

HILS를 위한 시뮬레이션 프로그램 설계 기법

A Design Method of the Simulation Program for HILS

박해리* 전인수* 송창준*
Haerhee Park In-Soo Jeon Chang-Joon Song

ABSTRACT

In this paper, a design method of the simulation program for HILS(Hardware-In-the-Loop Simulation) system is proposed. The present method consists of definition of requirements for HILS, development of specifications, and implementation of the program to satisfy the specifications. In the implementation of the program, the application of hardware interface and the concept of structural modularization are proposed to satisfy the specifications. The concepts of CSCI(Computer Software Configuration Item) and encapsulation are used for structural modularization.

The proposed method was practically applied to the development of the simulation program for the efficient operation in HILS of an anti-ship missile system.

Keywords : Simulation Program(시뮬레이션 프로그램), HILS(Hardware-In-the-Loop Simulation, 모의비행시험), Modularization(모듈화), CSCI(Computer Software Configuration Item, 형상항목), Encapsulation(캡슐화)

1. 서론

HILS(Hardware-In-the-Loop Simulation)는 복잡한 실시간 임베디드 시스템의 개발과 시험에 사용되는 기술이다. HILS에는 HILS 대상 장비가 운용되는 전체 시스템 및 환경을 수학적으로 모델링한 시뮬레이터와 HILS 대상 장비가 포함된다. 시뮬레이터는 실시간으로 HILS 대상 장비의 운용 환경을 모의하는 역할을 하며, HILS를 통하여 실제 운용 환경에 가까운 실시간 모의 운용 환경에서 HILS 대상 장비를 시험한다. 임베디드 시스템을 개발하는 가장 정확한 방법은 임

베디드 시스템이 속한 전체 시스템 및 실제 환경에서 시험하여 개발하는 방법이므로, 실제 환경과 유사한 시험 환경을 제공하는 HILS는 효과적인 방법이다. 그 뿐 아니라 HILS는 실제 환경 및 실제 시스템에서 시험하는 것보다 비용 절감의 효과가 크고, 안전 등의 문제로 실제 환경에서 수행하지 못하는 시험의 일부를 HILS에서 시험할 수 있으므로 장비 개발의 시험 방법으로서 가치가 높다. 따라서 자동차 공학, 항공우주공학, 무기공학, 전기공학 등의 많은 분야에서 사용되고 있다^{1~7)}. 유도탄 체계를 개발함에 있어서도 HILS는 중요한 필수 개발 과정 중의 하나이다. HILS를 통하여 실제 체계 개발 및 운용 시험 시 발생할 수 있는 오류 및 시행착오를 사전에 발견 및 조치함으로써 개발 기간 및 비용을 크게 줄일 수 있기 때문이다.

† 2013년 2월 22일 접수~2013년 6월 21일 게재승인

* 국방과학연구소(ADD)

책임저자 : 박해리(p_helen@hanmail.net)

HILS 시스템은 HILS 대상 장비가 속한 전체 체계의 특성, 즉 체계가 어떠한 하부 시스템(subsystem)으로 구성되어 있는가, 어떠한 기능 및 성능을 가진 시스템인가에 따라 구성이 달라지므로, 각각의 경우마다 적합한 HILS 시스템을 구축하여야 한다. 따라서 다양한 체계에 사용되는 HILS 시스템의 구축에 대한 연구가 이루어지고 있다^[8~12].

이러한 HILS 시스템에서 HILS 대상 장비(이하 대상 장비)가 속한 체계의 운용 환경을 모델링한 시뮬레이션 프로그램(이하 HILS 시뮬레이션 프로그램)은 HILS 시스템의 공통적인 필수 구성 요소이다. HILS의 목적이 대상 장비로 하여금 실제로 운용하지 않고도 유사한 환경을 제공하여 개발 시험을 수행하는 것이므로, HILS의 목표는 대상 장비로 하여금 실제로 운용되고 있지 않으나 운용되고 있는 것과 동일한 상황을 만들어주는 것이다. 따라서 HILS 시뮬레이션 프로그램이 체계를 포함한 운용 환경을 얼마나 잘 모의하는가, 그리고 이 프로그램을 활용하여 HILS 시스템을 얼마나 잘 구축하였는가 HILS의 질을 좌우하는 요소 중의 하나이다.

체계 개발 과정에서 설계 및 성능 분석을 목적으로 체계 시뮬레이션 프로그램이 개발되고 사용되는데, 이 프로그램은 체계 및 운용 환경을 모의한다는 점에서 HILS를 위한 시뮬레이션 프로그램과 성격이 같다. 따라서 체계 설계 및 성능 분석을 목적으로 개발된 체계 시뮬레이션 프로그램은 HILS를 위해서도 사용될 수 있다. 그러나, 체계 시뮬레이션 프로그램은 체계 개발의 초기 단계에서부터 개발 및 운용되는 반면, HILS는 체계 설계 후 관련 장비의 개발 및 시험 단계에서 수행되므로, 체계 시뮬레이션 프로그램이 HILS 시스템의 구성 요소로 운용되기 위해서는 프로그램 개발 단계에서부터 이를 고려하여야 추후 프로그램의 전체 구조 및 코드 등을 수정하는 큰 작업을 줄일 수 있다. 즉, 체계 시뮬레이션 프로그램이 개발 단계에서부터 HILS 시스템에의 적용을 고려하여 해당 요구 조건에 적합하도록 개발될 필요가 있다.

본 논문에서는 HILS를 위한 시뮬레이션 프로그램의 체계적인 개발 기법을 제안한다. 먼저 관련된 요구 사항을 분석하고, 이로부터 설계 사항을 도출하며, 도출된 설계 사항을 구현하기에 적합한 설계 기법을 제안한다. 제안한 설계 기법은 구조적 모듈화와 하드웨어 인터페이스 구조적용이며, 이 프로그램 개발 기법으로 체계 시뮬레이션 프로그램을 개발하면 프로그램 개발

후 HILS 시스템에 적용 시 프로그램의 큰 변경 없이 적용 가능하다. 본 논문에서 제시한 구체적인 사례는 함대함 유도무기체계 HILS 관련 사례로 국한하였다.

2. HILS를 위한 시뮬레이션 프로그램 설계 기법

가. 요구사항 분석

HILS를 위한 시뮬레이션 프로그램 개발을 위한 요구 사항은 “HILS에 적용 가능하여야 한다.”는 것이다. 이 요구 사항으로부터 프로그램이 갖추어야 할 설계 사항을 도출하기 위하여 먼저 HILS의 특성을 살펴보면, HILS는 Fig. 1과 같이 HILS 대상 장비(부체계)가 포함된 시뮬레이션이라는 것을 알 수 있다.

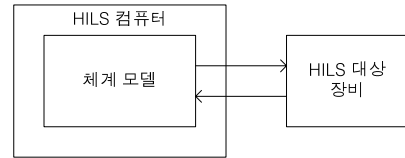


Fig. 1. Configuration of HILS

한 체계(system)에는 다양한 종류의 다수 HILS 대상 장비가 존재하고, 체계 시뮬레이션 프로그램에는 이들의 모델이 포함된다. 따라서 체계 시뮬레이션 프로그램에 구현되는 부체계 모델은 HILS 시에 하드웨어로 교체가 가능하여야 한다. 이를 하위 요구 사항으로 정의하고 위 내용을 정리하면 다음 Fig. 2와 같다.

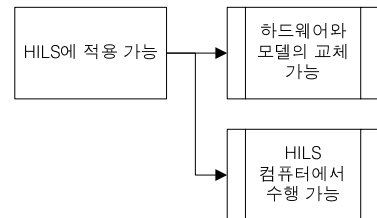


Fig. 2. Requirements for HILS

하드웨어와 모델이 교체가 가능하다는 것은 둘의 사양이 동일하다는 것을 뜻한다. 그리고 여기서 사양은 모듈의 인터페이스가 동일하고, 모듈의 기능과 동작 방식이 동일하다는 것을 뜻한다. 따라서 첫 번째 하부 요구 사항을 만족하기 위하여 제안하는 설계 사항은 다음 Fig. 3과 같다.

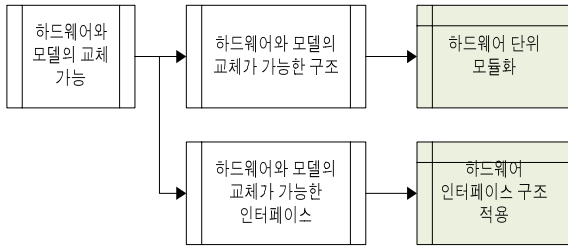


Fig. 3. Requirements and specifications for HILS

HILS에 적용되는 프로그램에서 위 두 개의 설계 사항은 매우 중요하다. 이 설계 사항을 사전에 고려하지 않고 개발할 경우에 비하여 프로그램 개발 시간 및 비용을 현격히 단축시킬 수 있기 때문이다. 다음으로 위 설계 사항에 대한 설계 및 구현 방법에 대하여 설명하고자 한다.

나. 설계 사항 구현

1) 구조적 모듈화

통상 유도탄 체계의 체계 시뮬레이션 프로그램은 운동 상태를 모의하는 운동 방정식 모델 및 항법 모델과 체계 모델로 구성되며, 체계 모델에는 체계 및 각 부체계(구성품) 모델이 포함된다. 그리고, 이 부체계 모델 중에는 HILS 대상 부체계가 포함된다. 앞에서 설명한 설계 사항 중 첫 번째 사항인 구조적 모듈화를 만족하기 위한 첫 번째 설계 방법으로 다음 Fig. 4와 같은 형상항목(CSCI : Computer Software Configuration Item) 구조를 제안한다. CSCI는 소프트웨어 공학의 형상 관리 분야에서 사용되는 개념으로 형상항목이란 형상 관리 대상으로 지정되는 항목을 말한다. CSCI는 구성요소로서 다수의 CSC(Computer Software Component)를 포함하고, CSC는 다수의 CSU(Computer Software Unit)로 구성된다. 본 설계 기법에서는 형상항목 개념을 프로그램의 구조를 설계하기 위하여 적용한다.

Fig. 4의 형상항목 구조에서 중요한 점은 유도탄 구성품인 각 부체계 모델을 프로그램 구동과 관련된 기능이 구현되는 시스템 모듈, 운동 방정식 모델 및 항법 모델이 구현되는 동역학 모듈과 동일한 레벨인 하나의 CSCI로 각각 설계하는 것이다. 위와 같이 설계함으로써 개념적으로 부체계 모델을 독립된 하나의 모듈로 정의하고 취급할 수 있으며, 각 부체계 모듈을 구성하는 다수의 서브 모듈 및 함수를 구조화하기에 용이하다.

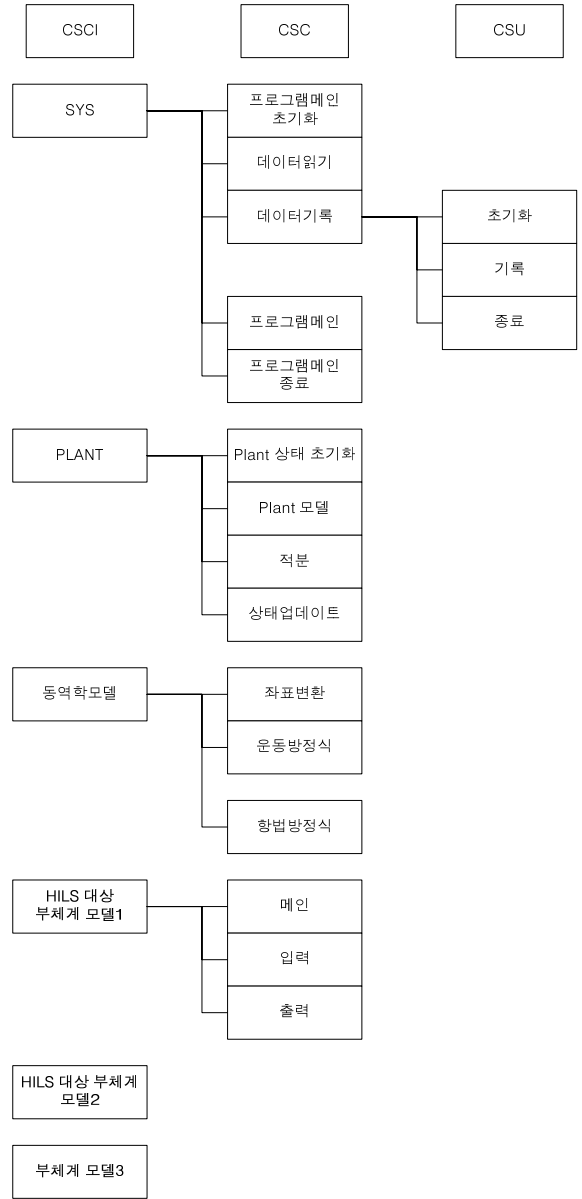


Fig. 4. Structure of Configuration Items for Structural modularization

두 번째 설계 방법으로 객체 지향 프로그래밍의 클래스의 개념을 도입하여 각 CSCI를 초기화 함수, 종료 함수, CSCI 내부 함수들과 인터페이스 함수, 메인 함수로 구성할 것을 제안한다. 초기화 함수는 클래스의 생성자 함수와 같은 기능을 갖고 종료 함수는 소

멀지 함수의 역할을 한다. 인터페이스 함수 및 CSCI 메인 함수는 클래스의 공개 멤버와 같은 성격을 갖고, 클래스의 비공개 멤버의 개념에 따라 CSCI 내부 함수와 변수를 CSCI 내부에서만 접근 가능하며 외부에서 접근할 수 없도록 한다. 이는 객체 지향 프로그래밍의 공통적 특징 중의 하나인 캡슐화(encapsulation)의 특성이므로 위 방법으로 설계된 CSCI는 캡슐화의 특성을 갖는다.^[13] 이와 같이 설계 및 구현함으로써 CSCI 외부로부터 내부 함수 및 변수의 간섭과 오용을 막고, 모듈 구동 시 독립성을 보장할 수 있다. 그러므로 하드웨어와 모델의 교체가 가능하도록 구조적 모듈화를 완전히 구현할 수 있다.

2) 하드웨어 인터페이스 구조 적용

두 번째 설계 사항인 “하드웨어 인터페이스 구조 적용”을 만족하기 위한 설계 방법으로, 앞서 제안한 방법으로 설계된 모듈의 인터페이스를 클래스의 인터페이스 개념과 하드웨어 인터페이스 구조 및 규약을 적용하여 설계할 것을 제안한다. 이와 같이 함으로써 하드웨어와 소프트웨어 모듈은 동일한 기능, 동일한 인터페이스를 가지므로 HILS 시 하드웨어와 모델의 즉시 교체가 가능하다. 부체계 모델의 인터페이스 설계 시 하드웨어의 인터페이스 규약을 적용하는 구현 방법은 여러 가지가 있으며, 한 예로 통신 방식 및 신호 종류별로 구조체나 배열을 다르게 선언하여 사용하는 방법이 있다.

위에서 제안한 설계 기법의 중요한 장점은 다음과 같다. 프로그램 개발자가 종종 경험하는 인터페이스 오류의 예는 다음 Fig. 5와 Fig. 6으로 쉽게 설명된다.

체계 시뮬레이션 프로그램은 한 프로그램 내에 모든 모듈이 구동되므로 Fig 5와 같이 A 모듈의 한 변수를 B 모듈에서 모듈 간 규정된 인터페이스를 고려하지 않고 자유롭게 사용할 수 있다. 그러나 Fig. 6과 같이 A 모듈이 하드웨어로 교체되고 A를 제외한 시뮬레이션 프로그램의 모듈이 모두 HILS 컴퓨터에서 구동되면, B 모듈에서는 하드웨어에서 송수신하는 정보 이외의 변수에 접근할 수 없다. 위에서 제안한 하드웨어 인터페이스 규약을 적용한 설계 기법을 적용하면 이와 같은 프로그램 개발 중 실수 할 수 있는 인터페이스 오류 문제를 프로그램 구현 시에 발견하여 해결할 수 있으므로, 프로그램 개발 시간 및 HILS 준비 시간을 단축시키며, 결과적으로 체계 개발 시 일정 지연, 오류 수정 비용 발생 등의 위험 요소를 줄여준다.

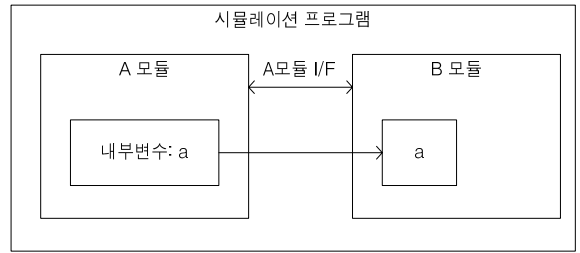


Fig. 5. An example of interface error #1

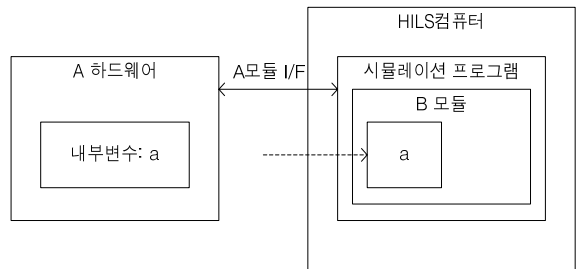


Fig. 6. An example of interface error #2

3. 적용 사례

본 논문에서 제안한 설계 기법을 함대함 유도탄 시뮬레이션 프로그램 개발에 적용하였다. 함대함 유도탄 시뮬레이션 프로그램의 HILS 대상 하드웨어는 유도조종장치(GCU), 관성항법장치(INS), 구동기(ACT), 구동기 제어기(DSC)이며, 다음 Fig. 7과 같은 형상항목 구조를 설계하였다.

본 프로그램에서는 전역 변수의 사용 금지를 원칙으로 하고 예외 없이 2개 이상의 CSCI가 공통 사용하는 전역 변수를 제거함으로써 모듈화를 구현하였다. 그러나 각 CSCI내에서는 불가피한 경우 전역 변수를 사용할 수 있도록 일부 허용했다.

한 CSCI가 인클루드하는 헤더 파일은 한 CSCI 내의 함수 및 파일에 한정하도록 하였고, 공통 인클루드 파일을 최소화하고 타 CSCI의 함수에서 해당 CSCI의 함수나 변수 선언문을 참조하는 경우를 명확히 선별하였다. 또한 부체계 모델의 헤더 파일이 다른 부체계 모델에 인클루드되는 경우는 모두 제거하였고, ‘SYS’, ‘동역학’ 그리고 ‘MSL’ CSCI에 속한 헤더 파일만 다른 CSCI에 인클루드가 허용된다. 이와 같이 구현함으로써 하나의 부체계 모델을 전체 시뮬레이션 프로그램으로부터 분리가 용이하도록 하였다.

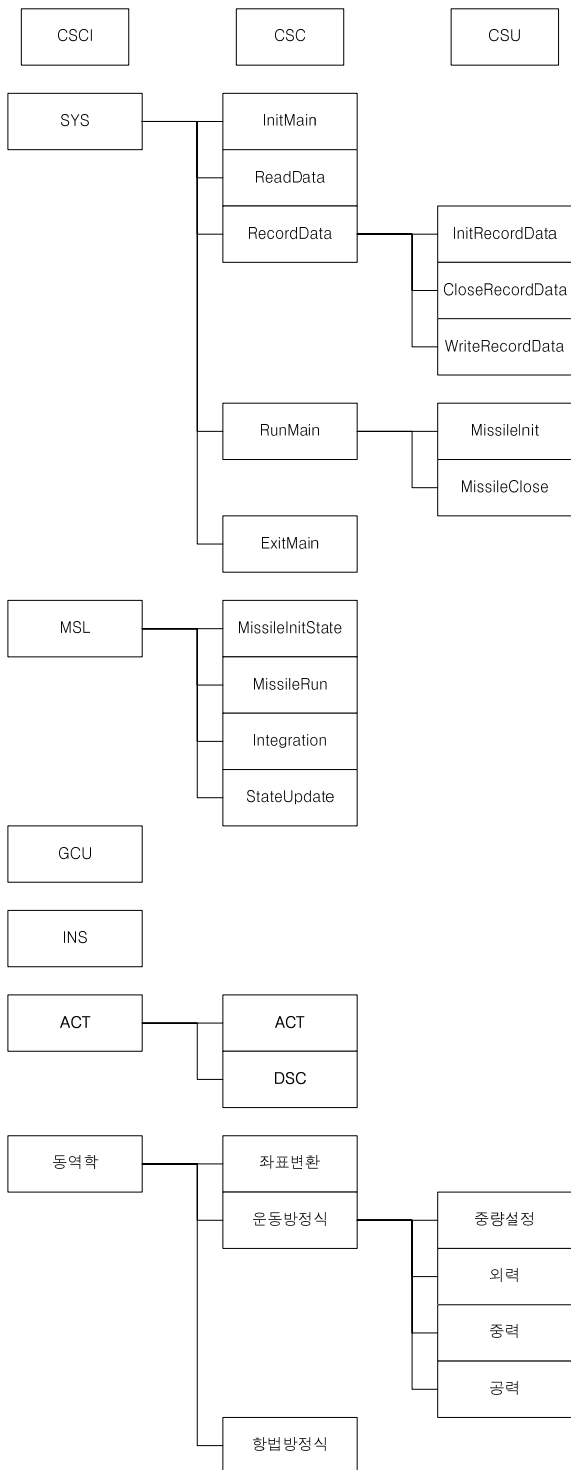


Fig. 7. Structure of Configuration Items for the Ship-to-Ship Missile

각 부체계 모델의 인터페이스는 본 논문에서 제안한 설계 기법을 적용하여 하드웨어 인터페이스 구조와 규약에 따라 구현하였다. 신호의 종류에 따라 구조체를 다르게 선언하였으며, 구현 내용은 다음과 같다.

Table 1. Interface of the program of the Ship-to-Ship Missile

구조체명	신호 종류	From	to
DataBus.GO	1553B통신	GCU	MSL
DataBus.LI	1553B통신	GCU	INS
DataBus.LO	1553B통신	INS	GCU
DataBus.DI	1553B통신	GCU	DSC
DataBus.DO	1553B통신	DSC	GCU
ActIn	Analog	DSC	ACT
ActOut	Analog	ACT	DSC

각 구조체의 멤버 변수명은 신호명을 고려하여 결정함으로써 변수명만으로도 어떤 신호인지 식별이 용이하도록 하였다.

이와 같은 구조로 구현된 함대함 유도탄 시뮬레이션 프로그램은 사업 기간 중 HILS에 적용되었고, 업무 효율화, HILS 준비 시간 단축의 성과를 이루어냈다.

시뮬레이션 프로그램과 HILS의 결과 차이에 대한 주요 원인 중 하나는 하드웨어 간 통신 딜레이이다. 본 설계기법으로 구현된 프로그램은 하드웨어 인터페이스 구조를 따르고 있으므로 하드웨어 포함 시 발생하는 이러한 현상을 쉽게 모델링할 수 있었고 HILS 결과를 검증하는데 용이하였다.

또한, 신호 사양에 대한 각 부체계 및 체계 간 확인 및 수정이 HILS 시작 이전부터 이루어질 수 있었고, HILS 수행을 완료하면서 설계 변경 및 결정되는 인터페이스 규약이 시뮬레이션 프로그램에 동일하게 구현됨으로써 인터페이스 규약 관리 업무의 효율성을 높일 수 있었다.

4. 결론

본 논문에서는 HILS에 적용하는 체계 시뮬레이션 프로그램을 구축하는 방법으로 요구 사항 분석, 설계 사항 도출의 과정을 통하여 목적에 부합하도록 하는

설계 기법을 제안하였다.

요구사항 분석을 통하여 도출한 HILS에 적용 가능한 프로그램의 설계 사항은 “하드웨어 단위 구조적 모듈화”와 “하드웨어 인터페이스 구조적용”이다.

제안된 설계 기법에서는 구조적 모듈화를 위하여 형상항목 구조 설계 방법 도입을 제안하였고, 객체 지향 프로그래밍의 클래스 개념을 적용하였으며, 하드웨어 인터페이스 구조적용을 위하여 모듈 인터페이스 설계 시 하드웨어 인터페이스 구조 및 규약을 반영하였다.

본 설계 기법으로 함대함 유도탄 시뮬레이션 프로그램을 개발, 적용함으로써 HILS 준비 시간 단축, 업무 효율화, 프로그램의 실제에 더 근접한 모델링의 성과를 이루어 낼 수 있었으며, 결과적으로 체계 개발 기간 중 HILS 수행 일정에 있을 수 있는 일정 지연 위험 요소를 줄이는 성과도 얻을 수 있었다.

References

- [1] Jie, X., Feng, W., Yongguang, Y. D. Z. and Qiang, G., “An Application of Real-time Engine Model in HILS Simulation and Testing Platform Based on SIMULINK”, International Conference on Digital Manufacturing & Automation/IEEE, pp. 176~180, 2010.
- [2] Kim, S. P., Park, M. H., Lim, J. H. and Kim, E. T., “HILS of Flight Control System for Unmanned KARI Airship”, Korea Institute of Military Science and Technology 2005 Annual Conference, pp. 562~564, 2005.
- [3] Kim, Y. D., Park, K. H., Chung, S. K. and Ha, D. H., “Rotational Angle Control of Smart Munition Using HILS”, Korea Institute of Military Science and Technology 2002 Annual Conference, pp. 281~284, 2002.
- [4] Kim, C., U., Kim, M., J., Rhee, E., J., Lee, K., H. and Woo, K., J., “A Study on the Application of Semi-Active Suspension Units for a Combat Vehicle by Using HILS”, Journal of the Korea Institute of Military Science and Technology, Vol. 13 No. 6, pp. 967~975, 2010. 12.
- [5] Hwang, A. R., Yoon, S., I., Choi, C., H. and Cho, H., J., “Verification of the Underwater Vehicle Control Algorithm with MEMS Inertial Sensor based on Hardware in the Loop Simulation”, Korea Institute of Military Science and Technology 2009 Annual Conference, pp. 1988~1991, 2009.
- [6] Jeon, J. H., Kim, J. Y., Kim, S. K., Ahn, O. B., Park and J. H., “Development of HILS(Hardware In-Loop Simulation) System for MMS(Microgrid Management System) by using RTDS, the 13th International Power Electronics and Motion Control Conference(EPE-PEMC 2008), pp. 2492~2497, 2008.
- [7] Ersal, T., Brudnak, M., Stein, J. L., Fathy and H. K., “Statistical Transparency Analysis in Internet-Distributed Hardware-in-the-loop Simulation”, IEEE/ASME Transactions on Mechatronics, Vol. 17, No. 2, pp. 228~238, April 2012.
- [8] Kim, W. S., Yang, T. S. and Kim, S. H., “A Simulation Technique of the Shipboard INS Transfer Alignment Environments using Hardware-In-the-Loop Simulation”, Journal of the Korea Institute of Military Science and Technology, Vol. 14 No. 2, pp.181~188, 2011. 4.
- [9] Hwang, H. S., Jang, I. G., Seo, I., Park, J. H., Jeon, J. W. and Hwang, S. H., “Development of HILS System for Testing Embedded System of the BCU for HEV”, IEEE/ASME International Conference on Mechatronics and Embedded Systems and Applications, pp. 286~291, 2008.
- [10] Sanada, K., “A Study on HILS of Fluid Switching Transmission”, SICE-ICASE International Joint Conference, pp. 4668~4671, 2006. 10.
- [11] Jeon, J. H., Kim, J. Y., Kim, H. M., Kim, S. K., Cho, C., Kim, J. M., Ahn, J. B. and Nam, K. Y., “Development of Hardware In-the-Loop Simulation System for Testing Operation and Control Functions of Microgrid”, IEEE Transactions on Power Electronics, Vol. 25, No. 12, pp. 2919~2929, 2010. 12.
- [12] Park, B. J., Kim, S. P. and Ju, G. H., “Development of HILS Software for KARI Unmanned Airship”, Korea Institute of Military Science and Technology 2003 Annual Conference, pp. 154~157, 2003.
- [13] Herbert Schildt저, 우진운/유해영 역(1988.1), “알기 쉽게 해설한 C++”.