Journal of Physics: Conference Series

PAPER • OPEN ACCESS

Research and Development of a Real-time UAV Flight Visualization Simulation System

To cite this article: Yang Xiao *et al* 2022 *J. Phys.: Conf. Ser.* **2218** 012081

Research and Development of a Real-time UAV Flight Visualization Simulation System

Yang Xiao[1,*], Shulin Dai[2], Chenfan Xiao[3] and Xinfeng Xu[1]

[1]Department of Mechanical Engineering, Southwest Petroleum University, Chengdu, China
[2]School of Automation Science and Electrical Engineering, Beihang University, Beijing, China
[3]Ant Group, Chengdu, China

*Corresponding author email: eastsun628@163.com

Abstract. On the premise of practical application, this paper describes the principle, technology and implementation methods of a real-time UAV flight visualization simulation system. The overall framework, hardware system composition, UAV flight model establishment, determination of flight control mode and control rate, design of 3D visual simulation system, management and processing of UAV flight state data are discussed, and finally, the actual application effect of the system is evaluated.
Keywords: UAV; Flight status; Flight simulation; Visualization; Data management.

1. Introduction
Unmanned aerial vehicle (UAV) are unmanned aircraft operated by radio remote control equipment and self-contained programmed control devices. In terms of civil use, the current application of UAV in aerial photography, traffic management, agricultural plant protection, surveying and mapping and other fields has greatly expanded the use of UAV itself. Whether controlling drones through remote controls or smartphone apps, they have the ability to reach the most remote areas, requiring little human support and little efforts, time and effort, this is one of the biggest reasons that drone are being adopted worldwide.
Small and medium-sized fixed-wing drones are heavy and wide used in civilian fields. For improving the development of small and medium-sized fixed-wing drones, beside to paying attention to the shape design, we should pay more attention to the research and development of flight control systems. The traditional research and development process of flight controllers is divided into object modeling, control law simulation, hardware design, software programming, integrated debugging[1]. Object modeling and control law simulation are based on the aerodynamic parameters of the UAV, establish a flight model and control model for engine characteristics and actuator characteristic parameters, then, the model was verified by the simulation software. Reference[2] establishes a non-linear six-degrees-freedom equation of motion, a guidance and control model with Matlab/ Simulink, the simulation results were analyzed by curves. Reference [3] similar to the former, the innovation point lies in the visual output of the simulator implements passed through Flightgear simulator. At present, most of the simulation process work on a computer platform. However, actual controllers used on drones are all embedded platforms. Considering the timeliness of the model simulation depends on the way the model is built, computing method and the operational power of the controller and the fixed-wing aircraft is operating during commissioning. Once a failure occurs, greater damage to the fuselage. So, it is urgent to add a

more realistic flight control simulation and debugging method in the development process of the flight controller, make up for the deficiencies of the traditional debugging process.
To more truly verify the accuracy of the flight model with the control algorithm, the response efficiency of the hardware platform, the software program to the control signals, the compatibility between the software and the hardware, to reduce development costs, purpose of improving the commissioning efficiency, using visual simulation technology, the visual real-time flight simulation system is developed based on embedded controller ARM.

2. System Architecture

The whole system mainly includes flight simulation module and 3D view simulation module. The flight simulation module is built on the kernel STM32F103 microcontroller based on the ARM and on the associated peripheral circuits (The following referred to as the lower machine), complete the command control signal acquisition, flight state calculation and data transfer. The 3D visual simulation module receives the uploaded flight status information in real time, drive quick update of the 3D view module, realize the purpose of visualization. The system function is showed in Figure 1.

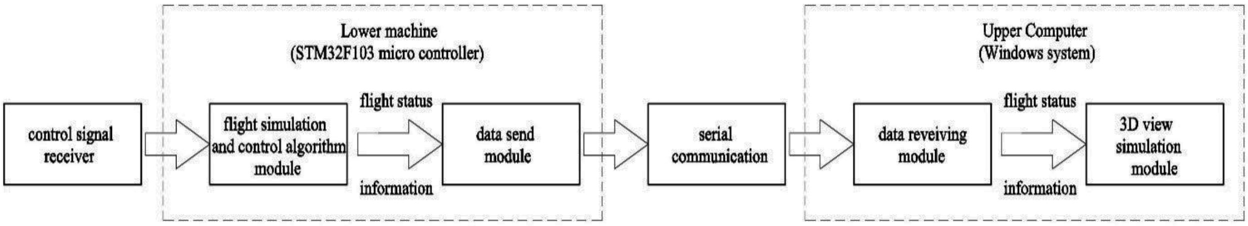

Figure 1. System Function.

3. Design and Implementation of the Flight Simulation Module

The flight simulation module is a core part of the system, flight models are mainly built using embedded microcontrollers, and under a given control signal and control law, UAV flight simulation was performed in real time using suitable numerical calculation methods, the continuous spatial position and posture information of the UAV are obtained. The functional structure is shown in Figure 2.

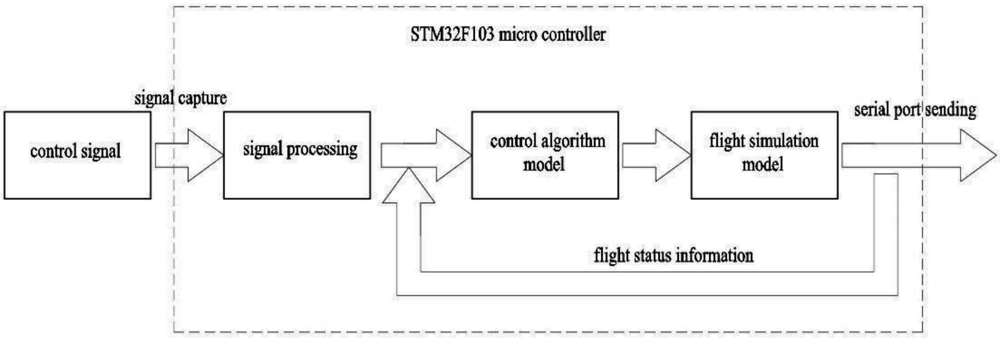

Figure 2. Module Function.

3.1. Hardware Design of the Flight Simulation Module

Considering the task requirements and functional requirements, the system uses Italian Semiconductor Inc. STM32F103 Series controllers serve as an embedded hardware platform. The series of microcontrollers include CPU, RAM, ROM, I/O Interface, Timer/Counter, UASRT, A/ D and D/ A converter. Its highest operating frequency is 72MHz, and can adjust the frequency of work. Single-cycle multiplication and hardware division can be achieved. The system utilizes its powerful operational capability and interface functions, realize the establishment and solution of flight simulation model and control model, control the signal acquisition and flight state data transmission.

3.2. Design of Flight Model and Control Law

3.2.1. Flight model establishment. The flight models established in this system include UAV dynamics model, kinematics mode, environmental model, engine model and steering machine model[4]. To simplify the equation reasonably, the following assumptions were made:

- Considering the drone flight altitude and mission radius, consider the Earth as an infinitely extended plane, ignoring the Earth's rotation and curvature;
- Considering the characteristics of the material and acting force of the UAV body, no deformation of the body, that is, the body as the ideal rigid body[5].

To ensure the fidelity of the UAV flight simulation, our system establishes the six degrees of freedom flight equations using the quaternions method as shown below:

$$\begin{cases} \mathrm{m}\frac{dV}{dt} = P\cos\alpha\cos\beta - X - mg\sin\theta \\ \mathrm{mV}\frac{d\theta}{dt} = P(\sin\alpha\cos\gamma_c + \cos\alpha\sin\beta\sin\gamma_c) + Y\cos\gamma_c - Z\sin\gamma_c - mg\cos\theta - mV\cos\theta \\ \frac{d\psi_c}{dt} = P\,(\sin\alpha\sin\gamma_c - \cos\alpha\sin\beta\sin\gamma_c) + Y\sin\gamma_c + Z\cos\gamma_c \end{cases} \quad (1)$$

$$\begin{cases} J_x\frac{dw_x}{dt} + (J_z - J_y)\omega_z\omega_y = M_x \\ J_y\frac{dw_y}{dt} + (J_x - J_z)\omega_x\omega_z = M_y \\ J_z\frac{dw_z}{dt} + (J_y - J_x)\omega_y\omega_x = M_z \end{cases} \quad (2)$$

$$\begin{cases} \frac{dq_0^*}{dt} = \frac{1}{2}(-\omega_x q_1 - \omega_y q_2 - \omega_z q_3) \\ \frac{dq_1^*}{dt} = \frac{1}{2}(\omega_x q_0 + \omega_z q_2 - \omega_y q_3) \\ \frac{dq_2^*}{dt} = \frac{1}{2}(\omega_y q_0 + \omega_x q_3 - \omega_z q_1) \\ \frac{dq_3^*}{dt} = \frac{1}{2}(\omega_z q_0 - \omega_y q_1 - \omega_x q_2) \end{cases} \quad (3)$$

$$\begin{cases} q_0 = \frac{q_0^*}{\sqrt{(q_0^*)^2 + (q_1^*)^2 + (q_2^*)^2 + (q_3^*)^2}} \\ q_1 = \frac{q_1^*}{\sqrt{(q_0^*)^2 + (q_1^*)^2 + (q_2^*)^2 + (q_3^*)^2}} \\ q_2 = \frac{q_2^*}{\sqrt{(q_0^*)^2 + (q_1^*)^2 + (q_2^*)^2 + (q_3^*)^2}} \\ q_3 = \frac{q_3^*}{\sqrt{(q_0^*)^2 + (q_1^*)^2 + (q_2^*)^2 + (q_3^*)^2}} \end{cases} \quad (4)$$

$$\begin{cases} \tan\psi = \frac{-2(q_1q_3 + q_0q_2)}{q_0^2 + q_1^2 - q_2^2 - q_3^2} \\ \tan\gamma = \frac{-2(q_2q_3 + q_0q_1)}{q_0^2 - q_1^2 + q_2^2 - q_3^2} \\ \sin\theta = 2(q_1q_2 + q_0q_3) \end{cases} \quad (5)$$

where q_0, q_1, q_2, q_3 are the quaternion, ω_x, ω_y, ω_z are the projection of angular velocity of rigid body on coordinate axis, ψ, γ, θ are yaw angle, inclination angle and pitch angles respectively.

In contrast to the system of equations of motion established by the directional cosine and Euler angle methods, the quaternionic method has outstanding advantages: Do not degenerate on any parameters, the number of parameters is only 4, the contact equation has only 1 [6].

The lower position machine uses a fourth-order Runge-Kutta method in the initial experiment to solve the system equations. The calculation results are smooth and highly realisti, but the computational rate is low. To improve the real-time nature of the computation, a second-order Runge-Kutta method is used later on the premise of ensuring smoothness and fidelity. The final results demonstrate that, in contrast to the Euler angle method, our proposed method combines the quaterniion method with the Runge-Kutta algorithm, the calculated time of the first-time pose information successfully reduce from about 60ms to 25ms, the computational rate was improved more than 2 times.
Both the engine model and the pneumatic data are determined by the specific UAV models, as known data. Procedures extract data from the corresponding interpolation table according to the atmospheric environment, flight status, throttle lever position and other states, interpolate current engine speed, thrust and aerodynamic coefficient.

3.2.2. Control mode and control law. The drone control mode includes manual control, instruction control, autonomous control. The system mainly adopts the instruction control mode. The control law used by the system is open-source, The control rules can be implanted according to the purpose of user development[7].The default control law is the classic PID Control law. The following are PID control law under three attitude angles of the command control mode.

1) Elevope angle holding and control loop closed loop control law:

$$\delta_e = P_\theta(\theta_r - \theta) + \int I_\theta(\theta_r - \theta)\,dt - D_\theta \omega_y \tag{6}$$

2) Roll angle holding and control loop closed-loop control law:

$$\begin{cases} \delta_a = P_\varphi(\varphi_r - \varphi) + \int I_\varphi(\varphi_r - \varphi)\,dt - D_\varphi \omega_x \\ \delta_r = D_r \omega_z \end{cases} \tag{7}$$

3) Yaw angle holding and control loop closed-loop control law:

$$\begin{cases} \varphi_r = P_\psi(\psi_r - \psi) \\ \delta_a = P_\varphi(\varphi_r - \varphi) + \int I_\varphi(\varphi_r - \varphi)\,dt - D_\varphi \omega_x \\ \delta_r = D_r \omega_z \end{cases} \tag{8}$$

where P, I, D are the proportional coefficients, integral and differential coefficients respectively. θ, r, ψ are the pitch angles, rolling angle and yaw angle respectively[8].

3.3. Programming of Flight Simulation Module

The Flight Simulation module is a multitask program, to ensure the reliability and real-time of the program operation, the operating system μC/OS-II is embedded in STM32F103, the microcontroller. The module mainly implements four functions: control signal acquisition, control model calculation, flight model calculation and serial port transmission of flight state data.

- Control signal acquisition task: Control signal is duty cycle from 5% to 10% adjustable 50Hz square wave signal. By configuring the control register of the microcontroller TIM(general timer), invokes its input capture function for pulse width acquisition.
- Control model calculation task: The program inputs the control amount and the flight state data at the previous moment as message, save the model calculation results in a preconfigured storage unit waiting for a call.
- The flight model computational task: Read the latest control model calculation results in the storage unit, input the flight mode, calculate current flight status, save the message, wait for the serial port transfer task extraction.
- Serial port transport task: The program calls the controller channel UASRT1 uploads the latest flight status data.

4. Design and Implementation of the UAV Flight 3D Visual Simulation Module

The 3D visual view simulation program is divided into three sub-module programs: the serial port communication module based on the multi-threaded and Windows customize message, the OpenGL3D scenario display module based on MFC and the UAV model establishment and import module.

4.1. The Serial Port Communication Based on Multi-thread and Windows Custom Messages

To ensure a higher asynchronous serial communication data transmission rate, at the same time, in order to ensure the flexible data interaction between the master and owner threads, the program builds on using shared storage for high-speed communication function between the threads by custom messages.

4.2. UAV Model Establishment and Lightweight

In the 3D visual simulation module, there are two problems for directly displaying 3D drone models in OpenGL:

1) Display the 3 D model with OpenGL code, the programming process is complex and the drone model quality is low;

2) It takes more time to load and update the models. Comprehensive consideration, the module uses 3 D software 3DMAX to build the drone model, ensure the quality of the model. At the same time, for the requirements of fluency display for the system, the drone model needs to be lightweight. We use Milkshape3D converts the model into a thin file format readable by the 3D View program[9].Figure 3 show the results of UAV model format transformation.

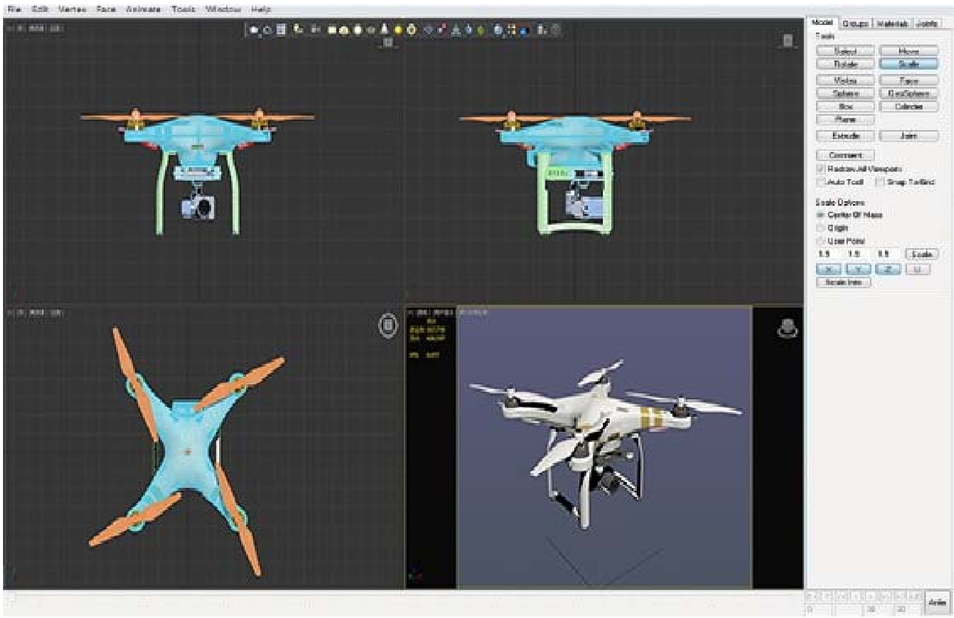

Figure 3. Modelling and Lightweight of the drone model.

4.3. The OpenGL 3D Scene Display Based on MFC

OpenGL is a high-performance 3D graphics processing system developed by SGI company, It is an open graphics software library, combined with Visual C++ programming, can implement relevant 3D objects computational and graphic algorithms, especially suitable for high-quality 3D dynamic display[10-11].

There are no window management functions in OpenGL library, so in Windows environment, build a 3D scene with OpenGL need to put window system with OpenGL. It can then be used to show and process the 3D scene[12]with OpenGL functions. MFC includes rich Windows application framework, Window and event management functions, so the combination of MFC and OpenGL can very well meet the requirements of the module.

When the 3D scene module display the image, the focus is on completing terrain and sky model building and flight state scenario-driven tasks. The specific implementation method is as follows:

The UAV model introduced in the 3D vision program will be moved according to the flight state data rules calculated by the flight simulation program. The motion of a 3D object in space can be described by its translation, rotation, and transformation matrix relations. Module Will UAV Flight state data is converted to the main thread OnDraw ( ). Translation function in the function glTranslate ( ), And the rotation function glRotate ( ), And the input parameter of the matrix transformation relation function, So as to realize the display and update of the UAV flight status[13].Figure 4 shows the flight of the 3D view program in the drone.

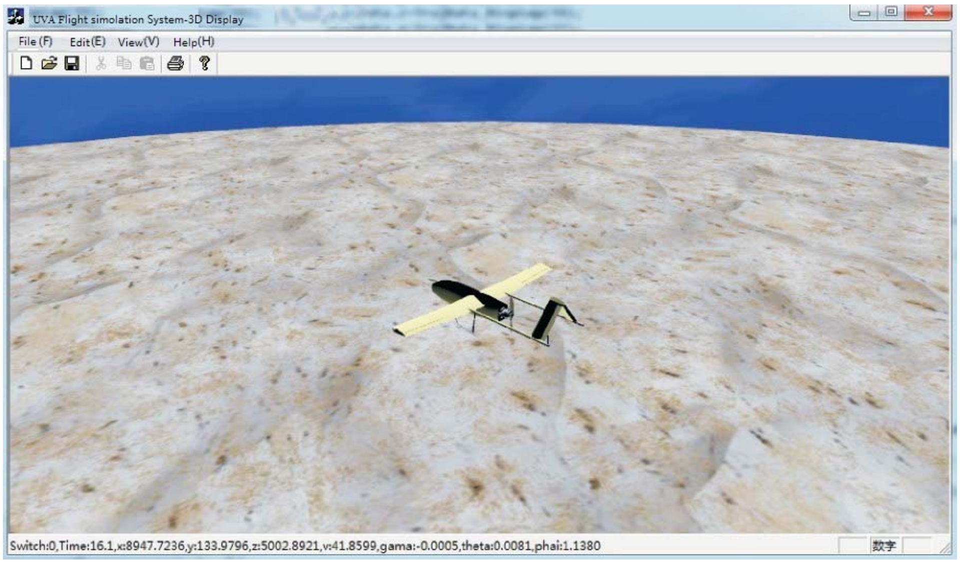

Figure. 4. The 3D visual display interface.

5. Management and Processing for the System Data

The system will have a lot of data generation and processing, with a module. Use the SQL Server database platform. SQL Server is a relational database management system that was developed by Microsoft Inc. SQL Server database has many advantages, making it the first in the database field and becoming the most popular database system. It graphics the user interface and makes system management and database management more intuitive and simple. Rich programming interface tools provide greater options for user programming and excellent scalability across multiple platforms from desktop computers running Windows to servers with large multiprocessors. Its support for Web technology makes it easy for users to publish data in the database to the Web page. It is easy to use, secure and available, fast performance, with an integrated development environment, running in a Windows operating system environment.

6. System Simulation Results

The 3D vision program contains functions to record flight state parameters in real time, these parameters record the results into the system's database for unified processing and management, to facilitate the later quantitative analysis of the data.

Figure 5 to figure 7 is the analysis curve of the results of the flight experiment, the traces of centroids of the UAV are shown separately, Speed change and attitude-angle response.

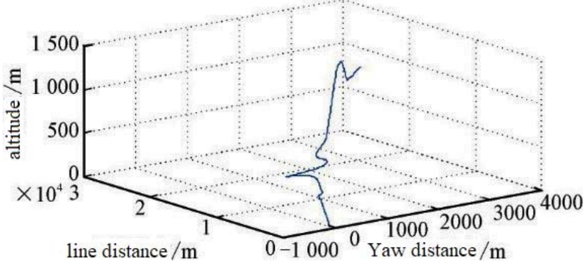

Figure 5. Track data curve.

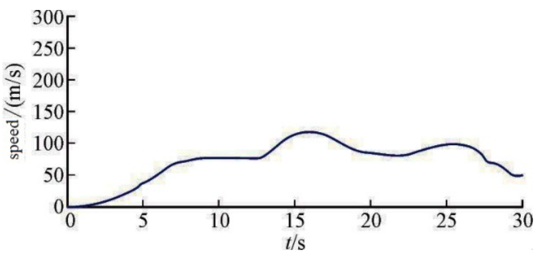

Figure 6. Speed response curve.

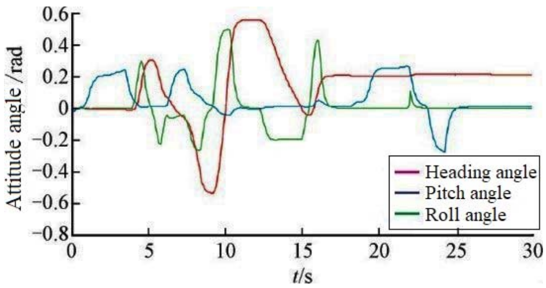

Figure 7. Posture-angle response curve.

Figure 8 is the main interface of the overall system for a real-time simulation of the UVA flight process.

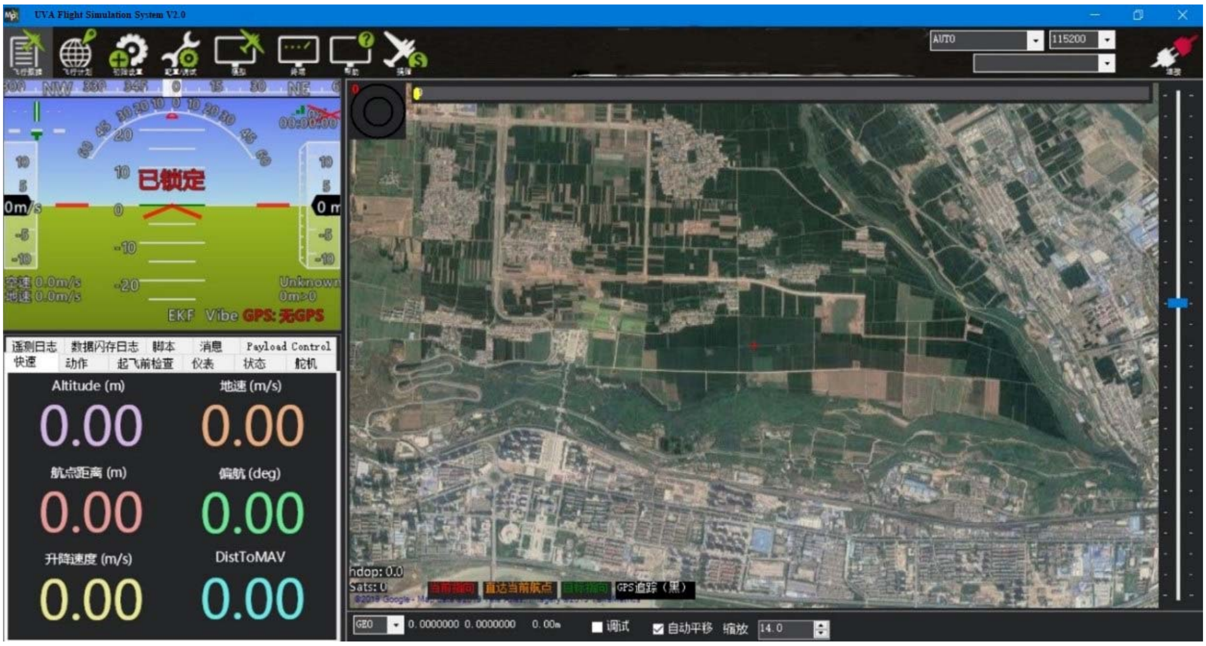

Figure 8. Main interface of the system.

7. Conclusions

This is known from the experimental data analysis, it is feasible to realize the visual flight simulation of the UAV by combining with 3D visual simulation and embedded development based on ARM. The method discussed in this paper lays the technical foundation for making physical UAV flight controllers.

- To develop an open-source small and medium-sized fixed-wing UAV flight controller, The embedded UAV flight simulation system based on ARM is designed as a trantransition system for flight controller. The system implanted the traditional flight simulation process into an embedded microprocessor, the real-time flight state solution is realized.
- For dynamic, vividly show the results of the microcontroller calculating the UAV flight state, exploitation Developed a 3D OpenGL view simulation program that can be quickly update the drone flight data.
- The platform provided by this paper is also applicable for the flight control development of other unmanned aircraft vehicles. At the same time, build different flight models and control models, this system can also be expanded to other types of control methods and flight object.

Acknowledgments

This research is supported by the Open Research Project of the National Key Laboratory of Virtual Reality Technology and Systems VRLAB 2020C07 and the Major science and technology R&D project "civil large bypass ratio turbine engine unit delivery" of Sichuan Province (2019YFG0384).

Journal of Physics: Conference Series **2218** (2022) 012081 doi:10.1088/1742-6596/2218/1/012081

References

[1] Yang Zhong, Shen Chunlin, Liu Jiufu. A UAV New technology for the virtual prototype design and verification of the flight controller [J]. Information and Control, 2005, 34 (1):81-85.
[2] Li Wenqiang, Peng Xuefeng, Zheng Zhiqiang. Six degrees freedom UAV simulation based on Simulink [J]. Systematic Simulation Journal, 2007, 19 (19): 4604-4606.
[3] Li Chao, Wang Jiangyun, Han Liang. Development of the flight simulation system of a fixed-wing aircraft based on Matlab [J]. Systematic Simulation Journal, 2013, 25 (8):1772-1777.
[4] Huang Hua, Xu Youping, Deng Zhiwu. A real-time visualization flight simulation system based on Flightgear simulator [J]. Systematic Simulation Journal, 2007, 19 (19): 4421-4423.
[5] Liu Xiongxiong, Zhang Weiguo, Huang Yijun, et al. Research on Advanced UAV Six Degree of Freedom Motion Modeling Technology [J]. Arrow and guidance paper, 2005, 25 (3):431-433.
[6] Yuan Zihuai, Qian Xingfang. Controlling flight mechanics and computer simulation [M]. Beijing: National Defense Industry Press, 2001: 38-80.
[7] Li Jun. Design and development of flight control software for small fixed-wing UAV [D]. Nanjing: Nanjing University of Aerospace Science and Astronautics, 2011.
[8] Hu Qing. Design of UAV flight control system for single-chip machine Based on STM32 [D]. Nanjing:Nanjing University of Aerospace Science and Astronautics, 2012.
[9] Ye Ge, Tian Zhaofeng, Yan Chuliang. Visualization of aircraft flight data based on OpenGL [J]. Aviation Journal, 2011, 32(6): 1050-1057.
[10] Payneb. Programming 3D Graphics with OpenGL [M] Houston: Rice University Press. , 2010: 325-362.
[11] Li Lirong, Shen Chunlin, Lu Yuping, et al. The 3D visual simulation system of ultra-low altitude/Intercept aircraft based on OpenGL [J]. Journal of Nanjing University of Aeronautics and Astronautics, 2005, 37 (3): 269-273.
[12] Hu Wei, Wang Hong. The OpenGL Programming in VC + +with MFC [J]. Computer application, 2001, 21 (8): 87-89.
[13] Li Haijun, Li Jiejie, Liu Xiao, et al. A visual simulation of the missile attack process Based on OpenGL [J]. Systematic Simulation Journal, 2004, 16 (3): 530-533.