θ_{des} using fsolve. These values must be recalculated at every timestep because if there are any disturbances that throw the drone off the steady straight path, a new steady straight path must be calculated.
4) The gain matrices will try to drive the system to the origin. Thus, the current state vector must be shifted by the current desired states. The shifted state vector is $X_{shifted} = X_{current} - X_{des}$.
5) It is possible that $\psi_{current}$ or ψ_{des} is greater than $\pm\pi$. Thus, the angles should be wrapped between $\pm\pi$. $\psi_{shifted}$ must also be wrapped between $\pm\pi$ by changing the direction of rotation.
6) The desired inputs can be calculated using the gain matrix and the shifted states, as shown in Equations 36 and 37.
7) The inputs must be clipped so that they do not exceed the limits calculated earlier.
8) The clipped inputs are then applied to the system, and the nonlinear dynamics are simulated.

$$\begin{bmatrix} Thrust \\ elevator \end{bmatrix} = K_{long} \begin{bmatrix} u_{shifted} \\ w_{shifted} \\ q_{shifted} \\ \theta_{shifted} \end{bmatrix} \tag{36}$$

$$\begin{bmatrix} aileron \\ rudder \end{bmatrix} = K_{lat} \begin{bmatrix} v_{shifted} \\ p_{shifted} \\ r_{shifted} \\ \phi_{shifted} \\ \psi_{shifted} \end{bmatrix} \tag{37}$$

C. Tuning

The Q and R matrices must be tuned to achieve good performance. The Q and R matrices were set to be diagonal. The weights for each of the states were scaled by the maximum value that state is expected to reach squared. The same scaling was done for the inputs. Through testing, it was determined that very small weights should be placed on u, w, p, q, and r. A small weight was placed on ϕ. Reducing the weight on ϕ allowed the drone to change heading quickly, but also lead to more cases where ϕ increased near $\pm\pi/2$, causing the drone to lose altitude and crash. Large weights were placed on θ, v, and ψ. The system performed best when it was able to rapidly change θ and ψ to the desired position, and then slowly adjust all other states. The weights on the inputs were kept low, since the priority was for the system to reach all waypoints.

IX. LQR Results

A. Following Waypoints

It can be confirmed that the gain matrices K_{long} and K_{lat} cause the closed loop system to have stable poles. Figure 10 shows the poles of the open and closed loop A matrices for the longitudinal and lateral dynamics. The open loop longitudinal dynamics have two poles that are close to or in the right half plane depending on theta, and the open loop lateral dynamics has two poles in the right half plane, but the closed loop dynamics has all poles in the left half plane.

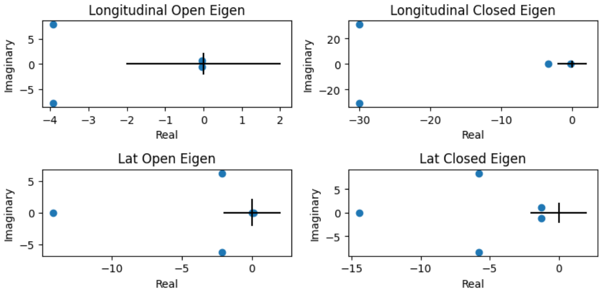

Fig. 10. Graph of the open and closed loop poles.

The LQR controller can be used to drive the drone through a series of waypoints. Two different paths were tested. Waypoints forming the shape of a helix are shown in Table II. The simulated trajectory is shown in Figure 11.

TABLE II
WAYPOINTS FOR HELIX

x	y	z
100	-28	-100
172	-100	-120
200	-200	-140
172	-300	-160
100	-372	-180
0	-400	-200
-100	-372	-220
-172	-300	-240
-200	-200	-260
-172	-100	-280
-100	-28	-300
0	0	-320

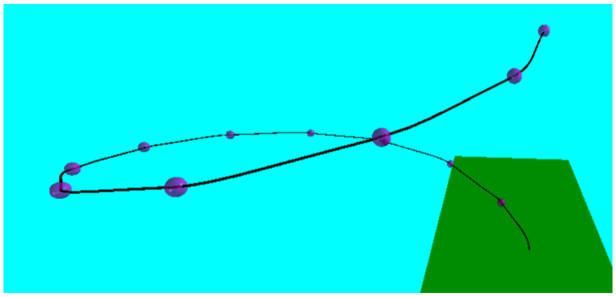

Fig. 11. The complete trajectory for the helix waypoints.

A wave pattern was also tested. The waypoints for the wave pattern can be found in Table III. The simulated trajectory is shown in Figure 12. The longitudinal state, lateral state, and input plots for the wave trajectory can be seen in Figures 13, 14, and 15.

TABLE III
WAYPOINTS FOR WAVE

x	y	z
100	100	-100
175	150	-120
250	150	-140
325	100	-110
400	50	-90
475	50	-120
550	100	-150

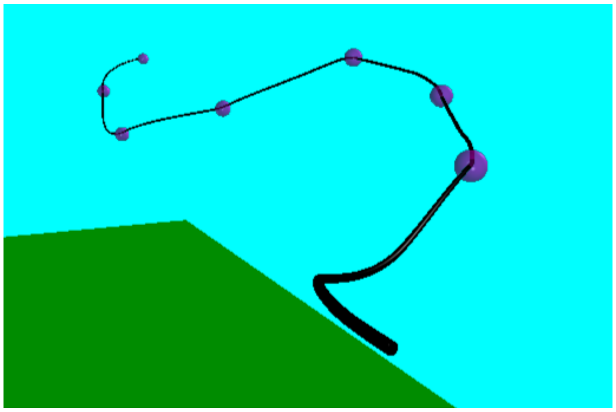

Fig. 12. The complete trajectory for the wave waypoints.

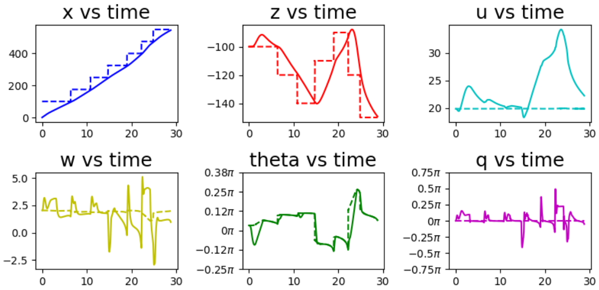

Fig. 13. A plot of the longitudinal states for the wave trajectory.

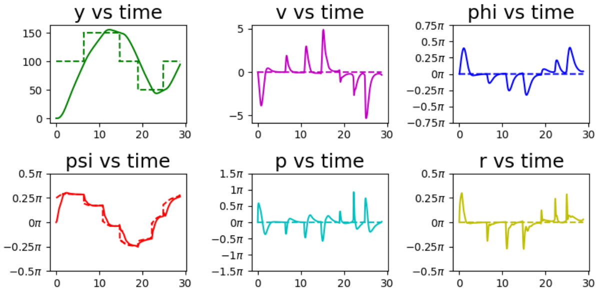

Fig. 14. A plot of the lateral states for the wave trajectory.

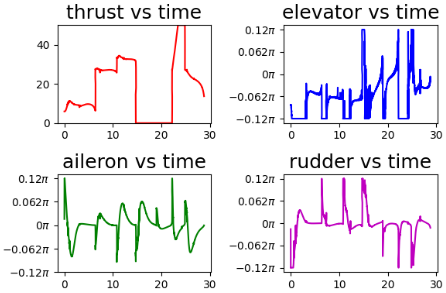

Fig. 15. A plot of the inputs for the wave trajectory.

B. Adding Disturbances

Every time the drone switches from going to one waypoint to another, this can be thought of as a disturbance on the system. The system is suddenly no longer located at the origin of the shifted coordinate system, so the gain matrices must be used to drive the system back to the origin. Other forms of disturbances can also be introduced, such as wind. Wind will be modeled simply as an additional change in the translation kinematics ($\dot{x}$, $\dot{y}$, and $\dot{z}$). A probability that a wind gust in the x, y, or z direction will start in a given timestep is set. If a wind gust occurred in the last timestep in this particular direction, it is more likely that it will keep going. Then the magnitude of the wind gust is also probabilistic.

Figure 16 and 17 show the drone reacting to a gust of wind. Figure 18 shows the wind magnitudes as the change in x, y, or z velocity.

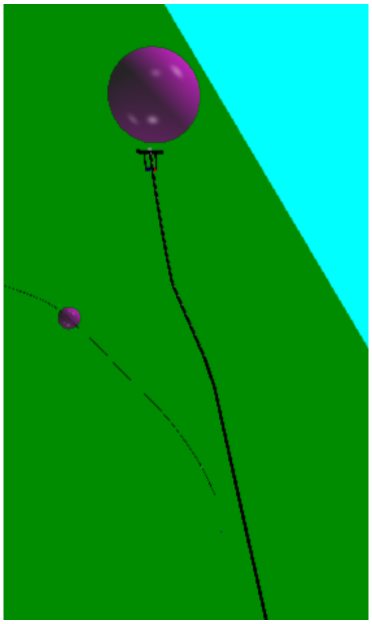

Fig. 16. A gust of wind pushes the drone off of the steady state path, but it is able to correct.

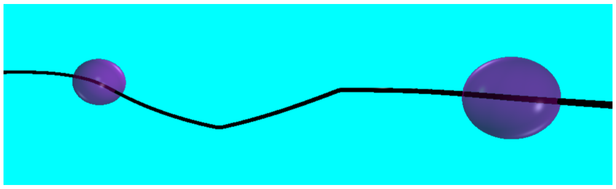

Fig. 17. A gust of wind forced the drone to quickly change the trajectory to the next waypoint.

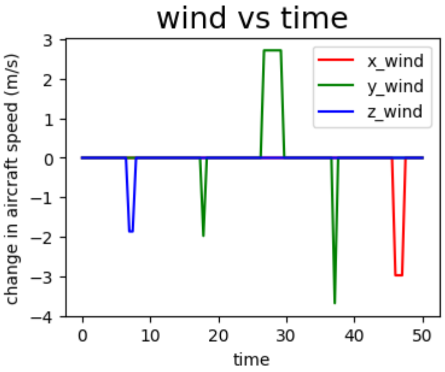

Fig. 18. A plot of the wind magnitudes as the change in x, y, or z velocity.

C. Limitations of LQR

It has been shown that LQR is an effective control algorithm for simple trajectories. However, poor results were obtained when the waypoints were placed in such a way as to require an aggressive trajectory with quick turns and steep ascents. Increasing the magnitude of the wind disturbances also lead to poor results. The drone would fail to reach all waypoints.

The biggest drawback of LQR is that there is no limit on the control inputs that the gain matrices will generate. This means that in certain situations, the LQR controller will output very large or very small control inputs that will be clipped to realistic values. The LQR controller will calculate the most optimal control inputs that may be invalid instead of calculating less optimal control inputs that are valid. Thus, the controller will fail when there are sharp turns or steep ascents because it is calculating control inputs that are not feasible.

Model Predictive Control (MPC) is a control method that can take into account the input constraints. Thus, this style of controller would most likely be able to handle more aggressive trajectories. MPC can also place contraints on the states, which would be useful. When using LQR, a larger weight must be placed on the ϕ state so that ϕ does not increase more than $\pm\pi/4$, which would cause the aircraft to spiral away from its current heading and is not desirable. However, it is not as important to drive ϕ to 0, and allowing ϕ to change quickly would allow the aircraft to turn quicker. MPC may help with this by allowing a small weight to be placed on driving ϕ to 0 while a constraint limiting ϕ to $\pm\pi/4$ will prevent the undesirable range of this state.

References

[1] M. Niculescu, "Lateral track control law for aerosonde uav," in *39th Aerospace Sciences Meeting and Exhibit*, 2001, p. 16.

[2] "Insitue aerosonde laima," in *The Muesume of Flight*, 2023.

[3] M. T. Burston, R. Sabatini, R. Clothier, A. Gardi, and S. Ramasamy, “Reverse engineering of a fixed wing unmanned aircraft 6-dof model for navigation and guidance applications,” *Applied Mechanics and Materials*, vol. 629, pp. 164–169, 2014.

[4] R. W. Beard and T. W. McLain, *Small unmanned aircraft: Theory and practice*. Princeton university press, 2012.

[5] Hiwonder LD 220MG Full Metal Gear Digital Servo, https://www.robotshop.com/products/hiwonder-ld-220mg-full-metal-gear-digital-servo-w-metal-servo-horn-bracket, 2023.

[6] A. Sarhan and M. Ashry, “Self-tuned pid controller for the aerosonde uav autopilot,” *International Journal of Engineering Research & Technology (IJERT)*, vol. 2, no. 12, pp. 2278–0181, 2013.

[7] AXI Motors, https://www.modelmotors.cz/product/detail/516/, 2023.

[8] APC Propellers. https://www.apcprop.com/product/20x10e/, 2023.

X. Appendices

The values for the elements of the A and B matrix for the linearized longitudinal equations.

$$\begin{aligned} A_{uu} = &-\frac{u_0\rho S}{M}(C_{D0} + C_{D\delta elevator}\delta elevator_0 \\ &+ C_{D\alpha}\alpha_0) + \frac{\rho S w_0 C_{D\alpha}}{2M} - \frac{\rho S c C_{Dq} q_0 u_0}{2mV_0} \end{aligned} \tag{38}$$

$$\begin{aligned} A_{uw} = &-\frac{w_0\rho S}{M}(C_{D0} + C_{D\delta elevator}\delta elevator_0 \\ &+ C_{D\alpha}\alpha_0) - \frac{\rho S u_0 C_{D\alpha}}{2M} - \frac{\rho S c C_{Dq} q_0 w_0}{2mV_0} - q_0 \end{aligned} \tag{39}$$

$$A_{uq} = -w_0 - \frac{\rho V_0 c S C_{Dq}}{4M} \tag{40}$$

$$A_{u\theta} = -g\cos\theta_0 \tag{41}$$

$$B_{u\delta elevator} = -\frac{\rho V_0^2 S C_{D\delta elevator}}{2M} \tag{42}$$

$$\begin{aligned} A_{wu} = &-\frac{u_0\rho S}{M}(C_{L0} + C_{L\delta elevator}\delta elevator_0 \\ &+ C_{L\alpha}\alpha_0) + \frac{\rho S w_0 C_{L\alpha}}{2M} - \frac{\rho S c C_{Lq} q_0 u_0}{2mV_0} + q_0 \end{aligned} \tag{43}$$

$$\begin{aligned} A_{ww} = &-\frac{w_0\rho S}{M}(C_{L0} + C_{L\delta elevator}\delta elevator_0 \\ &+ C_{L\alpha}\alpha_0) - \frac{\rho S u_0 C_{L\alpha}}{2M} - \frac{\rho S c C_{Lq} q_0 w_0}{2mV_0} \end{aligned} \tag{44}$$

$$A_{wq} = u_0 - \frac{\rho V_0 c S C_{Lq}}{4M} \tag{45}$$

$$A_{w\theta} = -g\sin\theta_0 \tag{46}$$

$$B_{w\delta elevator} = -\frac{\rho V_0^2 S C_{L\delta elevator}}{2M} \tag{47}$$

$$\begin{aligned} A_{qu} = &\frac{u_0\rho S c}{I_{yy}}(C_{m0} + C_{m\delta elevator}\delta elevator_0 \\ &+ C_{m\alpha}\alpha_0) - \frac{\rho S c w_0 C_{m\alpha}}{2I_{yy}} - \frac{\rho S c^2 C_{mq} q_0 u_0}{2I_{yy}V_0} \end{aligned} \tag{48}$$

$$\begin{aligned} A_{qw} = &\frac{w_0\rho S c}{I_{yy}}(C_{m0} + C_{m\delta elevator}\delta elevator_0 \\ &+ C_{m\alpha}\alpha_0) - \frac{\rho S c u_0 C_{m\alpha}}{2I_{yy}} - \frac{\rho S c^2 C_{mq} q_0 w_0}{2I_{yy}V_0} \end{aligned} \tag{49}$$

$$A_{qq} = \frac{\rho V_0 c^2 S C_{mq}}{4I_{yy}} \tag{50}$$

$$B_{q\delta elevator} = -\frac{\rho V_0^2 S c C_{m\delta elevator}}{2I_{yy}} \tag{51}$$

$$A_{xu} = \cos\theta_0 \tag{52}$$

$$A_{xw} = \sin\theta_0 \tag{53}$$

$$A_{x\theta} = w_0 \cos\theta_0 - u_0 \sin(\theta_0) \tag{54}$$

$$A_{zu} = -\sin\theta_0 \tag{55}$$

$$A_{zw} = \cos\theta_0 \tag{56}$$

$$A_{z\theta} = -u_0 \cos\theta_0 - w_0 \sin(\theta_0) \tag{57}$$