A REAL-TIME CONTROL ALGORITHM FOR FIXED-WING UAVS IN TWIN-BOOM INVERTED V-TAIL CONFIGURATION

THUẬT TOÁN ĐIỀU KHIỂN TỰ ĐỘNG CHO MÁY BAY KHÔNG NGƯỜI LÁI CẤU HÌNH V-TAIL FIXED-WING

Han Trong Thanh, Nguyen Huu Trung, Duong Minh Duc, Nguyen Thai Binh, Do Trong Tuan

School of Electronics and Telecommunications, Hanoi University of Science and Technology;
{thanh.hantrong, trung.nguyenhuu}@hust.edu.vn; {minhduc146, thethaibinh}@gmail.com; tuan.dotrong@hust.edu.vn

Abstract - Unmanned Aerial Vehicles (UAVs) have been widely used in many areas such as economy, security, military..., including aerial photo shooting, traffic status updating, surveillance of building under construction and entertainment… Nowaday, the research in uavs is the most focused area, especially in autonomous controllers. In this paper, we propose a model of a real-time control algorithm for a fixed-wing uav in inverted v-tail configuration, including automatic takeoff phase, waypoint tracking phase and auto-landing phase. The algorithm is built as a standardized model on the matlab/simulink as well as using PID controllers for implemention. The performance of algorithm is simulated by using X-Plane – a simulator developed by Laminar Research and certified by the Federal Aviation Administration (FAA- USA) to train pilots, which enables simulation flights with real time data and the highest degree of accuracy

Key words - UAV, inverted V-tail, fixed-wing, autonomous control, PID controller

Tóm tắt - Máy bay không người lái (UAV) được sử dụng rộng rãi trong nhiều lĩnh vực như kinh tế, an ninh, quân sự, bao gồm các ứng dụng chụp ảnh, kiểm tra mật độ giao thông, giám sát xây dựng công trình, giải trí... Ngày nay, nghiên cứu về UAV là một trong những lĩnh vực trọng tâm, đặc biệt là nghiên cứu các bộ điều khiển tự động. Trong bài báo này, các tác giả đề xuất một mô hình thuật toán tự động điều khiển theo thời gian thực với cấu hình máy bay cánh bằng đuôi V, bao gồm điều khiển tự động cất cánh, điều khiển dẫn đường và tự động hạ cánh. Thuật toán được xây dựng trên Matlab/Simulink và sử dụng bộ điều khiển PID để thực thi. Tính hiệu quả của thuật toán sẽ được kiểm chứng bằng mô phỏng thông qua phần mềm X-Plane – một phần mềm mô phỏng bay theo thời gian thực có độ chính xác cao, được phát triển bởi Laminar Research và chứng nhận của cục Hàng không Liên bang Mỹ cho việc đào tạo phi công.

Từ khóa - UAV, V-tail, cánh bằng, điều khiển tự động, bộ điều khiển PID.

1. Introduction

In recent years, Unmanned Aerial Vehicle - UAV technology has been developed and applied widely in fields of economy, military, etc. In particular, fixed-wing configurations are proved to be very effective in harsh environments and special situations. Fixed-wing UAVs consume a smaller amount of fuel compared to multi-rotor UAVs for the same length of a route. Being a fixed-wing UAV, twin-boom inverted V-tail aircraft is one of the most fuel-saving UAV configurations. The world ‘s longest recorded flight for a mini-class unmanned aircraft is held by a twin-boom inverted V-tail. Apart from being exceptionally fuel-saving, twin-boom inverted V-tail configuration features several advantages in control and stability:

V-tail has slightly lower drag.

- Keeping more of the control surfaces up away from the prop wash, at least the middle of the prop wash.

- Producing slightly better yaw characteristics (compared to a noninverted V-tail) in a coordinated turn [1].

- Also, being in the slipstream of a pusher powerplant, an inverted V-tail twin-boom may permit the use of smaller control surfaces and provide better low-speed responsiveness but at the cost of increased buffeting and parasite drag.

In this research work, we propose an ideal model based on a customized design called *ASEA v1.01* (Figure 1). The object has qualified many manual flight tests and has been experimentally proved to be aerodynamically stable. The defined model used in this research has been slightly modified for the purposes of calculation and simulation as been described in Table 1.

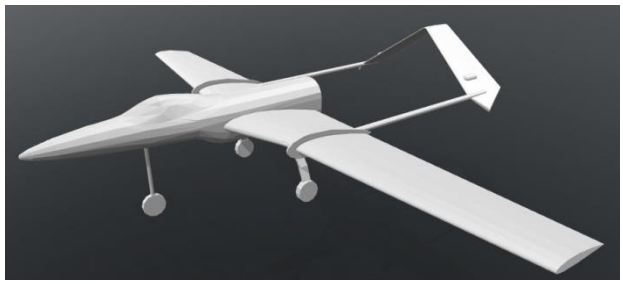

Figure 1. *3D object modeled by Solidwork*

Table 1. *ASEA v1.01 Description*

Description	Symbol	Value	Unit
Airfoil	*NACA 65-2415*	n/a	*n/a*
Maximum Lift Coefficient	$C_{L_{max}}$	1.4 [2]	*n/a*
Zero-lift Drag Coefficient	C_{D_0}	n/a	*n/a*
Angle of attack for zero-lift	α_0	−2.2	*degree*
Mass	m	16	*kg*
Wing area	S	0.9	m^2
Wing Span	b	2.74	*m*
Engine Power	*Zeonah G62*	3150	*W*
Propeller	*D×P*	20×10	*inch*
Mean aerodynamic chord (MAC)	$\bar{c}$	0.38	*m*
Wing area	S	0.9	m^2
Aspect ratio	AR	8.34	

In this paper, the design of an autonomous control

algorithm for a fixed-wing UAV in twin boom inverted V-tail configuration is presented in Section 2. Next, modelling of the UAV object and Integration Simulation Software System are analysed to simulate X-Plane. Then, simulations for scenarios of take-off way-point tracking and landing process are carried out with experiment results analysed in section 4. Finally, conclusion and future work are given in section 5.

2. Design The Autonomous Control Algorithm

2.1. The algorithm structure

Every single phase of the flight process will be managed by a separate controller (*stage-controller*), they are *Auto Take-off, Auto Enroute, Auto Landing,* depicted as in **Figure 2**. Stage-controllers calculate their own output control signals in two factors: *Longitudinal control* and *Directional control*. The two factors will be used as inputs of a *Attitude Stabilizer* (will be discussed in the next sub-session). Signal switches will be placed after stage-controllers in order to provide the exact control signals corresponding to the flight stage. The stage-controllers will collect UAV flight parameters to compute indicators for switches to determine its specific state, navigating precise control signals for the *Attitude Stabilizer*.

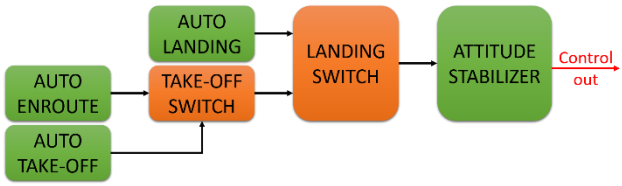

Figure 2. *The structure of the autonomous control* lgorithm

Runway direction, waypoints database, runway coordinates and sensors data are all of the system's inputs. All of the internal stage-controllers need sensors data input (Figure 3).

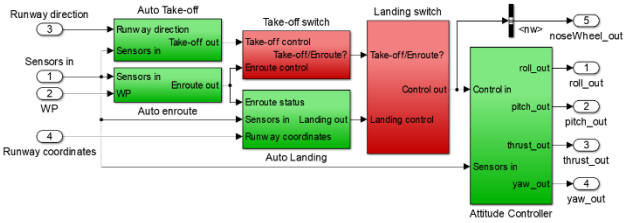

Figure 3. *Simulink model for the autonomous controller*

2.2. Attitude Stabilizer

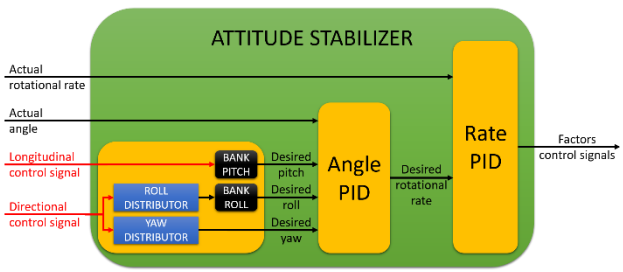

Figure 4. *Attitude Stabilizer diagram.*

Attitude Stabilizer is the "lowest controller", as shown in Figure 4, directly providing control signals for actuators or the actuators mixer. The core of *Attitude Stabilizer* are PID (Proportional Integral Derivative) controllers. Being structured according to a hierarchical architecture, *Attitude Stabilizer* has two sequential control levels: angle control and angle rate control, both are PID controllers.

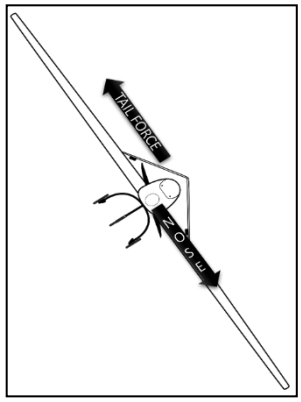

Figure 5. *Unstable nose at great roll if it has substantial yaw control*

Bank angle blocks act as limiters that keep pitch and roll control signals from exceeding safety limits corresponding to critical desired angles. The essence of using bank angle block is to convert the control signal into critical angles scale: $-40° <$ desired pitch, desired roll $< 40°$ (Figure 5). As a consequence, PID controllers will drive the UAV to remain in safety attitude.

When the UAV perform a turn, ailerons are mainly responsible for rolling the aircraft to one side, causing the adverse yaw effect [3], a result of differential drag due to the slight difference in the velocity of the left and right wings. To reduce this, yaw control will be correlatively mixed with roll control, coordinating the turn by V-tail (i.e., ruddervator). However, any substantial yaw control at great roll (close to roll bank angle) will cause a body's oscillation in relation to the yaw axis. Distributors hold the responsibility of coupling the two control factors roll and yaw in relation to the directional control signal (Figure 6).

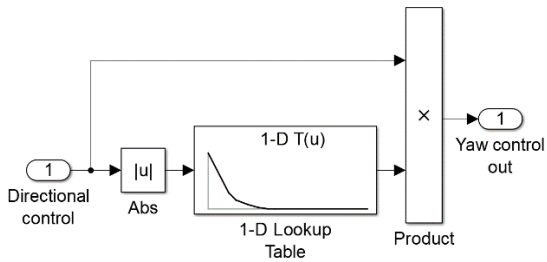

Figure 6. *Directional control signal distributor for yaw*

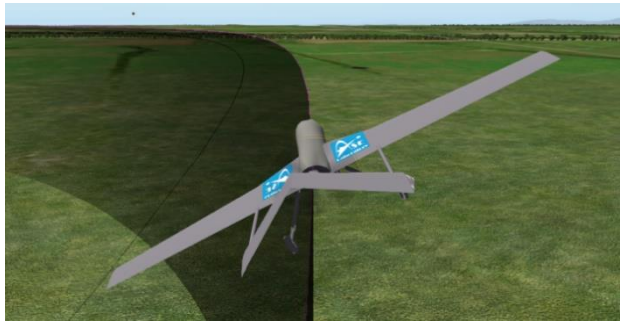

Figure 7. *Coordinated turn conducted by yaw control manipulations*

In a coordinated turn, the heading path conforms the velocity direction (as in Figure7).

2.3. Auto Take-off

The take-off is the procedure of flight in which an aircraft leaves the ground and becomes airborne. Based on a phase-breakdown of take-off, this process can be divided into three separate phases as in Figure 8.

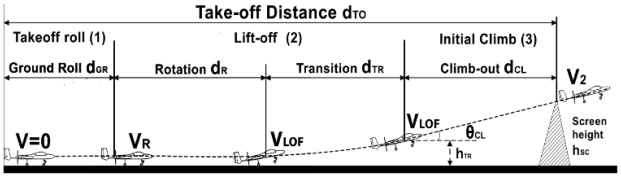

Figure 8. *Phases in take-off procedure*

The stall speed V_s is reference speed and defined as the minimum speed at which level fight can be maintained with zero acceleration. Stall velocity can be expressed as:

$$V_s = \sqrt{\frac{2mg}{\rho C_{L_{max}} S}} = 14.25\ (m/s)$$

Rotation velocity and lift-off velocity [4]:

$$V_R = 1.1V_S = 15.7\ (m/s)$$

$$V_{LOF} = 1.2V_S = 17.1\ (m/s)$$

The take-off algorithm uses P and PD controllers to compute the desired velocity and pitch angle in each phase. The only external input needed is the azimuthal orientation of the runway.

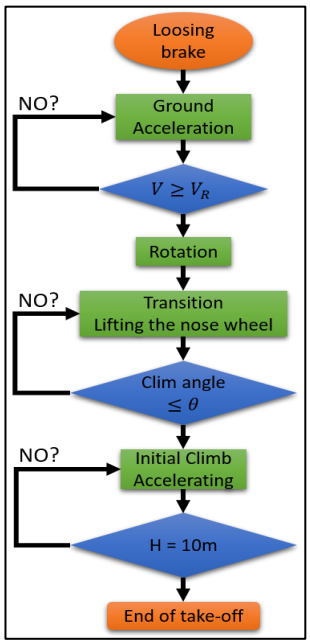

Figure 9. *Take-off algorithm*

During the acceleration in ground roll, the pitch control signal is set to be zero (the elevator is in neutral position). A dedicated PD controller will drive the nose-wheel to maintain heading direction in runway as well as control the roll angle signal in order to avoid longitudinal oscillation (because of crosswind during take-off). After the UAV departs the runway, yaw angle controller will manage UAV's heading by manipulating V-tail.

2.4. Waypoints Tracking

Gathering the coordinates of UAV position and the current waypoint, the *Waypoints tracking controller* computes the desired heading providing the directional control signal for the *Attitude Stabilizer*.

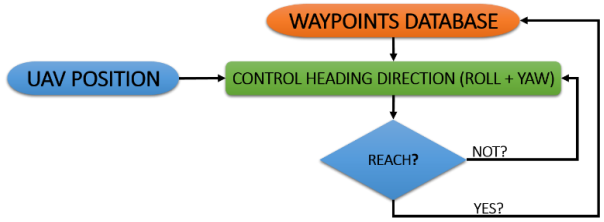

Figure 1. *Waypoints tracking controller algorithm*

The computation of desired heading angle is based on the *azimuth* function from Matlab Mapping Toolbox library:

$$targetHeading = azimuth(latUAV, lonUAV, latWP, lonWP) \quad (1)$$

The equation (1) result depends on which quadrant the waypoint is in relation to the UAV's current position and heading, ranging from 0^0 to 360^0. Therefore, it is necessary to transform the scale to -180^0 to $+180^0$ by a normalize block, forcing the aircraft to conform to the shortest possible path (Figure 11).

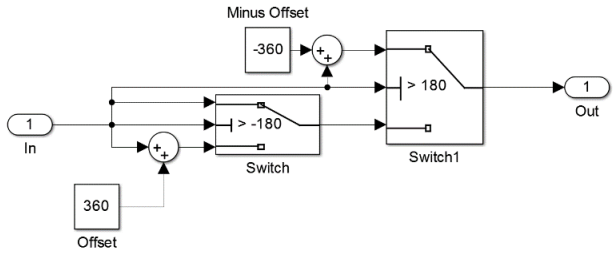

Figure 2. *Normalize block diagram*

2.5. Auto Landing

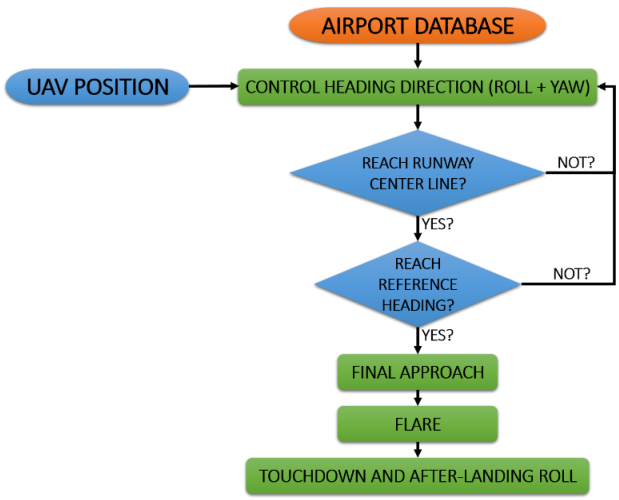

Figure 3. *Auto landing controller algorithm*

Sequential stages in the landing procedure and corresponding algorithm is as follows [5]:

The Base leg is the stage that UAV will approach the space area near the beginning point of the runway. The algorithm for this stage has no difference compared with

the algorithm for waypoints tracking to that point.

The Final approach: illustrated by Figure 13, the control algorithm for this stage is responsible for bringing both the UAV's position and heading angle to the value needed in order to start descending in the next phase when both of these conditions happen:

- The heading angle of the aircraft must be approximately equal to the azimuthal orientation of the runway.

- The azimuthal orientation generated by the coordinates of the aircraft and the endpoint of the runway must be approximately equal to the azimuthal orientation of the runway.

Figure 4. *Before and after the final approach*

The algorithm for the final approach updates the present coordinate value, the heading angle of the UAV, and differentiates it with the azimuthal orientation of the runway to give the directional control signal for the *Attitude Stabilizer*. The UAV achieves the needed position as well as the desired heading direction by the combination of roll and yaw controls.

The Round out (Flare): only when the UAV meets the above two conditions in the previous phase, will the phase **flare** be activated. The mission of the algorithm for this phase is to cause the UAV to slow down in airspeed and to decrease the descent rate (longitudinal velocity) as well. To keep the control aircraft in low speed, the angle of attack must be maintained to maximum value (maximum the lift produced). A pitch-angle mapping block (a 1-D look-up table in Simulink) between pitch control signal and the altitude will be used for progressively increasing the angle of attack, decreasing the descent rate, as depicted in Figure 14.

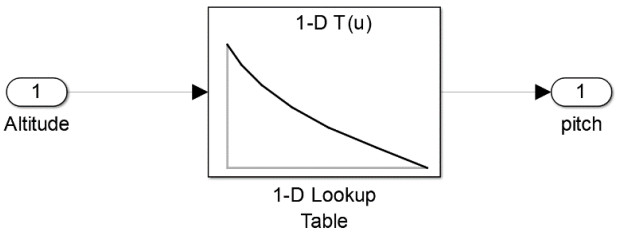

Figure 5. *Pitch control look-up table in the flare*

The altitude, at which the flare begins, must be high enough for the UAV to slow down, but still, must not be too high for it to hit the runway before it reaches the stall speed, equal to $40m$. The corresponding pitch control value has to keep the angle of attack not exceed the stall angle in the whole progress as well.

The Touchdown and after-landing roll: when the UAV has hit the runway, the algorithm is supposed to immediately shut off the engine and turn over the elevator, forcing the nose wheel to hit and force the runway. From this moment, we will control the nose wheel to adjust the running angle of the UAV complying to the direction of the runway. We can hit the brakes of the landing gear (if available) to make the roll on the runway finish faster.

3. Simulink – X-Plane Real-Time Simulation

3.1. X-Plane

X-Plane is a commercial software package that enables ultra-realistic simulation flights from Laminar Research, modeling accurately aerodynamics of flying vehicles. X-Plane is certified by the U.S. Agency of Aviation (FAA – Federal Aviation Administration) to train pilots because its method ensures a reliable system since it is much more detailed, flexible, and advanced than the flight model based on stability derivatives that are used by most other flight simulators. The controllers developed and tested in X-Plane platforms have been successful when embedded in real aircraft, adding more credibility to the results that are obtained in this work [6].

3.2. Modeling the UAV object by Plane Maker Tool

Plane Maker is a tool that comes along with X-plane that allows users to create new UAV models with detailed customized modification, following the user's designs. The created models are fully compatible with X-Plane environment (Figure 15). Plane Maker can model all the ranges of aircraft configurations, from biplanes to multicopters, supporting numerous features of airfoils, engines, and materials.

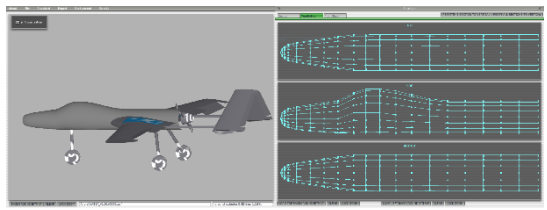

Figure 6. *Modelling the UAV object by Plane Maker tool*

3.3. Integrating Simulation Software System

Due to the purpose of evaluating an autonomous control algorithm, it is required to perform real-time simulation as we select the UDP communication between X-Plane and Simulink to transmit and receive data. To do this, we use a Simulink UDP library that allows sending control surfaces commands to X-Plane and extracting received data packages as shown in Figure 16 [7]. Control surfaces command will be normalized as Throttle in [0,1]; Elevator, Aileron, Rudder and Nose wheel in [–1,1].

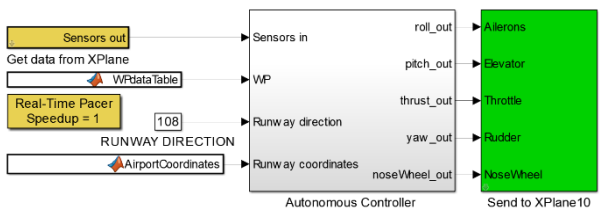

Figure 7. *Simulink model for simulation of software system*

Firstly, we configure IP and input/output UDP port for both X-Plane and Simulink; the software simulation system will be ready as the connection is established (Figure 17). For purposes of control effectiveness, UDP packages rate is set to 50Hz.

Figure 8. *Configuration IP address and ports in simulation*

Secondly, we initialize flight parameters:

1. Azimuthal direction of the runway.
2. Import waypoints database (Figure 18).
3. Beginning and ending points of the runway.

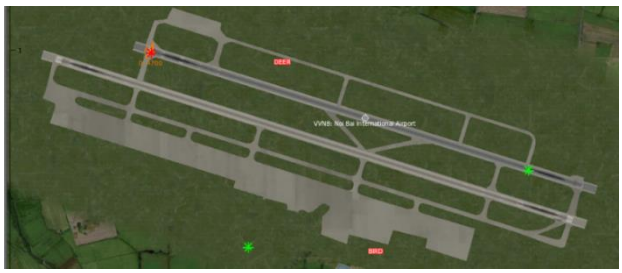

Figure 9. *Waypoints position in airport map*

Finally, we tune PID controllers to find optimal P, I and D gains, following an experimental method called Ziegler-Nichols [8].

3.4. Calculating Steady Flight Optimal Velocity

Drag coefficient versus lift coefficient is mathematically modeled with the following second-order parabolic curve with an acceptable accuracy [9]:

$$C_D = C_{D_0} + \frac{C_L^2}{\pi \times AR \times e} \quad (1)$$

Where: C_{D_0}: The *zero-lift drag coefficient*

e : The wing Oswald efficiency factor

AR : The wing aspect ratio

Symbolizing $C_{D_0} = a$ and $1/(\pi \times AR \times e) = b$, we have:

$$C_D = a + bC_L^2 \quad (2)$$

In a steady and stable flight, lift force is equal to gravity force as well as the drag equal to the engine thrust. Hence, in body-axis of an aircraft, the equation of motion is

$$\sum F_x = 0 \Leftrightarrow T = D \quad (3)$$

$$\sum F_z = 0 \Leftrightarrow W = L \quad (4)$$

Therefore, the lift-to-drag ratio is simple:

$$\frac{L}{D} = \frac{W}{T} \Rightarrow T = \frac{W}{L/D} = W\frac{C_D}{C_L} \quad (5)$$

If a pilot intends to have a minimum fuels consumption (minimum thrust T):

$$(L/D)_{max} \longrightarrow D_{min}$$

D is minimum when $\frac{C_D}{C_L}$ reach the minimum value, as it means:

$$\frac{d}{dC_L}\left(\frac{C_D}{C_L}\right) = 0 \Leftrightarrow \frac{d}{dC_L}\left(\frac{a+bC_L^2}{C_L}\right) = 0 \quad (6)$$

Solve the differential equation with C_L as the variable in $\boldsymbol{Eqn.6}$, we will find the lift coefficient, at which, the drag force is minimum:

$$C_{L_{(C_L/C_D)max}} = \sqrt{\frac{a}{b}} \quad (7)$$

The calculation of b is not a big deal, but the calculation of $a\ (i.e., C_{D_0})$ is very challenging, tedious, and difficult. Therefore, we apply a more feasible and experimental approach, manipulating *X-Plane*'s capabilities. Carrying out steady flight simulation in various velocity from *minimum speed = 18m/s* [10], we have the table and figure below

'V'	'AoA'	'L/D'	'Cl'	'Cd'	'Cd0'
18.0000	6.4178	16.0703	0.6498	0.0404	0.0203
18.9997	5.5050	16.9344	0.5861	0.0346	0.0182
19.9960	4.7133	17.6309	0.5310	0.0301	0.0167
20.9993	4.0238	18.1573	0.4829	0.0266	0.0155
21.9992	3.4206	18.5134	0.4410	0.0238	0.0145
23.0006	2.8899	18.7018	0.4041	0.0216	0.0138
24.0005	2.4233	18.7325	0.3717	0.0198	0.0133
25.0004	2.0089	18.6324	0.3429	0.0184	0.0128
26.0003	1.6421	18.4389	0.3174	0.0172	0.0124
27.0003	1.3095	18.0970	0.2945	0.0163	0.0121
28.0015	1.0177	17.7729	0.2740	0.0154	0.0118
29.0003	0.7524	17.3495	0.2557	0.0147	0.0116
30.0004	0.5127	16.8855	0.2390	0.0142	0.0114

Figure 10. *Cruise airspeed and corresponding drag coefficient*

From Figure 20, based on the geometric properties of the quadratic function and the theory of the slope of the graph, we line up a tangent and find the point at which, the lift-to-drag ratio reaches the maximum, and the value of optimal velocity is:

$$V_{Optimal} = 24\text{m/s}$$

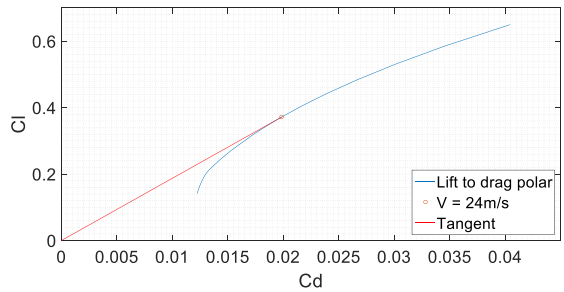

Figure 11. *Drag polar and special points*

4. Simulation Scenarios and Results

4.1. Auto Take-off

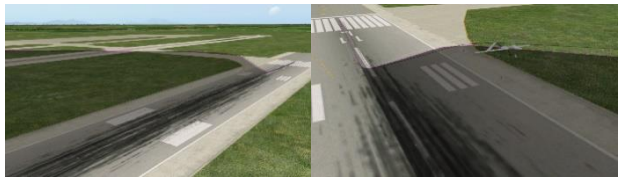

Figure 12. *3-D trajectory in the take-off process*

The result of the take-off scenario is approximately similar to the calculations of the theoretical procedure (Figure 21).

It takes *5.8* seconds for the UAV to leave the runway when the airspeed equals*19.2m/s*. Take-off distance is *96m*.

4.2. Waypoints Tracking

With the optimal cruise airspeed found above, UAV will be powered to perform a steady flight at *25m/s* in airspeed and at the altitude of *50m* in the waypoints tracking stage. We make a scenario including two waypoints due to the goal of forcing the UAV curve in

different directions to examine responsiveness as well as the control effectiveness. Post-simulation data proves that the algorithm works well, driving the UAV to pass through a circle of radius *1.5m*, circling around the current waypoint (the condition for considering a UAV reach the desired waypoint). However, most of the tests show even better results.The UAV completes the current waypoint within a *1m* radius circle.

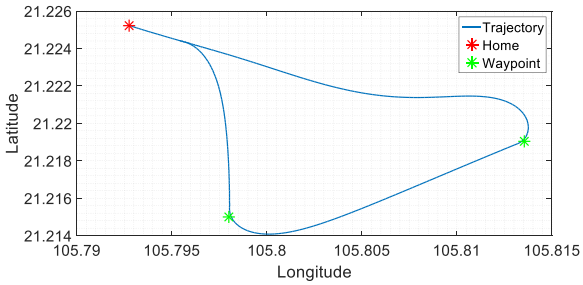

Figure 13. *Post-simulation waypoints tracking 2-D trajectory*

Another important criterion is the trajectory's radius of curvature (*RoC*) that reaches the minimum at the switching of a waypoint. The critical value for ROC: RoC_{min} = *50m* at bank roll angle = 40^0 and steady airspeed = *25m/s*, following the calculation in [11]. The minimum value is approximately equal to *125m*, being greater than the critical value $RoC_{CRITICAL}$. This means the UAV flight with safe bank roll angle in relation to steady flight airspeed. As a consequence, the UAV can perform the "hover" mode, circling around the current waypoint with the radius approximately *125m* (Figure 23).

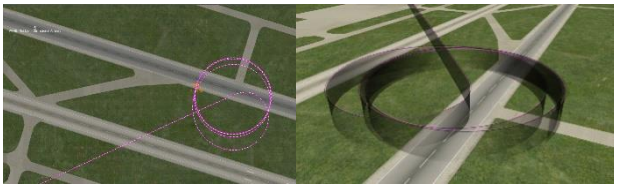

Figure 14. *3-D flight path in "hover" mode*

The feature "Cycle 3-D Flight Path" from X-Plane shows us the aggregated and visual trajectory in the whole stage, as in Figure 24.

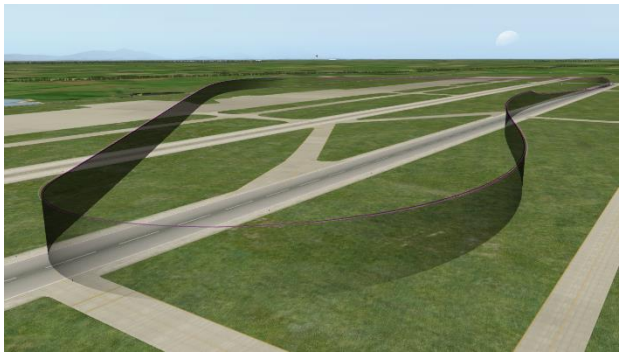

Figure 15. *Post-simulation waypoints tracking 3-D trajectory in X-Plane*

4.3. Auto Landing

The UAV makes the final approach with 1.5 *degree* maximum difference in comparison between heading angle and the runway azimuthal orientation, the maximum difference between the UAV's position and the runway reference line is *2 m*.

The descent rate of the UAV when touching the runway is *1.5m/s*, being within the critical longitudinal velocity, which is *5m/s*. This will secure a safe and successful touchdown, drawing a hyperbolic trajectory in the round out (flare) stage (shows in Figure25).

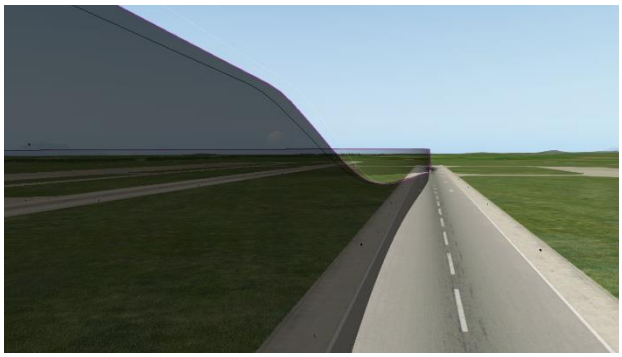

Figure 16. *The hyperbolic trajectory in the round out (flare) stage*

Touching the runway at the lowest possible descent rate, the phenomenon of the aircraft jump off back to the air does not happen. The main landing gear touches the ground first and the nose gear has been controlled before it reaches the runway, keeping the UAV moving on the runway direction.

The UAV continues to run on about *200m* before it completely stops. If it has brakes on all three wheels, the running distance is reduced to *90m*.

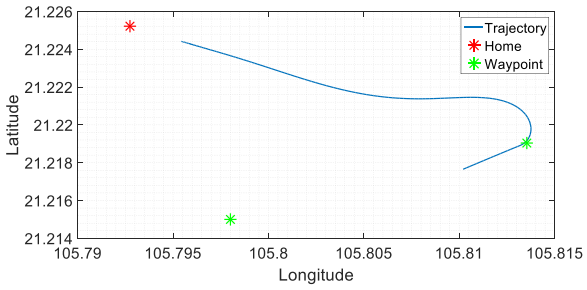

Figure 17. *Post-simulation landing 2-D trajectory*

5. Conclusion and Future Work

The paper presents a design of an autonomous control algorithm for a fixed-wing UAV in twin boom inverted V-tail configuration. Simulation results show that the proposed autonomous control algorithm for the twin-boom inverted V-tail UAV object is reliable when applying to various scenarios of take-off, waypoint tracking and landing process. Using a very aerodynamically precise simulation tool X-Plane, the approach allows developers to focus on designing control algorithms, and by being a real-time simulation, the shown method not only gives feedbacks for both algorithm evaluation and optimization process but also benefits the development of the UAV aerodynamic design. The future work will target at deploying the algorithm into a specific hardware platform with real sensors and devices to conduct HIL (Hardware in the Loop) simulation, and using X-plane features to evaluate effects of disturbances like lateral wind or turbulences on the aircraft.

Acknowledgment

This research is carried out in the framework of the project funded by the Ministry of Science and Technology (MOST), Vietnam under the grant number ĐTĐL.CN-02/16. The authors would like to thank the MOST for their financial support.

REFERENCES

[1] Douglas M. Marshall, R. Kurt Barnhart, Eric Shappee, Michael Most, in Introduction to Unmanned Aircraft Systems, New York, CRC Press, 2016, p. 187.

[2] "Airfoil Tools," [Online]. Available: http://airfoiltools.com/airfoil/details?airfoil=naca652415a05-il. [Accessed 14 06 2017].

[3] "Pilot's Handbook of Aeronautical Knowledge," U.S. Department of Transportation - Federal Aviation Administration, 2008, pp. 5-3.

[4] "Takeoff Path," in Federal Aviation Regulations part 23 and 25, U.S. Department of Transportation - Federal Aviation Administration.

[5] "Approaches and Landings," in Airplane flying handbook, U.S. Department of Transportation - Federal Aviation Administration, 2016, pp. 8-1.

[6] Richard Garcia, Laura Barnes, "Multi-UAV Simulator Utilizing X-Plane," Journal of Intelligent and Robotic Systems, vol. 57, pp. 393-406, 2010.

[7] M. A. Zahana, "Simulink-Xplane10 Communication Via UDP," MathWorks, [Online]. Available: https://www.mathworks.com/matlabcentral/fileexchange/47144-simulink-xplane10-communication-via-udp. [Accessed 14 06 2017].

[8] J.G. Ziegler, N. B. Nichols, "Optimum Settings for Automatic Controllers," 1942.

[9] Jan Roskam, Chuan-Tau Edward Lan, "Complete airplane drag polars," in Airplane Aerodynamics and Performance, DARcorporation, 1997, p. 137.

[10] Jan Roskam, Chuan-Tau Edward Lan, "Maneuvering and the flight envelope," in Airplane Aerodynamics and Performance, DARcorporation, 1997, p. 581.

[11] L. Quang, "Steady turn mode," in Cơ học vật bay - Aircraft Aerodynammics, 2012, Hanoi University of Science and Technology, pp. 113-117.

(The Board of Editors received the paper on 28/09/2017, its review was completed on 23/10/2017)