Hardware-in-the-Loop Simulation Platform for UAV Based on dSPACE

Hong-Mei ZHANG [1,a], Qiang ZHOU[2,b,*] and Guang-Yan XU [3,c]

[1,3]School of Aerospace Engineering, Shenyang Aerospace University, Shenyang, 110136, China

[2]School of Automation, Shenyang Aerospace University, Shenyang, 110136, China

[a]zhang.hongmei@163.com, [b]zhouqiangxt@163.com, [c]guangyan_xu@163.com

*Corresponding author

Keywords: dSPACE, Real-Time Simulation, RTW.

Abstract. In order to quicken the process of designing the Unmanned Aerial Vehicle (UAV) flight control system, this paper develops a hardware-in-the-loop Simulation (HILS) system of UAV flight control to complete the whole process of simulation from modeling to real-time code generating. Firstly, nonlinear six-degree-of-freedom (6-DOF) UAV model, environment model and sensor simulation model are built in the Matlab/Simulink environment. Then, the HILS system of UAV flight control system model based on dSPACE is built using the Real-Time Interface (RTI), and generates the C code with Real-Time Workshop (RTW). Secondly, the platform manager ControlDesk provided by dSPACE is used to develop the software of simulation control and data view. Finally, the software of path display and flight visual are developed using the C Interface Library (CLib). The result of the software design and test UAV dynamic model shows that the degree of automatization for developing the HILS system is increased and the cost and time used is decreased greatly.

Introduction

The approach of traditional flight simulation uses Simulink module construct UAV system model and implement simulation in Matlab/Simulink environment, and then uses C language reconstruct the whole system, finally, executes HILS or man-in-the-loop simulation. Because the manual programming takes up a lot of time, and generates code is not reliable. If there is a problem in the testing process, it is difficult to confirm whether the design scheme or the code is wrong. Germany dSPACE Company develops a real-time simulation system and provides a good solution. The dSPACE real-time system realizes the seamless connection with Matlab/Simulink which makes the Simulink model generate C code directly, and then executes real-time simulation. Due to the good interactive environment, the unreliable of manual programming and the waste of time are solved, and makes the developers only focus on simulation scheme and control algorithm.

Over the past few years, National Aeronautics and Space Administration have been using dSPACE for UAV rapid prototyping[1], flight control method verification[2], and control law testing of harsh conditions[3]. Columbia University uses dSPACE for developing Quad Rotor[4]. This article is based on dSPACE platform to achieve UAV rapid prototyping, and provide a good development environment for UAV electronic equipment and flight control system.

Platform design of simulation system

The overall structure of the UAV hardware-in-the-loop simulation system is shown in Fig.1. The manager computer of dSPACE is responsible to build the UAV model, create man-machine interface and develop application based CLib; The ground control station not only receives flight data to monitoring flight status and planning path, but also sends control command to the flight control computer through RS232; The actuator dispatch controlpiston deflexion angle signal to the UAV dynamic model through A/D collection card; The UAV dynamic model simulates the response process of real aircraft and real-time calculates attitude and position information according to the input signal, then transmits airspeed and altitude, position information to the corresponding atmospheric data simulation system and GPS simulator through RS232, and send Euler angle to the turntable computer through RS422; The GPS simulator receives location information and pack information according to NEMA-0183 agreement, and then sent it to flight control computer; The atmospheric data simualtion system converts altitude and airspeed to corresponding dynamic pressure and static pressure; The turntable computer transfers attitude information to the three-axis turntable and the Attitude and Heading Reference System (AHRS) is fixed by the turnable; The AHRS calculate attitude information and send it to the flight control system. The flight control computer receives data from each simulator and sensors, and calculates the output signal with its own control law. Now, the simulation platform forms a closed-loop control system. This paper mainly introduces the realization of the UAV model in the dSAPCE and the development of application by using CLib API.

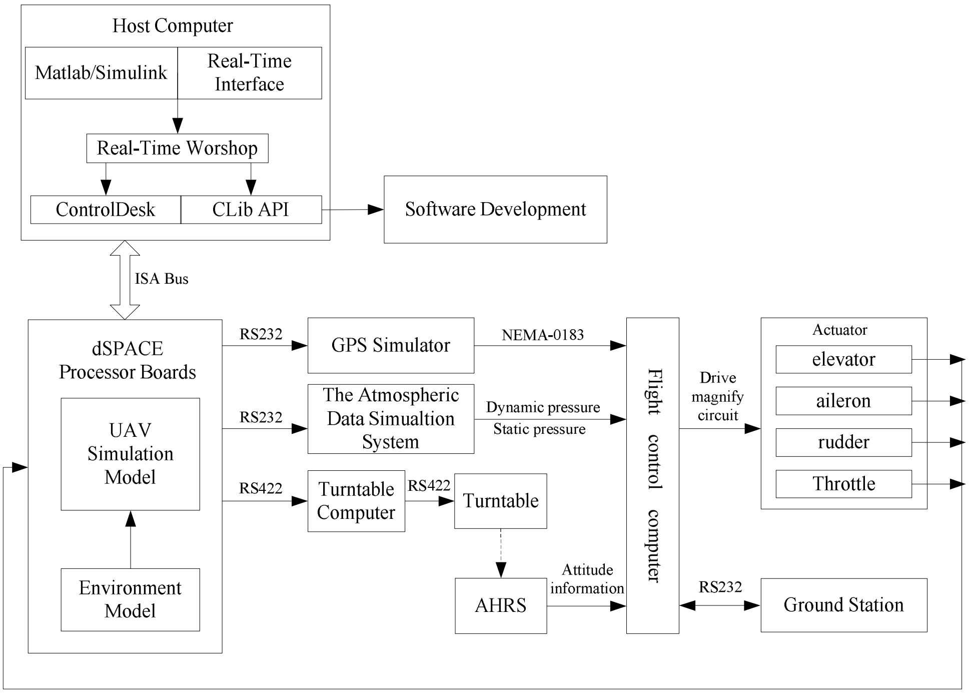


Fig. 1 The overall structure of the UAV flight control system

Simulation Mathematical Model Establish

Establishing the UAV mathematical model is the basis for HILS[5]. The standard 6-DOF equations of motion for a conventional aircraft are used for modeling and simulation of a small Fixed-Wing UAV. Flat Earth approximation provides a reasonable modeling assumption when the vehicle operates over a small area. The body-axes equations are as follows:

$$\begin{cases} \dot{u} = vr - \omega q - g\sin\theta + F_x/m. \\ \dot{v} = -ur + \omega p + g\cos\theta\sin\phi + F_y/m. \\ \dot{\omega} = uq - vp + g\cos\theta\cos\phi + F_z/m. \end{cases} \tag{1}$$

$$\begin{cases} \dot{p}I_x - (\dot{r} + pq)I_{xz} + qr(I_z - I_y) = \overline{L}. \\ \dot{q}I_y + pr(I_x - I_z) + (p^2 - r^2)I_{xz} = M. \\ \dot{r}I_z - (\dot{p} - qr)I_{xz} + pq(I_y - I_x) = N. \end{cases} \tag{2}$$

$$\begin{cases}\dot{\phi} = p + \tan\theta(r\cos\phi + q\sin\phi). \\ \dot{\theta} = q\cos\phi - r\sin\phi. \\ \dot{\psi} = (r\cos\phi + q\sin\phi)/\cos\theta.\end{cases} \tag{3}$$

$$\begin{cases}\dot{X}_g = u\cos\theta\cos\psi + v(-\cos\theta\sin\psi + \sin\phi\sin\theta\cos\psi) + w(\sin\phi\sin\psi + \cos\phi\sin\theta\cos\psi). \\ \dot{Y}_g = u\cos\theta\sin\psi + v(\cos\theta\cos\psi + \sin\phi\sin\theta\sin\psi) + w(-\sin\phi\sin\psi + \cos\phi\sin\theta\sin\psi). \\ \dot{Z}_g = u\sin\theta - v\sin\phi\cos\theta - w\cos\phi\cos\theta.\end{cases} \tag{4}$$

where Eq. 1 to Eq. 4 are the *force*, *momental*, *kinematic*, and *navigation* equations respectively. $F_x, F_y, F_z, \bar{L}, M, N$ refer to the corresponding components of the *force* and *moment* in the airframe axis; u, v, ω refer to the corresponding components of the *centroid speed* in the airframe axises; I_x, I_y, I_z are corresponding *moments of inertia*; I_{xz} is the *product of inertia*; $\dot{X}_g, \dot{Y}_g, \dot{Z}_g$ are coordinates of the centroid in geocentric coordinate system; θ, ψ, ϕ are the angles of *pitching*, *yaw* and *roll*; p, q, r are the corresponding *rotational angular velocity.*

In the air, the UAV is mainly suffered from the aerodynamic and engine thrust. Under taking the engine mounting angle into account, the force can be depicted as:

$$\begin{bmatrix} F_x \\ F_y \\ F_z \end{bmatrix} = \begin{bmatrix} \cos\alpha\cos\beta & -\cos\alpha\cos\beta & -\sin\alpha \\ \sin\beta & \cos\beta & 0 \\ \sin\alpha\cos\beta & -\sin\alpha\sin\beta & \cos\alpha \end{bmatrix} \cdot \begin{bmatrix} \text{-}D \\ Y \\ -L \end{bmatrix} + G \begin{bmatrix} -\sin\theta \\ \cos\theta\sin\phi \\ \cos\theta\cos\phi \end{bmatrix} + T \begin{bmatrix} \cos\sigma \\ 0 \\ -\sin\sigma \end{bmatrix} \tag{5}$$

where, D, Y, L are the forces of *drag*, *yaw* and *lift* respectively; G and T are the *gravity* and *thrust* of the UAV. σ refers to the engine mounting angle. The forces D, Y, L and the moment $\bar{L}, M, N$ are computed by using the following Eq. 6 and Eq. 7:

$$D = \bar{q}SC_D, \qquad Y = \bar{q}SC_Y, \qquad L = \bar{q}SC_L \tag{6}$$

$$\bar{L} = \bar{q}SbC_R, \qquad M = \bar{q}ScC_M + T\sigma, \qquad N = \bar{q}SbC_N \tag{7}$$

where, $\bar{q}, b, c, S$ are the corresponding *dynamic pressure*, *wingspan*, *aerodynamic chord*, and *wing area*; C_D, C_L, C_Y are the coefficients of *drag*, *yaw* and *lift* force*;* C_R, C_M, C_N are the coefficients of *roll*, *pitch* and *yaw* moment.

So far, from Eq. 1 to Eq. 7, the UAV real-time flight attitude information can be obtained.

The application of simulation platform

The simulink model of UAV flight control system which based on the dSPACE integrated development environment is constructed and shown in Fig. 2. All the modules of the UAV system are packaged separately for clear modeling and convenient modification. The UAV dynamics module is established by using 6-DOF equations of motion; the aerodynamic coefficient module using real experiment data saved in the *mat* file; the environment module is divided into three parts, the atmosphere module using *1976COSEA* American standard atmosphere model, the wind module using *MIL-HDBK-1797 discrete wind model*, and the gravity module using *WGS1984* world gravity system model[6]. After setting up the basic simulation loop, the Pilot and Sensor module uses the RTI to map each input and output signal to specific serial interface board.

Before generating real-time code, the item of *configuration parameters* in simulink needs to be corresponding configured. In the *Solver* option, because the RTW doesn't support variable step integral algorithm, the simulink solver type is chosen to be *fixed-step*, and the solver algorithms should be chosen the *Runge-Kutta integration method*; in the *Code Genration* option, the system target file is *rti1103.tlc*, the make command is *make_rti*, and the template makefile is *rti1103.tmf*. And then by clicking on *Build model* in the RTW, the *System Description File* (*SDF*) and *Variable Description File* (*VDF*) which correspondence with the UAV model are generated. The *SDF* is used to download C code to dPSACE processor and the *VDF* will be used in developing software.

After the simulink model successfully complied, the platform management software *ControlDesk* provided by dSPACE is used to verify UAV dynamic model. The *ControlDesk* providing professional virtual aviation instruments and data acquisition tools are employed to establish *the Real-time Platform of the UAV* shown in Fig.3. By dragging the flight variables to the corresponding virtual instruments, the visual management of flight data is achieved. This platform can monitor, record and change UAV flight data in real time.

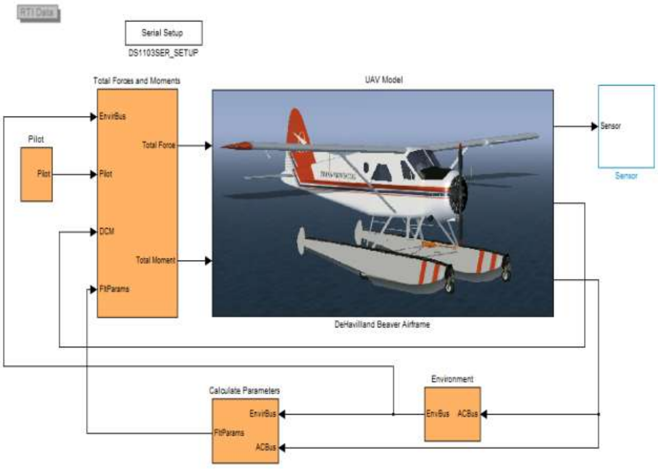
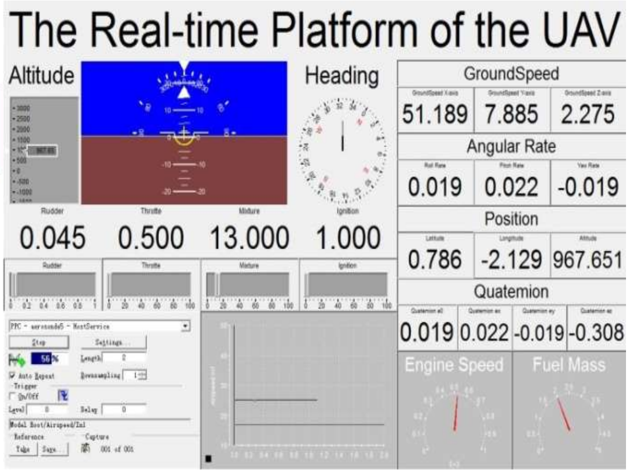


Fig. 2 Simulink mode


The Real-time Platform of the UAV

Altitude

Heading

GroundSpeed X-axis	GroundSpeed Y-axis	GroundSpeed Z-axis
51.189	7.885	2.275

Angular Rate

Roll Rate	Pitch Rate	Yaw Rate
0.019	0.022	-0.019

Position

Latitude	Longitude	Altitude
0.786	-2.129	967.651

Quatemion

Quaternion e0	Quaternion ex	Quaternion ey	Quaternion ez
0.019	0.022	-0.019	-0.308

Rudder	Throttle	Mixture	Ignition
0.045	0.500	13.000	1.000

Engine Speed

Fuel Mass

PPC - aerosonde5 - HostService

Stop Settings...

56%

Length 2

☑ Auto Repeat

Downsampling 1

Trigger

□ On/Off

Level 0

Delay 0

Model Root/Airspeed/In1

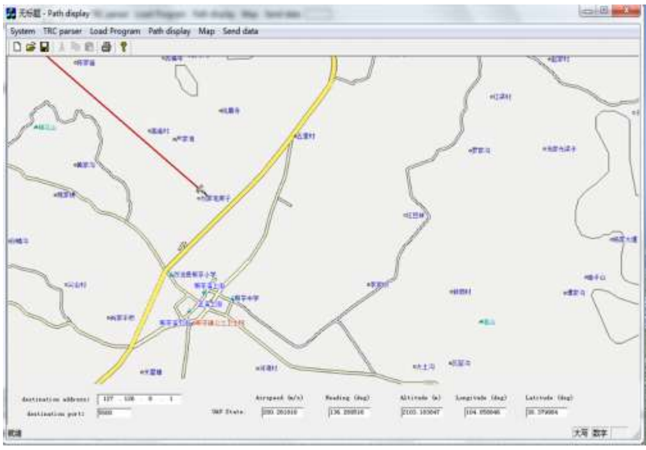
Reference Capture

Take Save... 001 of 001

Fig. 3 Real-time simulatin platform

Although *ControlDesk* can quickly establish an application to monitor and manage simulation experiment, it just achieves the simple visualization management of flight data and cannot communicate with other application. While the dSPACE real-time system is not a closed system and provides application programming interface that is *CLib*, so that PC can communicate with dSPACE real-time hardware. The *CLib* library provides more than 20 functions that can complete common operations, such as program registration and cancellation, board resources application, reading and writing memory data. So it is used to develop the specific requirements application. Using *CLib* to read and write a variable value in the dSPACE memory, which needs to provide the corresponding variable address in the function and the address also needs to provide a variable name to obtain. However the variable name stored in dSPACE is no longer the original name of the simulink environment, because the simulink module and the RTI module are compiled to generate a new variable name stored in *VDF* file. The *VDF* file uses ASCII character code to describe the simulink variable information and simulink model structure. Then a C++ class is used to parse the *VDF* file in order to map the new variable name to the corresponding original variable name. Finally, Socket programming method and Clib API function are employed to develop the software under Microsoft Visual C++6.0 environment, which is shown in Fig. 4. The function of the software is to display the flight path and as *flight visual* client which is used to communicate with the famous flight simulation software *FlightGear*[7] through *user datagram protocol*(UDP).

The *FlightGear* is used as the *flight visual* server and provides a class *FGnetFDM* found in Ref.6 that defines the data style of each variable. So the flight data that read from dSPACE memory are packaged accroding to *FGnetFDM* and sent to *FlightGear*. The *FlightGear* must be started with the appropriate command line arguments to interact with the upper computer software. In here, the command line *fgfs--native--fdm='socket, in, 50, 22221, udp'* is used to start *FlightGear* for receiving flight data. The Flight visual is shown in Fig. 5.



destination address: 127 . 126 . 0 . 1

destination port: 5500

UAV State: Airspeed (m/s) 200.261610 Heading (deg) 136.260516 Altitude (m) 2103.183047 Longitude (deg) 104.050046 Latitude (deg) 30.379984

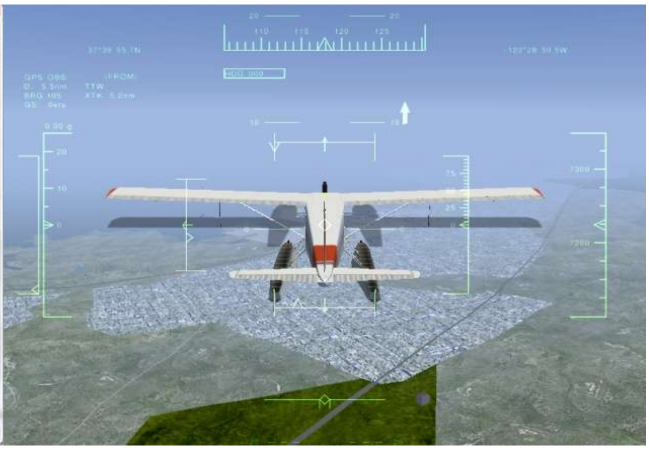


Fig. 4 Path display

Fig. 5 Flight visual

Summary

The build process of a UAV HILS platform based on dSPACE is described in detail. In order to improve the function of the HILS system, the dSPACE Clib and *FlightGear* flight simulator are used to develop the flight path display and flight visual software, which allow developers to get more intuitive UAV flight status information. Finally, the UAV dynamics model is verified under the HILS platform. The experiment results shows HILS platform can conveniently and quickly build UAV dynamic model and shorten the development cycle of UAV system, and the software design based CLib and ControlDesk which provides a realistic environment to verify UAV system model. The HILS platform is a relatively ideal integrated development environment which can be used to test UAV flight control system and develop avionics in future.

Acknowledgement

This research was financially supported by general project of Liaoning Education Department under Grant L2013075.

References

[1] T. L. Jordan, J. V. Foster, R. M. Bailey, AirSTAR: A UAV Platform for Flight Dynamics Control System Testing, AIAA Guidance, Aerodynamic Measurement Technology and Ground Testing Conference, San Francisco, California, 5-8 June, 2006-3307.

[2] A. M. Murch, D. E. Cox, K. Cunningham, Software Considerations for Subscale Flight Testing of Experimental Control Laws, Aerospace Conference, Seattle, Washington, 6-9 April, 2009-2054.

[3] T. L. Jordan, R. M. Bailey, NASA Langley's AirSTAR Testbed - A Subscale Flight Test Capability for Flight Dynamics and Control System Experiments, AIAA Guidance, Navigation and Control Conference and Exhibit, Honolulu, Hawaii, 18-21 August, 2008-6660.

[4] M. Chen, M. Huzmezan, A Combined MBPC/2 DOF H_∞ Controller for a Quad Rotor UAV, AIAA Guidance, Navigation, and Control Conference and Exhibit, Austin, Texas, 11-14 August, 2003-5520.

[5] D. Jung, P. Tsiotras, Modeling and Hardware-in-the-Loop Simulation for a Small Unmanned Aerial Vehicle, 2007 AIAA InfoTech at Aerospace Conference, Rohnert Park, California, 7-10 May, 2007-2768.

[6] G. Avanzini, G. Matties, Model and Simulation of A Shrounded-Fan UAV for Environment Monitoring, Technical Conference and Workshop on Unmanned Aerospace Vechicle, Portsmouth, Virginia, 20-23 May, 2002-3464

[7] E. F. Sorton, S. Hammaker, Simulated Flight Testing of an Autonomous Unmanned Aerial Vehicle Using FlightGear, Infotech at Aerospace, Arlington, Virginia, 26-29 September, 2005-7083.