Penerbit UTHM

PAAT

Journal homepage: http://penerbit.uthm.edu.my/ojs/index.php/paat
e-ISSN : 2821-2924

Progress in Aerospace and Aviation Technology

The Design of an Automatic Flight Control System and Dynamic Simulation for Fixed-Wing Unmanned Aerial Vehicle (UAV) using X-Plane and LabVIEW

Nur Ezzyana Ameera Mazlan[1], Syariful Syafiq Shamsudin[1*], Mohammad Fahmi Pairan[1], Mohd Fauzi Yaakub[1], Muhammad Faiz Ramli[2]

[1]Research Center for Unmanned Vehicle, Faculty of Mechanical and Manufacturing Engineering, Universiti Tun Hussein Onn Malaysia, Batu Pahat, 86400, MALAYSIA

[2]Aircraft System and Design Research, Faculty of Mechanical and Manufacturing Engineering, Universiti Tun Hussein Onn Malaysia, Batu Pahat, 86400, MALAYSIA

*Corresponding Author

DOI: https://doi.org/10.30880/paat.2021.01.01.007
Received 24 September 2021; Accepted 11 November 2021; Available online 21 December 2021

Abstract: This research focuses on developing an automatic flight control system for a fixed-wing unmanned aerial vehicle (UAV) using a software-in-the-loop method in which the PID controller is implemented in National Instruments LabVIEW software and the flight dynamics of the fixed-wing UAV are simulated using the X-Plane flight simulator. The fixed-wing UAV model is created using the Plane Maker software and is based on existing geometry and propulsion data from the literature. Gain tuning for the PID controller is accomplished using the pole placement technique. In this approach, the controller gain can be calculated using the dynamic parameters in the transfer function model and the desired characteristic equation. The proposed controller designs' performance is validated using attitude, altitude, and velocity hold simulations. The results demonstrate that the technique can be an effective tool for researchers to validate their UAV control algorithms by utilising the realistic UAV or manned aircraft models available in the X-Plane flight simulator.

Keywords: Flight simulation, PID Controller, Fixed-Wing UAV, Software-In-The-Loop Method

1. Introduction

An unmanned aerial vehicle (UAV) can be defined as an aircraft designed with no pilot on-board that can perform various roles of piloted aircraft. A complete UAV system includes the UAV platform, ground control station (GCS), payload and communication system which also known as the Unmanned Aerial System (UAS). The UAV can be categorized into different platforms such as the fixed-wing UAV, rotary-wing UAV and hybrid UAV. Typically, the fixed-wing aircraft have been preferred as the UAV platforms simply because of their simple structures, efficient in terms of aerodynamic design and easy to build and maintain. The autopilot design for the fixed-wing UAV is much easier to design because the fixed-wing aircraft have relatively simple, symmetric and decoupled dynamics.

The increased number of UAVs usage is primarily driven by the ease of use, low maintenance cost and the high maneuverability of the aircraft. The UAV have been widely used in numerous military and civilian applications such as the surveillance, aerial mapping, disaster prevention, border protection, cinematography or safety support missions [1], [2]. Automation of these applications requires development of an automatic flight control system (AFCS) with path/velocity tracking and attitude stabilization capabilities [3]. The UAV is typically equipped with the on-board

*Corresponding author: syafiq@uthm.edu.my

autopilot system that enables a UAV to perform the above mentioned missions automatically with lower accident risks and higher confidence in mission success.

This paper focuses on the design and verification of the control law developed for a fixed-wing UAV through the application of the Software-In-the-Loop (SITL) simulation. SITL simulation is used to predict the flight response of the UAV, and the measured flight data obtained during simulation is used to validate the performance of control algorithms. The study will describe all the steps that are needed to implement an SITL simulation, where the model of the aircraft runs on X-plane, and the control and guidance algorithm runs on LabVIEW.

The technology of SITL simulation plays a vital role in autopilot design. The standard standalone personal computer is used to simulate the plant's behavior in SITL simulation concept. The SITL simulator environment allows researchers to develop, evaluate, and validate multi-aircraft flight guidance and control algorithms using realistic unmanned aircraft models in real-world modeled environments through the application of commercial flight simulator such as X-Plane. The flight data and aircraft control signals can be easily monitored and such features would improve our insight into critical factors that affect flight performance [4].

The Proportional-Integral-Derivative (PID) controller is widely used in the closed-loop control mechanism. The conventional method used to tune a PID controller is typically based on the trial-and-error method, which involves manually adjusting the controller gain values without any prior knowledge of the dynamic system behavior. The approach can be dangerous, as the incorrect gain values can be accidentally used during the tuning process. There are various examples of trial-and-error method for finding PID gains (K_P, K_I, K_D) such as the Ziegler-Nichols (Z-N) oscillation method and the Cohen-Coon tuning method [5]–[7]. The controller development strategy for the UAV system can be improved and simplified by using the UAV dynamic model and the desired specification of the closed-loop system. In this paper, the PID controller tuned using the Pole Placement method will be proposed to accelerate the controller tuning process.

This research project aims to develop an SITL simulator for fixed-wing UAV's automatic flight control system (AFCS) design. The performance and effectiveness of the proposed controllers are evaluated in a series of simulated flight tests such as altitude hold, velocity hold, and attitude hold maneuver. The objectives of this research work are to develop a fixed-wing UAV model using the Plane Maker software according to the literature's established geometry and propulsion data and to propose a suitable automatic flight control system design by Pole Placement method to control the fixed- wing UAV in specific value of altitude hold, velocity hold, and attitude hold modes.

2. Methodology

In this section, the SITL simulation setup for testing and validation of fixed wing UAV control algorithm is described in detailed. The execution of the entire UAV control algorithm is shown through a realistic flight simulator environment known as X-Plane. The control algorithm is coded in LabVIEW, which schedules each task efficiently.

2.1 Aerial Platform

The unmanned aerial vehicle (UAV) platform which was used in this research is a conventional electric model fixed-wing aircraft known as MS Composit Maxi Swift. The Maxi is a direct descendent of the Swift and the Swift II but with larger wings, thicker airfoil and overall size. Fig. 1 shows the Maxi Swift RC model that will be constructed in Plane Maker software. The mass, geometry, propulsion, and aerodynamic parameters for the Maxi Swift are given in the Table 1 below.

Fig. 1 - Maxi Swift RC model

Table 1 - Geometry parameters and aerodynamic coefficients of Maxi Swift UAV [8]

Geometry Parameter	Value	Longitudinal Coefficient	Value	Lateral Coefficient	Value
m	1.56 kg	C_{L_0}	0.09167	C_{Y_o}	0
J_x	0.1147 kg m^2	C_{D_0}	0.01631	C_{l_0}	0
J_y	0.0576 kg m^2	C_{m_0}	-0.02338	C_{m_0}	0
J_z	0.1712 kg m^2	C_{L_α}	3.5016	C_{Y_β}	-0.07359
J_{xz}	0.0015 kg m^2	C_{D_α}	0.2108	C_{l_β}	-0.02854
S	0.2589 m^2	C_{m_α}	-0.5675	C_{n_β}	-0.00040
b	1.4224 m	C_{L_q}	2.8932	C_{Y_p}	0
c	0.3302 m	C_{D_q}	0	C_{l_p}	-0.3209
S_{prop}	0.0314 m^2	C_{m_q}	-1.399	C_{n_p}	-0.01297
ρ	1.2682 kg/m^3	$C_{L_{\delta e}}$	0.2724	C_{Y_r}	0
k_{motor}	20	$C_{D_{\delta e}}$	0.3045	C_{l_r}	0.03066
k_{T_p}	0	$C_{m_{\delta e}}$	-0.3254	C_{n_r}	-0.00434
k_Ω	0	C_{prop}	1.0	$C_{Y_{\delta a}}$	0
e	0.9	M	50	$C_{l_{\delta a}}$	0.1682
		α_0	0.4712	$C_{n_{\delta a}}$	-0.00328
		ϵ	0.1592		
		C_{D_p}	0.0254		

2.2 SITL Simulation

Flight tests were conducted in the simulation to validate the proposed control algorithm by assessing the UAV's flight performance. The main components of the test platform are LabVIEW software running the autopilot control system, and X-Plane software that responsible for simulating the response of the UAV model. The X-Plane graphics model and LabVIEW autopilot simulation are shown in Fig. 2. Both softwares will run on the same computer using the computer network card's IP address "127.0.0.1". The X-Plane is chosen as a flight simulator because of its ability to predict the aircraft's response and flying qualities with high accuracy. X-Plane flight simulator also has a comprehensive database of aircraft models and was certified by the Federal Aviation Administration (FAA) to be used as training devices for pilot training.

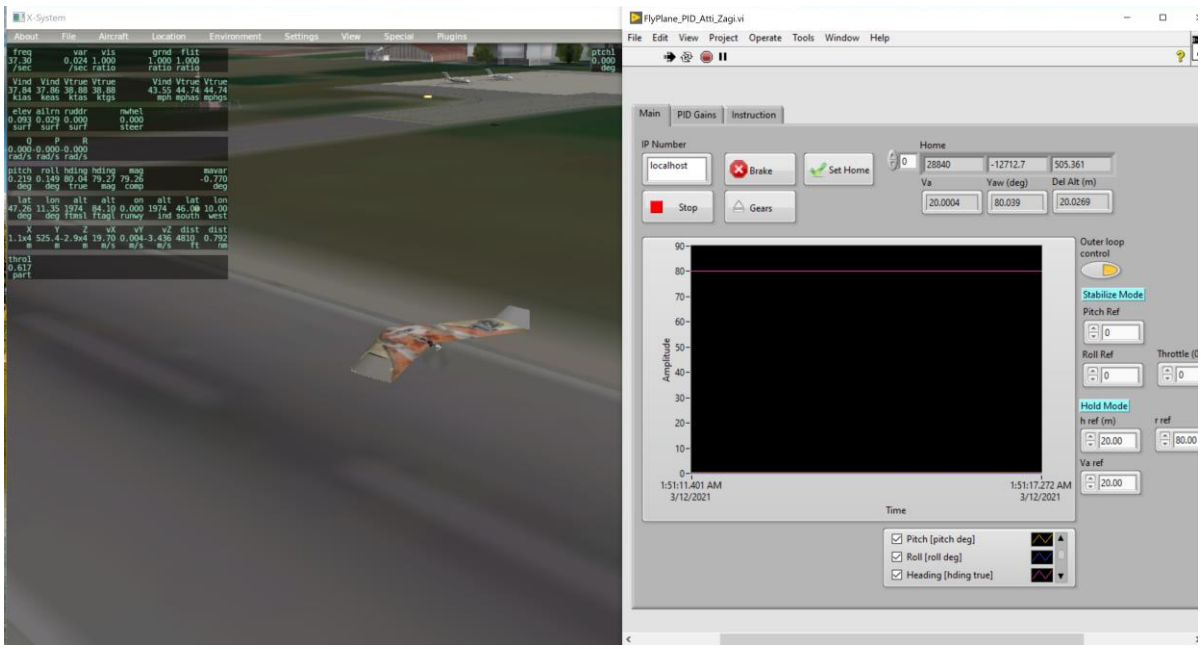


Fig. 2 - X-Plane Maxi Swift RC model with LabVIEW simulation front panel

The X-Plane software predicts the force acting on the aircraft using Blade Element Theory [9], [10] over several times per second as compared to other flight simulator such FlightGear that uses Stability Derivative Method [11], [12]. Based on the mass of the aircraft and center of gravity, the forces acting on the aircraft will be converted into accelerations, which are integrated to generate velocities and positions. The principle of operation of X-Plane is based on reading the geometric shape of any aircraft and before predicting the aircraft response. User can design their own aircraft by using Plane Maker program which was bundled together with X‑Plane installation. Once all the physical specifications of the aircraft have been entered (e.g., weight, engine power, wingspan, wing area, control surfaces, and the center of gravity), the X‑Plane simulator will predict aircraft response.

X-Plane can communicate directly with external software or machines via User Datagram Protocol (UDP). The UDP protocol is well suited for flight dynamic simulation, as the communication protocol is extremely fast with no guarantee of data delivery, error detection, or error correction. In this study, the UDP interface between LabVIEW and X-Plane is used in two directions where X-Plane accepts control signals to drive actuators and outputs flight information such as UAV's position, linear velocity, Euler angles and angular velocity. On the other hand, LabVIEW software will send actuator commands such as elevator, aileron, rudder and throttle input signals computed from the proposed flight controller [13]. The input data and flight sensor information can be selected according to user selection as shown in Fig. 3. Note that X-Plane can produce all the navigation data necessary to perform the simulation and allows users to adjust the update rate from 1 to 99 Hz. Once the flight sensor data and input data have been selected on X-Plane, we can transmit and receive control signals and flight data using the standard data packet format of X-Plane [14]–[16].

.

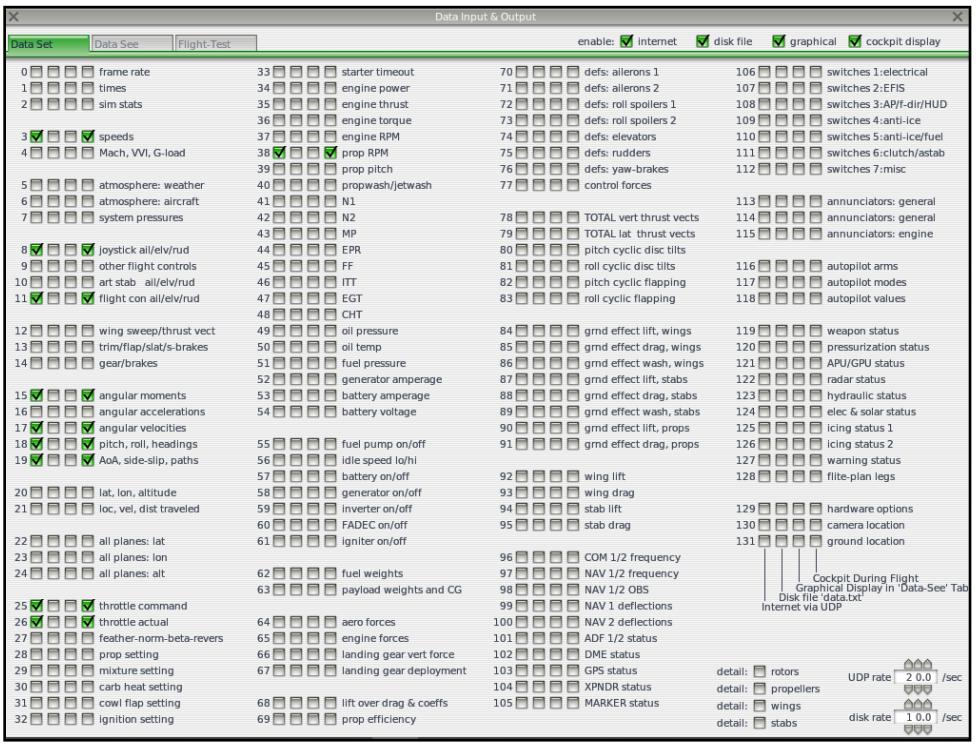


Fig. 3 - Selected data input and output produced by X-Plane

2.3 Flight Controller Structure

The flight controller structure can be partitioned into smaller control subsystems using the cascaded control approach where the complete fixed-wing UAV control problem is decomposed into cascaded loops as shown in Fig. 4. The main function of the inner loop control is to stabilize the UAV attitude by controlling the actuator. The inner loop control receives reference signals from the outer control loop, which operates at a slower rate compared with the inner loop's sampling rate. The outer loop control handles guidance and generation of attitude commands for the inner loop control.

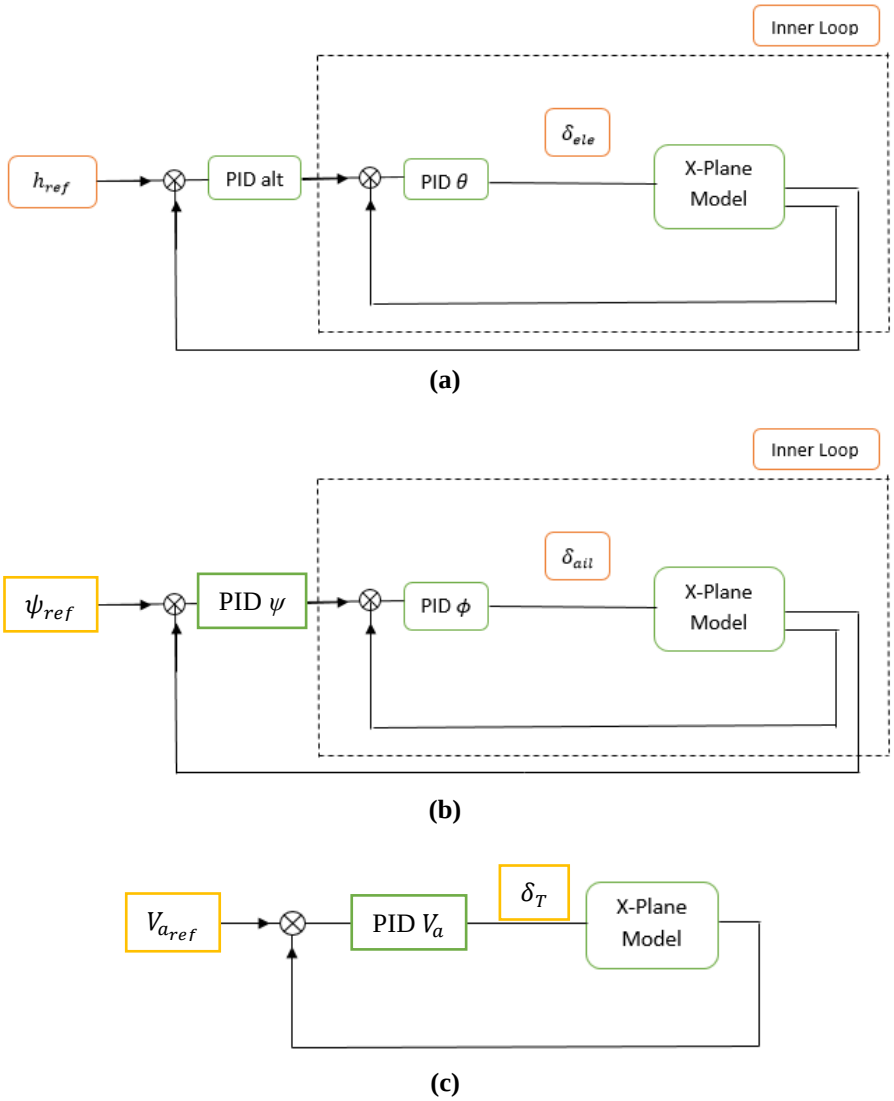


Fig. 4 – The overall flight controller structure used in the SITL simulation. (a) Altitude and pitch angle controller (b) Heading and roll angle controller (c) Airspeed controller.

2.4 Flight Controller Design

The proposed flight controller is based on PID controller design using pole placement design technique [17]. In this approach, the controller gain can be calculated using the dynamic parameters in transfer function and desired characteristic equation. The basic requirement when using pole placement design technique requires that the plant transfer function model be available. The transfer function models used in the design of PID controllers are limited for the first order or second-order model only. If the plant is consists of a higher order model, an approximation is often involved to obtain a first order or a second-order model so that a PID controller can be designed using pole-placement design approach [8], [18]. For example, if the plant model is a first-order model, then a PI controller is used for the feedback system, and if it is a second-order model, then a PID or PD controller is used.

When using pole-placement based design methods, a desired closed-loop performance specification is required in order to calculate the controller gain. The desired performance is chosen in terms of the desired characteristic equation that we want for the closed loop system. The desired closed-loop characteristic equation is often adjusted several times using closed-loop simulation or experimental validation before the designer finds the suitable closed-loop performance. Example of the PI controller design is given in the remaining of the section.

Consider a first order transfer function given by:

$$G(s) = \frac{b}{s + a} \tag{1}$$

and the PI controller is represented as:

$$C(s) = K_P + \frac{K_I}{s} = \frac{K_P s + K_I}{s} \tag{2}$$

where K_P is the proportional gain and K_I is the integral gain. The PI control system is shown in Fig. 5 where $R(s)$, $E(s)$, $U(s)$ and $Y(s)$ represent the Laplace transform variables of reference input, error signal, control signal and output signal, respectively.

The closed loop transfer function of the PI controller system can be expressed as:

$$\frac{Y(S)}{U(s)} = \frac{C(s)G(s)}{1 + C(s)G(s)}$$

$$= \frac{b(K_P s + K_I)}{s^2 + (a + bK_P)s + bK_I} \tag{3}$$

The characteristic equation of the modeled system can be obtained by setting the denominator of the transfer function to zero, which yield:

$$s^2 + (a + bK_P)s + bK_I = 0 \tag{4}$$

The solution to characteristic equation Eq. (4) would produce the actual closed-loop poles of the feedback system. Note that the dynamic model parameters a and b are given in the transfer function while controller gain K_P and K_I is unknown. To calculate the controller gain K_P and K_I, the following polynomial equation is set:

$$s^2 + (a + bK_P)s + bK_I = As^2 + Bs + C \tag{5}$$

where the right-hand side of Eq. (5) is the characteristic polynomial that represents the desired closed-loop poles. By equating these two polynomials, the desired closed-loop poles are assigned to the actual closed-loop poles. This controller design technique is called pole-placement controller design. By comparing the coefficients of the polynomial equation on both sides gives:

$$K_P = \frac{B - a}{b} \tag{6}$$

$$K_I = \frac{C}{b} \tag{7}$$

After assigning the controller gain, we need to further verify whether the control design goals have been achieved. This can be done by simulating the transient response of the feedback control system to confirm that the settling time and maximum overshoot does not exceed the specified requirement. If the dynamic plant under consideration has a second order transfer function model, PD or PID controller design can be applied to the feedback control system. Table 2 lists the PID tuning equation for first order and second order plant used in the study.

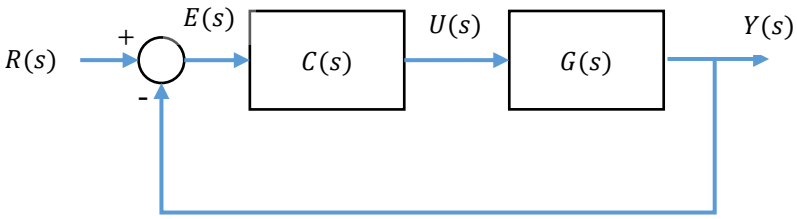


Fig. 5 – Block diagram of PI control system.

Table 2 - PID tuning equation for first order and second order transfer function model

First Order Transfer Function Model $G(s) = \frac{b}{s+a}$ **Controller Selection:** PI **Desired characteristic equation:** $As^2 + Bs + C$	
PI	$K_P = \frac{B-a}{b}$
	$K_I = \frac{C}{b}$
Second Order Transfer Function Model $G(s) = \frac{b}{s^2 + a_1 s + a_2}$ **Controller selection:** PD **Desired characteristic equation:** $As^2 + Bs + C$	
PD	$K_P = \frac{C-a_2}{b}$
	$K_D = \frac{B-a_1}{b}$
Second Order Transfer Function Model $G(s) = \frac{b}{s^2 + a_1 s + a_2}$ **Controller selection:** PID **Desired characteristic equation:** $As^3 + Bs^2 + Cs + D$	
PID	$K_P = \frac{C-a_2}{b}$
	$K_I = \frac{D}{b}$
	$K_D = \frac{B-a_1}{b}$

2.5 Transfer Function Model for Flight Controller Design

This section introduces the transfer function models used to design the flight controller system. The dynamics for fixed-wing aircraft can be approximately decomposed into longitudinal motion, which includes airspeed, pitch angle, and altitude, and into lateral motion, which includes roll angles. While there is coupling between longitudinal and lateral motion, for most airframes, the dynamic coupling is sufficiently small that its unwanted effects can be mitigated by control algorithms designed for disturbance rejection. In this study, we will follow the standard convention and decompose the dynamics into lateral and longitudinal motion.

2.5.1 Lateral Transfer Function Models

For the lateral dynamics, the variables of interest are the roll angle φ, the roll rate p, and the yaw angle ψ. The control surfaces used to influence the lateral dynamics are the aileron, δ_a and the rudder δ_{rud}. The ailerons are primarily used to influence the roll rate, p while the rudder is primarily used to control the yaw angle, ψ of the aircraft [8]. The corresponding lateral transfer function models are given as follows:

Roll Angle Transfer Function:

$$\phi(s) = \left(\frac{a_{\phi 2}}{s\left(s + a_{\phi 1}\right)}\right)\delta_a(s) \tag{8}$$

where,

$$a_{\phi 1} \triangleq -\frac{1}{2}\rho V_a^2 SbC_{p\rho}\frac{b}{2V_a} \tag{9}$$

$$a_{\phi 2} \triangleq -\frac{1}{2}\rho V_a^2 SbC_{p\delta_a} \tag{10}$$

Roll Rate Transfer Function:

$$p(s) = \left(\frac{a_{\phi 2}}{(s + a_{\phi 1})}\right)\delta_a(s) \tag{11}$$

Heading Transfer Function:

$$\psi(s) = \frac{g/V_a}{s}\phi(s) \tag{12}$$

2.5.2 Longitudinal Transfer Function Models

For longitudinal dynamic, variables such as pitch angle, θ, the pitch rate, q, the altitude, h, and the airspeed, V_a was used for the formation of longitudinal transfer function. The control signals used to influence the longitudinal dynamics are the elevator, δ_e and the throttle, δ_t. The corresponding longitudinal transfer function models are given as follows [8]:

Pitch Angle Transfer Function:

$$\theta(s) = \frac{a_{\theta 3}}{s^2 + a_{\theta 1}s + a_{\theta 2}}\delta_e(s) \tag{13}$$

where,

$$a_{\theta 1} \triangleq -\frac{\rho V_{a^2} cS}{2J_y}C_{m_q}\frac{c}{2V_a} \tag{14}$$

$$a_{\theta 2} \triangleq -\frac{\rho V_{a^2} cS}{2J_y}C_{m_\alpha} \tag{15}$$

$$a_{\theta 3} \triangleq \frac{\rho V_{a^2} cS}{2J_y}C_{m_{\delta e}} \tag{16}$$

Pitch Rate Transfer Function:

$$q(s) = (\frac{a_{\theta 3}s}{s^2 + a_{\theta 1}s + a_{\theta 2}})\delta_e(s) \tag{17}$$

Altitude Transfer Function:

$$h(s) = \frac{V_a}{s}\theta(s) \tag{18}$$

Velocity Transfer Function:

$$\bar{V_a}(s) = \frac{a_{v2}}{s + a_{v1}}\delta_t(s) \tag{19}$$

where,

$$a_{v1} = \frac{\rho {V_a}^* S}{2m}\left[C_{D_o} + C_{D_\alpha}\alpha^* + C_{D_{\delta e}}\delta e^*\right] + \frac{\rho S_{prop}}{m}C_{prop}{V_a}^* \tag{20}$$

$$a_{v2} = \frac{\rho S_{prop}}{m}C_{prop}k^2{\delta_t}^* \tag{21}$$

3. Results and Discussion

This section presents the simulation results of the proposed flight controller discussed in the previous section. The developed aircraft model, Maxi Swift and the proposed controllers are imported into the X-Plane simulation environment and LabVIEW respectively, and validated in a series of flight tests to analyze its flight performance characteristics and determine the efficiency of the control systems. In this section, we carry out the flight dynamic simulation for attitude control, altitude control and velocity control.

3.1.1 Attitude Control Results

The closed loop transfer function model of roll and pitch angle dynamics are given in Eq. (13). The value of the closed loop transfer function obtain for pitch angle transfer function are $a_{\theta 1} = 0.20805$ and $a_{\theta 2} = 0.53408$. Since the pitch angle dynamic is a second order system, we consider a PD controller for pitch angle dynamic as shown in Fig. 6. The transfer function models are then compared with the desired characteristic equation, with the value of damping ratio being set as $\xi = 1$ and natural frequency $\omega_n = 1.02$ rad/s. The resulting desired characteristic equation is given as: $s^2 + 2.04s + 1.0404$. Then, from the established PID tuning equation, the value of K_{P_θ} and K_{D_θ} will be obtained for the pitch angle controller.

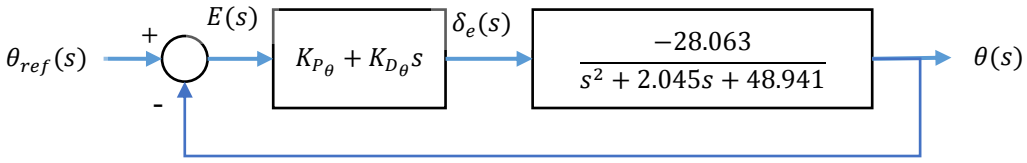


Fig. 6 - Pitch attitude hold feedback loops

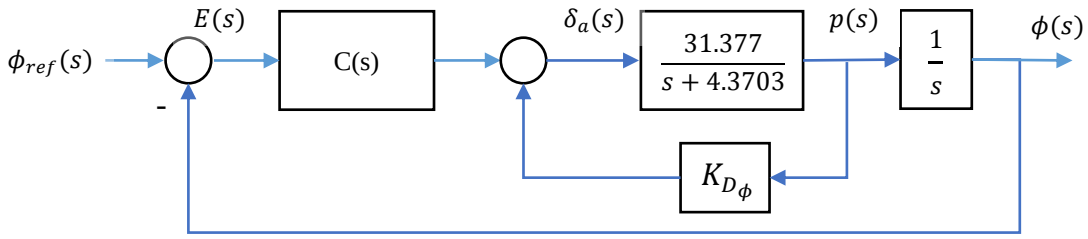
The closed loop transfer function model of roll angle dynamics is given in Eq. (8). The value of the closed loop transfer function obtain for roll angle transfer function are $a_{\phi 1} = 4.3703$, and $a_{\phi 2} = 31.377$. Fig. 7(a) shows roll angle control system with roll rate feedback (inner loop). To simplify the controller design, the inner loop will be transformed into a closed loop transform function and simplify as a DC gain [8]. From Fig. 7(a), the transfer function of the inner loop is given as:

$$\frac{p(s)}{\delta_a(s)} = \frac{31.377}{s + 4.3703 + 31.377K_{D_\phi}} \tag{22}$$

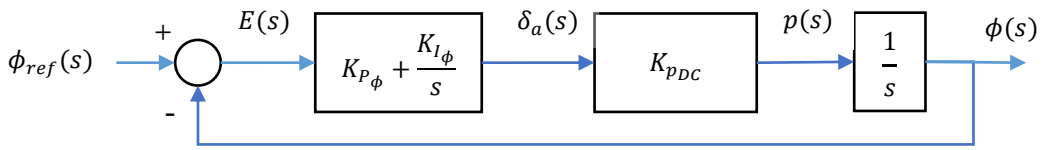
If the desired characteristic equation is given by $\Delta des = s + 4$, the controller gain can be found to be $K_{D_\phi} = 0.1275$. Eq. (8) can be modeled as a DC gain, $K_{\phi_{DC}}$ by evaluating Eq. (22) at $s = 0$:

$$\frac{p(s)}{\delta_a(s)} = K_{p_{DC}} \approx \frac{31.377}{4.3703 + 31.377K_{D_\phi}} \approx 7.111 \tag{23}$$

The design of the outer loops will use DC gain, $K_{p_{DC}}$ to represent the gain of the inner loop. For the outer loop design, we consider a PI controller to control the roll angle of the UAV, as shown in Fig. 7(b).



(a)



(b)

Fig. 7 - Autopilot design for roll angle control. (a) Autopilot design using roll rate feedback; (b) PI controller to control the roll angle of the UAV

Fig. 8 and Fig. 9 show the response of the simulated Maxi Swift to pitch and roll angle input of 10° without activating the velocity controller. The orange line represents the controller input values while the blue line signifies the Maxi Swift attitude responses. The graphs for pitch and roll responses show an under-damped second-order transfer function response. The rise time for pitch response is 0.41 seconds and the roll response is 0.52 seconds. The settling time for

pitch and roll and angle is measured at 0.93 seconds and 0.82 seconds, respectively. The value of peak time for pitch response is 0.31 seconds while for roll response is 0.29 seconds. Both pitch and roll response had a large overshoot of 23.3% and 23.04% respectively. The pitch and roll responses fluctuate because of the controller response to minor disturbance in the attitude values. The performance of the attitude controllers is shown in Table 3 and the PID controller values in **Table 4**.

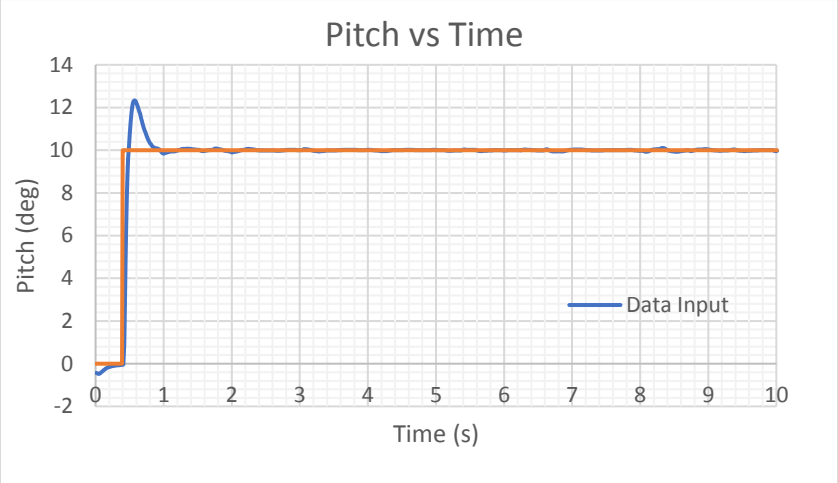


Fig. 8 - Pitch angle response

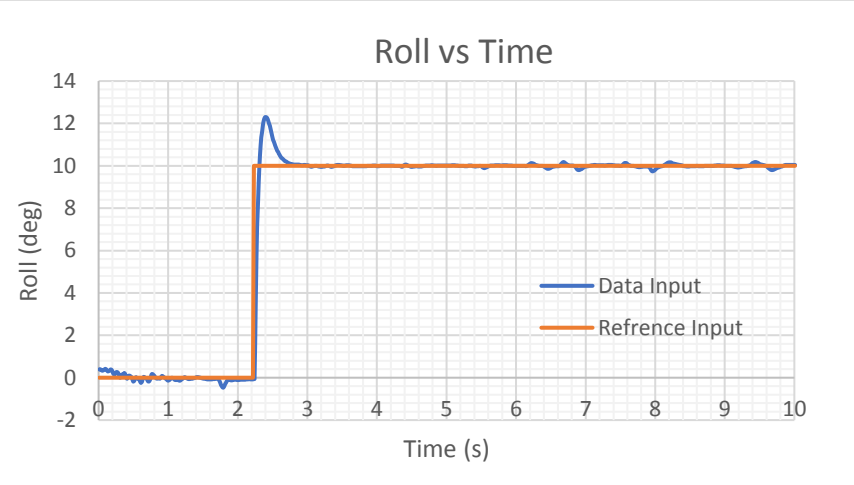


Fig. 9 - Roll angle response

Table 3 - Attitude controller performance

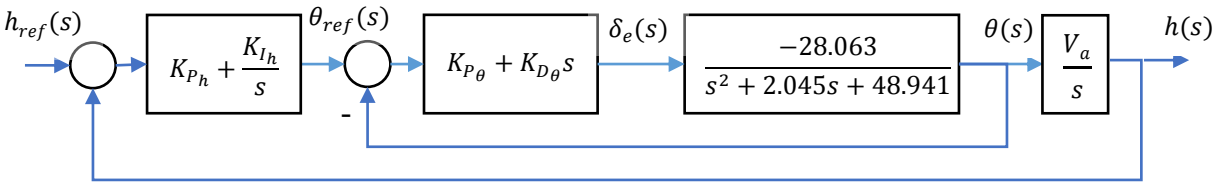
Performance parameter	Pitch, θ	Roll, ϕ
Rise time (s)	0.41	0.52
Settling time (s)	0.93	0.82
Peak time (s)	0.31	0.29
Overshoot (%)	23.34	23.05
Steady-state error (degree)	-	-

Table 4 - Attitude controller gain values

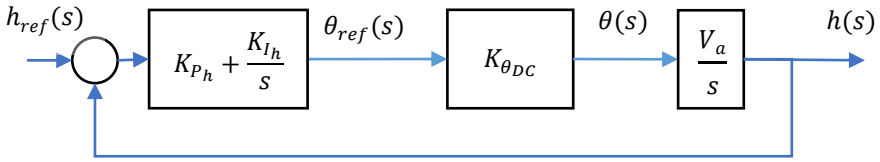
Channel/Controller Types	Proportional, K_p	Integral, K_i	Derivative, K_d
Pitch, θ	1.673	-	0.00018
Roll, ϕ	0.28	0.141	-

3.1.2 Altitude Control Results

The altitude hold control is designed using the successive loop closure approach with the pitch rate feedback system as an inner loop as shown in Fig. 10(a). The inner loop can be modeled as unity gain to simplify the outer loop design as shown in Fig. 10(b). The PI controller is then assigned to the altitude feedback system. The transfer function models are then compared with the desired characteristic equation given as: $s^2 + 5.25s + 1.25$. Then, from the established PID tuning equation, the value of K_{P_h} and K_{I_h} will be obtained for the altitude controller. Fig. 11 shows the response of the simulated Maxi Swift to the altitude input of 20 m. The orange line represents the controller input values while the blue line represents the Maxi Swift responses. There was a small overshoot about 12.1 percent in the altitude response of the aircraft. The rise time for altitude response is about 0.77 seconds while the settling time is about 1.75 seconds. The altitude graph shows an under-damped second-order transfer function response, with the response having a response delay of 0.5 seconds and peak time of 0.85 seconds. The altitude controller has a small steady-state error of 0.001 feet and a response delay of 0.1 seconds. The performance of the altitude controller is shown in Table 5 and the PID controller values in Table 6.



(a)



(b)

Fig. 10 - The altitude hold control design (a) The altitude hold controller with pitch rate feedback; (b) The simplified altitude controller design using PI gain and DC gain, $K_{\theta_{DC}} = 1$ representing inner loop

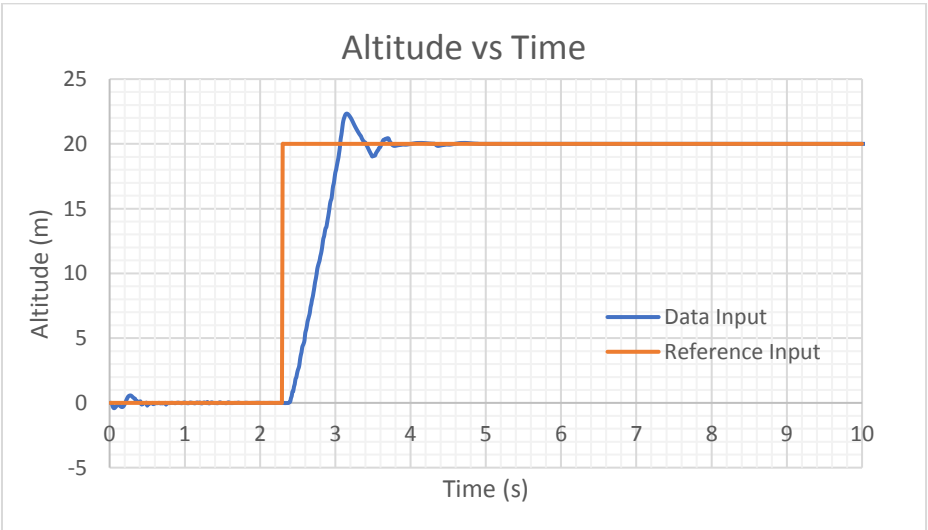


Fig. 11 - Altitude response

Table 5 - Altitude controller performance

Performance parameter	Altitude, h
Rise time (s)	0.77
Settling time (s)	1.75
Peak time (s)	0.85
Response delay (s)	0.5
Overshoot (%)	12.1
Steady-state error (feet)	0.001

Table 6 - Altitude controller gain values

Channel/Control Types	Proportional, K_p	Integral, K_i	Derivative, K_d
Altitude	0.525	0.125	-

3.1.3 Velocity Control Results

The transfer function model for velocity control is given in Eq. (19) with the values of model parameters is given as $a_{v1} = 0.3395$ and $a_{v2} = 5.6218$. Since the linear velocity dynamic is a first order system, we consider a PI controller for velocity dynamic as shown in Fig. 12. The characteristic equation for the closed transfer function model is then compared with the desired characteristic equation given as: $s^2 + 3s + 1.25$. From the established PID tuning equation, the value of $K_{P_{Va}}$ and $K_{I_{Va}}$ will be obtained for the velocity controller. Fig. 12 shows the response of the simulated Maxi Swift to the velocity input of 20 m/s. The orange lines represent the controller input values and yellow lines represent the Maxi Swift responses. There was no appreciable overshoot in the velocity responses of the aircraft. The velocity response rise time is approximately 0.8 seconds while the settling time is approximately 4.15 seconds. The tuned PID gain values enabled the simulated aircraft to have a stable response to the inputted speed changes. The shape of the velocity response graph shows an over-damped second-order transfer function response, with no steady state error. The performance of the velocity controllers is shown in Table 7 and the PID controller values in Table 8.

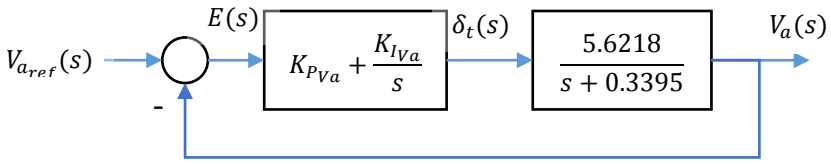


Fig. 12 - Velocity hold feedback loop with PI controller

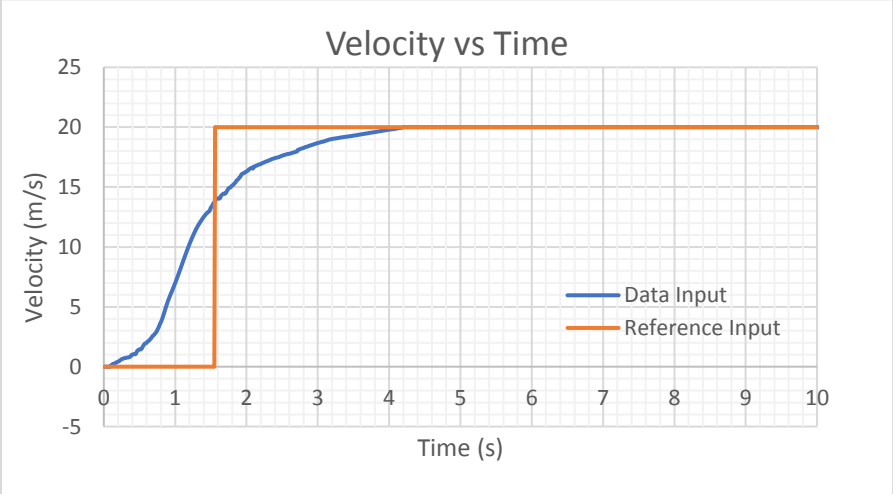


Fig. 13 - Velocity response

Table 7 - Velocity controller performance

Performance parameter	Velocity
Rise time (s)	0.8
Settling time (s)	4.15
Steady-state error (m/s)	-

Table 8 - Velocity controller gain values

Channel/Controller Type	Proportional, K_p	Integral, K_i	Derivative, K_d
Velocity	0.473	0.22	-

4. Conclusion

The aim of this work was to establish a controller and dynamic simulation for the Maxi Swift UAV in low-speed flight condition. The geometry model of a Maxi Swift UAV was created in the X-Plane flying simulator, and the controller was constructed in LabVIEW. The UDP connection was used to connect the controller to the X-Plane software that simulates the Maxi Swift UAV model. The controllers' PID values were then tuned using the Pole Placement tuning approach to get the desired performance. The proposed controller design could maintain the specified altitude, velocity, and attitude reference for the UAV. The controllers implemented in this work require further fine-tuning of the gain values to improve time response. Advanced tuning methods that are based on the optimization algorithm can be used to find the optimal gain of PID controller so that a more accurate positional tracking performance can be obtained. It is also recommended to extend the current approach to the hardware-in-the loop (HIL) approach by implementing the designed controller on the actual Maxi Swift hardware and autopilot system. The proposed SITL simulation can be used effectively to evaluate the response performance of the proposed controller before conducting actual flight tests. By adopting such an approach, engineers can shorten the development cycle, reduce cost, and improve the performance of such UAV system.

Acknowledgement

This research was supported by Universiti Tun Hussein Onn Malaysia (UTHM) through Tier 1 Grant (H924).

References

[1] M. F. Pairan and S. S. Shamsudin, "System identification of an unmanned quadcopter system using MRAN neural," *IOP Conf. Ser. Mater. Sci. Eng.*, vol. 270, no. 1, p. 12019, Dec. 2017

[2] S. S. Shamsudin, "The Development of Neural Network Based System Identification and Adaptive Flight Control for an Autonomous Helicopter System," Mechanical Engineering Department, 2013

[3] R. C. Nelson, *Flight stability and automatic control*, vol. 2. WCB/McGraw Hill New York, 1998

[4] M. J. Sidi, *Spacecraft dynamics and control: a practical engineering approach*, vol. 7. Cambridge University Press, 1997

[5] J. G. Ziegler and N. B. Nichols, "Optimum settings for automatic controllers," *trans. ASME*, vol. 64, no. 11, 1942

[6] B. R. Trilaksono, S. H. Nasution, and E. B. Purwanto, "Design and implementation of hardware-in-the-loop-simulation for UAV using PID control method," in *2013 3rd International Conference on Instrumentation, Communications, Information Technology and Biomedical Engineering (ICICI-BME)*, 2013, pp. 124–130

[7] Gh. Cohen, "Theoretical consideration of retarded control," *Trans. ASME*, vol. 75, pp. 827–834, 1953

[8] T. W. M. Randal W. Beard, *Small Unmanned Aircraft: Theory and Practice*. Princeton University Press, 2012

[9] M. F. Yaakub, A. A. Wahab, A. Abdullah, N. A. R. Nik Mohd, and S. S. Shamsuddin, "Aerodynamic Prediction of Rotor in Forward Flight using Blade Element Theory," in *Journal of Physics: Conference Series*, 2019, vol. 1150, no. 1

[10] M. Bangura, M. Melega, R. Naldi, and R. Mahony, "Aerodynamics of Rotor Blades for Quadrotors," Jan. 2016.

[11] M. V. Cook, *Flight Dynamics Principles: A Linear Systems Approach to Aircraft Stability and Control*, 3rd Ed. Butterworth-Heinemann, 2013

[12] M. B. Tischler and R. K. Remple, *Aircraft and rotorcraft system identification : engineering methods with flight-test examples*. Reston, VA: American Institute of Aeronautics and Astronautics, 2006

[13] Y. A. A. Zabidin, M. F. Pairan, and S. S. Shamsudin, "Dynamic Modelling and Control for Quadcopter UAV with LabVIEW and X-Plane Flight Simulator," *J. Complex Flow*, vol. 2, no. 2 SE-Articles, Oct. 2020

[14] A. Bittar, H. V Figuereido, P. A. Guimaraes, and A. C. Mendes, "Guidance software-in-the-loop simulation using X-plane and Simulink for UAVs," in *2014 International Conference on Unmanned Aircraft Systems (ICUAS)*, 2014, pp. 993–1002

[15] L. Yu, G. He, S. Zhao, X. Wang, and L. Shen, "Design and Implementation of a Hardware-in-the-Loop Simulation System for a Tilt Trirotor UAV," *J. Adv. Transp.*, vol. 2020, 2020

[16] A. Kaviyarasu, P. Sivaprakash, and K. Senthilkumar, "Design, Development and Evaluation of Longitudinal Autopilot for An Unmanned Aerial Vehicle Using X-Plane/Simulink," in *Recent Advancements in System Modelling Applications*, 2013, pp. 343–355

[17] L. Wang, S. Chai, D. Yoo, L. Gan, and K. Ng, *PID and predictive control of electrical drives and power converters using MATLAB®/Simulink®*. 2014

[18] L. Wang, *PID control system design and automatic tuning using MATLAB/Simulink*. John Wiley & Sons, 2020