

Distributed and Real-time Integrated Simulation System on Avionics

Yaoming Zhou*

School of Aeronautic Science and Engineering, Beihang University, Beijing 100191, China

Yaolong Liu**

Institute of Aerospace Systems (ILR), RWTH Aachen University, Aachen 52062, Germany

Shaowei Li* and Yuhong Jia******

School of Aeronautic Science and Engineering, Beihang University, Beijing 100191, China

Abstract

In order to achieve iterative design in early R&D period, a Distributed and Real-time Integrated Simulation System for avionics based on a Model-Based Systems Engineering (MBSE) method is proposed. The proposed simulation system includes driver, simulation model, monitor, flight visual model and aircraft external model. The effect of this simulation system in iterative design and system verification is testified by several use cases. The result shows that the simulation system, which can play an important role in iterative design and system verification, can reduce project costs and shorten the entire R&D period.

Key words: Simulation System, Distributed, Model-based System Engineering, Avionics

1. Introduction

The rapid development of electronic and network technology has promoted the upgrading of avionics system. The new generation of integrated avionics system based on Integrated Modular Avionics (IMA) or Distributed Integrated Modular Avionics (DIMA) architecture is complex. As the amount of onboard SLOC (Single Line Of Code) grows exponentially, the development and maintenance costs of onboard code become the bottleneck of avionics system [1]. At the beginning of the system development, the defects and incompatibilities may be found through simulation to avoid higher project costs and schedule delay during the later period of the project, especially when the system is running [2]. Model-based system engineering (MBSE) can improve predictability at an early stage and reduces integration and maintenance effort greatly throughout the life cycle [2]. There are many simulation methods and tools based on MBSE mainly in the machinery and electronics fields. However,

the simulation technology has low-maturity in electronic systems and software systems [3]. Thus, it is unable to meet the requirements like real-time, dynamic and complex environments in the modeling and simulation of the avionics systems. It is of great importance to establish an open, distributed and real-time embedded integrated simulation system that implements high-fidelity integrated avionics simulation. Professor Peter Feiler from SEI of Carnegie Mellon creates Architecture Analysis and Design Language (AADL) that could realize the modeling and analysis in avionics system [4]. AADL is an extensible modeling language. The case of Thales verified the simulation of AADL model based on SystemC [5-7]. Ref. [8] gives avionics system unified lifecycle model which are the basis of avionics simulation.

This paper builds an integrated avionics simulation system based on SystemC kernel with open structure, distributed and real-time features. It can realize logic event or excitation response of the continuous control system under the unified timeline. The simulation process parameters will be monitored

This is an Open Access article distributed under the terms of the Creative Commons Attribution Non-Commercial License (<http://creativecommons.org/licenses/by-nc/3.0/>) which permits unrestricted non-commercial use, distribution, and reproduction in any medium, provided the original work is properly cited.

© * Assistant Professor
** Post-doctoral
*** Ph. D Student
**** Professor, Corresponding author: jjayuhong@buaa.edu.cn

in real time and outputted graphically. It can not only verify the correctness of basic system logic design in the early days, but also conduct a complex closed-loop continuous simulation with adding the aircraft model, the target model and the radar model. Besides, the man-machine interface such as the cockpit display, battlefield and visual flight simulation will be fully showed. All in all, the simulation system is designed to provide an effective means for designers to confirm system functionality and performance.

To describe the establishment and application of the integrated simulation system, this paper is divided into three sections, which are architecture and principle of the simulation system, simulation system implementation and simulation applications.

2. The architecture and principle of the simulation system

2.1 The architecture of the simulation system

This paper presents a distributed and real-time integrated simulation system for the R&D (Research and Development) stage of avionics system. It aims to achieve the transmission of the signal in avionics system and the scheduling of logical events under the unified timeline based on simulation kernel. This simulation system can achieve logic simulation during the early R&D period of avionics system and functional confirmation during the later period.

As shown in Fig. 1, Distributed and real-time integrated simulation system on avionic is mainly composed of several parts as follows.

Avionics simulation model consists of four parts: Simulation Kernel, Driver Model, Device Model and Monitor Model. Simulation Kernel includes timeline and signal library that consists of various signal packages so that it can achieve parallel scheduling of device signals and logical functions. Driver Model provides driving source for the entire simulation process, including model data and model

logic of driving. Monitor model can obtain the changes of signal during the simulation process, including model data and model logic of monitoring. Device Model can simulate various devices of Avionics system, which includes device model data and input and output signals. It has to be noted that within general avionics system, Device Model consists of multiple device modules.

The driver model is used to simulate real avionics incentive and is made up of three parts: Pilot manipulation module, simulated cockpit control module and user-defined input interface module.

Pilot manipulation module can simulate how to achieve access and input of an aircraft operating incentives under real flight environment. Simulated cockpit control module can simulate real aircraft cockpit control equipment and generate common cockpit's control signal. User-defined input interface module can customize graphical input control related to common excitation and control signal based on simulation needs, such as switching signal and aircraft rocker signal. All above three modules' data can exchange with the data generated from the driver model in simulation model. Monitor module is used to simulate avionics data output and real cockpit display, which consists three parts: simulated cockpit displays, signal monitoring module and user-defined output interface.

Simulated cockpit displays, such as multi-function display (MFD) and head-up display (HUD), simulate output signal of various display devices in real operational status including status signal and continuous signal. Signal monitoring module can manage real-time monitoring of signals that change during the simulation, save their data files and output as waveform. In the user-defined output interface, users can customize graphical output controls based on simulation needs .Their data can also exchange with the monitoring model data.

Aircraft dynamics model, electronic device model and flight environment model can get related data of Flight Simulator by resolving itself model in the simulation process, for example Flsim model. It can simulate aircraft state data, device data and external environment data under real flight environment. Also, it can obtain the flight data such as altitude, latitude and longitude and air data. The real flight data obtained from Flsim can dock with avionics model. Visual flight module can obtain certain data through simulation kernel which could show a flight visual scene.

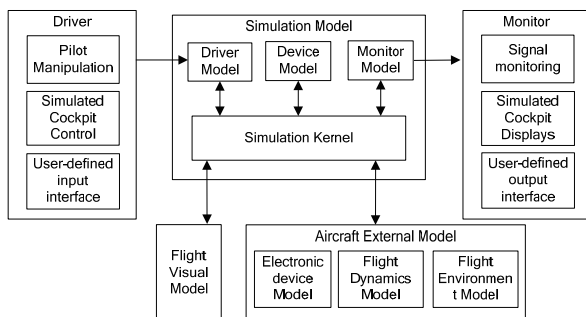


Fig. 1. The architecture of the simulation system

2.2 The principle of the simulation system

The principle of Integrated Simulation System is briefly introduced below, as shown in Fig. 2.

Firstly, the system architecture data, interface control

document (ICD) and device logical data can be obtained after the design of the entire physical architecture in avionics system and data-flow. Based on these data, the avionics simulation model is generated for each device according to the existing avionics model code templates. Code templates are model information that is made up of device, port, logic function, ICD etc. The specific code rules are based on C++. Device model is consisted of two parts: model data and logical functions. Device model data includes data that produced by avionics devices model itself. It has data exchange with relative real data that produced by external model, so that it can enhance the authenticity of the simulation. Device logic includes input and output ports of model and functions related to the ports, which are used to describe device state machine.

Secondly, simulation scenarios need to have driving signals and monitoring signals, which needs to configure related drive against specific scenario and to generate driver module by using driver model code templates. Driver data may be derived from flight control joystick, simulated cockpit control equipment and user-defined input interface. Driver module can read driver data in real time from above devices in order to refresh the data. Similarly, it can generate monitoring module which can be simulated using monitor model code templates and save the simulation result data in *.vcd format. In the mean time, cockpit display and the user-defined output interface using signal monitoring complete real-time processing and display of simulation data so that the purpose of signal monitoring can be fulfilled.

Thirdly, the system is able to simulate to produce related data under aircraft real operating state by resolving some external models such as aircraft dynamics model, flight environment model and aircraft electronic equipment model. It can also maintain validity by exchanging data in real time between simulation kernel and each avionics module so that the data from simulation is more real.

Finally, linking multiple avionics device modules, driver module and monitor module using ICD, simulation kernel can schedule these modules based on a unified timeline. Each signal has a corresponding signal packet in the simulation kernel. In the communication implementation, both Input port and output port interact with kernel signal packets. Once an output port writes value into a signal package, there is a corresponding time stamp. Similarly, the input port also read the value from the signal package in order to complete signal transmission between devices.

This integrated simulation system abstract devices of avionics system into device model that can be simulated. Simulation kernel can achieve a smaller interval and more precise clock scheduling. Incentive signals and monitoring signals of the real aircraft operations can be added to achieve closed-loop simulation. Simulation process can be displayed dynamically using cockpit control display device and flight vision system in order to enhance the simulation fidelity. The input and output signals of simulation can be customized to largely enhance human-machine interaction. Besides the system also has defined data interface of external model in order to add external aircraft model such as vehicle dynamics model, aircraft electronic equipment model, flight environment models etc. It can produce more real flight data to dock with simulation model. On the basis of signal scheduling and under the support of multiple models, multidisciplinary simulation of avionics system can be realized.

3. The implementation of the simulation system

The simulation model consists of driver model, device model, monitor model and simulation kernel, which can be implemented based on SystemC. It is a system-level modeling

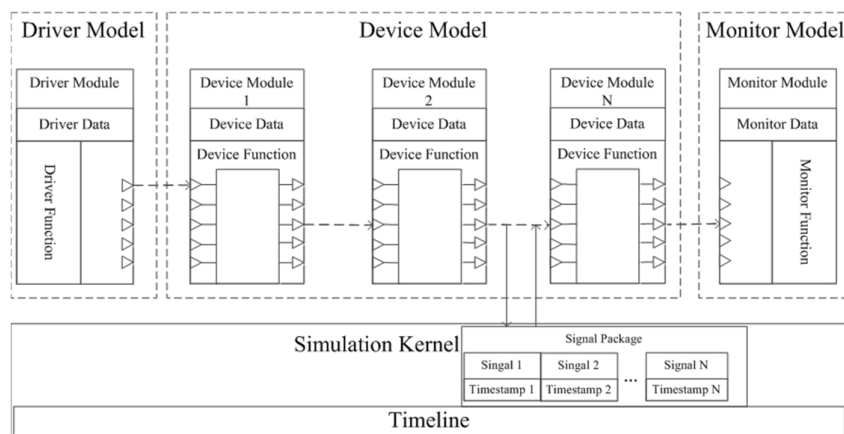


Fig. 2. The principle of the simulation system

language based on C++, accepted by IEEE as 1666 standard file. It generates the code template of diver model, device model and monitor model, and then completes communication among each module based on the SystemC kernel.

As shown in Fig. 2, according to the earlier design data, it can produce many device modules by the device code template, module1, module2 and module N. Device module, driver module, monitor module are generated by the corresponding model according to the design data. Each device module includes two parts, the device data and device logical function. Device module can generate data itself, for instance GPS, whose data usually includes longitude and latitude, and device logic refers to the function relationship based on input and output signals. For example, when the platform computer receives the self-check command, it would return to its self-check state. According to the requirements of the simulation, driver module are instantiated by the driver model, as same as monitor module.

Driver module, multiple device modules, monitoring module are connected to each other through ports, completing signal transmission with the support of the simulation kernel. Under the unified timeline, these different modules transmit signals by the method of writing and reading signal package. Such as the process of the control computer sending self-checking signal to the platform computer, in the previous moment the control computer firstly outputs signals to self-check package in the simulation kernel, and then platform computer reads the signal from the self-check package as a result of completing transmission of signal.

Diver data come from three parts, pilot manipulation module, simulated cockpit control device and user-defined input interface. And driver model can apply the data of these three parts. The signal of pilot manipulation module mainly refers to the throttle signal and six degrees of freedom signals. Simulated cockpit control device mainly refers to the various cockpit control button, such as the power switch and self-check switch, etc. User-defined input interface can design and synthesize signals according to the simulation requirements.

Monitor data mainly consist of three parts, signal monitoring module, simulated cockpit display and user-defined output interface. Monitor model can apply the data of these three parts. Signal monitoring mainly refers to the signal transmission between monitoring device, such as the self-check signal above. Simulated cockpit display mainly refers to display signals, such as head-up display, multi function display. User-defined output interface can design and synthesize display signals according to the simulation requirements.

Aircraft external model mainly includes three parts,

electronic equipment model, aircraft dynamics model, and flight environment model. Aircraft external model generates more real data docking with device module data. Flsim model is used to produce data as external model. Flight visual simulation simulates a real flight environment through the data interaction with simulation kernel that simulates the information of flight environment.

4. The application of the simulation system

Taking the weapon subsystem which is one of the subsystems of avionics system as an example and using the process of missile ground self-test “power up” as the design scene. This section will show the process of simulation.

This “power up” process contains two user scenarios, which are “Power up normal” and “Power up failure”. They are described clearly in ref. [8].

As shown in Fig. 3, according to the design of the two scenes, there are the state machines of SMS, IMP and DD and their state machine function which can be achieved based on SystemC kernel. As shown in Fig. 4, for each state machine function, there is a need to describe the input and output ports and the corresponding interface function.

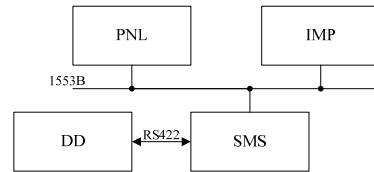


Fig. 3. The architecture model of the weapon subsystem

```

#ifndef SMS_INTFMODEL_01_H_
#define SMS_INTFMODEL_01_H_
#include "systemc.h"

SC_MODULE(SMS_intfmodel_01){
    sc_in<bool> SM_in[3];
    sc_out<sc_uint<16>> SM_out;

    void pro_SMS_intfmodel_01();

    SC_CTOR(SMS_intfmodel_01){
        SC_METHOD(pro_SMS_intfmodel_01);
        sensitive<< SM_in[0] << SM_in[1] << SM_in[2];
    }
};
#endif

#include "SMS_intfmodel_01.h"

void SMS_intfmodel_01::pro_SMS_intfmodel_01()
{
    if (SM_in[0].read()&&SM_in[1].read()&&SM_in[2].read())
    {
        SM_out = 0x0001;
    }
    else
    {
        SM_out = 0x0002;
    }
}
    
```

Fig. 4. The systemc code of SMS

In the process of simulating the two scenes, we can get human-machine interface by using the existing software such as VAPS, to connect with the corresponding kernel data. When it is needed, joystick and Flsim can generate driving signal and some other data to achieve the purpose of close-loop simulation. As shown in Fig. 5, for this scenario we use graphics to indicate the status of missiles.

During the simulation, the value of input and output signals are changing over time, and the critical signal value in VCD format is recorded for subsequent analysis of the simulation. As shown in Fig. 6, after generating the instruction of powering DC, DD will pass feedback information of its power-up state. Throughout the entire scene, we record five kinds of signals shown as DDState, WeapPowSch, CDCPowerSch, MilPosState and TitlePage. Two switch signals and DD power-up state are represented. Fig. 6 shows the change process of the five signals in two seconds.

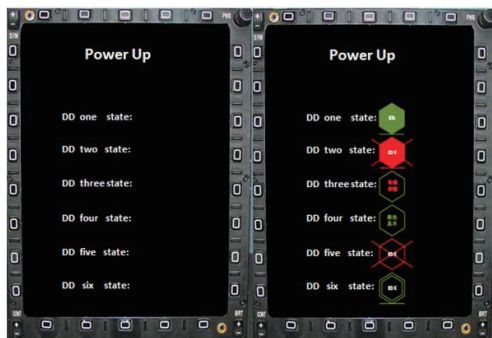


Fig. 5. The human machine interface of the scene

```

At time 9945 ms Sdate
DDState: 1 Aug 27, 2014 18:50:44
WeapPowerSch: 1 Send
CDCPowerSch: 1
MilPosState: 2
TitlePage: 0
Sversion
SystemC 2.1.v1 --- Apr 12 2013 16:49:55
Send
At time 9955 ms Stimescale
DDState: 1 1 ps
WeapPowerSch: 1 Send
CDCPowerSch: 1
MilPosState: 2
TitlePage: 0
Sscope module SystemC Send
DDState: 1 Svar wire 1 aaa DDState Send
WeapPowerSch: 1 Svar wire 1 aab WeapPowerSch Send
CDCPowerSch: 1 Svar wire 1 aac CDCPowerSch Send
MilPosState: 2 Svar wire 16 aad MilPosState [15:0] Send
TitlePage: 0 Svar wire 16 aae TitlePage [15:0] Send
Supscope Send
Senddefinitions Send
Scomment
All initial values are dumped below at time 0 sec = 0 timescale units.
Send
SDumpvars
1aaa
1aab
1aac
b1 aad
b0 aae
Send
    
```

Fig. 6. The simulation results of the scene

5. Conclusion

This paper has proposed a kind of avionics integrated simulation system, which can be used in early design stage. It can obtain state machine function of avionics devices by using earlier design data stage. And the state machine function is realized based on SystemC kernel. Closed-loop simulation can be implemented by adding external signal such as flight joystick, Flsim and man-machine interface. In this paper the simulation process was verified by “DC powered” scene, i.e. the simulation system can show changing process of this scene, which can be proved by recording signal data. As such, the simulation system can play an important role in earlier iterative design and the simulation verification in the design stage can find the logical problems as early as possible, to lower program costs and shorten the entire R&D period.

Acknowledgement

The authors gratefully acknowledge the support of the National Natural Science Foundation of China (No. 91538204) and the Aerospace Science and Industry Foundation.

References

- [1] Redman, D., “An Integrated Modeling Environment for Improved Design of Complex Systems”, *Safe & Secure Systems & Software Symposium*, Texas Engineering Experiment Station, 2010, pp. 4.
- [2] Lewis, B., “Impact of Architectural Model-Based Engineering with AADL”, 2008, pp. 4.
- [3] Friedenthal, S., “Model-Based Systems Engineering (MBSE)”, *Engineering Design Forum Emerging Software Innovation*, San Diego, CA, USA, 2012, pp. 19.
- [4] Feiler, P. H., Hansson, J., De Niz, D. and Wrage, L., “System Architecture Virtual Integration: An Industrial Case Study”, 2009, pp. 1-33.
- [5] Lafaye, M., Borde, E, Pautet, L. and Gatti, M. “Towards AADL to SystemC Mapping for Partitioned Systems”, *AADL Meeting*, Paris, 2011, pp. 1-17.
- [6] IEEE, “IEEE Std 1666™-2011- IEEE Standard for Standard SystemC Language Reference Manual”, *The Institute of Electrical and Electronics Engineers*, 2011.
- [7] Lafaye, M., “Model Driven Early Exploration of IMA Execution Platform”, *30th Digital Avionics Systems Conference*, 2011.
- [8] Yaoming, Z., Lei, Z., Yang, L. and Yongchao, W., “Avionics System Unified Lifecycle Model Architecting and Application”, *33rd Digital Avionics Systems Conference*, 2014.