

## 실시간 제어 시스템 설계 지원 기술동향과 실제 : CEMTool을 중심으로

권 옥 현, 김 기 백, 최 성 규, 김 형 진, 한 수 희

서울대학교 전기공학부 제어정보시스템 연구실

### 1. 서론

제어공학은 2차 대전이후 학문적으로 체계화되고 실용화되어 오늘날 산업의 모든 분야에 적용, 확대되어 가고 있으며, 최근 들어 제어공학은 상당한 발전이 있었고 실용적인 면에서도 이용범위가 넓어지고 있다. 이러한 제어 공학의 발전에 제어기 모의실험용 툴의 기능 확대가 중추적인 역할을 하고 있다. 초창기 제어기 모의실험용 툴은 초보적인 제어용 그래프를 제공하고, 몇 가지의 라이브러리를 포함한 것이 고작이었다. 하지만 현재의 제어 모의실험용 툴은 수많은 라이브러리를 포함하고 화려한 그래픽 사용자 연결성을 지니며, 다양한 기능을 제공하여 쉽고 간편하게 제어기를 설계할 수 있도록 많은 도움을 주고 있다. 그리고 수많은 제어 모의실험용 툴들이 각각 나름대로의 특색을 갖추며 상호 보완 및 견제를 하며 서로의 상승 효과를 나타내고 있다. 특히, 요즘의 모의실험용 툴은 모의실험 그 자체만을 중시하지 않으며, 모의실험이 바로 실시간 제어기 구현이 될 수 있는 야심작들을 만들어 내기에 이르렀다. 모의실험이 바로 실시간 제어기 구현이라는 목표를 이루기 위해서는 모의실험용 툴 개발을 위한 소프트웨어뿐만이 아니고 많은 복합적인 기술을 요구한다. 즉, 하드웨어 설계 기술, 자동 코드 생성 기술, 제어 기술, 시스템 프로그래밍 기술 등의 수많은 기술을 불과 수년 전만 해도 제어용 국산 모의실험용 툴은 없었으며, 국내의 제어 모의실험을 MATLAB이 거의 담당하고 있었다. 하지만 서울대학교 제어계측신기술연구센터에서 개발한 제어용 모의실험용 툴인 CEMTool은 국내에서 급격히 보급되고 있는 실정이다. CEMTool은 블록 다이어그램 기반과 명령어 기반의 프로그래밍 방법을 모두 사용 가능하게 지원하며, 제어용 라이브러리, DSP 보드를 위한 자동 코드 생성, 다양한 플랜트 라이브러리, 실시간 소프트웨어/하드웨어 모의실험 등의 기능 등을 갖추어 대학의 실험 실습용이나 제어기 설계 등에 많이 이용되고 있는 실정이다. 본 논문은 다음과 같은 구성으로 이루어져 있다. 제 2장에서는 실시간 제어 시스템 설계에 대한 전반적

인 개요를 설명하고, 제 3장에서는 모의실험의 종류 및 기능을 소개하며, 제 4장에서는 많은 모의실험용 툴들에 대해 비교 평가한다. 제 5장에서는 제어 시스템의 구성 요소들을 알아 보고, 제 6장에서는 Software-In-the-Loop 모의실험에 대해 자세히 알아 본다. 제 7장에서는 제어 실험의 예를 대모로 보여주며 제 8장에서는 결론을 내린다.

### 2. 실시간 제어시스템 설계 개요

실시간 제어 시스템을 설계하려면 다양한 모의실험과 실제 실험 과정을 거쳐야 한다. 먼저 대상 플랜트가 선정되면 물리적인 대상 플랜트를 수학적 식으로 나타내야 하는데, 이러한 과정을 대상 플랜트의 모델링이라고 한다. 모의실험을 거쳐서 대상 플랜트의 제어기를 설계하는 데에 있어서 모델링은 필수적이며, 특히 현대 제어 이론에서는 대부분이 이런 모델을 기반으로 제어기를 설계한다.

대상 플랜트에 대한 모델링이 끝나면 적당한 제어 알고리즘을 선정해야 한다. 고전 제어에서 현대 제어에 이르기까지 제어기법은 매우 다양하며 시스템 특성에 맞추어서 제어 알고리즘을 선정하게 된다. 주로 산업체 현장의 단일입출력 플랜트에서는 PID (비례-적분-미분) 제어기가 많이 사용되고 있으며, 이는 PID 제어기가 설계의 간편성, 강인성, 설계의 다양성 등의 많은 장점을 지녔기 때문이다. 다변수 시스템에는 최적 제어 기법인 LQ (linear quadratic) 제어, RH (receding horizon) 제어, 강인 제어 기법인  $H_{\infty}$  제어, 지능 제어기법인 fuzzy 제어, neural 제어 등 많은 제어기 등이 연구되고, 또 사용되고 있다.

제어기가 선정되면 플랜트의 모델링식과 제어식을 사용하여 제어기의 성능해석을 하여야 한다. 성능해석을 위해서는 모의실험이 필수적이며, 요즘은 모의실험용 툴들의 발달로 쉽고 빠르게 성능해석을 할 수 있다. 모의실험을 통하여 성능해석을 하는 방법에는 두 가지 방법을 사용할 수 있으며 먼저 컴퓨터 한대를 이용하여 실시간 성능을 고려하지 않고 시스템의 안정도, 시간영역 성능, 주파수 영역

성능 등을 알아보는 off-line 모의실험이 그 한가지이다. 이러한 모의실험이 끝나면, 이 제어기의 실시간 성이나 플랜트/제어기 연결성 등을 실험하기 위해 여러 대의 컴퓨터로 플랜트 동역학식/제어 알고리즘을 분리하여 실험하는 Software-In-the-Loop (SIL) 모의실험을 진행한다. 실제 설계과정에서 SIL 모의실험 과정은 생략되기도 한다.

제어기의 성능을 모의실험을 통하여 확인하였으면, 실제의 대상 플랜트에 제어기를 적용하는 실험을 한다. 이는 범용 제어기를 구현하는 과정이며, 또 구현된 제어기의 실시간으로 실제 플랜트 제어 성능을 확인할 수 있는 과정이다. 이를 위해 모의실험용 틀은 모의실험 언어를 실제의 컴퓨터에서 많이 사용하는 언어(예를 들면, C나 C++)로 변환하는 기능을 제공하여 주어야 하며, 요즘은 많은 제어용 모의실험 틀들이 이 기능을 제공하고 있다.

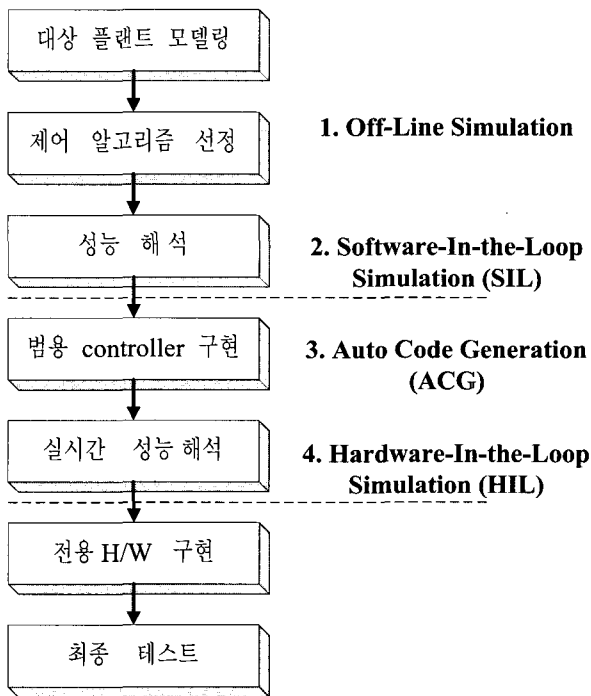


그림 1. 실시간 제어시스템 설계.

이렇게 생성한 언어와 적절한 인터페이스를 통하여 실제의 대상 플랜트의 제어 성능을 확인할 수 있다. 이러한 과정을 Hardware-In-the-Loop (HIL) 모의실험이라 부른다.

최종적으로, 모의실험을 통하여 설계한 제어 알고리즘은 각각의 대상 플랜트에 맞는 전용 하드웨어를 설계하여 전용 하드웨어에 맞는 언어를 이용하여 구현하여야 한다. 구현된 전용 제어기는 수많은 테스트를 거쳐서 실제 제어기로서 구실을 하게 된다. 이에 대한 전체 과정을 그림으로 나타내면 위와 같다.

### 3. Simulation 기능

시뮬레이션 (Simulation)에는 제어기 (controller)와 플랜트 (plant)의 역할을 무엇이 하는가에 따라 4가지 종류로 나누어진다.

#### 3.1 Off-Line Simulation

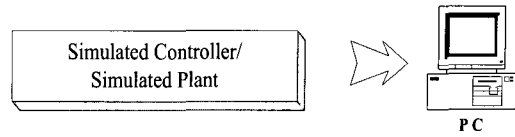


그림 2. Off-Line Simulation.

보통의 CACSD용 소프트웨어에서 지원하는 형태로 저렴한 가격으로 주변장치 필요 없이 시뮬레이션이 가능하다. 그러나 time-scaling과 real-time simulation같은 시간 변수에 대한 사용자의 접근이 용이하지 못한 단점이 있다.

#### 3.2 Software-In-the-Loop Simulation

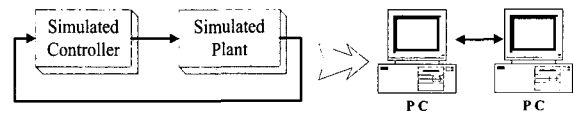


그림 3. On-Line Simulation.

제어기와 플랜트를 분리해서 두 PC에 그 기능을 각각 분담시켜 더욱 현실에 가까운 시뮬레이션을 할 수 있다. 두 PC간의 연결은 직렬(serial) 통신, 병렬(parallel) 통신, AD/DA 보드를 이용한 통신, LAN통신을 사용할 수 있고, LAN통신을 사용할 경우 소프트웨어와 LAN만을 가지고 저렴한 비용으로 Software-In-the-Loop의 시뮬레이션을 해서 time-scaling과 real-time 시뮬레이션 같은 기능을 구현할 수 있다. 사용자에게 시간 변수에 대한 좀 더 유연한 접근을 가능하게 한다. 따라서 현실에 더욱 근접한 실험을 가능하게 할뿐만 아니라 매우 빠른 동적 특성을 가진 시스템에 대해서도 시뮬레이션이 가능하다. 또한 고가의 하드웨어 장비 없이 실험실 내부에서 여러 대상모델에 대하여 시뮬레이션이 가능하므로 교육용으로서도 매우 가치가 있을 것이다. 아래 그림은 발전 계통에 이 시뮬레이션 방법이 어떻게 쓰일 수 있는가를 보여주고 있다.

보일러, 터빈과 발전기를 PC1, PC2, PC3가 각각 그 역할을 대신한다.

이와 같은 시뮬레이션에서는 각각의 PC간 동기성, 데이터 전송 속도 등이 문제될 수 있으며, 그것에 따르는 지연은 많은 분석의 여지를 남겨 놓고 있다.

실제의 제어기나 실제의 플랜트의 일부를 포함한

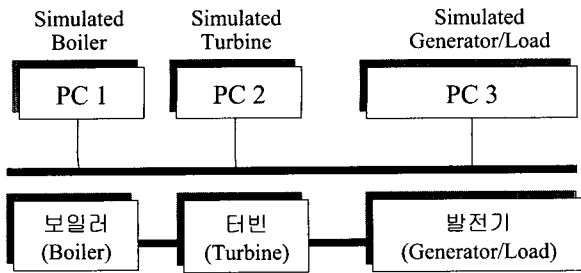


그림 4. 발전 계통에서의 시뮬레이션 응용.

### 3.3 Hardware-In-the-Loop Simulation

#### ■ Hardware-In-the-Loop Simulation ( I )

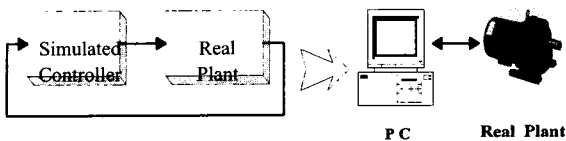


그림 5. Hardware-In-the-Loop.

#### ■ Hardware-In-the-Loop Simulation ( II )

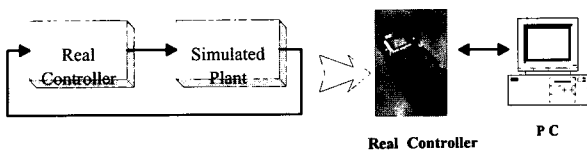


그림 6. Software-In-the-Loop.

실시간 시뮬레이션을 뜻한다. 새로 개발한 제어기나 플랜트를 위험 없이 실험실 내부에서 실제 환경과 동일한 환경에서 검사할 수 있다. 실제의 위험한 상황이나 검사하기 어려운 조건에서 매우 유용한 시뮬레이션이 될 것이다. 체계적인 제어 로직의 최적화 과정 및 검증을 통해 database 확보가 가능하게 되어 시스템의 설계 비용과 개발 시간을 단축시킬 수 있다. 실제로 McDonnell Douglas사는 차세대 위성 발사체를 위한 flight-control 제어기 개발에 이 시뮬레이션 방법을 써서 기존의 방법보다 50%이상 시간과 비용 절감의 효과를 가져왔고, Auto Code 기능을 이용하여 30,000줄 이상의 Ada 코드를 생성하였다.

## 4. Computer-Aided Control System Design

### 4.1 CACSD란?

최근 복잡한 제어 시스템의 증가에 따라 미분 방정식이나 행렬 등으로 표현되는 제어시스템의 해석 및 설계 문제들이 손으로 풀기에는 어려운 경우가 많아지고 있다. 또한, 제어시스템 해석의 방법론적 측면에서도 복잡한 계산 뿐만 아니라 여러 가지 종

류의 그래프 이용하는 작업을 동반하는 경우가 많으며, 제어시스템의 설계 문제에서는 반복적인 작업들을 수행하여야 한다는 특성이 내재되어 있다. 따라서, 최근 수십 년간의 제어 시스템 해석 설계에서 컴퓨터의 사용은 필수적이라고 할 수 있다. 그러나, 이런 문제들의 해결을 위하여 전통적인 방법인 C나 Fortran 등으로 직접 프로그램을 작성하는 것은 데이터 입출력, 그래프 출력 등의 부가적인 일에 많은 시간을 들여야 하므로 용이한 작업이 아니다. 따라서, 이러한 용도로 그림 7과 같은 기능 구성을 가진 전문 컴퓨터 패키지들이 개발되었는데, 그 예로 CEMTool을 비롯하여 MATLAB, MATRIXx, Program CC 등이 있으며 이와 같은 제어시스템 해석 및 설계용 패키지를 이용하여 문제들을 직접 풀어보고 실험해 보는 것이 점차 일반적인 경향으로 자리잡고 있다.

### 4.2 세계적인 CACSD 분야의 경향

앞서 언급한 바와 마찬가지로 최근까지 여러 가지 CACSD 패키지들이 세계적으로 개발되고 실제적으로 사용되고 있다. 표 1과 2에 나타난 바와 같이 몇 종류의 특징을 간단하게 살펴보면, MATLAB은 주로 대학교육, 연구용, 산업용에 널리 쓰이는 가장 광범위한 분야에서 이용되는 패키지이며, 다수의 범용 real-time processing unit 제작 회사들이 지원하는 모의 실험 환경 중 제어기 구현에 해당하는 부분에도 이용된다. 이에 반하여 MATRIXx는 산업 응용과 자체적인 실시간 실험 환경 구축에 강점을 가지고 있고, Mathematica는 수학 분야에 강점을 보이고 있으며, 공학용, 교육용, 기호

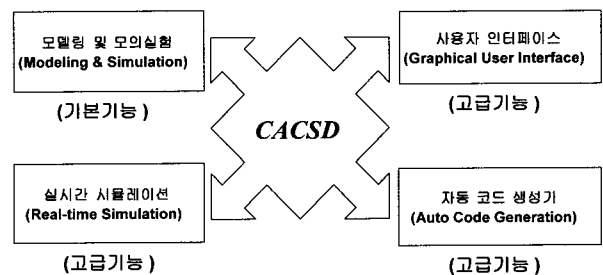


그림 7. CACSD의 기능 구성.

연산에도 응용된다. 또 ACSL은 여러 플랫폼을 제공하는 산업용 패키지로 통합 작업 환경을 제공한다. LabView는 가상 도구들을 이용하는 실험 환경을 제공하고, CEMTool은 교수 연구용 및 실시간 모의 실험용으로 발전을 계획 중인 패키지이다. 각각의 패키지들은 이와 같은 특화된 기능을 바탕으로 특정 분야에서의 적용을 목표로 발전해 가고 있으며 해당 분야에 대한 요구를 적극 수렴함으로써

표 1. 각종 CACSD의 특성별 비교.

	MATLAB	MATRIXx	LabView	OmSim	CEMTool
확장성	◎	◎	○	△	○
GUI	◎	◎	◎	△	○
블록 다이어그램	○	◎	◎	×	○
출력 형식	◎	○	○	△	◎
응용성	○	◎	△	×	△
실시간 모의실험	◎	◎	◎	×	○
예제	◎	○	○	△	○
개발자와 사용자 연계	◎	○	○	△	◎

◎: Excellent ○: Good △: Not Enough ×: Poor

표 2. 각종 CACSD의 기능별 비교.

	Mathworks	ACSL	MATRIXx	Hyperception	SNU/CISL
기본기능	MATLAB	ACSL model	Xmath	RIDE	CEMTool
블록 다이어그램	SIMULINK		SystemBuild	Block Diagram	SIMTool
자동 코드생성	Real-time Workshop		AutoCode		AUTOTool
Hardware In the Loop (HIL)	Real-time Workshop		RealSim	[Exist]	Real-time Processing Unit (가칭)
가상 도구 자료 취득			RealSim	VIDSP Studio	Variable Supervisor (가칭)
애니메이션	[Exist]	ACSL Real-time			Real-time Motion (가칭)

고정 사용자층을 확보하는데 패키지의 성패가 갈릴 정도로 중요한 요소가 되고 있다.

이와 같이 대부분의 패키지들은 모두 각기 다른 특징을 가진 서비스와 환경을 제공하지만 차세대CACSD의 기본적인 특징으로 규정할 수 있는 공통적인 몇 가지 점에서는 발전 방향이나 CACSD의 현황에서 일치하는 것을 발견할 수 있다.

첫 번째로 인터페이스 측면에서 통합 환경을 지향하는 그래픽 사용자 인터페이스의 강화가 PC나 워크 스테이션의 윈도우 환경의 일반화와 함께 가장 특징적인 것으로 생각할 수 있다. 대표적으로는 MATRIXx의 윈도우 환경이나 LabView의 실제 시스템을 모사한 형태의 시스템 구성 방식, MATLAB의 데모 기능을 생각할 수 있다.

두 번째는, 기본적으로 안정적인 수치 알고리즘 하에서, 전체 패키지의 객체 지향적 구조와 개방형 구조의 지향을 두드러진 특성이라고 할 수 있다. 그 예로 MATRIXx의 출력 데이터 형식이 일반적인 산업 시스템 형식에 직접 적용 가능한 점이나 독립

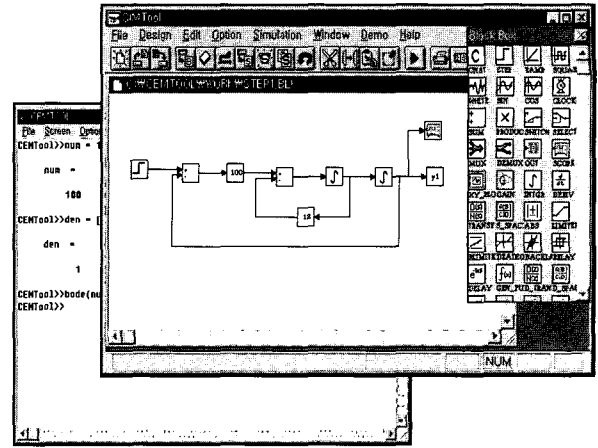


그림 8. CEMTool / SIMTool 실행 화면.

된 모듈간의 작업이 데이터 전달로 연계되는 점이 있다. MATLAB을 비롯한 다수의 패키지가 자동적으로 코드를 생성하는 것을 지원하고 있다.

이와 같은 CACSD의 기본적인 측면은, 특화 전략을 통하여 사용자를 확보하려는 각 CACSD 패키지들에서도 공통적으로 유지되고 있는 특성 추세라고 할 수 있다.

### 4.3 CEMTool / SIMTool

CEMTool은 제어계측신기술연구센터에서 개발한 CACSD 패키지로, 일반적인 CACSD의 특징들과 마찬가지로 행렬과 미적분을 위주로 하는 수학적인 문제들을 컴퓨터를 이용하여 손쉽게 풀 수 있게 함으로써 과학기술 분야의 문제의 해석과 설계를 돕는 기능을 하고 있다.

그림 8과 같은 실행 환경을 가지는 CEMTool은 일반적인 CACSD의 공통적인 추세인 사용자 인터페이스의 강화 및 객체 지향 및 개방형 구조에 대해 그래픽을 강화하여 시각적 측면에서 사용자의 작업 환경 구축을 용이하도록 구성되어 있으며 실시간 모의 실험을 위한 개별적인 패키지를 기존의 CEMTool 및 SIMTool의 구조에서 활용하고 외부 윈도우 어플리케이션들과의 연계를 통해 객체 지향적 구조로 발전하고 있다.

CEMTool은 CACSD로서의 기본적인 특징 이외에도 다른 패키지들과 구별되는 다음과 같은 특징들을 가지고 있다.

첫 번째로, 2 대의 PC 혹은 PC와 실제 플랜트를 이용한 실시간 실험을 DSP, AD/DA 보드 등의 네트워크 기반 시스템을 이용하여 그림 9와 같이 구성할 수 있다. 새로운 CEMTool 및 SIMTool에서 특성할 수 있다. 새로운 CEMTool 및 SIMTool에서 특히 네트워크 기능의 강화, 사용자 인터페이스의 보강으로 SIMTool의 블록 다이어그램의 형태로 제

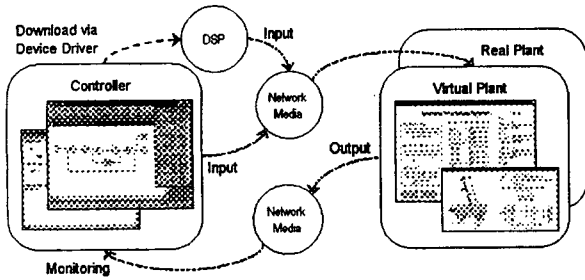


그림 9. CEMTool을 이용한 실시간 모의 실험.

표 3. 모의 실험의 측면에서의 비교

	Real-time	Software only	Cost	Time scaling	MATLAB	CEMTool
Off-line Simulation	Yes	Yes	Small	No	○	○
Software In the Loop (SIL)	Yes	Yes	Small	Yes	×	○
Hardware In the Loop (HIL)	Yes	No	Large	No	○	○

한되었던 실시간 모의 실험을 보다 복잡한 수식적 알고리즘 구현이 가능한 CEMTool을 이용하는 구조로 확장하였다. 이것은 대중적이고 실시간 모의 실험에 많이 이용되는 MATLAB과 비교할 때 표 3에서와 같이 Software-In-the-Loop (SIL) 에서 우수한 특성을 보이는 것을 확인할 수 있다. 따라서 CEMTool을 이용하면 새로 개발한 제어기나 플랜트를 고가의 하드웨어 장비 없이 실험실 내부에서, 여러 대상 모델을 검사할 수 있다.

두 번째로, CEMTool은 다른 CACSD 패키지들이 지원하지 않는 다양한 종류의 가상 플랜트를 제공하고 목적에 따라 교체 가능한 관련 예제들을 자체적으로 제공함으로써 제어 및 신호 분야에 대한 교육적 측면에서 강력한 기능을 보유하고 있다. 현재까지의 보고된 결과 및 사용자층으로부터 실제 강의나 개인적인 기본 개념의 습득에 활용되기에 적합함이 확인되었다.

### 5. 범용 Real-time Processing Unit

최근의 CACSD package들은 단순히 제어기 설계에 그치는 것이 아니라 범용 Real-time Processing Unit을 이용하여 직접 플랜트를 제어할 수 있는 환경을 구성하는 것을 목표로 하고 있다. 대표적인 예로 dSPACE는 MATLAB의 Real-time workshop 기능을 이용하여 이와 관련된 환경들을 제공하고 있다. Cemtool에서도 이러한 환경을 제공하고 있는데, 여기에서 필수적인 기능이 설계한 제어기를 C

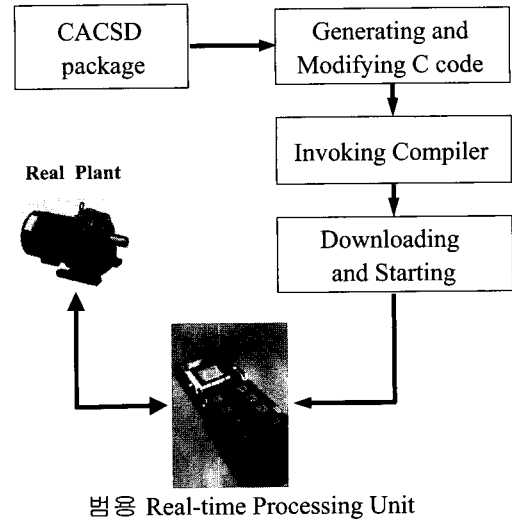


그림 10. 자동 C Code 생성과 실시간 제어.

표 4. PC Bus용 Real-time Processing Unit

	대정전자	dSPACE	SNU/CISL
DSP Board	○	○	○
Analog I/O Board	○	○	○
Digital I/O Board	○	○	○
High speed RAM Board	○	○	×
Motion Control Unit	×	○	×
Network	×	×	○

표 5. VME Bus용 Real-time Processing Unit

	VME Tech	명신 정보	우리 기술	SNU/CISL
DSP Board	○	○	○	○
Analog I/O Board	○	○	×	×
Digital I/O Board	○	○	○	×
High speed RAM Board	×	×	×	○
Motion Control Unit	×	×	○	×
Network	○	×	○	○

Code로 변환해주는 자동 C Code 생성과 생성된 C Code를 Real-time Processing Unit에 이식하여 Interfacing 하는 기능이다. 그림 10은 CACSD package를 이용하여 C 코드를 생성하고 Real-time Processing Unit을 이용하여 플랜트를 제어하는 일련의 과정을 보여준다. CACSD package에서 제공하는 이러한 기능들이 더욱 효율적으로 이용되기 위해서는 범용 Real-time Processing Unit에 모두 적용될 수 있는 일반적인 환경을 구축해야 한다는 것이다. 실제로 국내외에서 개발된 많은 Processing Unit들의 구조와 기능들이 약간씩 다르고, Interfacing library가 모두 다르기 때문에 일반적인 방법으로

이를 통합하여 사용자가 사용하기 쉬운 환경을 구성하는 것은 그리 간단한 문제는 아니다. 표 4와 표 5는 국내외에서 개발된 PC 및 VME Bus용 Real-time Processing Unit들의 구조와 지원기능을 비교하여 나타낸 것이다. 현재 국내의 CACSD package인 Cemtool에서는 몇 개 회사의 Real-time Processing Unit에 대하여 자동으로 C 코드를 생성, 이식하여 실시간 제어를 수행하는 통합기능을 제공하고 있으며, 최근에는 범용 Real-time Processing Unit에 손쉽게 적용할 수 있도록 C 코드 생성 및 Interfacing에 대한 일반화 작업을 진행 중이다. Cemtool이 이러한 일반화과정을 마치게 되면 Processing Unit을 보유한 회사들이 각자에 맞게 제어시스템을 개발하는 것이 가능하게 되어 HIL (Hardware-In-the-loop) 분야에서 외국의 어떤 CACSD package 못지 않은 역할을 할 수 있을 것으로 기대된다.

### 6. Software-In-the-Loop Simulation(SIL)기능

CEMTool에서는 두 대의 PC를 사용한 Software-In-the-Loop Simulation을 위해 네트워크 방식을 몇 가지 제공하고 있다. LAN 통신, AD/DA 보드를 이용한 통신, 직렬 통신, 병렬 통신을 지원한다. 사용자와의 편리한 인터페이스를 유지하기 위해 SIMTool에서는 블록 형태로 이들을 제공하고 있다. 각각의 블록에는 베이스 I/O 주소, 연결될 컴퓨터 이름, 입/출력의 제한, 채널 번호와 하드웨어 관련 데이터를 쉽게 입력할 수 있는 대화상자를 가지고 있어 프로그램을 용이하게 할 수 있다. 각각의 통신 방식은 다음과 같은 특징을 가지고 있다.

#### ■ LAN 통신

네트워크 상의 시간 지연을 최소화시킬 수 있으며, 빠른 플랜트와 다채널에서도 좋은 성능을 보인다. 대부분 학교에서는 LAN설치가 되었으므로 교육적으로 활용하기가 매우 용이하다.

#### ■ AD/DA

제어기와 플랜트 사이의 실제 인터페이스를 반영하고 있다. AD/DA와 같은 하드웨어 장치가 필요하지만, 빠른 플랜트에서 매우 좋은 성능을 보일 뿐 아니라, 실제 플랜트를 반영하고 있어 현실에 더욱 가까운 시뮬레이션을 할 수 있다.

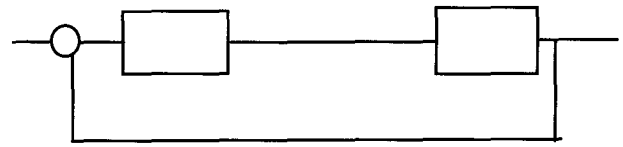
#### ■ Parallel / Serial

저렴한 가격으로 인터페이스가 가능하지만 네트워크상의 시간지연이 다른 방법보다 큰 단점이 있다.

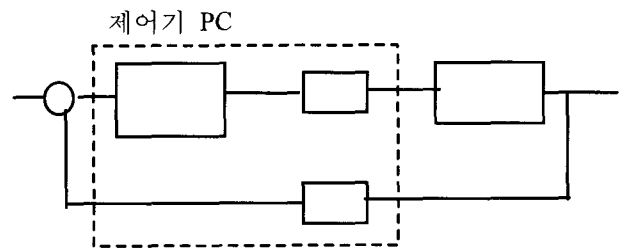
### 6.1 Delay에 관한 성능해석

네트워크의 특성에 따라 샘플링 시간 및 통신 지연, 데이터 동기화와 같은 문제가 발생한다. 특히 통신 지연이 시스템에 미치는 영향은 주목할 만하다. 아래 그림은 시간 지연 요소가 제어기와 플랜트에 들어가 있는 형태를 보이고 있다.

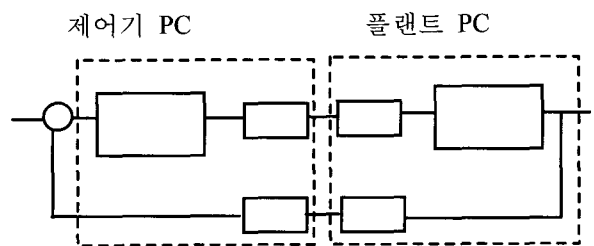
다음 그림의 (b)와 (c)에 대한 결과를 도시하면 시간지연에 시스템의 반응을 볼 수 있는데 특히



(a) 지연요소가 없는 이성적인 제어기와 플랜트



(b) 지연요소가 제어기에만 있는 구조



(c) 제어기와 플랜트 모두 지연요소가 있는 구조

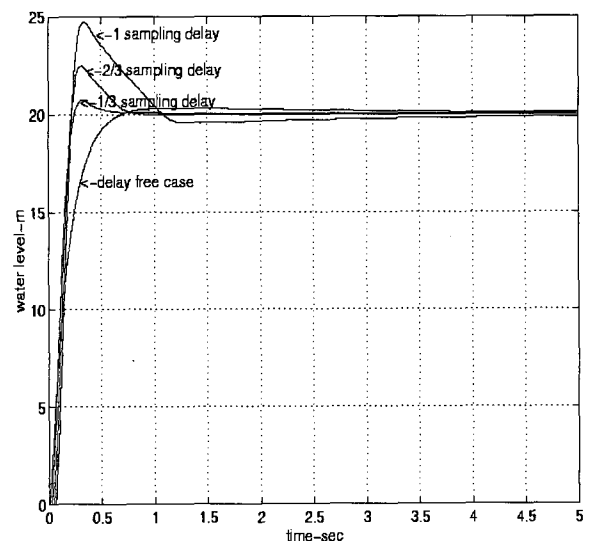


그림 11. 실제 디지털 시스템 시간 지연 효과.

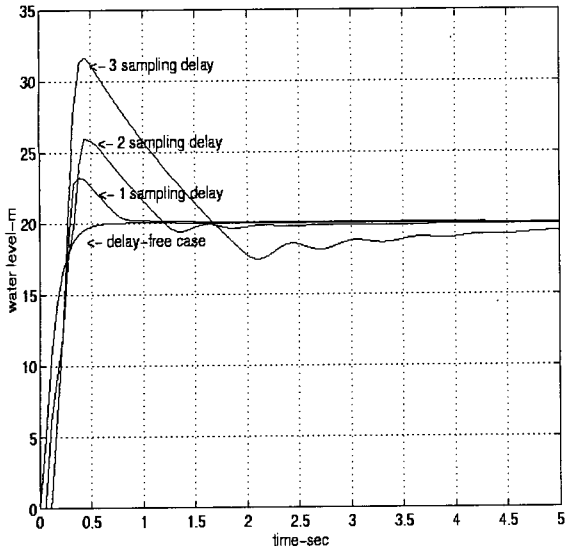


그림 12. SIL 시스템의 시간지연 효과.

SIL인 경우 네트워크 상의 시간지연이 시스템에 미치는 영향을 확연히 볼 수 있다. 그림 11과 12는 실제 디지털 시스템의 시간 지연과 네트워크 상의 시간 지연 효과를 보이고 있다. SIL인 경우 디지털 시스템과 비교해서 시간 지연이 시스템에 더욱 치명적인 영향을 보이고 있음을 알 수 있다. 과도 응답상태와 정상 상태 오차가 SIL인 경우 시간 지연에 매우 민감함을 알 수 있다.

### 6.2 Time-Sealing 방법

원래의 시스템에 대해  $a$ 배의 시간 스케일을 해서 SIL을 실행하고 다시 시간을  $1/a$ 배 하여 원 시스템으로 복귀하는 방법이다. 이 방법은 매우 빠른 동적 특성을 가진 시스템을 시뮬레이션 할 때 컴퓨터의 계산 속도를 어느 정도 극복할 수 있고, 시간 지연 효과도 억제할 수 있다. 따라서 SIL결과가 실제와 거의 비슷한 결과를 보인다. 그림 13에 있는 수식은 time-scaling을 보여주고 있다.

$\frac{dx(t)}{dt} = f(x(t), u(t))$ $y(t) = g(x(t))$	제어기	플랜트
	LQ, PID, $H_\infty$ 등을 소프트웨어로 구현	플랜트의 동적 특성 소프트웨어로 구현

$$\frac{dx(\frac{1}{a}t)}{dt} = \frac{1}{a} f(x(\frac{1}{a}t), u(\frac{1}{a}t))$$

$$y(\frac{1}{a}t) = g(x(\frac{1}{a}t))$$

그림 13.  $a$ 배 time-scaling 되는 과정.

### 7. Demo

SIL인 경우 demo를 보인다. 그림 14와 같이 제어기와 플랜트의 기능을 각각 노트북에 소프트웨어로 구현하고 네트워크를 통해 통신을 한다.

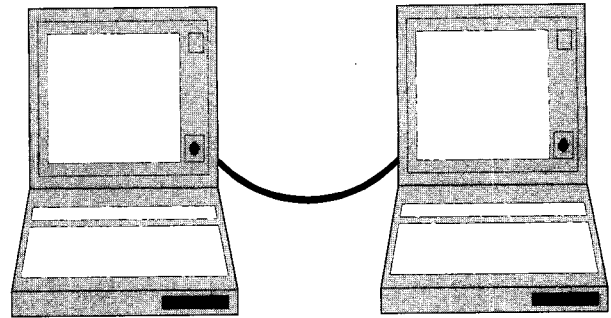


그림 14. SIL을 위한 두 노트북.

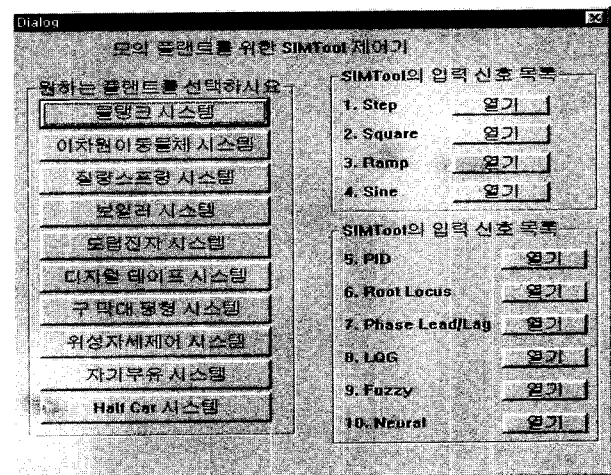


그림 15. 제어기와 플랜트를 선택할 수 있는 대화상자.

정해진 한가지 플랜트에 대해서 한쪽의 노트북의 제어기를 새로 설계함으로써 다양한 제어방법으로 플랜트를 제어할 수 있다. 이것은 제어기와 플랜트가 분리되어 있는 실제 상황을 반영할 뿐 아니라, 간단한 프로그래밍으로 새로운 제어기 설계가 용이하므로 교육적으로 매우 유용하게 사용될 수 있다. 현재 CEMTool에서는 자기부상, half car, 보일러 등 10개의 플랜트 애니메이션이 제공되고 있다. CEMTool에서는 원하는 플랜트와 제어기를 쉽게 선택할 수 있도록 아래와 같은 대화상자를 제공한다. 그림 15에서 10가지 플랜트에 대한 제어기와 입력신호 목록을 볼 수 있다. 입력 신호는 단위, 펄스, 램프, 사인 입력을 할 수 있고, 제어기는 PID, Root Locus, Phase lead/lag, LQ 등에서 선택할 수 있다.

10가지 애니메이션 플랜트 중에서 물탱크 시스템을 간단한 예제로 든다. 이 플랜트는 물탱크 수도의 유량을 조절하여 오른쪽 물탱크의 물높이를 일정하게 유지하는 것이 제어 목적이다. 그림 16은 기준 입력이 20으로 정해진 물탱크의 시뮬레이션된 결과이다. 제어기는 고전 제어 기법인 PID로 SIMTool을 이용해서 구성했다. PID 계수의 값으로 비례계수는 100, 미분 계수는 1, 적분 계수는 2로 설정했고, 통신 방식은 LAN을 선택했다.

위 물탱크 시스템의 사양과 통신방식을 알기 위해

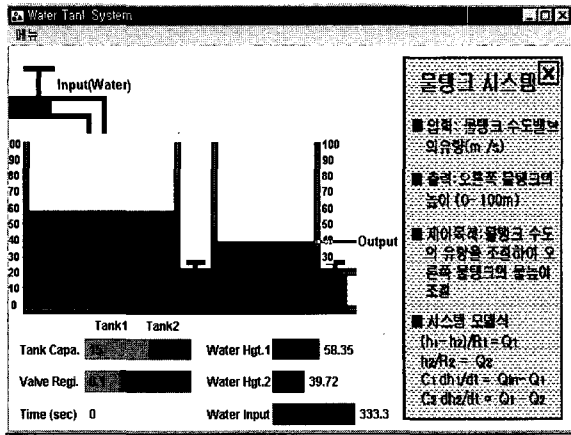


그림 16. 물탱크 애니메이션.

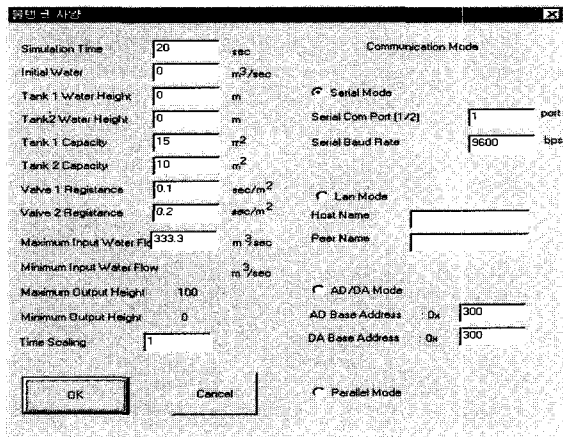


그림 17. 물탱크의 여러 가지 사양과 통신 방식을 선택할 수 있는 대화상자.

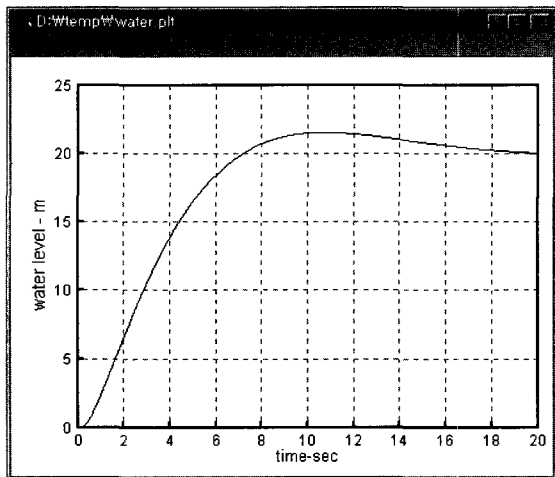


그림 18. 두번째 물탱크의 높이를 도시한 그림.

서 윈도우 화면에 더블클릭을 하면 다음과 같은 대화상자를 볼 수 있다.

대화상자에는 물탱크의 초기 수위와 용량, 물탱크 수도의 입력 한계를 명시할 수 있고, 네트워크 방식과 관련된 입력인자를 넣는다. 그림 18 은 두번째 물탱크의 수위가 20으로 수렴하는 것을 보이고 있다.

## 8. 결론

본 논고에서는 실시간 제어 시스템 설계에 대한 전반적인 개요 설명, 많은 모의실험용 툴의 소개 및 비교 검토, 제어 시스템의 구성 요소 설명, SIL 모의실험, 제어실험의 예등을 하나씩 알아 보았다.

실시간 제어 모의실험에서 CEMTool은 다른 모의실험 툴에서는 제공하지 않은 SIL 모의실험 등을 제공함으로써 제어기 설계를 체계적으로 할 수 있는 기본적인 툴을 제공하고 있다. 이와 더불어 하드웨어 장치와 인터페이스를 통하여 HIL 모의실험을 할 수 있으며 제어기 모의실험이 바로 실제 제어기의 구현이라는 목표 달성에 근접하고 있다.

CEMTool의 보급 및 사용은 국내의 제어 기술의 발전과 더불어 국내 산업 전반의 발전에 중대한 역할을 하리라 기대한다.

## 저자소개

### 권 옥 현(權 旭 鉉)

1943년 1월 19일생

1966년 서울대학교 전기공학과 (공학사)

1971년 서울대학교 전기공학과 (공학석사)

1975년 Brown University (공학박사)

1980년 - 1981년 Visiting Assistant Professor in Stanford Univ.

1991년 - 현재 제어계측신기술연구센터 소장

1977년 - 현재 서울대학교 전기공학부 교수

### <관심분야>

- 다변수 강인 및 예측 제어
- 분산 제어 시스템
- 네트워크 분석
- 공장자동화를 위한 컴퓨터 응용
- CACSD

### <연락처>

Email: whkwon@cisl.snu.ac.kr

### 김 기 백(金 起 伯)

1969년 11월 9일생

1993년 서울대학교 제어계측공학과 (공학사)

1995년 서울대학교 제어계측공학과 (공학석사)

1995년 - 현재 서울대학교 전기공학부 박사과정

### <관심분야>

- 예측제어, 강인제어
- 유도제어, 시간지연 시스템
- CACSD

### <연락처>

Email: kkb@cisl.snu.ac.kr



**최 성 규(崔 成 圭)**

1971년 1월 30일생

1992년 인하대학교 전기공학과 (공학사)

1994년 인하대학교 전기공학과 (공학석사)

1994년 - 1997년 서울대학교

제어계측신기술연구센터 전임연구원

1997년 - 현재 서울대학교 전기공학부 박사과정

<관심분야>

- SIL, DSIL simulation

- RTP 설계, CNC control

<연락처>

Email: csg@asri.snu.ac.kr

**김 형 진(金 炯 進)**

1974년 12월 27일생

1997년 서울대학교 전기공학부 (공학사)

1997년 - 현재 서울대학교 전기공학부 석사과정

<관심분야>

- 예측제어, 항법유도제어

- CACSD, 실시간 DB

<연락처>

Email: hyz kim@cisl.snu.ac.kr

**한 수 희(韓 水 熙)**

1974년 8월 26일생

1998년 서울대학교 전기공학부 (공학사)

1998년 - 현재 서울대학교 전기공학부 석사과정

<관심분야>

- 신호처리, 필터이론, 시간지연시스템

<연락처>

Email: hsh@cisl.snu.ac.kr