

PI-Filter compensator for LQG controller aimed to fixed-wings aircrafts

María del Carmen Claudio · Alida Ortíz-Pupo · Gloria Chicaiza-Claudio

Received: 20 Feb 2024 / Accepted: 07 Jun 2024 / Published: 15 Nov 2024

Abstract: Fixed-wing aircraft generate lift and propulsion using their wings, relying on forward motion for airflow instead of rotating blades like helicopters. They offer advantages such as extended range, higher velocities, stability in turbulent weather, and lower operational costs compared to rotary-wing aircraft. This study introduces a method to enhance control smoothness for fixed-wing aircraft using Linear Quadratic Gaussian control and Proportional-Integral filter compensation. Flight simulators like FlightGear are employed to test control algorithms, providing realistic flight dynamics and versatile options for various aircraft types. This approach offers a cost-effective and efficient means to develop and test controllers for challenging flight scenarios, while demonstrating the performance of the LQG+PI method by displaying the trends in longitudinal and lateral control errors.

Keywords PI+LQG · fixed-wing aircraft · FlightGear

1 Introduction

Fixed-wing aircraft are airplanes that generate lift and propulsion by directing airflow over their wings, which remain fixed in position during flight [1]. Unlike rotorcraft such as helicopters, which utilize rotating wings or blades to achieve lift, fixed-wing aircraft rely on forward motion to create the airflow necessary for lift generation. While rotary-wing aircraft offer enhanced maneuverability due to their ability to perform vertical takeoff and hovering, fixed-wing aircraft are the standard in aviation for various purposes, including long-distance

transportation, aerial surveillance, cargo transport, and military operations [2]. This preference is due to the inherent advantages rooted in the aerodynamic design and operational characteristics. The advantages of fixed-wing aircraft are numerous and encompass various aspects of performance, efficiency, versatility, and operational capability. Compared to rotary-wing aircraft, fixed-wing types offer extended flight range and endurance due to their design optimized for forward motion rather than vertical takeoff and hovering [3]. Additionally, fixed-wing aircraft can achieve significantly higher speeds, thanks to their aerodynamic configuration, and they demonstrate superior stability in turbulent weather conditions [4]. Moreover, fixed-wing aircraft can carry larger payloads and offer lower operational and maintenance costs than rotary-wing aircraft.

Flight simulators are a cost-effective and efficient way to calibrate, test, and improve control algorithms before conducting experiments on fixed-wing aircraft. Whether for military, entertainment, or commercial applications, a suitable simulator can be an excellent tool for proper vehicle handling, particularly when the vehicle can be damaged if the pilot loses control or is inexperienced [5]. In this context, the control of an aircraft that is challenging to test in a laboratory can be significantly enhanced by utilizing a simulator with flexible characteristics capable of interfacing with mathematical software, especially when assessing responses to wind disturbances that are difficult to measure and replicate experimentally. Simulation software such as X-Plane, AirSim, Gazebo, and FlightGear have such capabilities, with research studies employing them for various purposes. FlightGear features an intuitive user interface, the ability to communicate with external software, dynamic properties for an assortment of freely downloadable simulated prototypes, options for adding a debugging mode for communication errors, and the ability to display prototype states over LAN networks [6]. Furthermore, the versatility of FlightGear allows the use of different aircraft and flying objects and avoids the use of potentially oversimplified flight dynamics mod-

María del Carmen Claudio  · Gloria Chicaiza-Claudio 
Inmersoft Technologies
Quito, Ecuador
{mclaudio, gchicaiza}@inmersoft.com

Alida Ortíz-Pupo 
Instituto de Automática
Universidad Nacional de San Juan
San Juan, Argentina
aortiz@inaut.unsj.edu.ar

els. Given these features, FlightGear can communicate with mathematical software to modify the state of simulated prototype actuators to test controllers or calibrate them against simulated disturbances [7]. In this way, FlightGear performs the flight calculations, with the program treated as a black box to Matlab, akin to a real aircraft [8]. This approach leverages a high degree of realism provided by FlightGear, which utilizes established and realistic Flight Dynamics Models (FDMs) [9] based on nonlinear equations of motion.

Research has explored the use of LQG control for managing fixed-wing aircraft [10], yet a significant hurdle emerges during the linearization process and the dynamic alteration of linearization points, leading to unwanted abrupt maneuvers [11]. To overcome this challenge, this study introduces a PI filter compensation method aimed at enhancing the smoothness of LQG-generated control. As the aircraft, we used the F-104 Starfighter data to achieve the linearized model while employing its complete dynamics within FlightGear. The results show the lateral and longitudinal behavior of the system, indicating a trend of control errors towards zero.

The article is divided into five parts, including the Introduction and Conclusions. In Section 2, the linearization of the model is presented, while Section 3 develops the PI-Filter compensator. Furthermore, Section 4 includes the results of this work.

2 Modeling

When determining the acceleration of each mass element, we must consider the contributions to its velocity from both the linear velocities (u, v, w) in each of the coordinate directions and the contributions due to the rotational rates (p, q, r) about the axes (Fig. 1). Therefore, the time rates of change of the coordinates in an inertial frame that is instantaneously coincident with the body axes are:

$$\begin{aligned}\dot{x} &= u + qz - ry, \\ \dot{y} &= v + rx - pz, \\ \dot{z} &= w + py - qx.\end{aligned}$$

The linearized equations are derived from Caughey et al. [12]. In this way, for the longitudinal control, the equation is given by:

$$\begin{bmatrix} \dot{u} \\ \dot{\omega} \\ \dot{q} \\ \dot{\theta} \end{bmatrix} = \mathbf{F}_{\text{LO}} \begin{bmatrix} u \\ \omega \\ q \\ \theta \end{bmatrix} + \mathbf{G}_{\text{LO}} \begin{bmatrix} \delta_e \\ \delta_T \end{bmatrix}, \quad (1)$$

where

$$\mathbf{F}_{\text{LO}} = \begin{bmatrix} X_u & X_\omega & 0 & -g \\ Z_u & Z_\omega & u_0 & 0 \\ M_u + M_{\dot{\omega}}Z_u & M_\omega + M_{\dot{\omega}}Z_\omega & M_q + M_{\dot{\omega}}u_0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix}, \quad (2)$$

and

$$\mathbf{G}_{\text{LO}} = \begin{bmatrix} X_e & X_T \\ Z_e & Z_T \\ M_e + M_{\dot{\omega}}Z_e & M_T + M_{\dot{\omega}}Z_T \\ 0 & 0 \end{bmatrix}. \quad (3)$$

Here, the state vector is $\mathbf{x}_{\text{LO}} = [u \ \omega \ q \ \theta]^T$ and the longitudinal control vector (elevator and throttle) is $\eta = [\delta_e \ \delta_T]^T$. The orientation of the body is determined by (β, θ, ϕ) , with β representing the yaw rotation about the Z-axis, θ the pitch rotation about the Y-axis, and ϕ the roll rotation about the X-axis.

On the other hand, the equation for the lateral control is:

$$\begin{bmatrix} \dot{\beta} \\ \dot{p} \\ \dot{r} \\ \dot{\phi} \end{bmatrix} = \mathbf{F}_{\text{LA}} \begin{bmatrix} \beta \\ p \\ r \\ \phi \end{bmatrix} + \mathbf{G}_{\text{LA}} \begin{bmatrix} \delta_a \\ \delta_r \end{bmatrix}, \quad (4)$$

where

$$\mathbf{F}_{\text{LA}} = \begin{bmatrix} Y_\beta/u_0 & Y_p/u_0 & -(1 - Y_r/u_0) & -g/u_0 \\ L_\beta & L_p & L_r & 0 \\ N_\beta & N_p & N_r & 0 \\ 0 & 1 & 0 & 0 \end{bmatrix}, \quad (5)$$

and

$$\mathbf{G}_{\text{LA}} = \begin{bmatrix} 0 & Y_r/u_0 \\ L_a & L_r \\ N_a & N_r \\ 0 & 0 \end{bmatrix}. \quad (6)$$

Here, the state vector is $\mathbf{x}_{\text{LA}} = [\beta \ p \ r \ \phi]^T$ and the lateral control vector (aileron and rudder) is $\eta = [\delta_a \ \delta_r]^T$.

3 PI-Filter compensation

The non-zero point regulator is designed under the assumption that the system to be controlled is modeled without error and that any system disturbances are white random processes. However these conditions are violated because of slowly varying disturbances of uncertain magnitude, that makes the basic LQ regulation inadequate. Therefore there is the need to increase system robustness by providing dynamic compensation

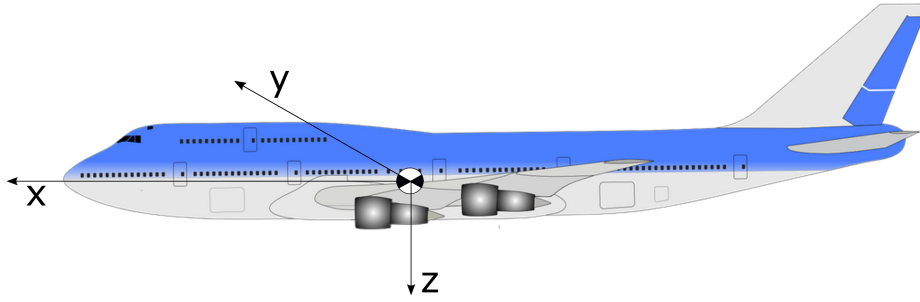


Fig. 1 The body axis system is centered at the center of gravity of the flight vehicle. The y-axis extends out towards the right wing.

which can be accomplished by adding new states to the closed-loop system. The state vector is augmented, corresponding differential equations are added to the system model and the control law that minimizes a quadratic cost function is computed for the augmented systems. In particular the proportional-integral compensation introduces command-error integrals to the LQ control law. The systems to be controlled is described by the linear, time-invariant model

$$\dot{\mathbf{x}}(t) = \mathbf{F}\mathbf{x}(t) + \mathbf{G}\mathbf{u}(t) + \mathbf{L}\mathbf{w}(t),$$

$$\mathbf{y}(t) = \mathbf{H}_x\mathbf{x}(t) + \mathbf{H}_u\mathbf{u}(t) + \mathbf{H}_w\mathbf{w}(t).$$

It is assumed that $\mathbf{F}$, $\mathbf{G}$, $\mathbf{H}_x$, $\mathbf{H}_u$, $\mathbf{L}$, and $\mathbf{H}_w$ are known without error and are a generalization of the lateral and longitudinal linearization.

The equilibrium of the system is reached when $\dot{\mathbf{x}}(t) = \mathbf{0}$. Therefore, we can represent the state system equations as follows

$$\begin{bmatrix} \mathbf{0} \\ \mathbf{y}^* \end{bmatrix} = \begin{bmatrix} \mathbf{F} & \mathbf{G} \\ \mathbf{H}_x & \mathbf{H}_u \end{bmatrix} \begin{bmatrix} \mathbf{x}^* \\ \mathbf{u}^* \end{bmatrix} + \begin{bmatrix} \mathbf{L} \\ \mathbf{H}_w \end{bmatrix} \mathbf{w}^*,$$

which can also be written as

$$\begin{bmatrix} \mathbf{0} - \mathbf{L}\mathbf{w}^* \\ \mathbf{y}^* - \mathbf{H}_w\mathbf{w}^* \end{bmatrix} = \begin{bmatrix} \mathbf{F} & \mathbf{G} \\ \mathbf{H}_x & \mathbf{H}_u \end{bmatrix} \begin{bmatrix} \mathbf{x}^* \\ \mathbf{u}^* \end{bmatrix}.$$

If the variables $\mathbf{x}^*$ and $\mathbf{u}^*$ are solved, then:

$$\begin{bmatrix} \mathbf{x}^* \\ \mathbf{u}^* \end{bmatrix} = \begin{bmatrix} \mathbf{F} & \mathbf{G} \\ \mathbf{H}_x & \mathbf{H}_u \end{bmatrix}^{-1} \begin{bmatrix} \mathbf{0} - \mathbf{L}\mathbf{w}^* \\ \mathbf{y}^* - \mathbf{H}_w\mathbf{w}^* \end{bmatrix}. \quad (7)$$

Let's define:

$$\mathbf{A} = \begin{bmatrix} \mathbf{F} & \mathbf{G} \\ \mathbf{H}_x & \mathbf{H}_u \end{bmatrix}$$

and

$$\mathbf{B} = \begin{bmatrix} \mathbf{B}_{11} & \mathbf{B}_{12} \\ \mathbf{B}_{21} & \mathbf{B}_{22} \end{bmatrix} = \mathbf{A}^{-1};$$

then, the equation (7) can be written as:

$$\begin{bmatrix} \mathbf{x}^* \\ \mathbf{u}^* \end{bmatrix} = \begin{bmatrix} \mathbf{B}_{11} & \mathbf{B}_{12} \\ \mathbf{B}_{21} & \mathbf{B}_{22} \end{bmatrix} \begin{bmatrix} \mathbf{0} - \mathbf{L}\mathbf{w}^* \\ \mathbf{y}^* - \mathbf{H}_w\mathbf{w}^* \end{bmatrix}.$$

The reference values $\mathbf{x}^*$ and $\mathbf{u}^*$, dependent on the desired input $\mathbf{y}^*$ (Fig. 2), are

$$\mathbf{x}^* = -\mathbf{B}_{11}\mathbf{L}\mathbf{w}^* + \mathbf{B}_{12}(\mathbf{y}^* - \mathbf{H}_w\mathbf{w}^*),$$

$$\mathbf{u}^* = -\mathbf{B}_{21}\mathbf{L}\mathbf{w}^* + \mathbf{B}_{22}(\mathbf{y}^* - \mathbf{H}_w\mathbf{w}^*);$$

with

$$\mathbf{B}_{11} = \mathbf{F}^{-1}(-\mathbf{G}\mathbf{B}_{21} + \mathbf{I}_n),$$

$$\mathbf{B}_{12} = -\mathbf{F}^{-1}\mathbf{G}\mathbf{B}_{22},$$

$$\mathbf{B}_{21} = -\mathbf{B}_{22}\mathbf{H}_x\mathbf{F}^{-1},$$

$$\mathbf{B}_{22} = (-\mathbf{H}_x\mathbf{F}^{-1}\mathbf{G} + \mathbf{H}_u)^{-1}.$$

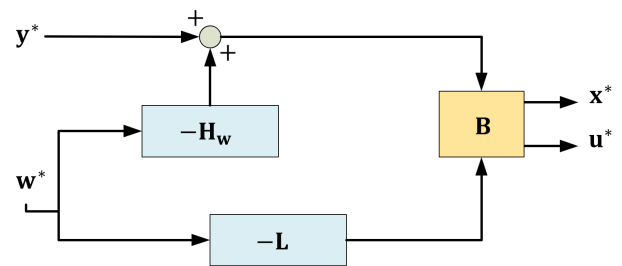


Fig. 2 Reference values depending on the desired inputs

The quadratic cost function for the Proportional-Integral-Filter Compensation is

$$J = \lim_{T \rightarrow \infty} \frac{1}{2T} \int_0^T \left\{ \begin{bmatrix} \tilde{\mathbf{x}}^T(t) & \tilde{\mathbf{u}}^T(t) & \xi^T(t) \end{bmatrix} \mathbf{\Gamma} \begin{bmatrix} \tilde{\mathbf{x}}(t) \\ \tilde{\mathbf{u}}(t) \\ \xi(t) \end{bmatrix} + \mathbf{v}^T(t) \mathbf{R}_2 \mathbf{v}(t) \right\} dt, \quad (8)$$

with

$$\Gamma = \begin{bmatrix} \mathbf{Q}_1 & \mathbf{M} & \mathbf{0} \\ \mathbf{M}^T & \mathbf{R}_1 & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{Q}_2 \end{bmatrix},$$

where $\mathbf{R}_1$, $\mathbf{M}$, $\mathbf{Q}_1$, and $\mathbf{Q}_2$ are gain matrices.

This equation may be reformulated considering

$$\dot{\tilde{\mathbf{x}}}(t) = \mathbf{F}\tilde{\mathbf{x}}(t) + \mathbf{G}\tilde{\mathbf{u}}(t),$$

$$\dot{\tilde{\mathbf{u}}}(t) = \dot{\mathbf{u}}_C(t) \triangleq \mathbf{v}(t),$$

and adding the integral-state vector to this, the augmented state equation can be formed

$$\begin{bmatrix} \dot{\tilde{\mathbf{x}}}(t) \\ \dot{\tilde{\mathbf{u}}}(t) \\ \dot{\xi}(t) \end{bmatrix} = \begin{bmatrix} \mathbf{F} & \mathbf{G} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{H}_x & \mathbf{H}_u & \mathbf{0} \end{bmatrix} \begin{bmatrix} \tilde{\mathbf{x}}(t) \\ \tilde{\mathbf{u}}(t) \\ \xi(t) \end{bmatrix} + \begin{bmatrix} \mathbf{0} \\ \mathbf{I}_m \\ \mathbf{0} \end{bmatrix} \mathbf{v}(t); \quad (9)$$

where

$$\tilde{\mathbf{u}}(t) = \mathbf{u}(t) - \mathbf{u}^*,$$

$$\tilde{\mathbf{x}}(t) = \mathbf{x}(t) - \mathbf{x}^*,$$

$$\xi(t) = \xi(0) + \int_0^t (\mathbf{y}(\tau) - \mathbf{y}^*) d\tau.$$

The augmented state equation (9) can also be written as

$$\begin{bmatrix} \dot{\chi}(t) \end{bmatrix} = \begin{bmatrix} \mathbf{F} & \mathbf{G} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{H}_x & \mathbf{H}_u & \mathbf{0} \end{bmatrix} \chi(t) + \begin{bmatrix} \mathbf{0} \\ \mathbf{I}_m \\ \mathbf{0} \end{bmatrix} \mathbf{v}(t),$$

with

$$\chi(t) \triangleq \begin{bmatrix} \tilde{\mathbf{x}}(t) \\ \tilde{\mathbf{u}}(t) \\ \xi(t) \end{bmatrix}.$$

The cost function in (8) then becomes

$$J = \lim_{T \rightarrow \infty} \frac{1}{2T} \int_0^T [\chi^T(t) \mathbf{Q}' \chi(t) + \mathbf{v}^T(t) \mathbf{R}' \mathbf{v}(t)] dt,$$

which leads to a control law of the form

$$\mathbf{v}(t) = -\mathbf{C}\chi(t)$$

or

$$\mathbf{v}(t) = -\mathbf{C}_1 \tilde{\mathbf{x}}(t) - \mathbf{C}_2 \tilde{\mathbf{u}}(t) - \mathbf{C}_3 \xi(t).$$

This equation is equivalent to

$$\begin{aligned} \dot{\mathbf{u}}(t) = & -\mathbf{C}_1 [\mathbf{x}(t) - \mathbf{x}^*] - \mathbf{C}_2 [\mathbf{u}(t) - \mathbf{u}^*] \\ & - \mathbf{C}_3 \left\{ \xi(0) + \int_0^t (\mathbf{y}(\tau) - \mathbf{y}^*) d\tau \right\}, \end{aligned} \quad (10)$$

with

$$\mathbf{y}^* = \mathbf{H}_x \mathbf{x}^* + \mathbf{H}_u \mathbf{u}^*.$$

The control law (10) can be rearranged as

$$\begin{aligned} \dot{\mathbf{u}}(t) = & (\mathbf{C}_1 \mathbf{B}_{12} + \mathbf{C}_2 \mathbf{B}_{22}) \mathbf{y}^* - \mathbf{C}_1 \mathbf{x}(t) - \mathbf{C}_2 \mathbf{u}(t) \\ & - \mathbf{C}_3 \left\{ \xi(0) + \int_0^t (\mathbf{y}(\tau) - \mathbf{y}^*) d\tau \right\} \end{aligned} \quad (11)$$

and

$$\begin{aligned} \dot{\mathbf{u}}(t) = & \mathbf{C}_F \mathbf{y}^* - \mathbf{C}_B \mathbf{x}(t) - \mathbf{C}_C \mathbf{u}(t) \\ & - \mathbf{C}_I \left\{ \xi(0) + \int_0^t (\mathbf{y}(\tau) - \mathbf{y}^*) d\tau \right\}, \end{aligned} \quad (12)$$

with $\mathbf{C}_F = \mathbf{B}_{22} + \mathbf{C}_1 \mathbf{B}_{12}$, $\mathbf{C}_B = \mathbf{C}_1$, $\mathbf{C}_C = \mathbf{C}_2$, $\mathbf{C}_I = \mathbf{C}_3$.

The implementation of the complete controller is visualized in Fig. 3, where the connection of the control outputs to FlightGear is shown, with the matrix calculations being performed in Matlab.

4 Results

4.1 Intercommunication

Based on the work of Aschauer et al. [13], we have programmed a UDP-based communication tunnel to exchange information between the mathematical software and the flight simulator (see Fig. 5) through an information frame. The parameters for obtaining the linearized model are taken from:

F-104 Starfighter parameters

<http://www.gnu-darwin.org/ProgramDocuments/f104/linear.html>

Additionally, the software is executed with the following command:

Command to execute FlightGear

```
C:\ProgramFiles\FlightGear_2017.3.1\bin\fgfs
-aircraft=F-104
-start-date-lat=2004:06:01:09:00:00
-generic=
socket,out,20,localhost,2054,udp,readUDP
-generic=
socket,in,20,localhost,2055,udp,writeUDP
```

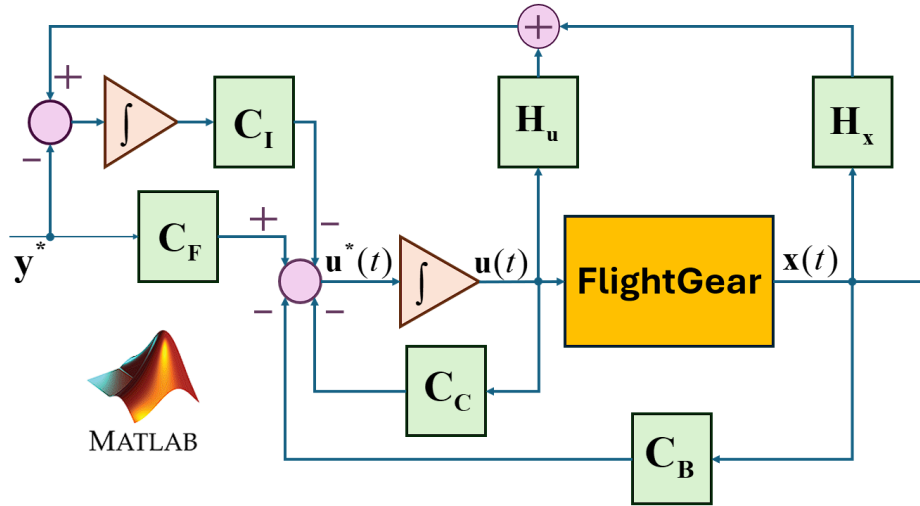


Fig. 3 Proportional-integral (PI) regulator for nonsingular command vector

where the aircraft type is specified (in our case, the F-104 Starfighter), along with the simulation start date and time, and the UDP ports for information exchange. Finally, the controllers defined in Section 3 are programmed in Simulink, as shown in Fig. 4.

4.2 States and control errors

The intercommunication between programs allows reading the states of the simulated aircraft in Matlab, while the control actions are reflected in FlightGear. Thus, we plot all states of both longitudinal and lateral behavior in Figures 6 and 7, respectively. For the longitudinal behavior, we assume a constant throttle, while setting an elevation of 20 degrees, starting from an initial elevation of -20 degrees. On the other hand, for the lateral behavior, both pitch and roll need to be controlled. Therefore, the roll starts from an angle of 5 degrees, with a desired roll of 20 degrees. Similarly, the pitch starts from an angle of 10 degrees and needs to reach 5 degrees. Both Figure 6 and Figure 7 demonstrate the trend of errors approaching zero.

5 Conclusions

In conclusion, this study focuses on the development and implementation of control techniques for fixed-wing aircraft, leveraging flight simulators as a primary tool for algorithm validation and testing. Key methodologies such as LQG control and PI filter compensation are employed to enhance control smoothness and efficiency. The integration of UDP-based communication tunnels facilitates seamless data exchange between the

flight simulator and mathematical software like Matlab, crucial for real-time control and simulation. Results presented demonstrate the longitudinal and lateral behavior of the controlled system, indicating a trend towards minimal control errors.

Appendix: Tables with constants required for linearization

Table 1 Constants for Longitudinal-Directional System

Parameter	Value
Stability Derivative	$X_u = -0.0093$
Angle of Attack Deriv.	$X_w = -0.0253$
Stability Derivative	$Z_u = -0.0236$
Angle of Attack Derivative	$Z_w = -0.1982$
Gravity in Slugs	$g = 32.174$
Initial vel.	$u_0 = 1740.81$
Compressibility Effect Deriv.	$M_u = 0.0$
Elev. Deflection	$X_e = 0.0$
Dimensional Pitching Mom. Deriv.	$M_w = -0.0104$
Dimensional Pitching Mom. Deriv.	$M_{\dot{w}} = 0.0$
Dimensionless Pitching Mom. Deriv.	$M_q = -0.1845$
Thrust Deflection	$X_T = 0$
Thrust Deflection	$Z_T = 0$
Pitching Mom. (Thrust Deflection)	$M_T = 0$
Pitching Mom. (Elevator Deflection)	$M_e = -18.1525$
Elevator Deflection	$Z_e = -87.9155$

Conflict of interest

The authors declare that they have no conflict of interest.

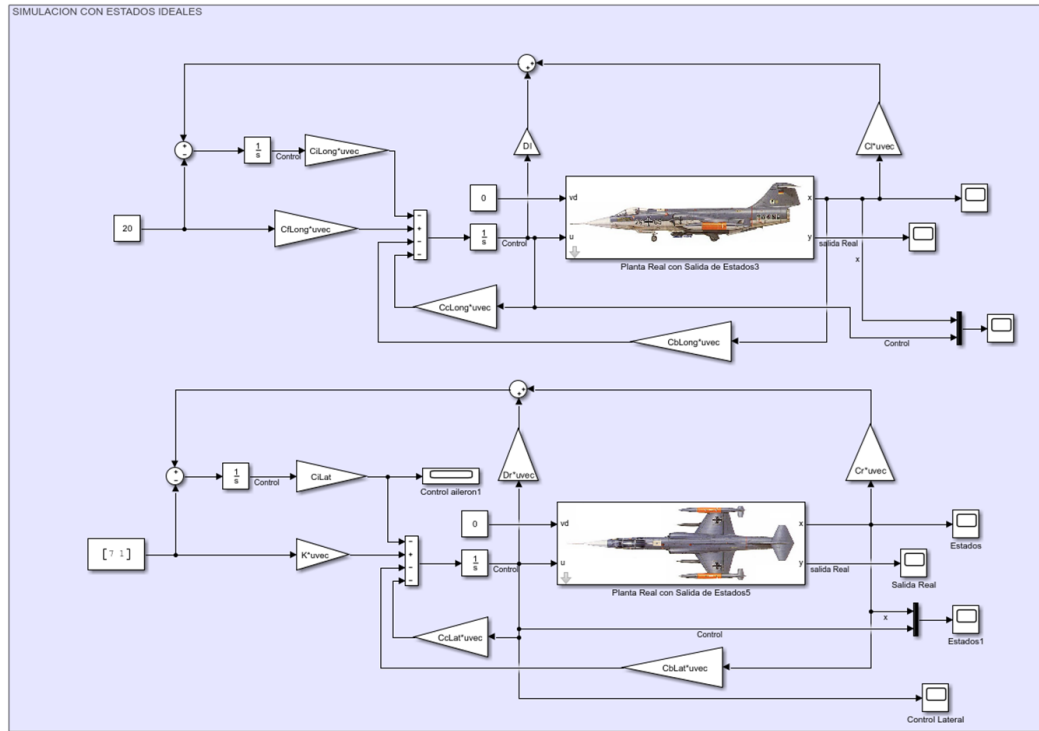


Fig. 4 Programming of the controller in Simulink

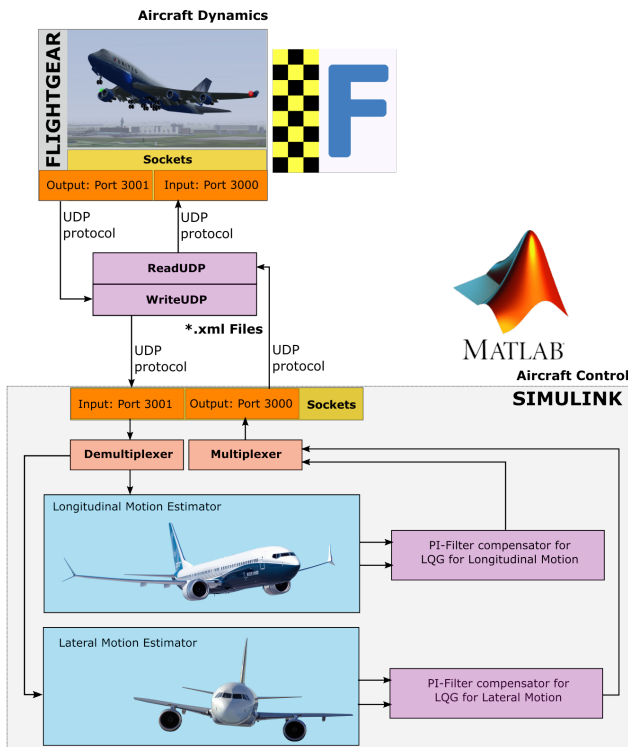


Fig. 5 Connection between Matlab and FlightGear

References

1. B. Sarlioglu and C. T. Morris, "More electric aircraft: Review, challenges, and opportunities for commercial

Table 2 Constants for Lateral–Directional System

Parameter	Value
Roll Rate	$Y_p = 0.0$
Sideslip Derivative	$Y_\beta = -175.6628$
Yaw Rate Derivative	$Y_r = 0$
Aileron Deflection Derivative	$Y_a = 0$
Rolling Moment	$L_p = -0.8864$
Rolling Moment	$L_r = 4.0927$
Rolling Moment	$L_a = -63.6874$
Roll Acceleration	$L_\beta = -48.1804$
Yawing Moment	$N_a = -0.0777$
Yawing Moment	$N_p = -0.0182$
Yawing Moment	$N_r = -1.3522$
Yaw Acceleration	$N_\beta = 7.5224$

transport aircraft," *IEEE transactions on Transportation Electrification*, vol. 1, no. 1, pp. 54–64, 2015.

2. S. G. Gupta, D. M. Ghonge, and P. M. Jawandhiya, "Review of unmanned aircraft system (uas)," *International Journal of Advanced Research in Computer Engineering & Technology (IJARCET) Volume*, vol. 2, 2013.
3. B. J. Brelje and J. R. Martins, "Electric, hybrid, and turboelectric fixed-wing aircraft: A review of concepts, models, and design approaches," *Progress in Aerospace Sciences*, vol. 104, pp. 1–19, 2019.
4. D. B. Barber, J. D. Redding, T. W. McLain, R. W. Beard, and C. N. Taylor, "Vision-based target geo-location using a fixed-wing miniature air vehicle," *Journal of Intelligent and Robotic Systems*, vol. 47, pp. 361–382, 2006.
5. J. Veltman, "A comparative study of psychophysiological reactions during simulator and real flight," *The Interna-*

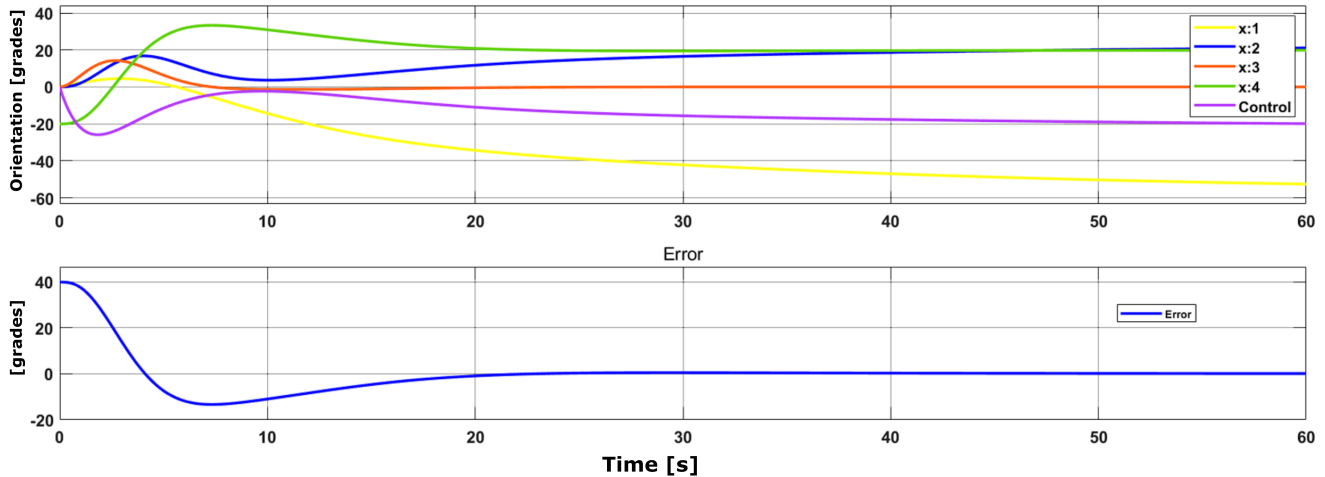


Fig. 6 Longitudinal states values and control error

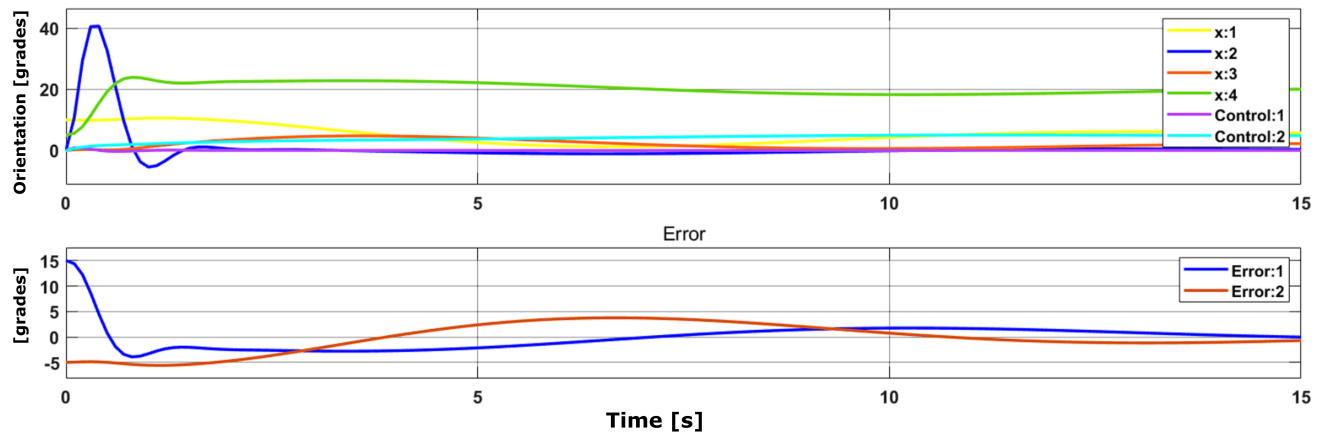


Fig. 7 Lateral states values and control errors

- tional Journal of Aviation Psychology*, vol. 12, no. 1, pp. 33–48, 2002.
6. A. Nisansala, M. Weerasinghe, G. Dias, D. Sandaruwan, C. Keppitiyagama, N. Kodikara, C. Perera, and P. Samarasinghe, “Flight simulator for serious gaming,” in *Information Science and Applications*. Springer, 2015, pp. 267–277.
 7. J. Ying, H. Luc, J. Dai, and H. Pan, “Visual flight simulation system based on matlab/flightgear,” in *2017 IEEE 2nd Advanced Information Technology, Electronic and Automation Control Conference (IAEAC)*. IEEE, 2017, pp. 2360–2363.
 8. N. Horri and M. Pietraszko, “A tutorial and review on flight control co-simulation using matlab/simulink and flight simulators,” *Automation*, vol. 3, no. 3, pp. 486–510, 2022.
 9. A. I. Hentati, L. Krichen, M. Fourati, and L. C. Fourati, “Simulation tools, environments and frameworks for uav systems performance analysis,” in *2018 14th International Wireless Communications & Mobile Computing Conference (IWCMC)*. IEEE, 2018, pp. 1495–1500.
 10. E. N. Demirhan, K. Ç. Coşkun, and C. Kasnaoğlu, “Lq control design with lqg regulator via ukf for a fixed-wing aircraft,” in *2020 24th International Conference on System Theory, Control and Computing (ICSTCC)*. IEEE, 2020, pp. 25–30.
 11. S. G. Clarke and I. Hwang, “Deep reinforcement learning control for aerobatic maneuvering of agile fixed-wing aircraft,” in *AIAA Scitech 2020 Forum*, 2020, p. 0136.
 12. D. A. Caughey, “Introduction to aircraft stability and control course notes for m&ae 5070,” *Sibley School of Mechanical & Aerospace Engineering Cornell University*, vol. 15, 2011.
 13. G. Aschauer, A. Schirrer, and M. Kozek, “Co-simulation of matlab and flightgear for identification and control of aircraft,” *IFAC-PapersOnLine*, vol. 48, no. 1, pp. 67–72, 2015.

License

Copyright (2024) © María del Carmen Claudio, Alida Ortiz, Gloria Chicaiza-Claudio.

This text is protected under an international Creative Commons 4.0 license.



You are free to share, copy, and redistribute the material in any medium or format — and adapt the document — remix, transform, and build upon the material — for any purpose, even commercially, provided you comply with the conditions of Attribution. You must give appropriate credit to the original work, provide a link to the license, and indicate if changes were made. You may do so in any reasonable manner, but not in a way that suggests endorsement by the licensor or approval of your use of the work.

License summary - Full text of the license