

A Low Cost Method for Establishing Hardware-In-Loop Platform for Aviation Piston Engine Electronic Controller

YU Bing^{1,a}, DONG MingChun^{1,b} and Bian WenChao^{2,c}
*1*Jiangsu Province Key Laboratory of Aerospace Power System,
Nanjing University of Aeronautics and Astronautics, Nanjing 210016, China
*2*School of Aeronautics, Shandong Jiaotong University, Jinan, 250023, China
ayb203@nuaa.edu.cn, bdmc1120@126.com, c15251873901@163.com

Abstract

To avoid the risk of bench test and reduce the validation cost when developing an aviation engine electronic controller, a low-cost method to build a hardware-in-loop platform, which includes an interface simulator and engine model system, is proposed. An engine model is built and validated in MATLAB via simulink. Then, the model is transplanted into a C/C++ development platform through RTW code automatic generation technology. Furthermore, an interface simulator employed to exchange information between engine model system and electronic controller is designed. Finally, the hardware-in-loop simulation platform is built with electronic controller and hardware-in-loop platform. The experiment's results show that the hardware-in-loop platform is efficient, which can meet the requirements of hardware-in-loop validation for the electronic controller with low cost.

Keywords: aero-engine; hardware-in-loop; software engine; electric controller; interface simulator

Introduction

Hardware-in-loop experiment is an efficient method to validate the aviation engine electronic controller[1]. Engine system simulator taking the place of the real one can avoid the risk of bench test and reduce the validation cost in the process of development[2,3]. There are many existing solutions for such applications, such as LabCar, dSPACE, ADI and so on. However, these systems are all too expensive, and unsuitable for some low weight situations. Unlike the above approach, a last and efficient method with low cost is proposed. An engine model is built and validated in MATLAB via simulink firstly. Then, the model is transplanted into a C/C++ development platform through RTW code automatic generation technology. Furthermore, an interface simulator employed to exchange

information between engine model system and electronic controller is designed. Finally, the hardware-in-loop simulation platform is built with electronic controller and hardware-in-loop platform. The experiment's results show that the hardware-in-loop platform is efficient, which can meet the requirements of hardware-in-loop validation for the electronic controller with low cost[4-6].

1. Establishment of Engine Model

Modeling the engine is the principal task to establish the hardware-in-loop platform[7], the schematic diagram of engine model is shown in Figure 1. The piston engine model is consisted of three main modules: air intake system; fuel system; combustion and power output system.

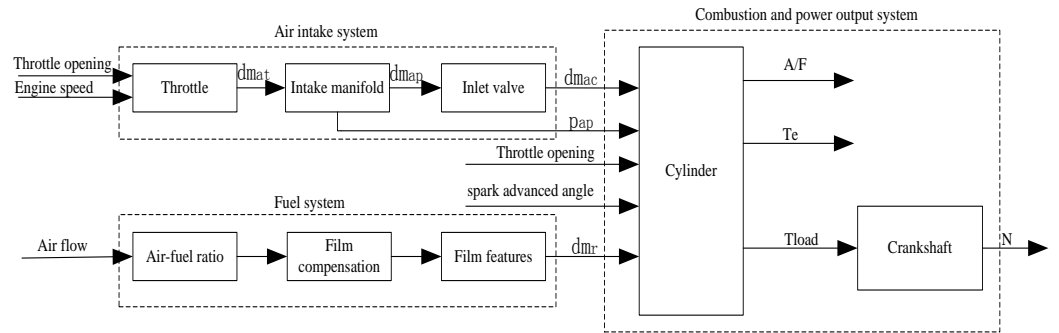


Figure 1 Schematic diagram of the engine model

1.1 Air intake system model

The intake system model includes the throttle air flow model, intake manifold pressure model and intake valve air flow model. For throttle air flow model, the model divides the air flow at throttle into two parallel master-slave isentropic flows, which converging in the back of the throttle body[8], the model is presented as follows:

$$\dot{m}_{at} = \frac{P_{amb}}{\sqrt{T_{amb}}} \beta_1(\alpha) \cdot \beta_2(p_r) \quad (1)$$

$$\beta_1(\alpha) = 1 - \varphi_1 \cos(\alpha) + \varphi_2 \cos^2(\alpha) \quad (2)$$

Where \dot{m}_{at} is the Mass flow, P_{amb} is the air pressure, T_{amp} is the air temperature.

$$\beta_2(p_r) = \begin{cases} \sqrt{p_r^{p_1} - p_r^{p_2}} / p_n, & \text{if } p_r \geq p_c \\ 1, & \text{if } p_r < p_c \end{cases} \left(p_c = \left(\frac{p_1}{p_2} \right)^{\frac{1}{p_2 - p_1}}, p_n = \sqrt{p_c^{p_1} - p_c^{p_2}} \right) \quad (3)$$

Where $P_1, P_2, \varphi_1, \varphi_2, P_c, P_n$ are constants used in the simplified process of the throttle valve air flow model. These constants are independent of the specific throttle structure parameters.

Given the input and output mass air flow into the intake are \dot{m}_{at} and \dot{m}_{ac} respectively, the air flow \dot{m}_{ap} inside the intake manifold can be calculated by the following formula:

$$\dot{m}_{ap} = \dot{m}_{at} - \dot{m}_{ac} \quad (4)$$

Considering the ideal gas state equation $pV = mRT$, air quality inside the intake

manifold can be obtained by $m_{ap} = \frac{p_{ap} V_{ap}}{RT_{ap}}$, derivation on both sides:

$$\dot{m}_{ap} = \frac{V_{ap}}{RT_{ap}} \dot{p}_{ap} - \frac{p_{ap} - V_{ap}}{RT_{ap}^2} \dot{T}_{ap}$$

(5)

Since the change of air temperature in the intake manifold is little, it can be neglected comparing with that of air pressure in intake manifold, resulting $\dot{T}_{ap} \approx 0$. Substituting equation (5) into equation (6), an intake manifold pressure model expression is obtained:

$$\dot{p}_{ap} = \frac{RT_{ap}}{V_{ap}} (\dot{m}_{at} - \dot{m}_{ac}) \quad (6)$$

In order to establish a intake valve air flow model with good portability and avoid collecting a lot of experimental data to identify and analysis for different engines, a common intake valve air flow model is adopted, namely to use speed-density method to establish air flow model for four-stroke engine, expression is as follows:

$$\dot{m}_{ac} = \frac{NV_s p_{ap} \phi_v}{120RT_{ap}} \quad (7)$$

Integration on equation (8) in the engine intake stroke, intake air mass of the engine single cylinder in each work cycle can be obtained:

$$m_{ac} = \int_{i_o}^{i_c} \dot{m}_{ap} dt \approx \frac{NV_s p_{ap} \phi_v}{120RT_{ac}} \frac{30}{N} = \frac{V_s}{4RT_{ac}} (s_i p_{ap} - y_i)$$

(8)

Fuel system model

As to Fuel system model, the methods from Elbert Hendricks and Thomas Vesterholm are adopted. The fuel transfer dynamic characteristics with one order inertial is presented as follows:

$$\begin{cases} \dot{m}_{ff} = \frac{dm_{ff}}{dt} = -\frac{m_{ff}}{\tau_{ff}} + x \cdot \dot{m}_{fi} \\ \dot{m}_{fv} = (1-x)\dot{m}_{fi} \\ \dot{m}_f = \dot{m}_{fv} + \frac{m_{ff}}{\tau_{ff}} \end{cases} \quad (9)$$

In the formulas, m_{ff} is the oil film quality, unit g ; \dot{m}_{fi} is the fuel mass flow ejecting from fuel injector, unit g/s ; \dot{m}_f is the actual fuel mass flow into the cylinder, unit g/s ; \dot{m}_{ff} is the fuel mass flow formed by oil film evaporation, unit g/s ; \dot{m}_{fv} is the fuel mass flow from fuel injector directly into cylinder, unit g/s ; τ_{ff} is time constant of oil film evaporation, unit s, generally taking 0.25~0.4s; x is the ratio of jet fuel deposition on the wall surface, generally taking 0.5~0.8. The values of τ_{ff} and x is associated with the engine speed and intake manifold pressure, and can calibrate by using the small deviation perturbation method.

1.2 Combustion and power output model

Engine power system operating state is modeled from the perspective of energy conversion and working medium flow. Engine output torque and power are calculated according to air amount and fuel amount entering the cylinders, and other important parameters (e.g. engine speed, the ignition advance angle, etc.). According to the law of conservation of energy and Newton's second law, energy change rate used to push the piston movement is the accelerating power of crankshaft rotation^[9]. Engine power output model is as follows:

$$\dot{\omega} = \left[H_u \eta_i \dot{m}_f (t - \tau_d) / \omega - (T_f + T_p + T_b) \right] / J \quad (10)$$

Transplantation of engine model

Engine model created in Matlab/Simulink is non-real-time, and human-machine interface functions have significant limitations. Semi-physical simulation test not only requires real-time performance for data acquisition and serial communication task, but also convenient human-machine interface and hardware control functions. VC development environment has the functions of high efficiency, flexible human-computer interface design and powerful hardware control. Matlab /Simulink and VC can not call each other directly. To solve this problem, the previous approach is to model the controlled object with Matlab/Simulink and verify its correctness, and then re-implement the entire system in VC. That is using programming language to realize modeling, control and simulation results recording. However, it is extremely difficult and tedious. Engine model in Matlab/Simulink is converted to the C++ code through the automatic code generation toolbox RTW in this chapter. To transplant the C++

code to VC development environment and do a little modification for the code, then the whole engine model system is formed by adding real-time control mechanisms and simulation interface. Validating the engine model under the same input conditions, it turns out that the results of the two simulation environment are consistent. Such programs give full play to the Matlab/Simulink software powerful modeling capabilities. It also plays the high timeliness, flexible human-machine interface design in VC++ development environment. This approach makes Matlab/Simulink model transit to VC development environment smoothly, and facilitates the integration and the secondary development of the system greatly.

For the establishment of aviation piston engine model simulation system in this paper, in order to call porting code conveniently, object code is loaded into VC development environment in the form of a dynamic link library, Then an MFC interface is built, simulation calculation module, data processing module, interface module and serial communication module is written. Next, multimedia timer is joined multimedia timer in the simulation calculation module to drive the model, so that the Matlab/Simulink engine model can run timely in the VC development environment. Specific development program is as shown Figure 2:

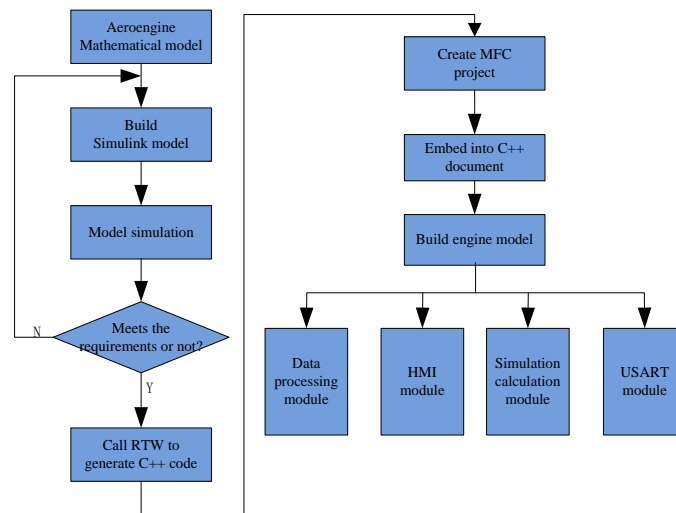


Figure 2 Engine model simulation system development program flowchart

2. Development of Interface Simulator

An interfaces simulator with good performance requires each subsystem division independently and cooperates with each other. In order to reduce mutual interference between subsystems and be easy for circuit error detection, modular design for the interface simulator is needed. The hardware circuit of interface simulator is divided into the minimum microprocessor systems, power modules, signal acquisition module, signal simulation module and communication module. Hardware circuit block diagram is described in Figure 3.

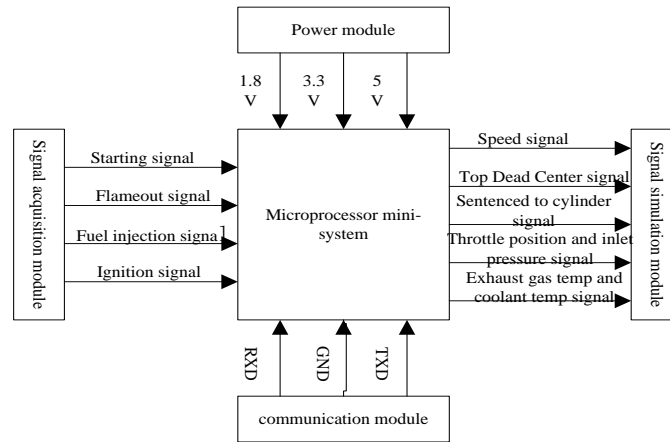


Figure 3 hardware circuit block diagram of interface simulator

Minimum microprocessor system manages and schedules every task, and ensures the program running in real time; Power module powers to other modules of interface simulator. External input is 24V, converted into 5V, 3.3V and 1.8V through various voltage converter chips; Signal acquisition module will isolate and convert the start-up, flame, fuel injection and ignition signals from the controller; Signal simulation module will simulate the speed, up-ending, sub-cylinder, throttle position, intake pressure, exhaust temperature and coolant temperature; Communication module will exchange data between interface simulator and engine model.

As the communication interface of "software engine", the microprocessor embedded on the interface simulator must have stability, temperature resistance and immunity. Control system processes a capacity of information and real-time requirements, therefore microprocessor must have a higher frequency. The whole engine control systems must have the function of A/D acquisition, the external interrupt acquisition, timer interrupt, SPI communication, SCI communication, PWM signal output, etc... So the microprocessor must have wealth I/O function. Form the point of energy consumption and price, we should choose low-power and low-cost microprocessor. Currently, the microprocessor architectures of engine control system are mainly 16-bit and 32-bit. The main production companies are TI, Intel, Motorola and Infineon. Considering the above factors, we take TI's 32-bit TMS320F28335 DSP chip as the microprocessor. The chip has a higher frequency, rich peripherals and strong anti-jamming capability, so that it is suitable as core unit of an interface simulator.

3. Research on Hardware-in-loop Simulation Test

Because the electronic control system of aviation piston engine requires high reliability, so in the development process, the stability and correctness of the electronic controller hardware and software is need to be examined. To optimize the system parameters and control rules, and analyze the fault handling capability of the electronic

control system. Hardware-in-loop simulation research plays an important role in engine electronic controller development process.

Test of electronic controller on hardware-in-loop simulation platform, can record the whole process. Using "software engine" can get data which is difficult to be measured in bench test. The calculation results can be carried in depth analysis on controller. At the same time, you can set fault point in the "software engine", for testing the controller's response to fault signal.

In hardware-in-loop simulation system, connecting the electronic controller with engine model system through the interface simulator to test and verify the accuracy of the model system under various working conditions. As a closed-loop testing system, the hardware-in-loop simulation system can verify the steady and transient state of the engine repeatedly, and have advantages in simulating the limit state and fault conditions.

3.1 Construction of Hardware-in-loop simulation test platform

The hardware-in-loop simulation system platform has following equipment: (1)electronic controller; (2) interface simulator; (3) computer installed with engine model system ; (4) computer installed with controller monitoring system; (5) switching stable power supply; (6)serial communication lines; (7) the wires ,etc..

Hardware-in-loop simulation system platform is formed through connecting the above equipment in correct order according to simulation test requirements. The connection of interface circuit and signal wires on hardware-in-loop simulation system platform is shown as Figure 4:

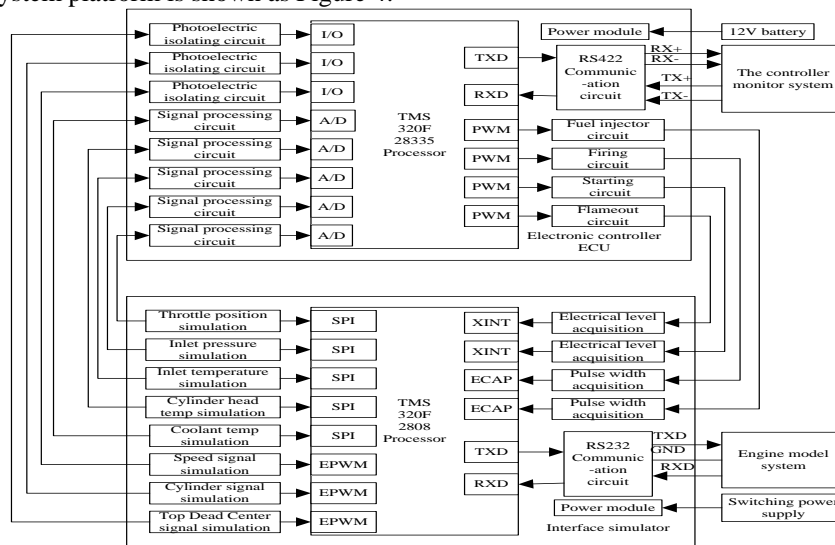


Figure 4 Hardware in the loop simulation system platform connection diagram

Interface simulator is connected to the engine model system through the RS232 serial line. The actuator control signals from electric controller ECU are sent to the engine model to regulate the engine operation state. And the state data got from operation of engine model is simulated into corresponding sensor signal for the controller to collect.

The electronic controller ECU collects throttle position, inlet pressure, exhaust temperature, coolant temperature and other sensor signals and outputs the control signal of the actuator through analyzing these above the signals. Meanwhile, sensor signal data and the output control signal data are sent to the monitoring system via RS422 serial line for data preservation and engine state monitoring. After the system platform connection is completed, we can conduct hardware-in-loop simulation test of aviation piston engine following the test procedures.

3.2 Hardware-in-loop simulation experiment

Through the previous introduction, the hardware and software designs required in hardware-in-loop simulation experiment have been completed gradually, including the engine model system, electronic controller, interface simulator and monitor calibration software. To link these hardware and software by signal lines, power lines and communication lines as Figure 4. The semi physical simulation platform is shown as Figure 5.

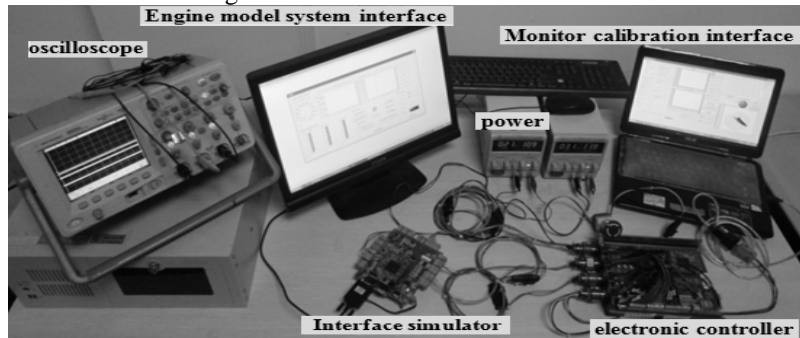


Figure 5 Hardware in the loop simulation system platform

After building the hardware-in-loop system platform, the performances of electric controller can be detected. In order to check the error and verify the controller roundly, simulation tests will be carried out step by step in this section. The order of the tests is as follows: speed pulse width signal, throttle opening, fuel injection pulse width, ignition advance angle, fuel injection phase and MAP verification. We conduct comprehensive test on the basis of the above tests correctly which is acceleration and deceleration test.

In acceleration and deceleration test, observe the changes about engine speed and fuel injection pulse width in the condition which is to set load as $60 N \cdot m$, throttle opening increasing from 10 to 45 deg in 1.5s, then reducing from 45 to 10 deg in 1.5s after the engine state is stable, throttle opening, speed, torque and fuel injection pulse width curves in the acceleration and deceleration condition are shown as Figure 6.

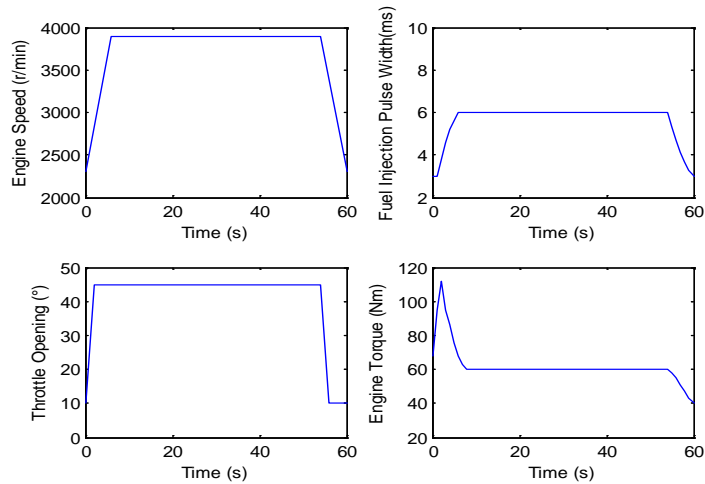


Figure 6 Throttle, speed, torque and fuel injection pulse width curves in the acceleration and deceleration condition

As shown in the above figures, when the throttle opening rises from 10 to 45 deg in 1.5s, then the engine speed rises from the 2300 *r/min* to 3900 *r/min* rapidly within 6s, and corresponding injection pulse width increases from 3ms to 6ms presenting an upward trend. The reason is that the mixed gas needs to be enriched when the engine accelerates. When the throttle opening decreases from 45 to 10 deg in 1.5s, the engine speed reduces rapidly from 3900 *r/min* to 2300 *r/min* within 6s, and then corresponding injection pulse width decreases from 6ms to 3ms presenting a downward trend. The reason is that the mixed gas needs to be diluted when the engine decelerates. The whole process reflects control strategies in the acceleration and deceleration conditions. However, the engine speed change is relatively slow, so we need to improve the dynamic response performance.

Conclusions

Based on the average model method, the mathematical model of aviation piston engine is established in MATLAB, and transplanted into VC by RTW toolbox to establish the engine model system. The “software engine” is formed by combining the engine model system and interface simulator. Hardware-in-loop simulation experiment shows that the “software engine” can simulate the real engine behavior process well. It improves the research and development efficiency of the electronic control system, and verifies that the electronic controller is feasible and able to conduct bench test. In addition, the method described in this paper has short development cycle and low development cost.

References

- [1] Isermann R, Schaffnit J. Hardware-in-the-loop simulation for the design and testing of engine-control system[J]. *Control Engineering Practice*. 1999. 7: 643~653.
- [2] Szuch J R. Advancement in Real-Time Engine Simulation Technology[J]. *I-8AIAA*. 1982: 82~1075.
- [3] Frangos C. Control system analysis of a hardware-in-loop simulation[J]. *IEEE Transactions on Aerospace and Electronic Systems*. 1990. 26(4): 666~668.
- [4] Geiger I, Theuerkauf H J, Woermann R J. Real-time Simulation of Gasoline Engine-A Development Tool for New Exhaust Emission Concept[J]. *SAE 901537*.
- [5] Chevalier A, Cristofaro F De, Ferrara F, et al. Benefits of Using a Real-time Engine Model During Engine ECU Development[J]. *SAE Paper 2003-01-1049*.
- [6] A.S.Lange. Semi-physical simulation of guide missile[J]. *Comp and Electr Eng I*. 1973(1):119~142.
- [7] Hendricks E, Sorenson S C. Mean Value Engine Modeling of Spark Ignition Engine[J]. *SAE Paper 900161*.
- [8] Hendricks E, Chevalier A, Jensen M. Modeling of the Intake Manifold Filling Dynamics[J]. *SAE960037*. 1996.
- [9] Hendricks E, Sorenson S C. SI Engine Control and Mean Value Engine Modeling[J]. *SAE Paper 930859*.