Aircraft Control System Using LQG and LQR controller with Optimal Estimation-Kalman Filter Design

Vijay Dev Reddy Chevireddi
119491485

Naga Venkata Siva Sai Gopal Kambhmapati
119484128

***Abstract*—The paper discusses a robust controller for an aircraft control system that uses LQG and LQR techniques to ensure stability and good dynamic performance even when aircraft parameters vary. The controller is implemented using MATLAB/Simulink program and simulation results confirm its effectiveness. Additionally, the paper explains how Kalman filters can be used to combine multiple measurements to produce more accurate estimates of unknown variables in autonomous or assisted navigation systems.**

The proposed controller is designed to maintain stability and good dynamic performance of an aircraft control system even when there are variations in aircraft parameters. The controller uses LQG and LQR techniques and is implemented using MATLAB/Simulink program. Simulation results confirm the effectiveness of the proposed controller. The paper also discusses how Kalman filters can be used to combine multiple measurements to produce more accurate estimates of unknown variables in autonomous or assisted navigation systems.

Keywords*—*Airplane Dynamics, LQR, LQG, Kalman Filtering

I. INTRODUCTION

Feedback control systems play a crucial role in various domains such as manufacturing, mining, automobiles, and military hardware. These systems face the challenge of delivering improved performance and robustness under changing conditions. To achieve this, control engineers need new design tools and better theory.

One common approach to enhance control system performance is by adding extra sensors and actuators, leading to multi-input multi-output systems. Modern feedback control methodologies must handle such complexity. Linear Quadratic Gaussian (LQG) optimal control theory is a significant achievement in control engineering. It allows synthesizing controllers that optimize a specified quadratic performance index. Importantly, LQG accounts for Gaussian white noise disturbances affecting the system. In practical scenarios, minimizing quadratic cost functions aligns with desired performance objectives, even when the system faces disturbances and measurement noise modeled as stochastic white noise processes.

II. AIRCRAFT CONTROL AND MOVEMENT

There are three Primary ways for an aircraft to change its orientation relative to the passing air. Pitch (movement of the nose up or down), Roll (Rotation around the longitudinal axis, that is, the axis which runs along the length of the aircraft) and Yaw (movement of the nose to left or right.) Turning the aircraft (change of heading) requires the aircraft firstly to roll to achieve an angle of bank; when the desired change of heading has been accomplished the aircraft must again be rolled in the opposite direction to reduce the angle of bank to zero.

III. AIRCRAFT LONGITUDINAL DYNAMICS

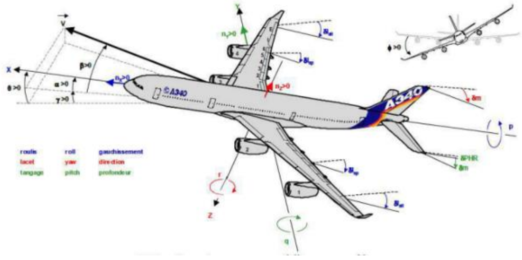

Figure 1 Aerodynamic Reference

1) *Equation of Movementnts:*

The Rigid body equation of motion is obtained from Newton's Second law:

$$\sum F = \frac{d}{dt}(mv)$$

$$\sum M = \frac{d}{dt}H$$

The Vector equations can be written in scalar form and consist of three forces equations and three moment equations.

$$F_x = \frac{d}{dt}(mu)$$

$$F_y = \frac{d}{dt}(mv)$$

$$F_z = \frac{d}{dt}(mw)$$

The Moment equation can also be written in the above manner:

$$L = \frac{d}{dt}H_x$$

$$M = \frac{d}{dt}H_y$$

$$N = \frac{d}{dt}H_z$$

If we let δm be an element of mass of the airplane, v be the velocity of the elemental mass relative inertial frame, and δFbe the resulting force acting on the elemental mass:

$$\delta F = \delta m \frac{dv}{dt}$$

Now, we take total external forces acting on the airplane is:

$$\sum \delta F = F$$

The Velocity of the differential mass δm is:

$$v = v_c + \frac{dr}{dt}$$

Now Substituting the equation, we get:

$$\sum \delta F = F = \frac{d}{dt}\sum\left(v_c + \frac{dr}{dt}\right)\delta m$$

Assume that the mass of vehicle is constant:

$$F = m\frac{dv_c}{dt} + \frac{d}{dt}\sum \frac{dr}{dt}\ \delta m$$

Where in the above equation $\frac{dr}{dt} = r$:

$$F = m\frac{dv_c}{dt} + \frac{d}{dt}\sum r\delta m$$

Where $\sum r\delta m = o$ because r is measured from the center of mass

$$F = m\frac{dv_c}{dt}$$

We going to get the equation of moment in the similar manner which can be written as:

$$\delta_m = \frac{d}{dt}\ \delta H = \frac{d}{dt}(r \times v)\delta m$$

The velocity of the mass element can be expressed in terms of the velocity of the center of mass and the relative velocity of the mass element to the center of mass:

$$V = v_c + \frac{dr}{dt} = v_c + \omega \times r$$

The Total moment of momentum can be written as:

$$H = \sum \delta H = \sum (r \times v_c)\delta m + \sum [r(\omega \times r)]\delta m$$

The velocity component v_c is constant can be written as:

$$H = \sum r \times \delta m \times v_c + \sum [r \times (\omega \times r)]\delta m$$

Where

$$\sum r \times \delta m = 0$$

we can express the angular velocity and position vectors as:

$$\omega = pi + qi + rk$$
$$r = xi + yj + zk$$

Where H can be written as:

$$H = (pi + qj + rk)\sum (x^2 + y^2 + z^2)\delta m - \sum (xi + yj + zk)(px + qy + rz)\delta m$$

The Scalar components of the H :

$$H_x = p\sum (y^2 + z^2)\delta m - q\sum xy\ \delta m - r\sum x_z \delta m$$

$$H_y = -p\sum xy\delta m + q\sum (x^2 + z^2)\delta m - r\sum yz\delta m$$

$$H_z = -p\sum xz\delta m - q\sum yz\delta m + r\sum (x^2 + y^2)\delta m$$

The summation in these equations are the mass moment and products of inertia of the airplane and are defined as follows:

$$I_x = \iiint (y^2 + z^2)\delta m\ \ ; I_{xy} = \iiint xy\delta m$$

$$I_y = \iiint (x^2 + z^2)\delta m\ ;$$
$$I_{xz} = \iiint xz\delta m$$
$$I_z = \iiint (x^2 + y^2)\delta m;$$
$$I_{yz} = \iiint yz\delta m$$

The Scalar equation for the moment of momentum follows:

$$H_x = pI_x - qI_{xy} - rI_{xz}$$
$$H_y = -pI_{xy} + qI_y - rI_{yz}$$
$$H_z = -pI_{xz} - qI_{yz} + rI_z$$

If the reference frame is not rotating, then as the airplane rotates the moments and products of inertia will vary with time. Now we must determine the derivates of the vectors v and H referred to the rotating frame of reference.

$$\frac{dA}{dt}|_I = \frac{dA}{dt}|_B + \omega \times\ A$$

Where the subscripts I and B refer to the inertial and body fixed frame of reference.

$$F = m\frac{dv_c}{dt}|_B + m(\omega \times v_c)$$

$$M = \frac{dH}{dt}|_B + \omega \times H$$

The Scalar equations are:

$$F_x = m\left(\frac{du}{dt} + qw - rv\right)\ ; F_y = m\left(\frac{dv}{dt} + ru - pw\right)\ ; F_z = m\left(\frac{dw}{dt} + pv - qu\right)$$

$$L = \frac{dH_x}{dt} + qH_z - rH_y\ ;\ M = \frac{dH_y}{dt} + rH_x - pH_z\ ; N = \frac{dH_z}{dt} + pH_y - qH_x$$

The Components of the force and moment acting on the airplane are composed of aerodynamics, gravitational, and propulsive contribution.

$$L = I_x\frac{dp}{dt} - I_{xz}\frac{dr}{dt} + qr(I_z - I_y) - I_{xy}pq$$

$$M = I_y\frac{dq}{dt} + rp(I_x - I_z) + I_{xz}(p^2 - r^2)$$

$$N = -I_{xz}\frac{dp}{dt} + I_z\frac{dr}{dt} + pq(I_y - I_x) + I_{xz}qr$$

Orientation and Position of the airplane can be determined about $0z_f$ through yaw angle to the frame; frame about $0y_1$through the pitch angle bringing the frame, about the $0x_2$ through the roll angle we can show that:

$$\frac{dx}{dt} = u_1\cos\psi - v_1\sin\psi$$
$$\frac{dy}{dt} = u_1\sin\psi + v_1\cos\psi$$
$$\frac{dz}{dt} = w_1$$

Where u_1, v_1,and w_1 can be expressed in terms of u_2, v_2, w_2

$$u_1 = u_2C_\theta + w_2S_\theta\ ; v_1 = v_2; w_1 = -u_2S_\theta + w_2C_\theta$$
$$u_2 = u\ \ ; v_2 = vC_\phi - wS_\phi;\ w_2 = vS_\phi + wC_\phi$$

We can determine the absolute velocity in terms of the Euler angles and velocity components in the body frame:

$$\begin{bmatrix}\frac{\mathrm{d}x}{\mathrm{d}t}\\ \frac{\mathrm{d}y}{\mathrm{d}t}\\ \frac{\mathrm{d}z}{\mathrm{d}t}\end{bmatrix} = \begin{bmatrix} C_\theta C_\psi & S_\Phi S_\theta C_\psi - C_\Phi S_\psi & C_\Phi S_\theta C_\psi + S_\Phi S_\psi \\ C_\theta S_\psi & S_\Phi S_\theta S_\psi + C_\Phi C_\psi & C_\Phi S_\theta S_\psi - S_\Phi C_\psi \\ -S_\theta & S_\theta C_\theta & C_\Phi C_\theta \end{bmatrix}\begin{bmatrix}u\\v\\w\end{bmatrix}$$

The relationship between the angular velocities in the body frame (p, q and r) and Euler rates ($\psi^{\cdot}, \theta^{\cdot}$ and $\phi^{\cdot}$) also can be determined by:

$$\begin{bmatrix}p\\q\\r\end{bmatrix} = \begin{bmatrix}1 & 0 & -S_\theta \\ 0 & C_\Phi & C_\theta S_\Phi \\ 0 & -S_\Phi & C_\theta C_\Phi\end{bmatrix}\begin{bmatrix}\dot{\Phi}\\ \dot{\theta}\\ \dot{\psi}\end{bmatrix}$$

The above equation can be solved for the Euler rates in terms of the body angular velocities:

$$\begin{bmatrix}\dot{\Phi}\\ \dot{\theta}\\ \dot{\psi}\end{bmatrix} = \begin{bmatrix}1 & S_\Phi \tan\theta & C_\Phi \tan\theta \\ 0 & C_\Phi & -S_\Phi \\ 0 & S_\Phi \sec\theta & C_\Phi \sec\theta\end{bmatrix}\begin{bmatrix}p\\q\\r\end{bmatrix}$$

Gravitational Forces acting on the body which can be written as:

$$(F_x)_{gravity} = -mgsin\ \theta$$
$$(F_y)_{gravity} = mgcos\theta sin\phi$$
$$(F_z)_{gravity} = mgcos\theta cos\phi$$

Small- Disturbance Theory are introduced in the equation we can get:

The summary of the Kinematics and dynamics equations:

$$X - mgSin\theta = m\left(\frac{du}{dt} + qw - rv\right)$$
$$Y + mgCos\theta Sin\phi = m\left(\frac{dv}{dt} + ru - pw\right)$$
$$Z + mgCos\theta Cos\phi = m\left(\frac{dw}{dt} + pv - qu\right)$$
$$L = I_x\frac{dp}{dt} - I_{xz}\frac{dr}{dt} + qr(I_z - I_y) - I_{xz}pq$$
$$M = I_y\frac{dq}{dt} + rq(I_x - I_y) - I_{xz}pq$$
$$N = -I_{xy}\frac{dp}{dt} + I_z\frac{dr}{dt} + pq(I_y - I_x) + I_{xz}qr$$
$$p = \dot{\Phi} - \dot{\psi}S_\theta$$
$$q = \dot{\theta}C_\Phi + \dot{\psi}C_\theta S_\Phi$$
$$r = \dot{\psi}C_\theta C_\Phi - \dot{\theta}S_\Phi$$
$$\dot{\theta} = qC_\Phi - rS_\Phi$$
$$\dot{\Phi} = p + qS_\Phi T_\theta + rC_\Phi T_\theta$$
$$\dot{\psi} = (qS_\Phi + rC_\Phi)\sec\ \theta$$

Velocity of aircraft in the fixed frame in terms of Euler angles and body velocity components:

$$\begin{bmatrix}\frac{dx}{dt}\\ \frac{dx}{dt}\\ \frac{dz}{dt}\end{bmatrix} = \begin{bmatrix} C_\theta C_\psi & S_\Phi S_\theta C_\psi - C_\Phi S_\psi & C_\psi S_\theta C_\psi + S_\Phi S_\psi \\ C_\theta S_\psi & S_\Phi S_\theta S_\psi + C_\Phi C_\psi & C_\Phi S_\theta S_\psi - S_\Phi C_\psi \\ -S_\theta & S_\Phi C_\theta & C_\Phi C_\theta \end{bmatrix}\begin{bmatrix}u\\v\\w\end{bmatrix}$$

All the variable in the equations of motion is replaced by a reference value plus a perturbation or disturbance:

$$u = u_0 + \Delta u v = v_0 + \Delta v w = w_0 + \Delta w$$
$$p = p_0 + \Delta p q = q_0 + \Delta q r = r_0 + \Delta r$$
$$X = X_0 + \Delta X Y = Y_0 + \Delta Y Z = Z_0 + \Delta Z$$
$$M = M_0 + \Delta M N = N_0 + \Delta N L = L_0 + \Delta L$$
$$\delta = \delta_0 + \Delta\delta$$

Now, we introduce the small disturbance notation into the equation of motion,

$$X - \mathrm{mgsin}\,\theta = m(\dot{u} + qw - rv)$$

Substituting the small disturbance variable into the equation:

$$X_0 + \Delta X - \mathrm{mgsin}(\theta_0 + \Delta\theta)$$
$$= m\left[\frac{d}{dt}(u_0 + \Delta u) + (q_0 + \Delta q)(w_0 + \Delta w) - (r_0 + \Delta r)(v_0 + \Delta v)\right]$$

We neglect products of the disturbance and assume that:

$$w_0 = v_0 = p_0 = q_0 = r_0 = \Phi_0 = \psi_0 = 0$$

The equation becomes:

$$X_0 + \Delta X - mg\sin(\theta_0 + \Delta\theta) = m\,\Delta\frac{du}{dt}$$

For further reduced equation trigonometric identity:

$$\sin(\theta_0 + \Delta\theta) = \sin\theta_0\cos\Delta\theta + \cos\theta_0\sin\Delta\theta = \sin\theta_0 + \Delta\theta\cos\theta_0$$
$$X_0 + \Delta X - mg(\sin\theta_0 + \Delta\theta\cos\theta_0) = m\Delta\dot{u}$$

The disturbance quantities are set equal to 0 in this equation, we have the reference flight conditions:

$$X_0 - mg\ sin\theta_0 = 0$$

The reduced X- force equation to:

$$\Delta X - mg\Delta\theta\ \cos\theta_0 = m\Delta\frac{du}{dt}$$

The force ΔX is the change in aerodynamics and propulsive forces in the x direction and can be expressed by means of a Taylor series in terms of the perturbation variables. If we assume that ΔX is a function only $u, w, \delta_e and\ \delta_T\ then\ \Delta X$ can be expressed as:

$$\Delta X = \frac{\partial X}{\partial u}\Delta u + \frac{\partial X}{\partial w}\Delta w + \frac{\partial X}{\partial \delta_e}\Delta\delta_e + \frac{\partial X}{\partial \delta_e}\Delta\delta_e + \frac{\partial X}{\partial \delta_T}\Delta\delta_T$$

If a Canard or all-moveable stabilator is used for longitudinal control; the control term would be replaced by :

$\frac{\partial X}{\partial u}\Delta\delta_H$ or $\frac{\partial X}{\partial \delta_e}\Delta\delta_c$

Substituting the expression for ΔX into the force equation yields:

$$\frac{\partial X}{\partial u}\Delta u + \frac{\partial X}{\partial w}\Delta w + \frac{\partial X}{\partial \delta_e}\Delta\delta_e + \frac{\partial X}{\partial \delta_T}\Delta\delta_T - mg\Delta\theta cos\theta_0 = m\,\Delta\frac{du}{dt};$$

on rearranging you will get:

$$\left(m\frac{\mathrm{d}}{\mathrm{d}t} - \frac{\partial X}{\partial u}\right)\Delta u - \left(\frac{\partial X}{\partial w}\right)\Delta w + (\mathrm{mgcos}\,\theta_0)\Delta\theta = \frac{\partial X}{\partial \delta_e}\Delta\delta_e + \frac{\partial X}{\partial \delta_T}\Delta\delta_T$$

The change in aerodynamics forces and moments are functions of the motion variable$\Delta u, \Delta w$, and so on. The aerodynamics derivatives usually the most important for conventional airplane motion analysis follows:

$$\Delta X = \frac{\partial X}{\partial u}\Delta u + \frac{\partial X}{\partial w}\Delta w + \frac{\partial X}{\partial \delta_e}\Delta\delta_e + \frac{\partial X}{\partial \delta_T}\Delta\delta_T$$
$$\Delta Y = \frac{\partial Y}{\partial v}\Delta v + \frac{\partial Y}{\partial p}\Delta p + \frac{\partial Y}{\partial r}\Delta r + \frac{\partial Y}{\partial \delta_r}\Delta\delta_r$$
$$\Delta Z = \frac{\partial Z}{\partial u}\Delta u + \frac{\partial Z}{\partial w}\Delta w + \frac{\partial Z}{\partial \dot{w}}\Delta\dot{w} + \frac{\partial Z}{\partial q}\Delta q + \frac{\partial Z}{\partial \delta_e}\Delta\delta_e + \frac{\partial Z}{\partial \delta_T}\Delta\delta_T$$

$$\Delta L = \frac{\partial L}{\partial v}\Delta v + \frac{\partial L}{\partial p}\Delta p + \frac{\partial L}{\partial r}\Delta r + \frac{\partial L}{\partial \delta_r}\Delta\delta_r + \frac{\partial L}{\partial \delta_a}\Delta\delta_a$$

$$\Delta M = \frac{\partial M}{\partial u}\Delta u + \frac{\partial M}{\partial w}\Delta w + \frac{\partial M}{\partial \dot{w}}\Delta\dot{w} + \frac{\partial M}{\partial q}\Delta q + \frac{\partial M}{\partial \delta_e}\Delta\delta_e + \frac{\partial M}{\partial \delta_T}\Delta\delta_T$$

$$\Delta N = \frac{\partial N}{\partial v}\Delta v + \frac{\partial N}{\partial p}\Delta p + \frac{\partial N}{\partial r}\Delta r + \frac{\partial N}{\partial \delta_r}\Delta\delta_r + \frac{\partial N}{\partial \delta_a}\Delta\delta_a$$

The Linearized small-disturbance Longitudinal and lateral rigid body equation of motion:

The longitudinal Equations:

$$\dot{u} = \frac{\mathrm{X_u}}{m}u + \frac{\mathrm{X_w}}{m}w - \frac{\mathrm{gcos}\,\Theta_0}{m}\theta + \frac{X_{\delta_e}}{m}\delta_e + \frac{X_p}{m}\delta_p$$

$$\dot{W} = \frac{Z_u}{m - Z_{\dot{w}}}u + \frac{Z_w}{m - Z_{\dot{w}}}w + \frac{Z_{q+mU_0}}{m - Z_{\dot{w}}}q - \frac{\mathrm{mgsin}\,\Theta_0}{m - Z_{\dot{w}}}\theta + \frac{Z_{\delta_e}}{m - Z_{\dot{\omega}}}\delta_e + \frac{Z_{\delta_p}}{m - Z_{\dot{\omega}}}\delta_p$$

$$\dot{q} = \frac{[M_u + Z_u\Gamma]}{I_{yy}}u + \frac{[M_u + Z_u\Gamma]}{I_{yy}}w + \frac{[M_q + (Z_q + mU_0)\Gamma]}{I_{yy}} - \frac{\mathrm{mgsin}\,\Theta_0\Gamma}{I_{yy}}\theta + \frac{M_{\delta_e + Z_{\delta_e}\Gamma}\Gamma}{I_{yy}}\delta_e + \frac{M_{\delta_p} + Z_{\delta_p}\Gamma}{I_{yy}}\delta_p$$

$$\dot{\theta} = q$$

Now Rewriting in the state Space form as:

Since$u \approx 0$ in this mode, then $\frac{du}{dt} \approx 0$ and can eliminate the X force equation:

$$\begin{bmatrix}\dot{W}\\ \dot{q}\\ \dot{\theta}\end{bmatrix} = \begin{bmatrix}\frac{Z_w}{m - Z_{\dot{w}}} & \frac{Z_q + mU_0}{m - Z_{\dot{w}}} & \frac{-mg\sin\,\Theta_0}{m - Z_{\dot{w}}}\\ \frac{[M_w + Z_w\Gamma]}{I_{yy}} & \frac{[M_q + (Z_q + mU_0)\Gamma]}{I_{yy}} & \frac{-mg\sin\,\Theta_0}{I_{yy}}\\ 0 & 1 & 0\end{bmatrix}\begin{bmatrix}W\\ q\\ \theta\end{bmatrix} + \begin{bmatrix}\Delta Z^c\\ \Delta \mathrm{M}^c\\ 0\end{bmatrix}$$

$$\begin{bmatrix}\dot{W}\\ \dot{q}\\ \dot{\theta}\end{bmatrix} = \begin{bmatrix}\frac{Z_w}{m} & U_0 & -g\sin\,\Theta_0\\ \frac{M_w + Z_w\frac{\mathrm{M}_{\dot{w}}}{\mathrm{m}}]}{I_{yy}} & \frac{[M_q + (mU_0)\frac{\mathrm{M}_{\dot{w}}}{\mathrm{m}}]}{I_{yy}} & \frac{-mg\sin\,\Theta_0}{I_{yy}}\frac{M_{\dot{w}}}{m}\end{bmatrix}\begin{bmatrix}W\\ q\\ \theta\end{bmatrix} + \begin{bmatrix}\Delta Z^c\\ \Delta \mathrm{M}^c\\ 0\end{bmatrix}$$

The author of the paper took the values for the variables present in the state space matrices for the A340 Boeing flight and the resulting Transfer function after inputting the values for the variables that are determined from Flight Characteristics can be represented in state space form and output equation as state by equation:

$$\begin{bmatrix}\dot{W}\\ \dot{q}\\ \dot{\theta}\end{bmatrix} = \begin{bmatrix}-0.3149 & 235.8928 & 0\\ -0.0034 & -0.4282 & 0\\ 0 & 1 & 0\end{bmatrix}\begin{bmatrix}W\\ q\\ \theta\end{bmatrix} + \begin{bmatrix}-5.5079\\ 0.0021\\ 0\end{bmatrix}\delta_e$$

$$y = \begin{bmatrix}0 & 0 & 1\end{bmatrix}\begin{bmatrix}w\\ q\\ \theta\end{bmatrix} + [0]$$

We tried to get the variables of A340 flight that we can get from the flight characteristics of that flight. However, we failed to retrieve this data from the internet. This work presents an investigation into the development of pitch control schemes for pitch angles and pitch rate of an aircraft system. Pitch control with full state space feedback controller is investigated. A modern Controller Control the pitch of an aircraft system.

X=$[u, \omega, q, \gamma]^T$ and $\gamma = \theta - \alpha$ represents flight path angle, with $\alpha = \omega, u = \begin{bmatrix}\delta_e\\ \delta_p\end{bmatrix}$

The input (elevator deflection angle, δ_e) will be 0.2 rad and the output is the pitch angle(theta).

$X = [u, \omega, q, \gamma]^T$ and $\gamma = \theta - \alpha$ represents flight path angle with $\alpha = \omega, u = \begin{bmatrix}\delta_e\\ \delta_p\end{bmatrix}$

The input (elevator deflection angle, δ_e) will be 0.2 rad and the output is the pitch angle(theta).

There are three types of possible lateral-directional dynamics motion: roll subsidence mode, Dutch roll mode, and spiral mode.

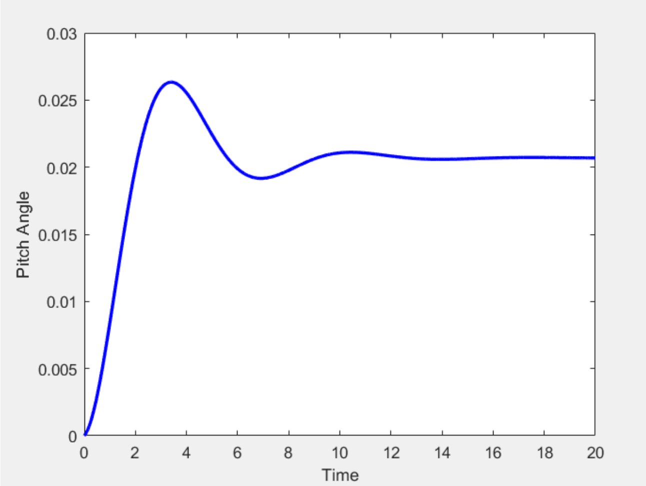

Figure 2 Open Loop Impulse Response (Pitch angle)

IV. Aircraft Lateral dynamics

Using Procedure like the longitudinal mode, we can develop the equation of motion for the lateral dynamic equations:

$$\frac{d\beta}{dt} = \frac{Y_\beta}{u_0}(\beta) + \frac{Y_p}{u_0}(p) + \frac{Y_r}{u_0}(r) + \frac{g\cos\theta}{u_0}(\phi) + \frac{Y_{\delta_r}}{u_0}\delta_a + \frac{Y_{\delta_r}}{u_0}\delta_r$$

$$\frac{dp}{dt} = L_\beta(\beta) + L_p(p) + L_r(r) + L_{\delta_r}(\delta_a) + L_{\delta_a}(\delta_r)$$

$$\frac{dr}{dt} = N_\beta(\beta) + N_p(p) + N_r(r) + N_{\delta_r}(\delta_a) + N_{\delta_a}(\delta_r)$$

$$\frac{d\phi}{dt} = p + \tan\theta_0(r)$$

Now we can write:

X^T=$[\beta\ p\ r\ \phi]^T$: state vector.

$u^T = [\delta_a\ \delta_r]^T$: control vector.

δ_a, δ_r : aileron and rudder deflection

β, ϕ : sideslip and roll angle.

p, r: roll and yaw rate.

$$A = \begin{bmatrix}\frac{Y_\beta}{u_0} & \frac{Y_p}{u_0} & \frac{Y_r}{u_0} & \frac{g\cos\theta_0}{u_0}\\ L_\beta & L_P & L_r & 0\\ N_\beta & N_p & N_r & 0\\ 0 & 1 & tan\theta_0 & 0\end{bmatrix}, B = \begin{bmatrix}\frac{Y_{\delta_r}}{u_0} & \frac{Y_{\delta_a}}{u_0}\\ L_{\delta_r} & L_{\delta_a}\\ N_{\delta_r} & N_{\delta_a}\\ 0 & 0\end{bmatrix},$$

$$C = \begin{bmatrix}1 & 0 & 0 & 0\\ 0 & 0 & 0 & 1\end{bmatrix}, D = \begin{bmatrix}0 & 0\\ 0 & 0\end{bmatrix}$$

The author of the paper took the values for the variables present in the state space matrices for the A340 Boeing flight and the resulting Transfer function after inputting the values for the variables that are determined from Flight Characteristics can be represented in state space form and output equation as state by equation:

We assume that the measurable output is the sideslip and β and roll angle ϕ, the matrix A, B and C are:

$$A = \begin{bmatrix} -0.0558 & -0.9968 & 0.0802 & 0.0415 \\ 0.5980 & -0.1150 & -0.0318 & 0 \\ -3.0500 & 0.3880 & -0.4650 & 0 \\ 0 & 0.0805 & 1.000 & 0 \end{bmatrix},$$

$$B = \begin{bmatrix} 0.0729 & 0.000 \\ -4.7500 & 0.00775 \\ 0.15300 & 0.1430 \\ 0 & 0 \end{bmatrix}, C = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

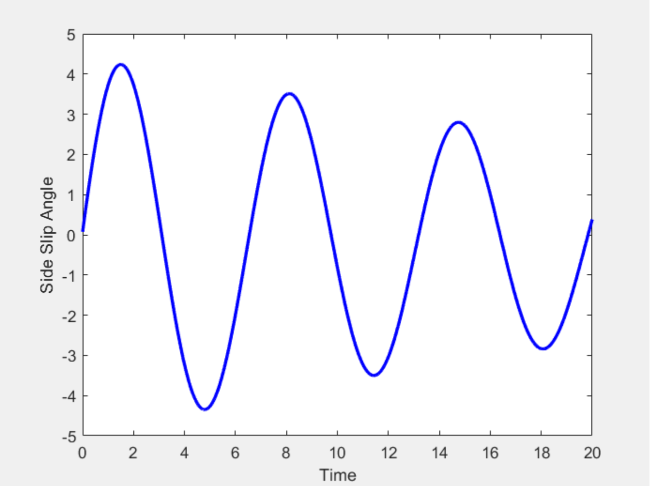

Figure 3 Open Loop Impulse response (Sideslip angle)

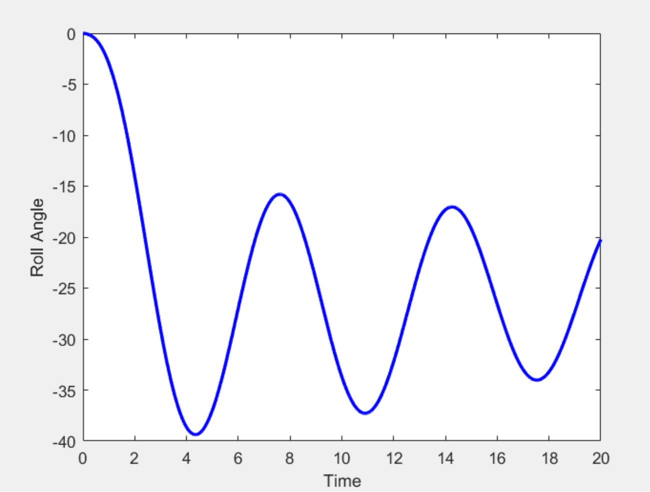

Figure 4 Open Loop Impulse Response (Roll angle)

V. Linear Quadratic Gaussian Controller

Linear Quadratic Gaussian (LQG) control represents a contemporary approach within the state space framework for creating optimal dynamic regulators. This methodology allows for a balanced consideration between regulation effectiveness and control input, while also accommodating variations arising from process and measurement noise. Similar to pole placement, the design of an LQG system necessitates a state-space model of the plant. In this discussion, emphasis is placed on the discrete-time scenario. To construct the LQG regulator, one can seamlessly integrate the Kalman filter and the LQ-optimal gain K, as depicted below:

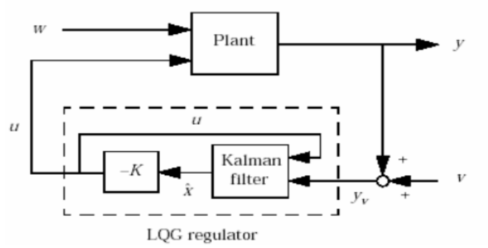

Figure 5 Block Diagram Of LQG Controller

The Kalman filter can also be used to get an estimate of the full state of a system when finding out the full state of the system is not possible or infeasible. The Kalman filter is an optimal full state estimator where the output of the Kalman filter is going to be the estimate of the full state $\hat{x}$ and the inputs to it are the control inputs u and noisy measurements y_v.

The plant in this model observed in Figure 5 experiences disturbances (process noise) *w* and measurement noise *v*. Here, *w* and *v* are modelled as white noise.

The Kalman filter itself can be treated as a system with the state space equations:

$$\dot{x} = Ax + Bu + Gw$$
$$y_v = Cx + Du + Hw + v$$

Kalman state estimator using the above state space representation of the plant, Kalman state estimator(kest) can be constructed using the following command in MATLAB:

$$[kest, L, P] = kalman(sys, Qn, Rn, Nn)$$

where *w* and *v* are modeled as white noise. L is the Kalman gain. Qn, Rn and Nn are the gaussian noise covariance data and P is the covariance matrix.

Estimate, $E(ww^T) = Q_n$
Estimate, $E(vv^T) = R_n$
Estimate, $E(wv^T) = N_n$

The Kalman filter is an optimal estimator when dealing with Gaussian white noise. Specifically, it minimizes the asymptotic covariance of the estimation error $x - \hat{x}$.

$$\lim_{t\to\infty} E((x - \hat{x})(x - \hat{x})^T)$$

The correct choice of Kalman Gain, L makes the error to drive towards zero.

The command $[kest, L, P] = kalman(sys, Qn, Rn, Nn)$ in MATLAB determines the Kalman gain L through an algebraic Ricatti equation as follows:

L, Kalman Gain is represented as $L = PC^T R_N^{-1}$

Where, the covariance matrix P that we also get from the $kalman$ command in MATLAB, is the solution to the Algebraic Ricatti Equation:

$$A^T P + PA - PBR^{-1}B^T P + Q = 0$$

Now let's build the regulator.

From Figure 5 it is obvious that the state space equation of the regulator is

$$\Rightarrow \ \frac{d}{dt}\hat{x} = A\hat{x} + Bu + L(y - C\hat{x} - Du)$$

$$\Rightarrow \ \frac{d}{dt}\hat{x} = [A - LC - (B - LD)K]\hat{x} + Ly_v$$
$$u = -Kx$$

The LQ-gain, K can be calculated using the MATLAB command $k = lqr(A, B, C, D)$. In the back end the solution to an Algebraic Ricatti equation is found to get the value of K as follows:

As we know in LQG control, the regulation performance is measured by a quadratic performance criterion of the form:

$$J(u) = \int_0^\infty \{x^T Qx + 2x^T Nx + u^T Ru\}$$

And Q and R are used to penalize performance and control input respectively. N is also a weighing matrix. The matrix N, Q and value R are user specified for a system based on what ones wants to prioritize between performance (how fast the system drives its states to zero or to a desired place) and control input.

K is represented as, $K = R^{-1}B^T P$
Where P is the solution to the Algebraic Ricatti Equation: $A^T P + PA - PBR^{-1}B^T P + Q = 0$.

The Kalman Gain, L

The goal is to regulate the output y around zero. The plant is subject to disturbance and is driven by controls. The regulator relies on the noisy measurements $y_v = y + v$ to generate these controls.

The input disturbance d is low frequency with power spectral density (PSD) concentrated below 10 rad/s. For LQG design purposes, it is modeled as white noise driving a lowpass filter with a cutoff at 10 rad/s, shown in the following Figure 6.

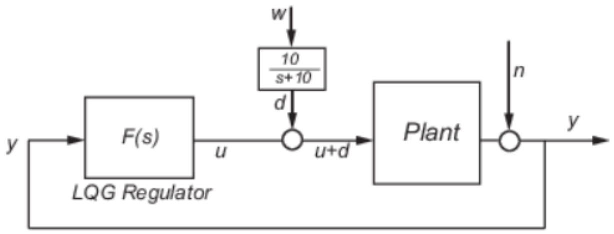

Figure 6 LQG Regulator Driving a low Pass Filter with a cut off at 10 rad/s

For simplicity, this noise is modeled as Gaussian white noise with variance of 1. There is some measurement noise n, with noise intensity given by

$$E(n^2) = 0.01$$

The cost function is

$$J(u) = \int_0^\infty (10y^2 + u^2)dt$$

to specify the trade-off between regulation performance and cost control. Note that an open-loop state space model is:

$$\dot{x} = Ax + Bu + Bd$$
$$y_v = Cx + n$$

Simulation results are shown in Figure 7, Figure 8, and Figure 9

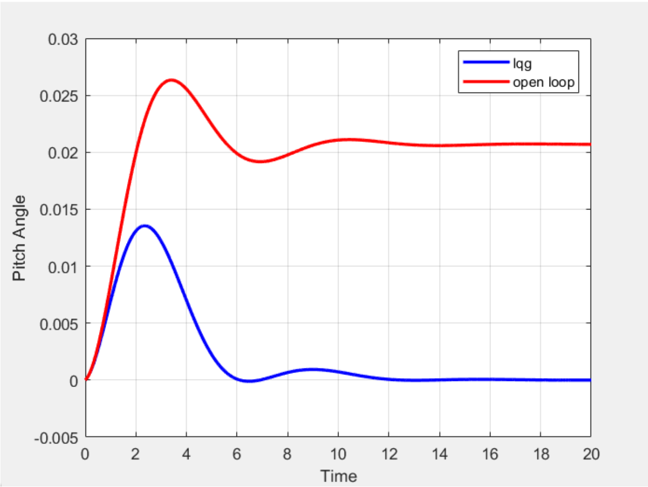

Figure 7 Comparison of Open-loop and Closed-Loop Impulse Response for the LQG (Pitch angle)

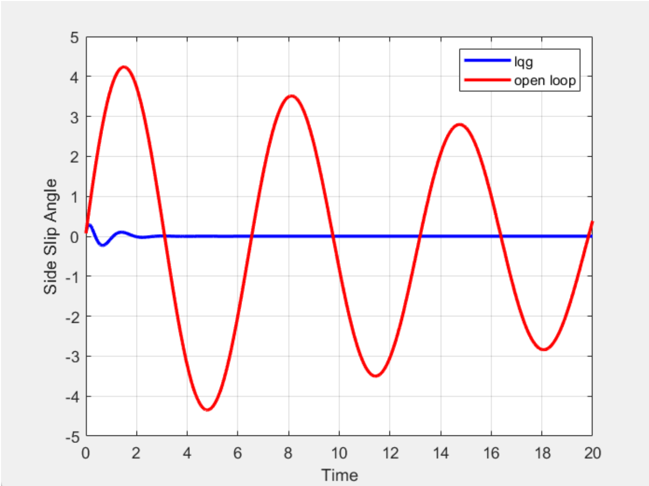

Figure 8 Comparison of Open-Loop and Closed-Loop Impulse response for the LQG (sideslip angle)

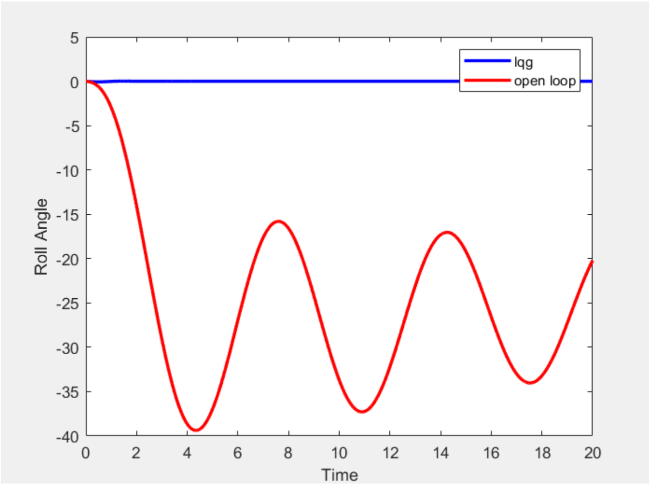

Figure 9 . Comparison of Open- and Closed-Loop Impulse Response for the LQG Example (Roll angle)

VI. Linear Quadratic Regulator Controller

Recent advancements in modern control theory have had a notable influence on the aviation sector. One of the techniques employed in modern control theory is the Linear Quadratic Regulator (LQR), which adopts a state-space approach for system analysis. Leveraging state-space methods offers a straightforward means of handling multi-output systems. Achieving system stabilization can be facilitated through the implementation of a full-state feedback system. Additionally, this approach enhances the overall understanding and control of complex aircraft systems. . The configuration of this control system is shown in Figure 10.

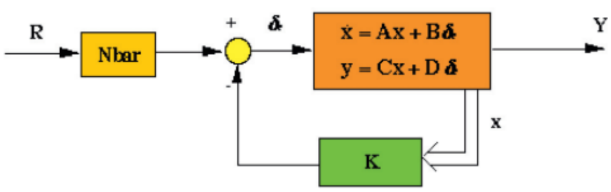

Figure 10 Full-state feedback controller with reference input

If the system is controllable then the using the LQR controller we can find the optimal gain matrix, K based on the cost function such that the eigen values of the closed system are placed in a perfect place, Using LQR controller we can trade-off regulation performance and control effort.

This is done by choosing two parameter values, input $R = 1$ and $Q = C^T \times C$ where C^T is the matrix transpose of C from state equations. The controller can be tuned by changing the nonzero elements in q matrix which is done in m-file code as obtained.

$$R = 1$$

$$Q = [0\ 0\ 0; 0\ 0\ 0; 0\ 0\ x]$$

$$K = lqr(A, B, Q, R)$$

Consequently, by tuning the value of $x = 500$, the following values of matrix K are obtained. If x is increased even higher, improvement to the response should be obtained even more. But for this case, the values of $x = 500$ is chosen because it satisfied the design requirements while keep x as small as possible.

The LQ-gain, K can be calculated using the MATLAB command $k = lqr(A, B, C, D)$. In the back end the solution to an Algebraic Ricatti equation is found to get the value of K as follows:

As we know in LQG control, the regulation performance is measured by a quadratic performance criterion of the form:

$$J(u) = \int_0^\infty \{x^T Q x + u^T R u\}$$

And Q and $R$are used to penalize performance and control input respectively. The matrix Q and value R are user specified for a system based on what ones wants to prioritize between performance (how fast the system drives its states to zero or to a desired place) and control input.

K is represented as, $K = R^{-1} B^T P$

Where P is the solution to the Algebraic Ricatti Equation:

$$A^T P + PA - PBR^{-1} B^T P + Q = 0\ .$$

To reduce steady state error of the system output, a value of constant gain Nbar should be added after the reference. With a full-state feedback controller all the states are feedback. The steady-state value of the states should be computed, multiply that by the chosen gain ک, and used a new value as the reference for computing the input.

Calculating Nbar:

Consider a linear time-invariant system described by the following state-space equations:

$$\dot{x} = Ax + Bu$$

$$y = Cx$$

The LQR control law is given by:

$$u = -Kx + r_{scaled}$$

where K is the LQR gain and r_{scaled} is the scaled reference input.

Let us determine r_{scaled} such that the output y tracks a reference signal r with minimal steady-state error. In steady state, $\dot{x} = 0$, so we have:

$$0 = Ax + Bu$$

Substituting the control law $u = -Kx + r_{scaled}$ into the above, we get:

$$0 = Ax - BKx + Br_{scaled}$$

This implies, $(A - BK)x = Br_{scaled}$

At steady-state, we want the output y to be equal to the reference r , so $y = r$ or $Cx = r$. We need to express x in terms of r and substitute it back into the equation to solve for r_{scaled}.

Rearranging the equation, we get the steady state value of x:

$$x = (A - BK)^{-1}Br_{scaled}$$

Now, substituting x in $Cx = r$, we have:

$$C(A - BK)^{-1}Br_{scaled} = r$$

Now, $r_{scaled} = \frac{1}{C(A-BK)^{-1}B} \times r$

For a unit reference input (when r = 1) the formula becomes:

$$r_{scaled} = \frac{1}{C(A - BK)^{-1}B}$$

Where, $r_{scaled} = Nbar$ in our discussion.

Simulation results are shown in Figure 11, Figure 12, and Figure 13

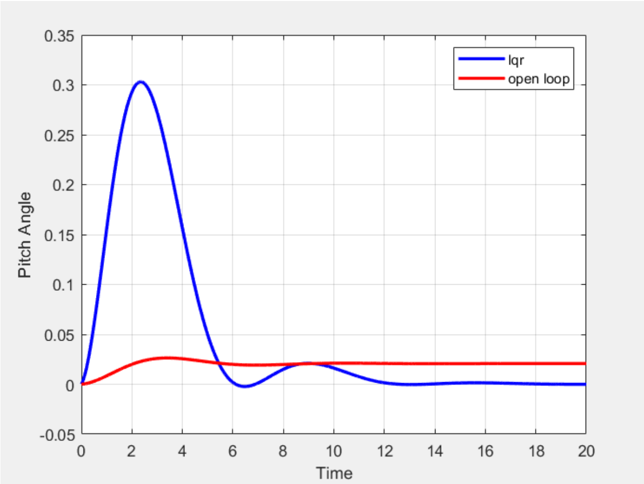

Figure 11 Comparison of Open-loop and Closed-Loop Impulse Response for the LQR (Pitch angle)

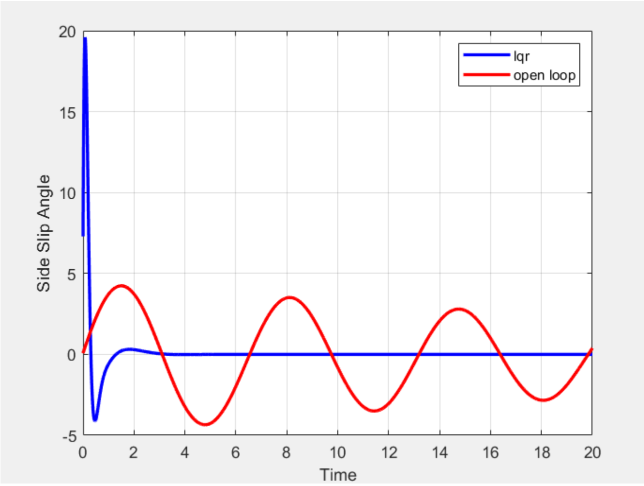

Figure 12 Comparison of Open-loop and Closed-Loop Impulse Response for the LQR (Sideslip angle)

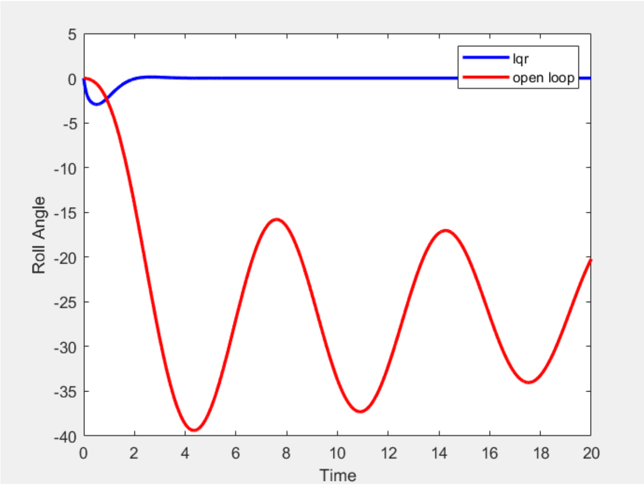

Figure 13 Comparison of Open- and Closed-Loop Impulse Response for the LQR Example (roll angle)

VII. KALMAN FILTERING

A discrete-time signal is a sequence of values that correspond to instants in time. The time instants at which the signal is defined are the signal's sample times, and the associated signal values are the signal's samples.

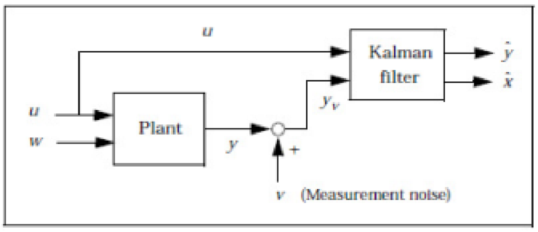

Figure 14 Kalman estimator

For a discrete plant the state space representation is as follows:

$$x(n+1) = Ax(n) + B(u(n) + w(n))$$
$$y(n) = Cx(n)$$

with additive Gaussian noise $w(n)$ on the input $u(n)$ and data. Our goal is to design a Kalman filter that estimates the output $y(n)$ given the inputs $u(n)$ and the noisy output measurements.

$$y_v(n) = Cx(n) + v(n)$$

Where $v(n)$ is measurement gaussian noise.

For a discrete Kalman Filter the steady state representation is given as follows:

Measurement update:
$\hat{x}(n/n) = \hat{x}(n/(n-1)) + M(y_v(n) - C\hat{x}(n/(n-1))$

Time update:
$\hat{x}((n+1)/n) = A\hat{x}(n) + Bu(n)$

Here,
$\hat{x}(n/(n-1))$ is the estimate $x(n)$ given past measurements up to $y_v(n-1)$ and $\hat{x}(n/n)$ is the updated estimate based on the last measurement $y_v(n)$.

Given the current estimate $\hat{x}(n/n)$, the time update predicts the state value at the next sample $n+1$ (one-step-ahead predictor). The measurement update then adjusts this prediction based on the new measurement $y_v(n-1)$. The correction term is a function of the innovation, that is, the discrepancy.

$$y_v(n-1) - C\hat{x}(n/(n-1) = C(x(n+1) - \hat{x}((n+1)/n)$$

between the measured and predicted values o $y(n+1)$. The innovation gain M is chosen to minimize the steady-state covariance of the estimation error given the noise covariances.

$$Estimate, E(w(n)w(n)^T) = Q,$$

$$Estimate, E(v(n)v(n)^T) = R$$

The covariance values for process noise (Q) and sensor noise (R) are positive quantities usually acquired through system studies or measurements. For our simulation, let's take $Q = 2.3$ and $R = 1$.

The time and measurement update equations can be updated into one state-space model (the Kalman filter)

$$\hat{x}((n+1)/n) = A(I - MC)\hat{x}(n/(n-1) + [B\ AM]\begin{bmatrix} u(n) \\ y_v(n) \end{bmatrix}$$

$$\hat{y}(n/n) = C(I - MC)\hat{x}(n/(n-1) + CMy_v(n)$$

The $kalman$ command can be used to design the Kalman filter in MATLAB using the command:
[kalmf,L,~,Mx,Z] = Kalman(sys,Q,R);

This above state space model of the Kalman filter created in the MATLAB implements the measurement update and time update equations. The filter inputs are the plant input u and the noisy plant output y. The first output of kalmf is the estimate $\hat{y}$ of the true plant output, and the remaining outputs are the state estimates.

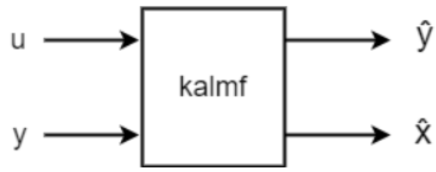

Figure 15 Implement of Kalmf

To see the working of the filter some data is generated, and the filtered response is compared with the true plant response. The complete system is shown in Figure 16.

A known sinusoidal input vector is generated to simulate the filter behavior.

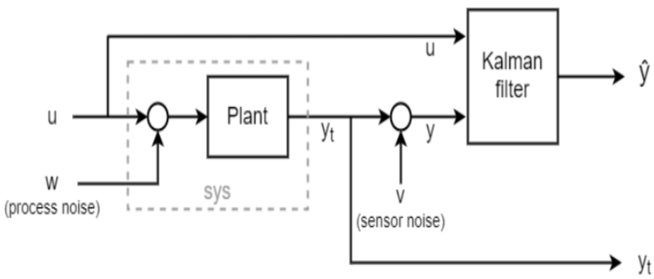

Figure 16 Complete System with Kalman Filter and Plant connected

This filter generates an optimal estimate $\hat{y}(n/n)$ of $y(n)$. Filter state is $\hat{x}(n/(n-1))$

Simulation Results are shown in Figure 18 and Figure 19

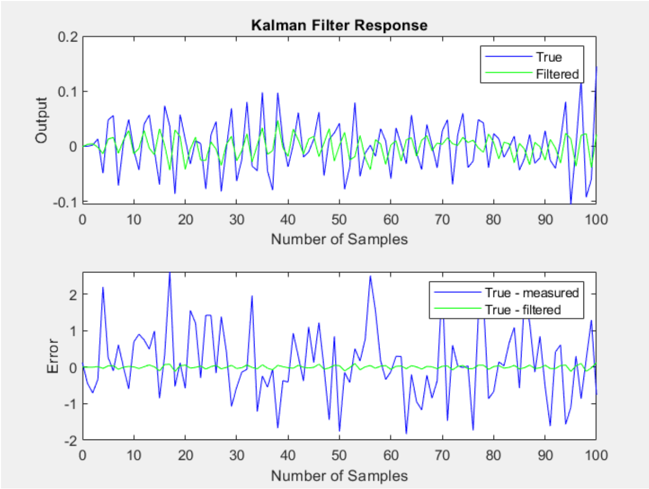

Figure 17 Kalman Filter Response for the pitch angle θ

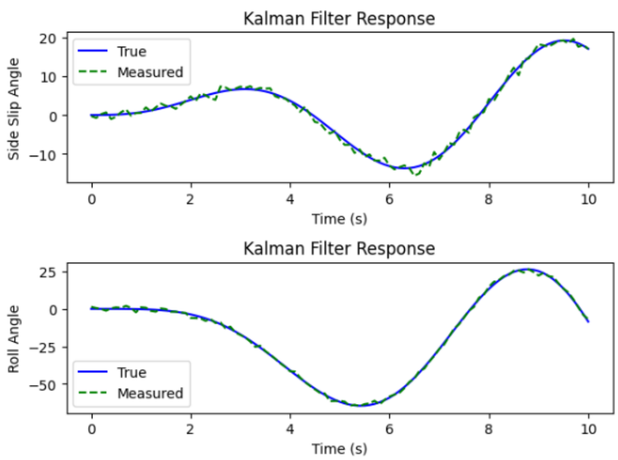

Figure 18 Kalman Filter Response for the sideslip angle β and roll angle ϕ

VIII. RESULTS AND DISCUSSION

A. Comparison of Open-loop and Closed-Loop Impulse Response for the LQG

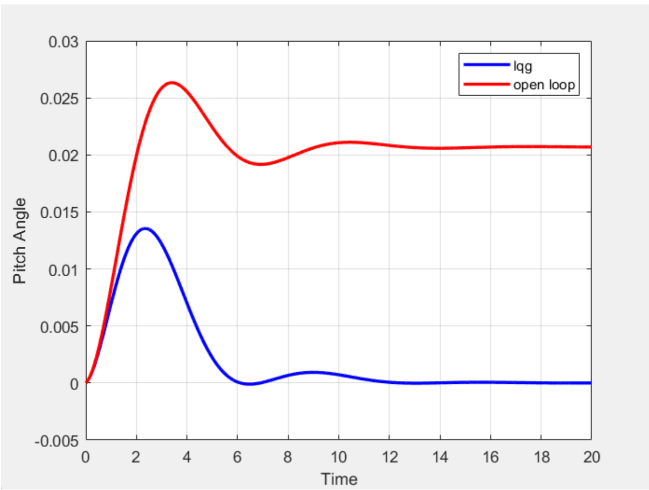

Figure 19 Comparison of Open-loop and Closed-Loop Impulse Response for the LQG (Pitch angle)

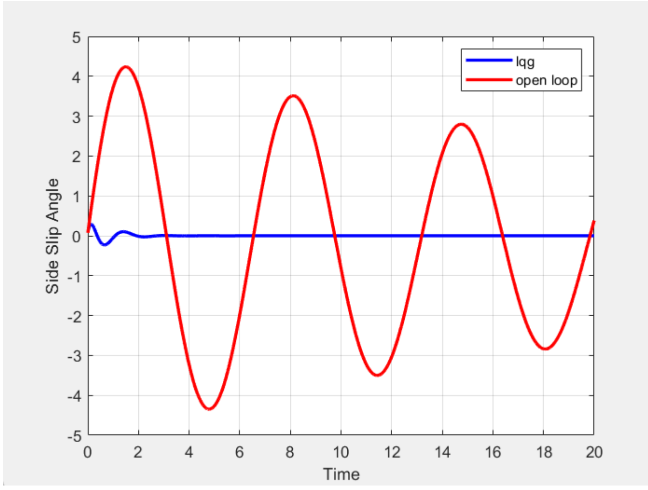

Figure 20 Comparison of Open-Loop and Closed-Loop Impulse response for the LQG (sideslip angle)

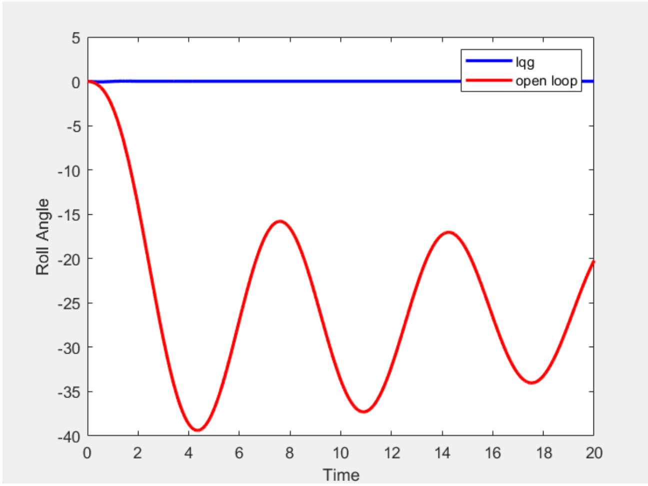

Figure 21Comparison of Open- and Closed-Loop Impulse Response for the LQG Example (Roll angle)

From Figure 19, Figure 20, and Figure 21 it's obvious that the LQG controller is doing a very good job of driving the system to zero after an impulse input was given to the system. Whereas the open loop system in contrast is having good performance. LQG gives a very good following to the outputs of plants with a steady shift error limited and the Kalman filter is an optimal estimator when dealing with Gaussian white noise. Optimal estimation provides an alternative rationale for the choice of observer gains in the current estimator which is based on observer performance in the presence of process noise and measurement errors.

B. Comparison of Open-loop and Closed-Loop Impulse Response for the LQR

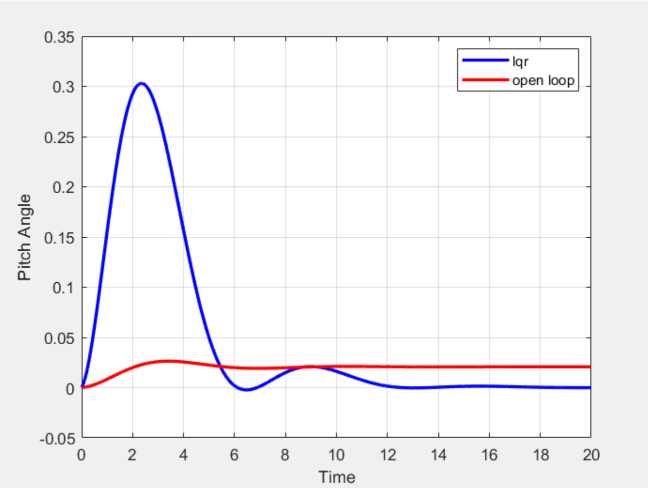

Figure 22 Comparison of Open-loop and Closed-Loop Impulse Response for the LQR (Pitch angle)

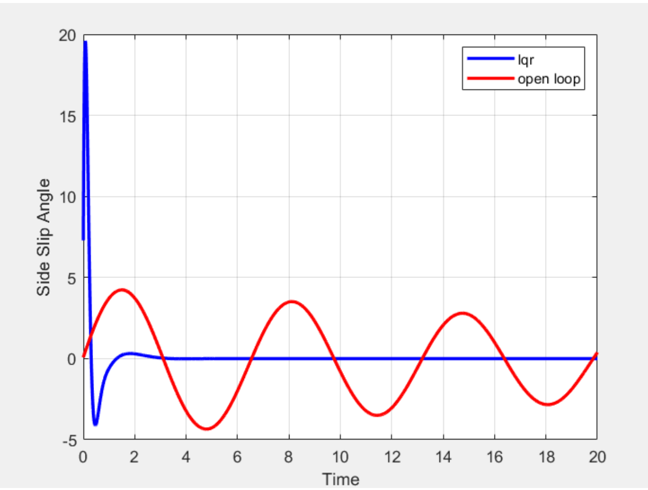

Figure 23 Comparison of Open-loop and Closed-Loop Impulse Response for the LQR (Sideslip angle)

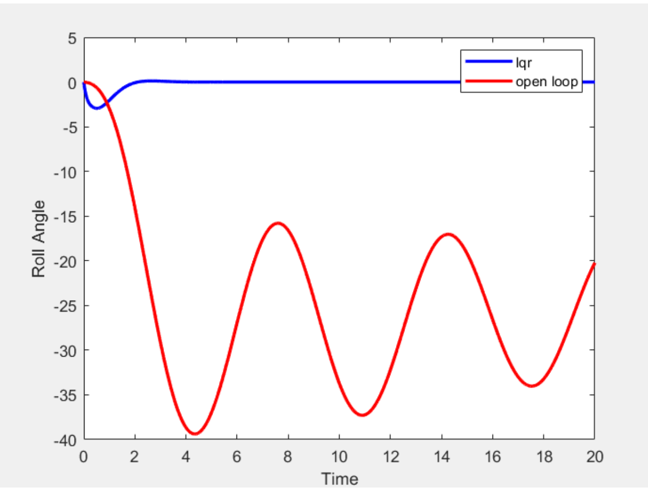

Figure 24 Comparison of Open- and Closed-Loop Impulse Response for the LQR Example (roll angle)

From Figure 22, Figure 23, and Figure 24 we can say that LQR controller is also doing a good job and regulating the system. We can see that LQG is relatively controlling the system in a better way because the control effort for LQR controller is higher than the control effort for the LQG controller which differs from what the author of the current paper we are reviewing got. However, this discrepancy might be due to the difference in values taken for noise and disturbance entering the system for the LQG controller, or maybe due to the difference in choice of the regulation matrices Q, R, and N for both the LQG and LQR controllers.

As LQG controller is robust to Noise and Disturbance too, this makes LQG controller a better controller compared to LQR in my view.

C. Kalman Filtering

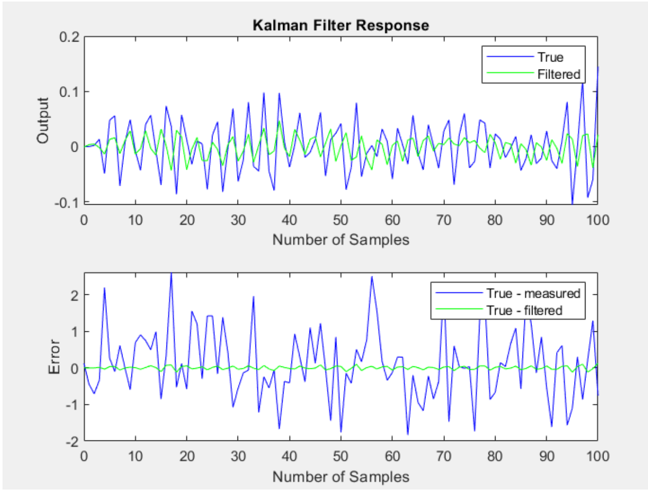

Figure 25 Kalman Filter Response for the pitch angle θ

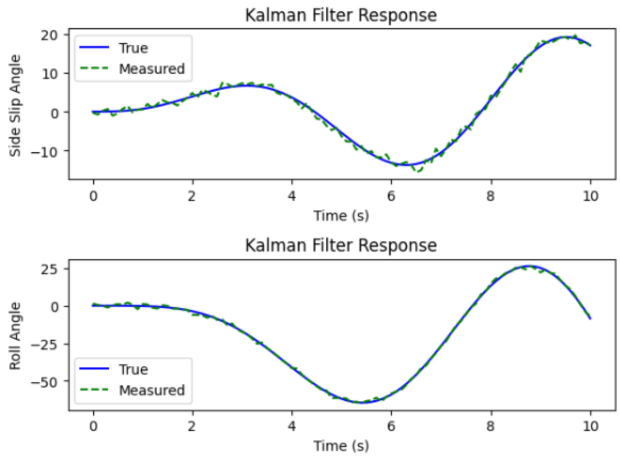

Figure 26 Kalman Filter Response for the sideslip angle β and roll angle ϕ

From Figure 25 and Figure 26 we can say that Kalman Filter is making a pretty good estimate of the state. It filters out the noise and disturbance the system is prone to. The simulation for Figure 26, Kalman Filter Response for the sideslip angle β and roll angle ϕ was done using Python rather than MATLAB, that was used for all other simulations. This is because we were facing multiple errors while running the code for Kalman Filter Response for the sideslip angle β and roll angle ϕ in MATLAB.

IX. Conclusion

The validated model for controlling an aircraft's pitch, roll, and sideslip plays a crucial role in shaping the control strategy for real-world systems.

The control of these angles necessitates dedicated controllers to maintain them at desired values. This involves minimizing the error signal, which represents the disparity between the output angle and the desired angle. The LQR control approach proves effective in managing the pitch, roll, and sideslip angles of the aircraft, achieving a system response of 0.2 radians (11.5 degrees). Analysis results highlight the superior performance of the LQR controller. Future endeavors could focus on advancing control techniques for greater robustness. Implementation of the proposed control algorithm in practical settings is crucial for validating theoretical outcomes. Additionally, the LQG demonstrates commendable tracking of plant outputs with limited steady-state errors, and the Kalman filter emerges as an optimal estimator in handling Gaussian white noise. Optimal estimation offers an alternative rationale for selecting observer gains based on observer performance in the presence of process noise and measurement errors.

The Kalman filter operates by estimating a process through feedback control, wherein it estimates the process state at a given time and incorporates feedback in the form of noisy measurements. The equations governing the Kalman filter fall into two categories: time update equations, projecting the current state and error covariance estimates forward in time, and measurement update equations, responsible for incorporating new measurements into the a priori estimate to derive an improved a posteriori estimate.

Acknowledgment

For the Dynamics equation of motions for solving the longitudinal and lateral dynamics book Flight Stability and Automatic Control by Robert Nelson. The Controls Playlist on YouTube of Steve Brunton and Christopher Lum really helped us learned the concepts of the control theory implemented in this paper, moreover the book machine learning control, by Duriez, Brunton, and Noack, and the MathWorks documentation of controls related functions really helped us to complete the math lab implementation of the LQG, LQR and Kalman Filtering.

References

[1] Robert C.Nelson.1998. Flight Stability and Automatic Control. University of Notre Dame.

[2] Anonymous. 2006: uPSD3400 PWM API with R/C Servomotors Control Example (PCM), AN2283 Application note, June, http://www.st.com.

[3] Wong, K.C .2006. UAV Design Activities in a University Environment. University of Sydney.

[4] Aly El-Osery.2004. MATLAB Tutorial, October. . http://www.ee.nmt.edu

[5] Michael, T. Rusnell. 2004. Morphing UAV Pareto Curve Shift for Enhanced Performance. University of Notre Dame.

[6] Reed Siefert Christiansen (2004). Design of an Autopilot for Small Unmanned Aerial Vehicle. Brigham Young University.

[7] Tom Bateman, Barry Blakeley, Adam Dean, Jake Hanft and Mike Sheek (2004). UAV Stability Augmentation System (USAS), University of Colorado.

[8] Bahadir Karuv (2003). Building Blocks, Bergen County Academies Invention Lab.

[9] A. Maddi, “Modélisation et contrôle du vol latéral d’un avion”, Magister Thesis, Electronics Department, University of Blida, Algeia, 1997.

[10] M. Grewal and A. Andrews, “Kalman Filtering Theory and Practice Using MATLAB”, (Second ed.), New York, NY USA: John Wiley & Sons, Inc., 2001.

[11] P.C. Murphy and V. Klein, “Estimation of Aircraft Unsteady Aerodynamic Parameters From Dynamic Wind Tunnel Testing” , AIAA 2001- 4016, August 2001

[12] G. Bishop, G. Welch, “An Introduction to the Kalman Filter”, University of North Carolina at Chapel Hill, Department of Computer Science, SIGGRAPH 2001.

[13] D. Simon, “Optimal state estimation: Kalman, H-infinity, and nonlinear approaches”, John Wiley & Sons, 2006.

[14] P. S. Maybeck, “Applied Optimal Estimation—Kalman Filter Design and Implementation”, notes for a continuing education course offered by the Air Force Institute of Technology, Wright-Patterson AFB, Ohio, semiannually since December 1974

X. Appendices

```
%% (1) Simulation of Aircraft’s Longitudanal Motion State Space equations without any
control signal (OPEN LOOP)

clear all
% State Space Matrices
A = [-0.3149,235.8928,0; -0.0034,-0.4282,0; 0,1,0];
B = [-5.5079; 0.0021; 0];
C = [0,0,1];
D = 0;
sys = ss(A,B,C,D);

% Time Interval
t_impulse = 0:0.01:20;  % Time vector for impulse response

% Impulse Response
impulse_response_openloop = impulse(sys, t_impulse);
plot(t_impulse, impulse_response_openloop, 'b', 'LineWidth', 2);
xlabel('Time');
ylabel('Pitch Angle');

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

%% (2) Simulation of Aircraft’s Lateral Motion State Space equations without any control
signal (OPEN LOOP)

clear all
% State Space Matrices
A = [-0.0558,-0.9968,0.0802,0.0415;0.5980,-0.1150,-0.0318,0;-3.0500,0.3880,-
0.4650,0;0,0.0805,1,0 ];
B = [0.0729,0;-4.75,0.00775;0.15300,0.1430;0,0];
C = [1 0 0 0;0 0 0 1];
D = [0 0; 0 0];

% State Space Equations
sys = ss(A,B,C,D);
% Time Interval
t_impulse = 0:0.01:20;  % Time vector for impulse response

% Impulse Response
impulse_response_openloop1 = impulse(sys(1,1), t_impulse);
impulse_response_openloop2 = impulse(sys(2,1), t_impulse);

figure(1);
plot(t_impulse, impulse_response_openloop1, 'b', 'LineWidth', 2);
xlabel('Time');
ylabel('Side Slip Angle');

figure(2);
plot(t_impulse, impulse_response_openloop2, 'b', 'LineWidth', 2);
xlabel('Time');
ylabel('Roll Angle');


%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
```

```matlab
%% (3) Simulation for Longitudanal Dynamics of An Aircraft in State Space Form controlled using LQG controller

clear all
% Define the system matrices
A = [-0.3149,235.8928,0; -0.0034,-0.4282,0; 0,1,0];
B = [-5.5079; 0.0021; 0];
C = [0,0,1];
D = 0;

% Define the disturbance matrices
% Assume a small positive value for process noise covariance
Qn = 1; % Modify as per your system's characteristics
Rn = 1; % Measurement noise covariance
Nn = 0; % Cross-correlation between process and measurement noise (assuming zero)

% Define the LQR weighting matrices
Qlqr = [0,0,0; 0,0,0; 0,0,500];
Rlqr = 1;

% Compute the LQR gain
K = lqr(A, B, Qlqr, Rlqr);

% Create state-space model Open Loop
sys = ss(A, B, C, D);

% Design the Kalman filter
[kest,L,P] = kalman(sys, Qn, Rn, Nn);

% Form the LQG regulator
Acl = [A-B*K B*K; zeros(size(A)) A-L*C];
Bcl = [B; zeros(size(B))];
Ccl = [C zeros(size(C))];
Dcl = D;

% Create state-space model Closed Loop
sys_cl = ss(Acl, Bcl, Ccl, Dcl);

% Simulate the impulse response
t_impulse = 0:0.01:20;  % Time vector for impulse response
impulse_response_lqg = impulse(sys_cl, t_impulse);
impulse_response_openloop = impulse(sys, t_impulse);


% Plot the impulse response
figure;
plot(t_impulse, impulse_response_lqg, 'b', 'LineWidth', 2);

hold on
plot(t_impulse, impulse_response_openloop, 'r', 'LineWidth', 2);
xlabel('Time');
ylabel('Pitch Angle');

legend({'lqg','open loop'},'Location','northeast')
grid on;

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
```

```matlab
%% (4) Simulation for Lateral Dynamics of An Aircraft in State Space Form controlled using
LQG controller

clear all
% Define the system matrices
A = [-0.0558,-0.9968,0.0802,0.0415;0.5980,-0.1150,-0.0318,0;-3.0500,0.3880,-
0.4650,0;0,0.0805,1,0 ];
B = [0.0729,0;-4.75,0.00775;0.15300,0.1430;0,0];
C = [1 0 0 0;0 0 0 1];
D = [0 0; 0 0];

% Define the disturbance matrices
% Assume a small positive value for process noise covariance
Qn = 1; % Modify as per your system's characteristics
Rn = 1; % Measurement noise covariance
Nn = 0; % Cross-correlation between process and measurement noise (assuming zero)

% Define the LQR weighting matrices
Qlqr = [0,0,0,0; 0,0,0,0;0,0,0,0; 0,0,0,500];
Rlqr = 1;

% Compute the LQR gain
K = lqr(A, B, Qlqr, Rlqr);

% Create state-space model Open Loop
sys = ss(A, B, C, D);

% Design the Kalman filter
[kest,L,P] = kalman(sys, Qn, Rn, Nn);

% Form the LQG regulator
Acl = [A-B*K B*K; zeros(size(A)) A-L*C];
Bcl = [B; zeros(size(B))];
Ccl = [C zeros(size(C))];
Dcl = D;

% Create state-space model Closed Loop
sys_cl = ss(Acl, Bcl, Ccl, Dcl);

% Simulate the impulse response
t_impulse = 0:0.01:20;  % Time vector for impulse response

impulse_response_lqg1 = impulse(sys_cl(1,1), t_impulse);
impulse_response_openloop1 = impulse(sys(1,1), t_impulse);


% Plot the impulse response
figure(1);
plot(t_impulse, impulse_response_lqg1, 'b', 'LineWidth', 2);

hold on
plot(t_impulse, impulse_response_openloop1, 'r', 'LineWidth', 2);
xlabel('Time');
ylabel('Side Slip Angle');

legend({'lqg','open loop'},'Location','northeast')
grid on;


impulse_response_lqg2 = impulse(sys_cl(2,1), t_impulse);
impulse_response_openloop2 = impulse(sys(2,1), t_impulse);
```

```
figure(2);
plot(t_impulse, impulse_response_lqg2, 'b', 'LineWidth', 2);

hold on
plot(t_impulse, impulse_response_openloop2, 'r', 'LineWidth', 2);
xlabel('Time');
ylabel('Roll Angle');

legend({'lqg','open loop'},'Location','northeast')

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%



%% (5) Simulation for Longitudanal Dynamics of An Aircraft in State Space Form controlled using LQR controller

clear all
% Define the state-space system matrices (A, B, C, D)
A = [-0.3149,235.8928 0; -0.0034, -0.4282, 0; 0, 1, 0];
B = [-5.5079; 0.0021; 0];
C = [0 0 1];
D = 0;

% Open Loop System
sys = ss(A, B, C, D);

% Define the Q and R matrices for LQR controller design
R = 1;
Q = 500*(C'*C);


% Design the LQR controller
K = lqr(A, B, Q, R);

% Calculate the constant gain Nbar to eliminate steady-state error
Nbar = -1 / (C * (A - B * K)^(-1) * B); %For unit reference input

% Create the closed-loop system with Nbar
A_cl = A - B * K;
B_cl = B*Nbar;
C_cl = C;
D_cl = D;

% Define the state-space system for closed-loop control with Nbar
sys_cl = ss(A_cl, B_cl, C_cl, D_cl);

% Generate the impulse response of the closed-loop system
t_impulse = 0:0.01:20;  % Time vector for impulse response
impulse_response_lqr = impulse(sys_cl, t_impulse);
impulse_response_openloop = impulse(sys, t_impulse);


% Plot the impulse response
figure;
plot(t_impulse, impulse_response_lqr, 'b', 'LineWidth', 2);

hold on
plot(t_impulse, impulse_response_openloop, 'r', 'LineWidth', 2);
xlabel('Time');
```

```
ylabel('Pitch Angle');

legend({'lqr','open loop'},'Location','northeast')
grid on;

% Display the calculated Nbar
fprintf('Constant Gain Nbar: %f\n', Nbar);



%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

%% (6) Simulation for Lateral Dynamics of An Aircraft in State Space Form controlled using
LQR controller


clear all
% Define the state-space system matrices (A, B, C, D)
A = [-0.0558,-0.9968,0.0802,0.0415;0.5980,-0.1150,-0.0318,0;-3.0500,0.3880,-
0.4650,0;0,0.0805,1,0 ];
B = [0.0729,0;-4.75,0.00775;0.15300,0.1430;0,0];
C = [1 0 0 0;0 0 0 1];
D = [0 0; 0 0];

% Open Loop System
sys = ss(A,B,C,D);

% Define the Q and R matrices for LQR controller desig
R = 1;
Q = 500*(C'*C);

% Design the LQR controller
K = lqr(A, B, Q, R);

% Calculate the constant gain Nbar to eliminate steady-state error
Nbar = -inv(C*inv(A-B*K)*B); %For unit reference input

% Create the closed-loop system with Nbar
A_cl = A - B * K;
B_cl = B*100;
C_cl = C;
D_cl = D;

% Define the state-space system for closed-loop control with Nbar
sys_cl = ss(A_cl, B_cl, C_cl, D_cl);
t_impulse = 0:0.01:20;

impulse_response_lqr1 = impulse(sys_cl(1,1), t_impulse);
impulse_response_openloop1 = impulse(sys(1,1), t_impulse);


% Plot the impulse response
figure(1);
plot(t_impulse, impulse_response_lqr1, 'b', 'LineWidth', 2);

hold on
plot(t_impulse, impulse_response_openloop1, 'r', 'LineWidth', 2);
xlabel('Time');
ylabel('Side Slip Angle');

legend({'lqr','open loop'},'Location','northeast')
```

```
grid on;


impulse_response_lqr2 = impulse(sys_cl(2,1), t_impulse);
impulse_response_openloop2 = impulse(sys(2,1), t_impulse);
figure(2);
plot(t_impulse, impulse_response_lqr2, 'b', 'LineWidth', 2);

hold on
plot(t_impulse, impulse_response_openloop2, 'r', 'LineWidth', 2);
xlabel('Time');
ylabel('Roll Angle');

legend({'lqr','open loop'},'Location','northeast')
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%



%% (7) Kalman Filtering for Aircraft’s Longitudanal Motion State Space Equations


clear all
% Define the system matrices
A = [-0.3149,235.8928,0; -0.0034,-0.4282,0; 0,1,0];
B = [-5.5079; 0.0021; 0];
C = [0,0,1];
D = 0;

% Sample Time = -1 to mark discrete time
Ts = -1;
% Discrete Plant Model
sys = ss(A,[B B],C,D,Ts,'InputName',{'u' 'w'},'OutputName','y');  % Plant dynamics and
additive input noise w

% noise covariance Q and the sensor noise covariance R are values greater than zero
Q = 2.3;
R = 1;

% Design the Kalman Filter
[kalmf,L,~,Mx,Z] = kalman(sys,Q,R);
%  discard the state estimates and keep only the first output,y_hat
kalmf = kalmf(1,:);


sys.InputName = {'u','w'};
sys.OutputName = {'yt'};

% sumblk to create an input for the measurement noise v
vIn = sumblk('y=yt+v');

kalmf.InputName = {'u','y'};
kalmf.OutputName = 'ye';

% Using connect to join sys and the Kalman filter together such that u is a shared input and
the noisy plant output y feeds into the other filter input
SimModel = connect(sys,vIn,kalmf,{'u','w','v'},{'yt','ye'});

t = (0:100)';
% Sinusoidal input Vector
u = sin(t/5);
```

```
rng(10,'twister');
w = sqrt(Q)*randn(length(t),1);
v = sqrt(R)*randn(length(t),1);

% Simulate the response
out = lsim(SimModel,[u,w,v]);

yt = out(:,1);   % true response
ye = out(:,2);  % filtered response
y = yt + v;     % measured response

% Comparing the true response with the filtered response
clf
subplot(211), plot(t,yt,'b',t,ye,'g'),
xlabel('Number of Samples'), ylabel('Output')
title('Kalman Filter Response')
legend('True','Filtered')
subplot(212), plot(t,yt-y,'b',t,yt-ye,'g'),
xlabel('Number of Samples'), ylabel('Error')
legend('True - measured','True - filtered')


%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%


%% (8) Kalman Filtering for Aircraft's Lateral Motion State Space Equations



import numpy as np
import matplotlib.pyplot as plt
from scipy.signal import lti, lsim

# State Space matrices
A = np.array([[-0.0558, -0.9968, 0.0802, 0.0415],
              [0.5980, -0.1150, -0.0318, 0],
              [-3.0500, 0.3880, -0.4650, 0],
              [0, 0.0805, 1, 0]])
B = np.array([[0.0729, 0],
              [-4.75, 0.00775],
              [0.15300, 0.1430],
              [0, 0]])
C = np.array([[1, 0, 0, 0],
              [0, 0, 0, 1]])
D = np.array([[0, 0],
              [0, 0]])

# Define system
sys = lti(A, B, C, D)

# Process and Measurement noise covariance
Qn = 2.3
Rn = 1

# Time vector
t = np.arange(0, 10.1, 0.1)  # Time vector

# Sinusoidal input
u = np.column_stack((np.sin(t), np.zeros_like(t)))

# Simulate the true system
```

```
_, x_true, _ = lsim(sys, u, t)

# Add noise to measurements (for simulation)
y_measured = lsim(sys, u, t)[1] + np.sqrt(Rn) * np.random.randn(len(t), C.shape[0])

state = ["Side Slip Angle", "Roll Angle"]
# Kalman filter simulation.
# Plotting

for i in range(x_true.shape[1]):
    plt.subplot(x_true.shape[1], 1, i + 1)
    plt.plot(t, x_true[:, i], 'b', label='True')
    plt.plot(t, y_measured[:, i], 'g--', label='Measured')
    plt.legend()
    plt.xlabel('Time (s)')
    plt.ylabel(state[i])
    plt.title('Kalman Filter Response')

plt.tight_layout()
plt.show()

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
```