

論文

J. of The Korean Society for Aeronautical and Space Sciences 44(8), 657-665(2016)

DOI:http://dx.doi.org/10.5139/JKSAS.2016.44.8.657

ISSN 1225-1348(print), 2287-6871(online)

발사관제시스템 시뮬레이터의 설계 및 구현

안재철*, 문경록**, 오일석***

Design and Implementation of Simulator of Launch Control System

Jae-Chel An*, Kyung-Rok Moon**, Il-Seok Oh***

Korea Aerospace Research Institute* , **, Chonbuk National University* , ***

ABSTRACT

Launch Control System(LCS) performs the pre-launch preparation and launch operation during launch campaign. The successful launch operation is basically influenced by hardware and software of LCS. Especially, a trivial errors in control algorithm can cause critical problem or disaster in launch operation. Therefore, the hidden or implicit errors should be distinguished and eliminated by the verification test in advance. In this paper, the design and implementation of hardware and software simulator which have already been used in LCS verification will be introduced. By presenting the detailed design and flowchart-based algorithms, we make other similar systems adopt the implementation philosophies of this paper. Especially, this paper emphasizes that all the simulation algorithms work on the self-controller in LCS without using separated computer or PLC.

초 록

발사관제시스템은 발사캠페인 기간 중 지상의 발사 준비 및 발사 운용 임무를 수행한다. 보통 발사 운용의 성공 여부는 발사관제시스템의 하드웨어나 소프트웨어에 의해 크게 영향을 받는다. 특히 제어 알고리즘의 사소한 오류조차 발사 운용에 심각한 문제나 재난을 유발시킬 수 있다. 따라서 내재된 오류를 사전 검증시험을 통해 찾아내 제거해야 한다. 본 논문은 발사관제시스템 검증에 사용된 하드웨어와 소프트웨어 시뮬레이터의 설계와 구현에 대해 소개한다. 시뮬레이터의 상세한 설계와 플로차트 형식의 구체적인 알고리즘을 기술하여 유사한 시스템의 구현에 적용할 수 있도록 하였다. 특히 별도의 컴퓨터나 PLC를 사용하지 않고 자체 제어기에서 모든 시뮬레이션 알고리즘이 동작하도록 개발한 점을 강조한다.

Key Words : Launch Control System(발사관제시스템), Launch Control Center(발사관제소), Launch Building(발사동), Centralized Building(중앙공용동), Lift Off Contact(발사체 이륙신호)

1. 서 론

발사관제시스템은 발사캠페인 기간 중 지상의

발사 준비 및 발사 운용과 관련된 주요 시스템을 감시하고 제어한다. 발사관제시스템의 오류는 발사 운용 전체에 큰 영향을 주기 때문에 시스템별

† Received : April 1, 2016 Revised : July 11, 2016 Accepted : July 12, 2016

* Corresponding author, E-mail : jcan@kari.re.kr

제어기와 제어공정 알고리즘이 요구사항에 맞게 설계 및 구현이 되어 있는지 사전에 면밀하게 검증해야 한다. 따라서 발사관제시스템의 자체 검증 방법이 수립되어야 하며, 검증을 통해 사전에 오류를 제거해야 한다. 이러한 검증을 위해 제어 대상이 되는 설비나 발사체에 대한 시뮬레이터를 제작하여 발사관제시스템의 성능 시험 및 검증에 사용하고 있다[1].

일반적으로 제어기와 제어공정 알고리즘 검증에 사용되는 방법은 두 가지가 있다[2, 3, 4, 5]. 첫째, 제어 대상과 동일 기능을 수행하는 하드웨어 시뮬레이터를 밸브나 센서와 같은 제어 대상이 위치한 곳 또는 제어기에 근접 구성하여 상태 신호를 모사하는 방법이 있다. 둘째, 소프트웨어를 이용해 제어 대상의 상태 값을 모사하는 방법이 있다. 전자의 경우 제어 대상의 수가 많거나 여러 장소로 분산되어 있는 경우 동시에 다수의 상태 신호 모사가 어려워 알고리즘 검증 보다는 제어기 입출력 회로와 제어 케이블의 무결성 점검 용도로 국한하여 사용된다. 반면, 후자의 방법은 제어 대상의 상태 값들을 소프트웨어를 이용해 동시 모사가 가능하기 때문에 석유화학, 철강, 자동차 생산라인, 발전소 등 복잡한 제어를 필요로 하는 분야의 알고리즘의 검증 방법으로 많이 사용된다.

본 논문이 설명하는 발사관제시스템은 제어기 입출력 회로와 케이블 검증을 위해 전자의 방법을 사용했고, 제어공정 알고리즘 검증을 위해 전자와 후자의 방법을 혼합하여 사용하였다. 시뮬레이터의 상세한 설계와 플로차트 형식의 구체적인 알고리즘을 기술하여 유사한 시스템의 설계와 구현에 적용할 수 있도록 하였다. 특히 일반적인 소프트웨어 시뮬레이터는 별도의 컴퓨터나 PLC(Programmable Logic Controller)에 제어 응답로직을 탑재하여 구현 되지만, 본 논문에서는 자체 제어기에서 모든 시뮬레이션이 이루어 질 수 있도록 제어시스템을 개발하였다.

2장에서 발사관제시스템의 구성, 기능 및 운용을 설명한다. 3장은 하드웨어 시뮬레이터의 설계와 구현 그리고 시험 및 결과를 기술한다. 4장에서 소프트웨어 시뮬레이터의 설계 및 구현을 기술한다. 5장은 결론으로서 시뮬레이터의 구현 결과를 고찰하고 향후 개선 방안에 대해 기술한다.

II. 시스템 구성 및 운용

2.1 시스템 구성

발사관제시스템은 발사체의 지상 발사준비를

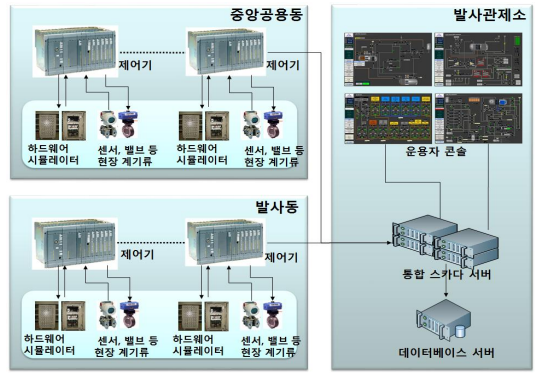


Fig. 1. Composition of Launch Control System



Fig. 2. Launch Control Center(Left), Local Control Room(Right)

수행하며 발사체와 발사대의 기계설비 및 추진제 공급설비를 제어한다. 제어시스템은 발사동(Launch Building)과 중앙공용동(Centralized Building)에 분산 설치 및 운용되며, Fig. 1과 같은 구성으로 설계하였다. 시스템은 설비별 운용자 콘솔, 통합 스카다(SCADA, Supervisory Control And Data Acquisition) 서버, 데이터베이스 서버, 설비별 제어기 및 시뮬레이터로 구성된다.

운용자 콘솔은 현장상태를 실시간으로 감시 및 제어하거나 필요 시 운용자 교육용 콘솔의 기능을 수행하도록 개발하였고, Fig. 2와 같이 발사관제소(Launch Control Center)와 현장 운용실(Local Control Room)에 배치되어 있다. 발사관제소는 발사운용 시 사용되고 현장 운용실은 현장 점검 및 시험을 위해 사용된다. 통합 스카다 서버는 제어기 데이터를 수집하여 운용자 콘솔로 전달하는 기능과 운용자 콘솔 명령을 수신하여 제어기에 전달하는 기능을 수행한다. 데이터베이스 서버는 통합 스카다 서버를 통해 전달되는 모든 정보 저장 기능을 수행한다.

설비별 제어기는 발사장의 주요 7개 설비제어를 위해 제어공정 알고리즘과 시뮬레이션 알고리즘을 탑재하고 있으며, 고 신뢰도 가용성을 유지하기 위해 삼중화로 개발하였다. 그리고 제어기와 제어 케이블의 무결성 검증을 위해 하드웨어



Fig. 3. SCADA Server(Left), Controller(Right)

시뮬레이터를 개발하여 활용하였다.

본 시스템 개발 시 콘솔과 서버에는 GE Fanuc사의 iFix 4.5, Proficy Historian 3.1a 및 Proficy SQL 2000 소프트웨어가 사용되었다. 제어기는 ICS Triplex사의 Trusted 시스템으로 구성하고, 알고리즘 개발 툴은 IEC 61131-3 기반의 FBD(Functional Block Diagram), LD(Ladder Diagram), ST(Structured Text), SFC(Sequential Function Chart) 언어 사용이 가능한 ISaGRAF 3.51을 사용하였다[6].

2.2 시스템 운용

발사관제시스템은 지상장비제어계와 발사준비제어계로 구분된다. 지상장비제어계는 Fig. 4와 같이 발사준비제어계로부터 명령을 받아 발사대 설비를 제어하며, 발사대 설비의 추진제 및 고압가스 등의 공급 준비가 완료되면 전자식유공압패널 제어를 통해 발사체로 공급을 수행한다.

발사 운용은 Fig. 5와 같다. D-2일째 발사체 이송, 기립, 전기 케이블 접속 및 기능점검, 온도제어를 수행하며, D-1일째 발사 운용 리허설 수행, D-0일 발사 2시간 전 추진제 충전의 착수와 발사 15분전 자동 발사가능 수행 및 발사를 수행한다.

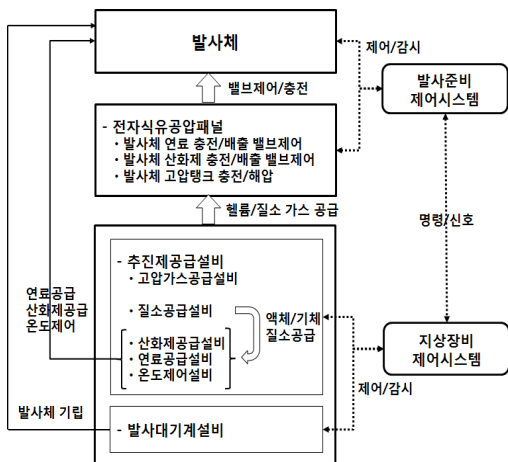


Fig. 4. Configuration of Control System and Control Object



Fig. 5. The Launch Preparation and Launch Operation

시뮬레이터는 성공적인 발사운용을 위해 공장 시험, 현장인증시험 그리고 시운전 기간에 제어기와 설비운용 알고리즘 검증 용도로 사용된다. 이 단계의 검증이 완료되면 발사운용에 필요한 최종 통합 연계시험 검증이 이루어지게 된다. 하드웨어 시뮬레이터와 소프트웨어 시뮬레이터는 시스템 및 운용시험 검증을 위해 개발된 시뮬레이터이며, 다음의 요구조건을 만족한다.

- 운전자 교육을 위한 모사
- 타 시스템과의 인터페이스 신호 모사
- 발사 운용에 필요한 시퀀스 모사
- 제어기 입출력 회로의 무결성 검증
- 제어 대상 케이블의 무결성 검증
- 제어공정 알고리즘 검증

III. 하드웨어 시뮬레이터

본 절에서는 발사관제시스템 제어기, 입출력 모듈 및 케이블 회로 점검에 사용된 하드웨어 시뮬레이터에 대해 기술한다. 하드웨어 시뮬레이터는 일반적으로 전기적인 디스크리트 또는 아날로그로 신호 입출력을 모사하기 위해 사용되지만, 본 논문에서는 일반신호 입출력 시뮬레이터 외에도 발사대 및 발사체의 특정 기능을 수행하는 시뮬레이터도 제시한다. 본 논문에 제시된 시뮬레이터는 다음과 같다.

- 일반신호 입출력 시뮬레이터
- 전자식유공압패널 시뮬레이터
- 기폭신호 입출력 시뮬레이터
- 발사체 탑재 시뮬레이터

3.1 일반신호 입출력 시뮬레이터

일반신호 입출력 시뮬레이터는 제어기의 터미

Table 1. Number of Channels for Systems

시스템	디지털		아날로그	
	입력	출력	입력	출력
- A 설비	440	280	0	0
- B 설비	920	360	160	0
- C 설비	200	80	40	0
- D 설비	480	160	80	40
- E 설비	360	80	40	0
- F 설비	400	200	320	120
- G 설비	600	480	0	0
합계	3,400	1,640	640	160

Table 2. Number of Channels for Simulators

시스템	디지털		아날로그	
	입력	출력	입력	출력
시뮬레이터 #1	320	640	0	98
시뮬레이터 #2	320	640	0	98
시뮬레이터 #3	192	384	196	272
시뮬레이터 #4	192	384	196	294
합계	1,024	2,048	392	762



Fig. 6. General Purpose Input-Output Signal Simulators

널 블록 접속부에 커넥터를 이용하여 연결이 된다. 본 시뮬레이터는 다수의 디지털 및 아날로그 신호를 생성하거나 수신할 수 있어 제어기 및 입출력 모듈의 동작 상태점검을 위한 용도로 사용된다. Table 1은 설비 제어를 위해 제어기에 탑재된 가용 입출력 모듈의 총 채널수를 나타낸다. 이 채널의 건전성은 공장시험 또는 현장시험 시 시뮬레이터를 사용하여 점검이 이루어진다.

시뮬레이터는 Table 1의 시스템별 채널의 수와 시뮬레이터 판넬에 장착 가능한 커넥터의 수를 고려하여 Table 2와 같이 4개의 시뮬레이터로

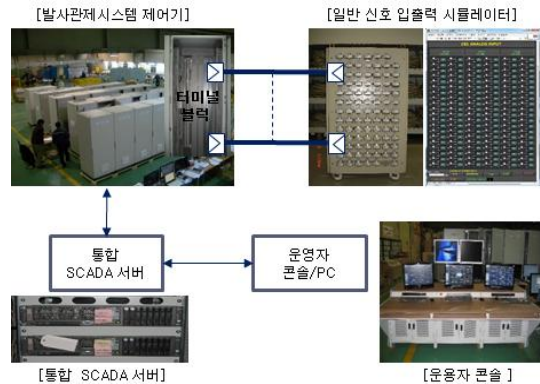


Fig. 7. Verification Test of Controllers and Input-Output Modules using Simulator

분리하여 개발하였다. 개발된 시뮬레이터의 형상 및 관련 케이블은 Fig. 6과 같다. 시뮬레이터의 하드웨어 장비는 NI사 제품을 소프트웨어는 LabView 8.2를 사용하였다.

시뮬레이터를 이용한 점검 과정은 Fig. 7과 같다. 운영자 콘솔에서 보낸 출력 신호는 통합서버와 제어기를 통해 최종 시뮬레이터로 전달된다. 반대로 시뮬레이터에서 발생된 신호는 제어기와 통합서버를 거쳐 최종 운영자 콘솔의 입출력 모듈의 점검 창에 그 결과가 표시된다.

3.2 전자식유공압패널 시뮬레이터

전자식유공압패널은 총 11개로 구성되며 지상에서 발사체로의 가스 공급을 제어하는 시스템이다. 전자식유공압패널 시뮬레이터는 전자식유공압패널 내부에 설치된 솔레노이드 밸브 코일의 특성 값을 시뮬레이터에 반영하여 동작되도록 Fig. 8과 같은 형상으로 설계 개발되었다.

시뮬레이터를 이용한 점검 과정은 Fig. 9와 같다. 즉, 시뮬레이터를 이용한 모사입력 값은 제어기를 거쳐 운영자 콘솔 화면의 센서의 상태를 변경 시키며, 운영자 콘솔 화면에서의 조작된 밸브의 상태 값은 시뮬레이터에 LED, 전압, 전류로 표시된다. 또한, 운용 알고리즘과 연계하여 알고



Fig. 8. Electro-Pneumatic Panel Simulator

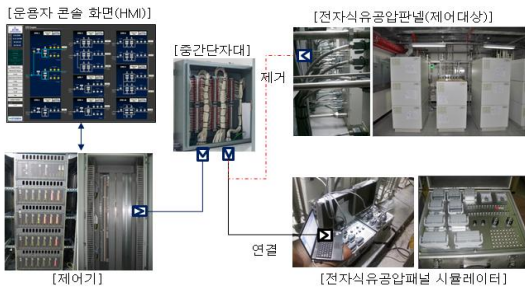


Fig. 9. Verification Test of Signals and Algorithms using Simulator

리즘 정상 동작 유무를 운용자 화면을 통해 확인이 가능하다.

전자식유공압패널 시뮬레이터가 일반신호 입출력 시뮬레이터와 다른 점은 다음과 같다. 전자는 시뮬레이션을 통해 제어기와 제어 대상 간 연결된 전 구간의 케이블 및 제어회로의 점검을 수행하는 반면, 후자는 제어기와 입출력 모듈의 점검 용도로만 사용되며, 점검 구간이 입출력 모듈을 포함한 제어기로부터 제어기 터미널 블록까지로 제한된다.

3.3 기폭신호 입출력 시뮬레이터

기폭신호 입출력 시뮬레이터는 발사체 이륙시 발사체와 연결된 지상 장치와 분리를 위해 기폭제어기 점검과 Pyrobolts 기폭 시험을 위한 발사체 이륙신호(Lift off Contact) 모사 용도로 사용된다.

Figure 10은 기폭제어기 점검과 Pyrobolts 기

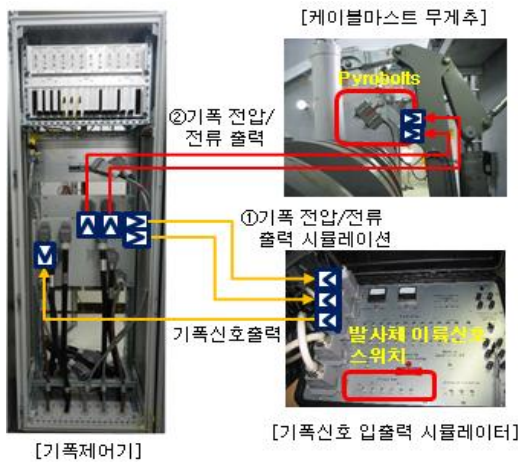


Fig. 10. Verification Test of Explosion Controller and Pyrobolts using Simulator

폭 시험을 위한 시뮬레이터의 연결 구성도이다. 시뮬레이터는 Pyrobolts와 동일한 저항 값을 갖도록 하여 Pyrobolts에 걸리는 전압과 전류를 시뮬레이터를 통해 측정을 한다. 이 측정을 통해 기폭제어기 상태 점검을 수행한다. 이 점검 과정이 완료되면, 시뮬레이터로 연결된 기폭 전압/전류 출력 케이블을 제거하고 Pyrobolts 케이블을 기폭제어기에 연결하여 기폭시험을 수행한다. 이 과정을 통해 기폭제어기, Pyrobolts 케이블 그리고 Pyrobolts 간 전반적인 동작 상태에 대한 검증을 수행한다.

3.4 발사체 탑재 시뮬레이터

발사체 탑재 시뮬레이터는 발사체 탑재 컴퓨터 구동용 전원공급 상태, 밸브구동 신호를 LED로 표시하며, 발사체 탑재 아날로그 센서 및 발사체 이륙신호 등을 모사한다. 시뮬레이터를 이용한 케이블 및 동작 회로의 점검과정은 Fig. 11과 같다. 센서 모사입력 값은 기폭제어기 및 발사준비제어기를 거쳐 운용자 콘솔의 센서 값을 변경 시키며, 운용자 콘솔에서 밸브 및 전원공급 구동 명령은 시뮬레이터의 LED와 전압, 전류계를 구동하여 케이블 및 동작 회로의 점검을 수행한다.

발사체 이륙신호는 Fig. 11과 같이 발사체가 이륙 시 발사체에 부착된 커넥터가 분리됨과 동시에 발생된다. 이 신호는 발사체 이륙 시 발사체에 부착된 전기 엄빌리컬 플레이트 제거를 위한 Pyrobolts 기폭 구동 신호로 사용된다.

발사체 이륙신호가 발생되면, Pyrobolts 기폭을 위한 구동 전압과 전류가 기폭제어기에서 출력된다. 발사체 이륙신호 발생 후 기폭신호의 출

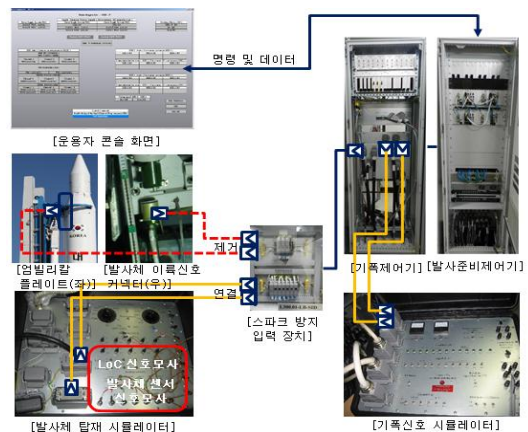


Fig. 11. Verification Test of Control Cables and Circuits using Simulator

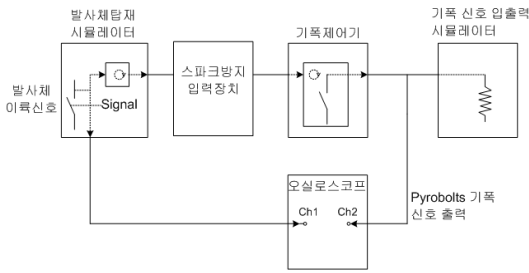


Fig. 12. Configuration of Measuring Explosion Delay Time

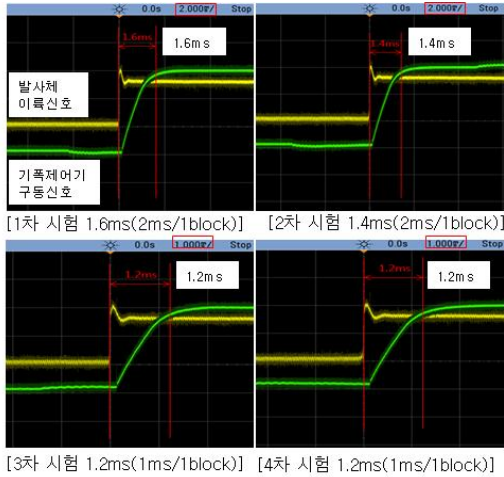


Fig. 13. Test Result of Measuring Explosion Delay Time

력 지연시간 요구조건은 50ms 이하이다. 요구 조건을 기준으로 Fig. 12와 같은 시험 구성을 통해 네 차례 시험 수행결과 최대 1.6ms 최소 1.2ms의 결과를 얻었다. 측정 결과 그래프는 Fig. 13과 같다. Fig. 13에서 노란색 선은 발사체 이륙 신호이며, 녹색 선은 기록제어기의 출력신호이다. 지연시간은 발사체 이륙신호 발생 후 기록제어기 출력신호의 발생 시간까지의 지연시간을 측정 한 결과이다.

IV. 소프트웨어 시뮬레이터

본 절에서는 발사관제시스템 알고리즘 검증에 사용된 소프트웨어 시뮬레이터의 설계 및 구현 방법에 대해 기술한다. 소프트웨어 시뮬레이터는 일반적으로 별도의 컴퓨터나 PLC에 제어 응답모직을 탑재하여 구현되지만, 본 논문에서 제시한 소프트웨어 시뮬레이터는 자체 제어기에서 모든 시뮬레이션이 이루어 질 수 있도록 개발 하였다.

이러한 프로그램을 개발하기 위해 신호 입력과 출력 함수 그리고 시뮬레이션 알고리즘이 포함된 운용 알고리즘의 설계 및 구현이 수행 되었다.

4.1 입출력 함수의 설계

일반적인 제어기 입출력 신호의 흐름은 Fig. 14와 같이 동작한다. 즉, 센서의 상태정보는 제어기의 입력 신호로 전달되며, 운용 알고리즘은 센서의 상태정보와 운용 조건에 따라 출력 신호를 생성한다. 출력 신호로 구동된 밸브, 펌프 등의 상태정보와 밸브 및 펌프 등 주변의 센서정보가 입력 신호로 제어기에 다시 전달된다. 따라서 소프트웨어 시뮬레이터는 이러한 과정을 모사해야 한다.

이러한 과정을 모사하기 위해 먼저 Fig. 15와 같은 신호 입출력 함수를 설계 하였다. 입출력 함수의 운용모드는 실제운용 모드와 시뮬레이션 모드로 구성되며, 운용자에 의해 모드 전환이 이루어진다. 입출력 함수의 변수는 실제 입출력 변수와 시뮬레이션 입출력 변수로 구분되며, 시뮬레이션 모드의 경우 모든 입출력은 시뮬레이션 입출력 변수를 통해 수행되도록 하여 안전을 보장한다.

밸브, 펌프와 같은 구동기는 구동 제어신호를

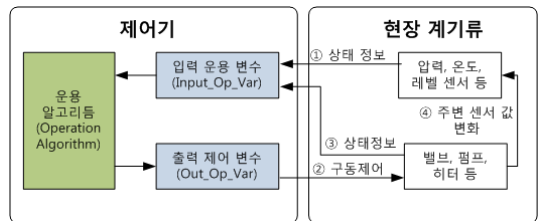


Fig. 14. Operating Procedure of General Control Input-Output Signal

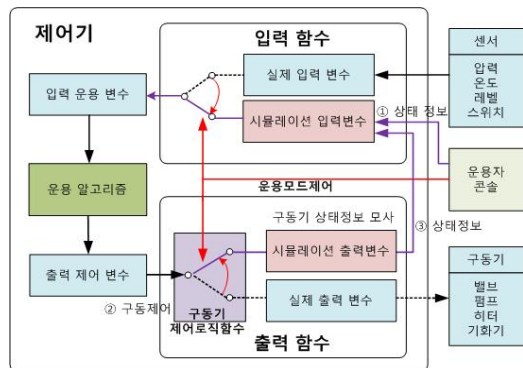


Fig. 15. Design of Control Input-Output Simulation Function



Fig. 16. Input Window of Discrete and Analog Value Simulation

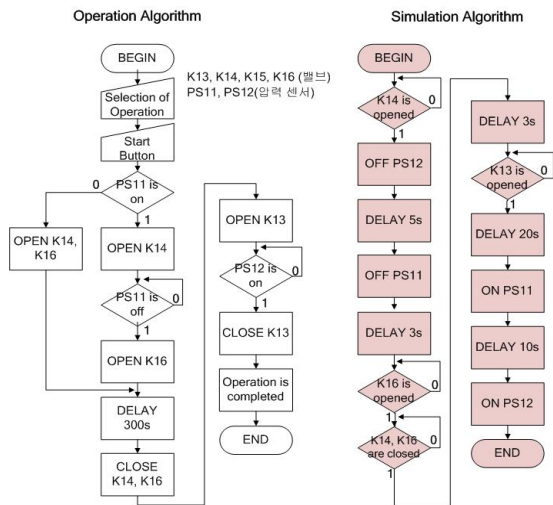


Fig. 18. Example of Operation(Left) and Simulation(Right) Algorithm (Desaturation)

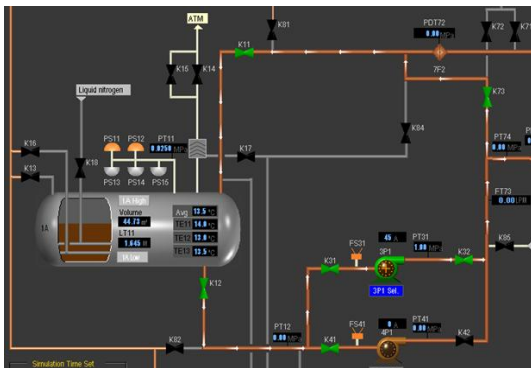


Fig. 17. Simulation Display Window

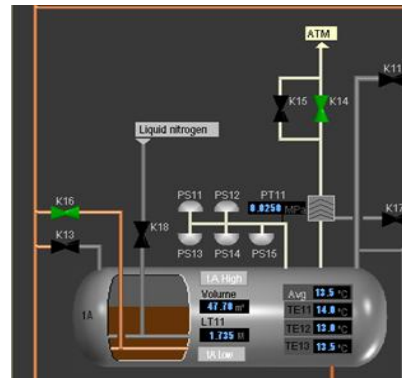


Fig. 19. Example of Operator Console Display

받으면 동작 후 동작 상태에 대한 응답 신호가 제어기의 입력 신호로 전달되어야 한다. 이에 대한 출력 함수의 모사 과정이 Fig. 15와 같다. 각 구동기는 대상별로 제어로직이 구성되며, 이 로직이 구동기에 대한 상태 정보를 자동으로 모사해 준다. 구동기에 대한 상태 정보로 밸브는 리미트 스위치 신호, 펌프는 정상 가동 또는 오류 신호 등이 된다. 이러한 신호는 운용자 콘솔에서도 수동 모사가 가능하며, Fig. 16과 같은 입력창들을 사용한다.

구동기 제어로직이나 운용자를 통해 모사되는 정보는 운용자 콘솔 화면의 밸브, 펌프, 센서의 구동 상태 정보로 표시되며, 동작 결과를 Fig. 17과 같은 화면을 통해 확인한다.

4.2. 시뮬레이션 알고리즘의 설계

시뮬레이션 알고리즘은 밸브, 펌프 등과 같은

구동기 동작 시 영향을 주는 주변 센서들의 값을 모사하여 운용 알고리즘이 진행 되도록 설계하였다. Fig. 18은 개발 적용된 운용 알고리즘과 시뮬레이션 알고리즘의 예를 나타낸다.

Figure 18의 운용 알고리즘은 Fig. 19 탱크 하부에 질소가스를 불어넣어 유증기를 제거하는 기능을 수행하는 알고리즘이다. 이 알고리즘이 자동으로 운영되기 위해 Fig. 18에 제시된 시뮬레이션 알고리즘이 운용 알고리즘과 동시에 구동 되도록 설계 하였다. 알고리즘 동시 운용을 위해 제어 프로그램 코딩에 사용된 운용 알고리즘은 Fig. 20과 같다.

Figure 19의 PS11, PS12는 탱크의 압력을 측정하는 센서를 나타낸다. K16과 K13 탱크 내에 질소가스 공급을 위한 밸브로 K16은 유증기 제거

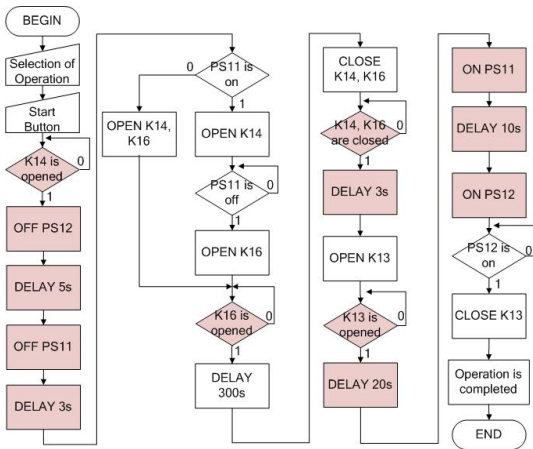


Fig. 20. Example of Operation Algorithm including Simulation Algorithm (Desaturation)

용이며, K13은 탱크 가압용이다. 또한, K14는 탱크 내부 압력 해압 밸브이다.

Figure 20의 알고리즘을 설명하면 다음과 같다. 알고리즘이 시뮬레이션 모드로 구동되면 먼저 K14번 해압용 밸브가 열려 있는지를 검사한다. 그 후 압력센서 PS11, PS12를 OFF로 모사하고, PS11이 OFF 상태이면 K14, K16 밸브를 열어 탱크 내 압을 해압 및 질소 가스 공급을 시작한다. 300초 후 K14, K16 밸브를 닫고 3초 후 K13 밸브를 열어 탱크를 가압한다. PS11과 PS12를 ON으로 모사하고, PS12가 ON이 되면 K13 밸브를 닫아 탱크 내 압을 일정 수준으로 유지되도록 시뮬레이션을 수행한다.

운영자는 필요에 따라 PS11이나 PS12의 값을 변경 모사하여 운용 흐름을 변경하기도 한다. 실제 운용모드로 알고리즘이 구동되면 시뮬레이션 블록에서 모사된 모든 값들은 무시되며, 실제 입력 변수의 값을 기준으로 수행된다.

발사장의 주요 7개 설비 제어를 위한 발사관제시스템의 모든 운용 알고리즘은 상기 소프트웨어 시뮬레이터의 설계 및 구현에서 소개된 방법으로 개발된다. 즉, Fig. 20과 같은 방법으로 개발하여 발사관제시스템 각 운용 알고리즘 검증을 수행한다.

V. 결 론

본 논문에서는 발사체 발사관제설비의 오류를 사전에 제거하기 위한 시뮬레이터의 설계와 구현 결과를 다루었다. 결론부에서는 하드웨어 시뮬레

이터와 소프트웨어 시뮬레이터를 개발하고 이용하는 과정에서 도출된 단점 및 제약 사항을 확인하고 그의 개선 방향을 제시하고자 한다.

하드웨어 시뮬레이터 중 일반신호 입출력 시뮬레이터는 질량과 부피가 커져 이동성에 제약사항이 많았다. 또한 점검을 위한 사전 준비 작업 중 케이블 연결 작업에 많은 시간이 소요되는 단점이 발견 되었다. 특정 목적을 가진 시뮬레이터는 이동이 용이하고 결선작업이 단순한 반면, 그 기능 이외의 다른 용도로의 재활용이 불가함을 확인하였다. 따라서 발사관제시스템과 같은 설비에는 범용 적으로 원하는 신호를 모사할 수 있고 휴대가 용이한 하드웨어 시뮬레이터의 개발이 필요하다.

소프트웨어 시뮬레이터의 경우 복잡한 알고리즘을 사용하며, 병렬처리를 수행해야 함으로써 LD나 ST 언어로는 구현에 어려움이 있었다. 이를 해결하기 위해 SFC 언어를 적용하였으나 실행코드가 LD나 ST로 개발된 언어에 비해 매우 커지는 단점이 발견되었다. 또한 시뮬레이션 변수나 시뮬레이션 알고리즘의 추가도 실행 코드의 크기를 증가 시키는 한 요인으로 작용되었다. 이는 저장 메모리 공간의 제약 사항을 가진 상용 PLC의 경우 불리한 점으로 작용한다. 이 경우 처리속도가 빠르고 메모리가 큰 고가의 PLC를 사용할 수 있으나 개발 비용 상승과 결부되어 적절한 해결방안이라 볼 수 없다. 따라서 설계 단계부터 제어 대상의 규모와 제어알고리즘의 분량을 사전 검토하고 이에 맞는 적절한 상용 제어기를 선정하거나 적합한 제어기를 개발해야 한다. 더 우선적으로는 알고리즘이 복잡하지 않도록 설계해야 하며, 알고리즘 구현단계에서 단순한 알고리즘은 ST나 LD로, 디버깅 및 구현이 어려운 복잡한 알고리즘은 SFC언어를 선택하여 개발하는 방법을 사용할 수 있을 것이다.

References

- 1) W. Choi, H. U. Chung, J. H. Seo, I. H. Hong, "KSLV-I Launch Control Center Simulation System Prototype Test for the Real Time data Processing", *Proceedings of KSAS Conference*, 2004, pp.603~608
- 2) M. P. Jung, K. N. Kim, "The Design and Implement of Program Simulator for testing PLC-based control systems", *Proceedings of KIISE Conference*, Vol. 26, No. 2, 1999, pp.63~

65

3) S. H. Bae, Y. M. Kim, "A Case Study of Virtual PLC Validation System's Implementation: In Case of An Automobile Trim Line", *Journal of the Korea Society for Simulation*, Vol. 19, No. 2, 2010, pp. 9~16.

4) H. S. Park, S. Y. Jeong, D. B. ANH, J. W. Park, H. W. Choi, "Simulation-based verification

of PLC program", *Proceedings of KSPE Spring Conference*, 2009. pp.25~26.

5) C. G. Lime, Y. S. Ahn, S. C. Park, G. N. Wang, "A Study of PLC Simulation Applying for Steel Product Line", *Proceedings of SCCE Conference*, 2011. pp.120~124

6) "ISaGRAF User's Guide", ICS Triplex, pp.A-138