

A Real-time HWIL Simulation Control System Architecture for Implementing Evaluation Environment of Complex Embedded Systems

WoonSik KIM*, ByungSun LEE*, KyungSoo KIM*, TaeSoo YANG*, SangHa KIM**

* The 1st(PGM) R&D Institute, Guidance & Control Directorate, ADD(Agency for Defense Development),
111 SunamDong, Yuseong, Daejeon, Korea

**Department of Computer Science and Engineering, ChungNam National University,
220 GungDong, Yuseong, Daejeon, Korea

wskim0106@hanmail.net, zeekzion@naver.com, justme7@hanmail.net, tsyang@hanafos.com, shkim@cnu.ac.kr

Abstract— The complex embedded systems used in aircraft and guided weapon systems have complex tasks and diverse I/O interfaces for controlling the interconnected slave systems. Some complex systems consist of many embedded systems which are closely cooperated each other to perform designated mission. Hardware-in-the-Loop(HWIL) simulation system of complex embedded system suppose to be implemented as a complex system, which has a complex software architecture and hardware configuration, because it needs to support real-time computation of complex embedded system models within a strictly constrained time interval and real-time I/O interfaces. With the HWIL simulation system, the users may run HWIL simulation on the various test scenarios and may implement many sorts of software in accordance with the test scenarios. Therefore, it should have the well-designed software architecture to provide convenient way for developing HWIL simulation environment and support real-time control environment. It also should support interactive simulation control and real-time data acquisition capabilities to provide simulation state control and data analysing environment. This paper suggests layered HWIL simulation control system architecture, which comprises I/O device driver, I/O library, HWIL Simulation control executive, and application, for building evaluation environment of the complex embedded system. In this architecture, real-time control mechanism using internal CPU timer is included. It, also, suggests two protocols, named Command Processing Protocol(CPP) and Multi-rate Data Acquisition Protocol(MDAP), for interactive simulation control and real-time simulation data acquisition. The proposed architecture is implemented as tasks based on the Commercial Off-The-Shelf(COTS) real-time operating system which runs on the computing processor installed on VME bus platform.

Keywords— HWIL Simulation, Real-time Control, Software Architecture, Data Acquisition, Embedded System

I. INTRODUCTION

Embedded systems are widely used in the industry, defense and telecommunication field and act as a core control system. Hardware-in-the-Loop(HWIL) Simulation is a Modelling and Simulation(M&S) technique that can be used to test and evaluation of the performance of the embedded systems. It is performed under the environment that real Unit Under

Test(UUT), which means the embedded systems involved in HWIL simulation as a being tested component, and mathematically modelled dynamic systems are calculated using high performance real-time computer system. Since HWIL simulation can provide the fidelity evaluation environment in the laboratory, complex embedded systems, such as guided weapon systems[1], vehicle's braking system[2], robotics, and autonomous system[3], adapt this evaluation technique to verify and validate the performance. This technique also gives an advantage of reducing the risk and the development cost and periods. Guided weapon systems[1] typically comprise many sub-components which are implemented in the form of complex embedded system, and being asked for high reliability. So, HWIL simulation is regarded as one of the most important procedure to be processed for evaluating the system performance when systems are developed.

In order to perform HWIL simulation, high performance computer system called HWIL simulation control system must be prepared. This system can be implemented based on the various types of hardware and operating system platforms like PCs. The studies for building a real-time computer system to be applied HWIL simulation keep going on in diverse part of applications areas[4][5][6][7][8]. In the HWIL simulation environment of complex embedded system like guide weapon system, many UUT hardware and emulated UUT models are involved in the simulation configuration. Therefore, the HWIL simulation control system should support lots of I/O interfaces to UUTs such as actuator, guidance and control unit, seekers and HWIL simulation facilities. It also should support high performance computing capability to calculate complex dynamics and UUTs models within a strictly constrained time interval. The system should have a lot of I/O boards for interface, and CPU boards for computing the complex UUTs and dynamics model so that it is hard to implement with a simple PC-based real-time system. Hence, the system should be developed under the environment where many I/O and computing boards are integrated within a unified system.

The software for HWIL simulation control system may become complex when lots of UUT models, kinematics and dynamics are involved in the HWIL simulation environment. When the number of UUTs increased, the more test scenario will be needed to evaluate the performance of whole UUTs. To support test environment of various scenarios, it need to provide easily replaceable simulation environment that can support real-time simulation of various applications which can be replaced according to the simulation scenario changing. Therefore, the HWIL simulation control system for complex embedded system should have not only the real-time control capability, but also the well-designed software architecture to provide convenient way for developing HWIL simulation environment. To support convenient HWIL simulation environment, it also need to provide interactive simulation control and real-time data acquisition capabilities for user friendly simulation state control and simulation data analysis.

This paper suggests the real-time control system architecture, interactive simulation control mechanism, and real-time data acquisition mechanism. These can be applied to build a HWIL simulation environment for complex embedded system, mostly applicable guided weapons system. It also presents practical example of implementation of HWIL simulation system with VME bus and COTS real-time operating system.

II. EVALUATION ENVIRONMENT OF COMPLEX EMBEDDED SYSTEM

Usually, Complex embedded systems of typical guided weapon system are evaluated under the HWIL simulation environment that consists of five(5) components, including Complex embedded systems(UUTs), HWIL simulation facilities, real-time HWIL simulation control system, interactive simulation control system, data analysis system, as seen in figure 1.

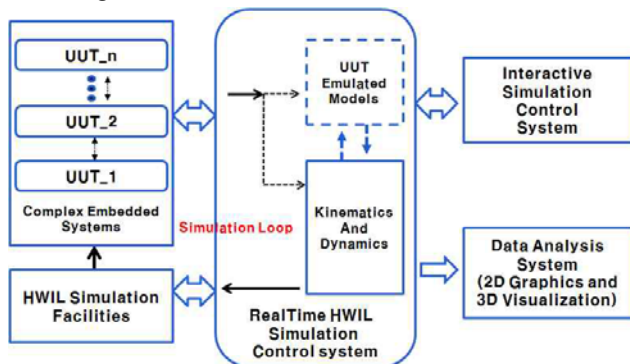


Figure 1. HWIL Simulation Environment of Complex Embedded System

Each subsystems, depicted in figure1, has following characteristics.

- **HWIL Simulation Facilities:** These components simulate real world environments in the laboratory so that UUTs feel as if they are in the real field. So, many types of facilities, like Radio Frequency Simulation system (RFSS), Infra-Red Simulation System(IRSS)[9],

and Flight Motion Simulator(FMS)[9], can be used according as the type of sensors are adopted.

- **Complex Embedded Systems:** These are Sub-components, such as seeker, actuator, inertia navigation, and guidance and control computer, of complex guided weapon system being tested in the real-time HWIL simulation environment. We call it UUT.
- **Real-Time HWIL Simulation Control system:** This is the hub of HWIL simulation environment. It calculates the kinematics and dynamics, and UUT's mathematical models of guided weapon system in real-time. It provides time-dependent continuous simulation synchronized with real time clock and closed simulation loop environment by supporting the real-time interface with HWIL simulation facilities and UUTs. It provides interactive simulation control and data acquisition for analysis of simulation result. Numerous application softwares may be loaded in this system in accordance with the test scenario.
- **Interactive Simulation Control system:** This system provides interactive control capability that users are able to control the simulation process by Graphic User Interface(GUI).
- **Data Analysis System:** This system has a capability to analyse the simulation result by 2-D graph and 3-D visualization using HWIL simulation data captured in real-time during the simulation.

In figure1 environment, the HWIL simulation control system makes a command to control the HWIL simulation facilities. Then, HWIL simulation facilities make a simulated environment that UUTs can operate as if it is in the real field. UUT's operation results are fed back to HWIL simulation control system. And these are used to calculate next frame system state information and to calculate HWIL simulation facility commands. With this command flow, the simulation proceeds continuously synchronized with real world time, and make a simulation loop which can evaluate the performance of UUTs.

Among the above components, Real-Time HWIL simulation control system plays an important role to construct a simulation environment. It provides real-time runtime environment of applications, I/O interfaces for controlling HWIL simulation facilities and UUTs. To support HWIL simulation operation, it also provide interactive simulation control mechanisms for interactive simulation state control and data acquisition mechanisms for analysing the simulation result using 2-D and 3-D graphics.

III. REAL-TIME HWIL SIMULATION CONTROL SYSTEM ARCHITECTURE

To support real-time HWIL simulation, HWIL simulation control system should support well-defined software architecture, interactive simulation state control, and real-time data acquisition mechanism.

A. Real-Time HWIL simulation control Software Architecture

Since HWIL simulation software will be used to evaluate various test scenarios in real-time, HWIL simulation control system should have the capabilities for real-time computation, real-time I/O interface, application program interface, convenient simulation control mechanism, and data acquisition mechanism. To meet above requirements, this paper proposes hierarchical software architecture as shown in figure 2. Suggested architecture consists of four layered software components, and has following characteristics.

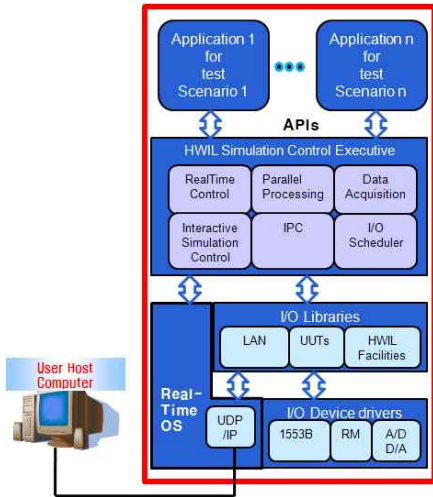


Figure 2. Real-Time HWIL Simulation control Software Architecture

1) **Application programs:** This is a mathematical six degree of freedom model that calculates kinematics, dynamics, and modelled UUTs. In order to get a fast and correct state information of system, the Adams-Bashforth 2nd order(AB2) method[10] that has a following numerical formula is adopted as a simulation integration method.

$$y_{n+1} = y_n + \frac{h}{2} (3f_n - f_{n-1})$$

, where h is an integration step_time

UUTs usually have different operation intervals so that multi-rate simulation is needed. Multi-rate simulation can be implemented by distributing the UUT model tasks to the idle processor or multi-tasking within a same processor by using preemptive priority task scheduling if it has an enough computing power. All this jobs are processed in the below HWIL simulation control executive.

2) **HWIL simulation control executive:** During the HWIL simulation, the integration step size, h in the above formula, should be synchronized with real world time. This component provides time control mechanism using internal CPU timer. All the processors participated in the computing should be synchronized its running interval. To support synchronization of processor's running interval, it uses internal timer interrupt and binary semaphore. The time control mechanism is worked as shown in figure3. When the timer control processor receives a wakeup signal from timer(`semBinWait(Timer)`) by timer interrupt, it wakes up all other processor by sending semaphore wakeup signal(`semBinGive(syncSem)`) through

software interrupt. Other processors which wait a wakeup signal in the semaphore(`semBinWait(syncSem)`) will be woken up after receiving it and process its designated tasks.

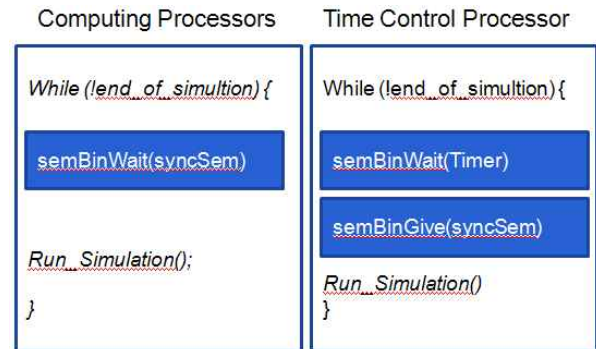


Figure 3. Time control mechanism

This component also supports high performance computation and simulation control capabilities, such as parallel computation of application, Inter-Processor Communication(IPC). To support multiple I/O interfaces, it provides IO scheduler for redirecting the I/O paths and selecting the appropriate UUTs and HWIL facilities in accordance with the scenario. For simulation state control and data analysis, it provides data acquisition and interactive simulation control capabilities. For application programs, HWIL simulation control executive provides two types of interface: Control Application program Interface(API)s, and Information APIs. Control APIs let the application know what simulation status is being processed. Information APIs are used as a communication port between application and Simulation executive for transferring simulation data.

3) **I/O Libraries:** These are library programs which provide interface between the HWIL simulation control system and all other HWIL simulation constituent, such as UUTs and HWIL simulation equipments. These libraries are called by I/O scheduler in order to control the UUT and HWIL equipment in accordance with the simulation configuration.

4) **I/O device drivers:** These programs control I/O boards, such as RS-422, A/D, D/A, 1553B, mounted on the HWIL simulation system, and called by I/O Libraries.

B. Interactive Simulation Control

In figure2, the simulation operator, who performs the HWIL simulation using GUI which runs on the host PC, should progress the simulation step interactively by selecting the simulation phase menu or button. In order to control simulation phase, this paper defines six(6) simulation states.

- **Ready:** This phase is for preparing the simulation
- **PreInit:** Applications initialize the simulation, and I/O device drivers initialize the I/O boards
- **Init:** UUTs and HWIL equipments are initialized by initial data generated by application program

- **PreRun:** Do some auxiliary tasks before run the simulation if any.
- **Run:** Real-time HWIL simulation running phase
- **Exit:** End of simulation phase

When operator selects above simulation phase, the command is packed into UDP/IP[11] PDU, and sent to the HWIL simulation control system. For interactive simulation control, this paper designs a Command Processing Protocol(CPP) operating as shown in figure 4. When the simulation control message sent by operator is reached to interface processor, which has a responsibility of simulation control interaction, it sends this command message to all other processors involved in simulation through interrupt message.

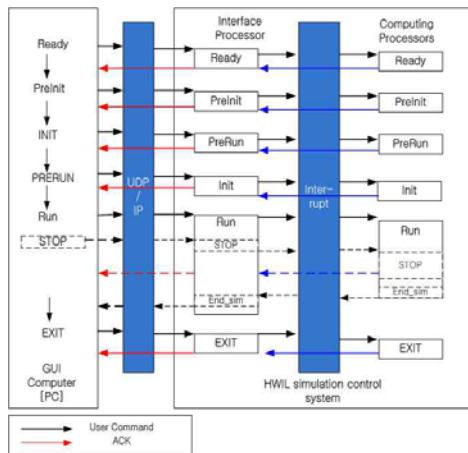


Figure 4. Simulation State Flow for Interactive Simulation Control

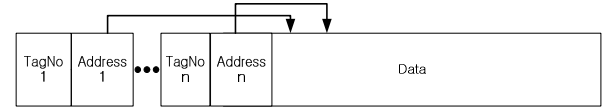
The interface processor makes a collection of reaction results of command from all other processors through interrupt, and sends it to the operator through UDP/IP packet as an acknowledgement message. The simulation could be terminated by stop button of user intervention or end of simulation message originated from application program.

C. Real-Time Data Acquisition

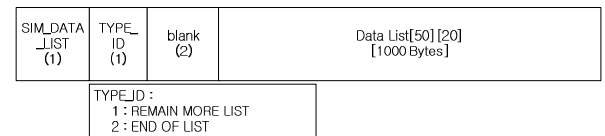
There are two methods to analyse simulation data. The one is detailed examination of captured data using 2D graph, and the other is qualitative analysis using 3D visualization. The goal of data acquisition is to capture the simulation results, which can be used 2D graph and 3D visualization, in real-time. For detailed examination, the data should be acquired every simulation step. However, for 3D visualization data, it only needs to be acquired about 50Hz rates. To support different rate of data acquisition, this paper suggests Multi-rate Data Acquisition Protocol(MDAP) designed using UDP/IP. This protocol has three types of messages, named SIM_DATA_FRAME_LIST, SIM_DATA_FRAME, SIM_3D_DATA as shown in Figure 5. SIM_DATA_FRAME_LIST message is used for transmitting data lists which will be contained in SIM_DATA_FRAME. It only used in "preinit" phase, where the simulation is initialized point. SIM_DATA_FRAME message is transmitted from HWIL simulation control system to data analysis system

at every milli-second. Two simulation step data are packed into one SIM_DATA_FRAME message. SIM_3D_DATA message, however, is transmitted every 20 milli-seconds and contains header information and simulation data in a packet. These two messages are only generated in simulation run phase and transmitted to data analysis system with constant bit rate traffic.

SIM_3D_DATA Message Format



SIM_DATA_FRAME_LIST Message Format



SIM_DATA_FRAME Message Format

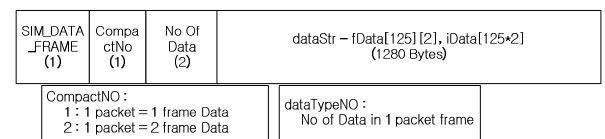


Figure 5. The Packet Structure for Real-Time Data Acquisition

IV. IMPLEMENTATION

In order to use many types of I/O boards and high performance processors, this paper uses VME bus as a backplane, and integrates whole boards in a unified system as shown in Figure 6. Many types of I/O Boards, such as RS-422, 1553B, D/A, A/D, are being used for interfacing between UUTs and HWIL simulation facilities, and it is controlled by I/O agent processor through VME bus. For high performance computing, multi-processor board, which uses PCI bus between the CPUs, is selected. With this configuration, one computing processor, named I/O and Interaction Agent, is designated for I/O board control and interactive simulation control. The others are participated in computing the application program in real-time. The I/O and Interaction Agent acts as a main controller of simulation. It controls I/O boards using I/O libraries in the figure 2 in addition to control the real-time clocks. When the timer interrupt occurred, it wakes up the all processors using software interrupt depicted in figure 3. The CPP and MDAP protocols described in chapter III.B and chapter III.C are run on this processor.

The software components described in figure 2 are implemented as tasks based on COTS real-time operating system: VxWorks[12]. The application is divided into many sub-tasks and assigned to each processor for parallel processing. It is downloaded into each processor with I/O device driver, I/O Library, and HWIL simulation control Executive. The step time is set to 0.0005 second.

To support interactive simulation control, GUI program is developed as shown in figure 7. It has a capability of CPP and MDAP connection control, and simulation state control.

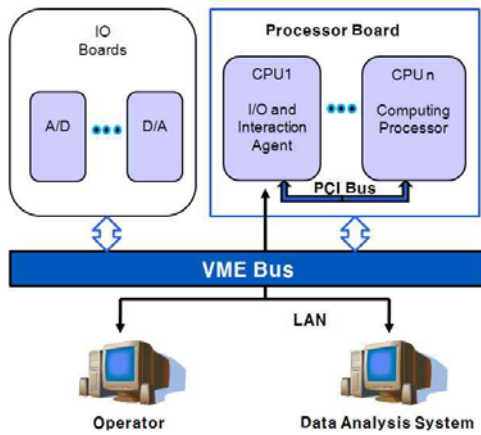


Figure 6. The hardware configuration of HWIL simulation control system



Figure 7. GUI for HWIL simulation state control

V. PERFORMANCE ANALYSIS

To verify the real-time performance of implemented system, the step time jittering of I/O and interaction agent and one of the computing processors is measured using local CPU timer for 200 seconds simulation time. The timer located at right after the semBinWait() function in figure 3. As shown in figure 8 and figure 9, the step time is in the boundary of ± 5 micro second, and shows the following characteristics.

Real-time performance of I/O and Interaction Agent

- Mean: 499.8122 micro second
- Standard deviation: 0.4363
- Sum of elapsed time: 199.925811 seconds

Real-time Performance of computing processor

- Mean: 499.8257 micro second
- Standard deviation: 0.8817
- Sum of elapsed time: 199.93085seconds

With above results, we can find out that the I/O and Interaction Agent processor has less jittering than computing processor. It's because I/O and Interaction Agent is woken up by the timer interrupt directly, but the computing processors are woken up by the software interrupt from I/O and Interaction Agent. So, more jittering can be contained in computing processor's step time due to the processing time of software interrupt service routine and semaphore operation.

The sum of elapsed time is calculated by summing all the measured step time. It doesn't mean the whole simulation time from start to end, but mean the pure operation time of simulation. With the applied measurement scheme, the time consumed by timer measurement function can't be included so

that this value is always less than 200 seconds. The mean value also can't be 500 micro second with this measurement scheme. But if we measure the step time with external timer, we may get 500 micro second of mean value. We are able to guess that the difference is occurred due to the measurement scheme, and it is a natural result that can be happened when the internal CPU timer is used as a measurement clock.

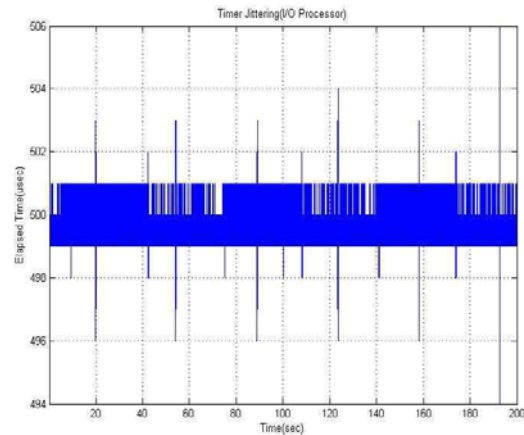


Figure 8. The measured step time of I/O and interaction agent

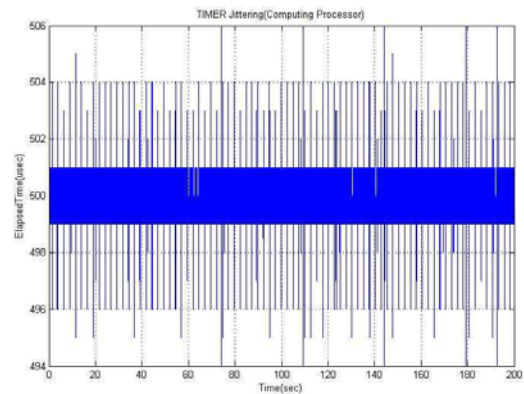


Figure 9. The measured step time of computing processor

To verify the real-time data acquisition performance, the loss ratio and received packet number of SIM_DATA_FRAME message and SIM_3D_DATA message are measured for 200 second simulation period. The loss ratio of SIM_DATA_FRAME and SIM_3D_DATA message can be guessed by figure 10. The figure presents the difference between the current frame's simulation time and right previous frame's simulation time in the message. On the time information contained in the SIM_DATA_FRAME message, the ideal difference would be 0.0005 second if there is no loss of packet since the step time is set 0.0005 second. On the SIM_3D_DATA message, it would be 0.02 second because it is transmitted 50Hz rates. In the figure, it shows that the difference is 0.0005 second for the SIM_DATA_FRAME message, and 0.02 second for the SIM_3D_DATA message during the whole simulation time. It means that there is no loss of messages, and the data acquisition protocol work perfectly.

The estimated number of packet can be calculated by following formula, and can get 200001 and 10001packet for each SIM_DATA_FRAME and SIM_3D_DATA packets.

$$\text{The estimated number of packet} = \text{simulation duration} / \text{capture\_interval} + 1$$

The number of SIM_DATA_FRAME message packets captured during the simulation is 200001, and the number of SIM_3D_DATA message packets is 10001. With the above results, we are able to estimate that there is no loss of packet with the CPP and MDAP protocol.

The needed bandwidth for SIM_DATA_FRAME and SIM_3D_DATA message can be calculated following formula.

$$\text{Bandwidth} = \text{SIM\_DATA\_FRAME PDU}(1326 \text{ Bytes}) * 8 * \text{DataRate} + \text{SIM\_3D\_DATA PDU}(800 \text{ Bytes}) * 8 * 3\text{DRate}$$

The DataRate and 3DRate stand for the communication frequency of SIM_DATA_FRAEM and SIM_3D_DATA message. When the DataRate and 3Drate are set by 1KHz and 50Hz respectively, the needed bandwidth would be 10,928 Kbps. We have used the gigabit Ethernet so that there is no loss of packets.

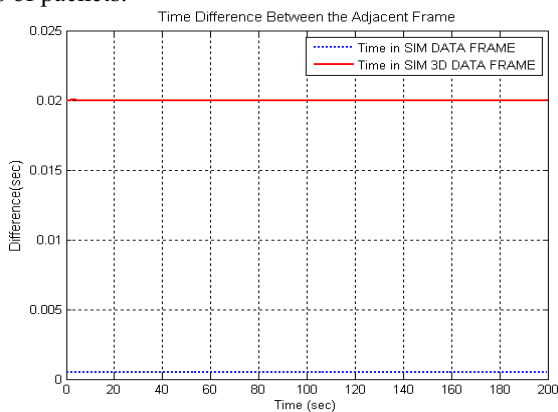


Figure 10. The performance of data acquisition system

VI. SUMMARY AND CONCLUSION

The HWIL simulation is the most efficient test tool to verify and validate the performance of complex embedded system. In the HWIL simulation environment of complex embedded system like guide weapon system, many UUT hardwares and emulated UUT models are involved in the simulation configuration. It leads to increase the complexity of software architecture, and increase the number of run of simulation. To reduce the implementation burden of HWIL simulation environment, HWIL simulation control system should support well-defined software architecture. In this architecture, the real-time control mechanism is indispensable. To provide convenient simulation operation environment, interactive simulation control and real-time data acquisition mechanisms are necessary.

This paper suggests layered HWIL simulation control system software architecture which comprises I/O device

driver, I/O library, HWIL Simulation control executive, and application with real-time control methods. In the HWIL Simulation control executive, the real-time control mechanism using internal CPU timer is included. It also suggests interactive simulation control method named CPP and real time data acquisition method named MDAP.

The hardware system is implemented based on VME bus with various I/O boards and high performance computing boards. The suggested software architecture is developed based on COTS real-time operating system, VxWorks.

The suggested architecture gives sufficient functionalities and convenient methods to the HWIL simulation developer and operator who are going to implement HWIL simulation environment and run the HWIL simulation on the complex embedded system. But this architecture doesn't take into consideration of using state-of-the-art I/O boards, such as PMC module. So, to support more universal I/O boards, more study on extension of this architecture is needed. And, to support accurate simulation time, precision external timer should be applied to implement the HWIL simulation control system.

REFERENCES

- [1] M.D. Trotsky, "PAC-3 Missile Ground Testing for Success In Flight," *The First Annual International Missiles and Rockets Symposium and Exhibition*, February 2000.
- [2] K. C. Lee, J. W. Jeon, and D. H. Hwang, "Comparison between Software Simulation and HIL Simulation of Antiskid Brake System for Aircrafts," *IECON 2005 31st Annual conference of IEEE*, pp.1993-1998, November 2005.
- [3] W. Deng, and Y. H. Lee. "Hardware-in-the-loop Simulation for Autonomous Driving," *IECON 2008 34th Annual conference of IEEE*, pp1742-1747, November 2008.
- [4] Wu xun, Yang Shu-xing, and Zhao lei, "Study on Evaluation Techniques for Embedded Control Computer of Aircraft," *International Conference on computer science and information technology*, pp.675-679, August 2008.
- [5] C. Godoy, "High Performance Computational Environment for Real-Time Hardware-in-the Loop Simulation," *HPCMP Users Group Conference*, pp.333-336, June 2006.
- [6] C. E. Hall, "A Real-time linux system for autonomous navigation and flight attitude control of an uninhabited aerial vehicle," *Digital avionics systems, 2001 DASC The 20th conference*, pp1a1/a -1a1/9, October 2001.
- [7] D. Beck, and R. Bennett, "Implementation of a Personal Computer Based Real-Time Hardware-in-the-Loop U.S. Army Aviation and Missile Command Simulator", *Proc. SPIE Vol.4717*, pp.24-31, July 2002.
- [8] J. A. Buford Jr., T. A. Garcia, and M. H. Bowden, "Re-configurable visualization for HWIL simulation," *Proc. SPIE, Vol3367*, pp.122-128, August 1998.
- [9] C. M. Edwing, "The Advanced Guided Weapon Testbed(AGWT) at the Air Force Research Laboratory Munitions Directorate," *ITEA Journal*, 31(1), March 2010.
- [10] S.D Conte, C. D. Boor, *Elementary numerical analysis : An algorithmic approach*, McGraw-Hill, 1980.
- [11] W.R. Stevens, *TCP/IP Illustrated, Volume I The Protocols*, Addison Wesley, 1994.
- [12] *VxWorks programmer's guide 5.5*, Windriver system inc, July 2002.