

■ 論 文 ■

# 교통신호제어시스템 성능평가 시뮬레이션 시스템 개발

Development of the Simulation System for Testing and Evaluating  
of Traffic Signal Control Systems

**정준하**

(도로교통안전관리공단 선임연구원)

**하동익**

((주)제온시스템 대표이사)

**이돈주**

((주)제온시스템 제어전략팀 팀장)

## 목 차

- I. 서론
- II. 기존 개발사례 고찰
  - 1. 미국사례
  - 2. 유럽사례
  - 3. HILSS의 방식별 분류
- III. 시뮬레이터시스템의 개발 및 활용방안
  - 1. 개발시스템의 구성 및 기능
- 2. 교통신호제어시스템 평가원리
- IV. 개발시스템 신뢰도 및 적용사례
  - 1. 개발시스템의 신뢰도 평가
  - 2. 교통신호제어시스템에의 적용사례
- V. 결론 및 향후 연구과제
- 참고문헌
- 부록

Key Words : HILSS, 검사장치, 평가장치, 시뮬레이터, 검지기에뮬레이터

## 요 약

본 연구에서는 ITS의 각종 시스템에 대한 효율성 평가를 목표로 개발된 시뮬레이션 시스템을 소개하고 이를 교통신호제어시스템의 성능평가 도구로 적용한 결과를 제시한다.

개발된 시뮬레이션 시스템은 종래의 소프트웨어 기반 시뮬레이션 프로그램에 인터페이스 장치를 통해 실제 교통제어시스템의 하드웨어를 직접 연결하여 실시간으로 성능검사와 교통관리시스템의 기능성을 평가하는 것으로 크게 시뮬레이터 운영시스템과 인터페이스 장치의 두 부분으로 구성된다.

교통신호제어시스템을 대상으로 개발 시뮬레이션 시스템 성능을 시험한바 정확도 및 안정성 면에서 우수하고, 별도의 추가장비 없이도 모든 신호제어기에 호환성을 확보하는 것으로 평가되었다. 차량 검지신호에 대한 출력 정확도 평가 및 신호등의 동기신호 검출 정확도 평가에서는 80msec 이내의 오차를 기록하여 성능검사 장치로서의 신뢰도를 확인할 수 있었다.

본 시뮬레이션 시스템 개발로 교통관리시스템의 운영효율을 과학적이고 객관적인 방법에 의해 평가함으로써 문제점의 진단, 개선내용 및 개선범위 등에 대한 조기파악이 가능하다.

향후 개발된 시뮬레이션 시스템에 교통류 재현 모형을 탑재하면 ITS 시스템에 대한 효율성 평가도구로 확장 사용이 가능하여 시스템 개발, 도입 시스템 선정, 설치 후 시스템의 조기 안정화 및 유지관리 등 각 단계에서 검증수단으로 유용하게 활용될 것이다.

## 1. 서론

체계구성요소의 변화나 새로운 제어기술 혹은 기법의 도입시 그 효과를 사전에 분석하기 위하여 컴퓨터를 이용한 시뮬레이션만큼 비용대비 효과적이고 유용한 도구도 없을 것이다. 교통분야에서도 보다 과학적인 분석과 예측을 위하여 계획에서부터 설계 및 운영의 단계에 이르기까지 다양하고 수많은 시뮬레이션 프로그램을 이용하고 있다. 그러나, 전자통신을 비롯한 다양한 기술을 집적한 ITS의 시스템에 대해서는 소프트웨어에 기반한 시뮬레이션만으로는 한계가 있다.

교통시스템은 개발자마다 구현방식이 상이하고, 업체의 경우 자사의 알고리즘이나 소스코드에 대한 공개를 기피하는 성향이 있으며, 공개를 한다고 하더라도 복잡하고 방대한 제어알고리즘을 시뮬레이션 내부에 탑재하 기란 결코 쉬운 작업이 아니기 때문이다. 그리고, 시스템마다 동일한 제어알고리즘이 탑재되어 있을지라도 하드웨어적인 구성상태, 정보처리방법 및 통신체계에 따라 실제 교통관리의 효과는 큰 차이를 나타낼 수 있다.

ITS 추진과정에서 나타난 이러한 평가방법상의 문제점을 해결하기 위한 방안으로 미국, 유럽 등에서는 1990년대 중반이후 종래의 소프트웨어 기반 시뮬레이션 프로그램에 인터페이스 장치를 통해 실제 시스템의 하드웨어를 직접 연결하여 실시간으로 교통관리의 효율성을 평가할 수 있는 기법을 연구하고 있다.

이 분야의 연구는 전세계적으로 개발의 초기단계라고 할 수 있다. 각국의 상황에 따라 연구의 착수동기는

상이한 면이 없지 않으나 궁극적으로는 시스템의 효율성 평가를 목표로 하고 있다. 실제 구현방식에 있어서는 시뮬레이션 프로그램과 실제 교통시스템을 연결하는 장치와 통신방식에 있어 가장 두드러진 차이를 보이고 있다.

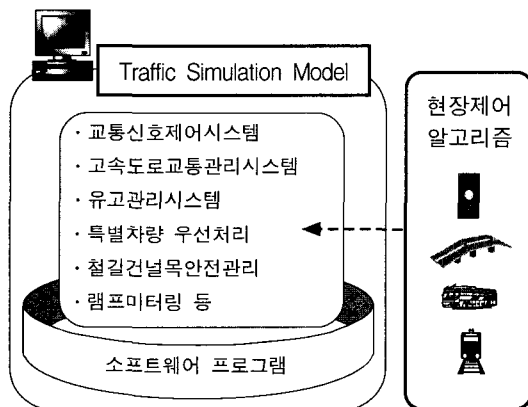
미국의 경우 기존 우주항공 및 방위산업분야에서 활용되고 있던 HILSS(Hardware-In-the-Loop Simulation System)의 개념 즉, 실세계에 존재하거나 구상중인 목표(Target) 시스템을 모형에 연결하여 실시간으로 시뮬레이션 하는 기술을 신호제어기 성능검사 목적으로 응용하였으며, 점차 ITS 부문으로 확대하고 있다(Emgelbrecht, 1999).

유럽은 실시간 교통신호제어전략의 평가를 위하여 기존 소프트웨어 기반 시뮬레이션 모형을 발전시켜 나가는 형태를 취하고 있다(ITS, 2000).

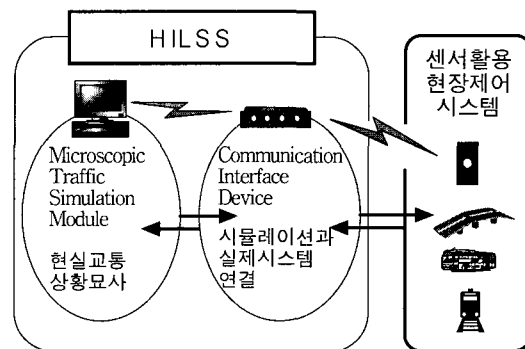
국내의 경우 본 연구이외에는 관련 기술개발 사례가 전무한 실정이다.

본 연구에서는 첫째, HILSS<sup>1)</sup>와 관련한 기존 개발 사례를 고찰하고, 둘째, 본 연구에서 개발한 시뮬레이션 시스템의 원리 및 체계를 설명하며, 셋째, 개발시스템의 평가를 통한 신뢰도 및 교통신호제어시스템에 대한 적용사례, 마지막으로 결론 및 향후연구과제를 제시하고자 한다.

본 연구는 1999년부터 시작되었으며, 시스템 요구 분석 및 상세설계(도로교통안전관리공단, 1999)를 거쳐, 2000년 시제품을 제작하고 안정화 및 사례적용(도로교통안전관리공단, 2000)을 통하여 서울시에 설치되는 교통신호제어기에 대한 성능평가에 활용하고 있다.



〈그림 1〉 소프트웨어 기반 시뮬레이션 평가 개념도



〈그림 2〉 HILSS를 이용한 실시간 평가 개념도

1) 시뮬레이션에 실제 교통시스템의 하드웨어를 직접 연결하여 실시간으로 평가하는 개념은 미국의 'HILSS'가 가장 선행이며 이외에는 별도의 명칭이 없으므로 해당기술을 HILSS로 통일하여 부르기로 한다.

## II. 기존 개발사례 고찰

### 1. 미국사례

HILSS는 실제 하드웨어 부분을 대체하여 시물레이션 하는 것으로 수년동안 우주항공 및 방위산업 분야에 이용되어 왔다.

1995년 처음으로 텍사스주립대 교통연구소(TTI)의 Urbanik과 Venglar가 HILSS를 근간으로 하여 실시간 교통류 시물레이션을 포함하는 SMART diamond system을 구상하였다. 그리고, 1998년 Louisiana 대학의 Bullock과 Catarella가 FHWA의 지원으로 HILSS를 기본으로 교통신호시스템을 평가하는 실시간 시물레이션 환경을 구축하였다(Emgelbrecht et al, 1999).

HILSS는 CID(Controller Interface Device)라 불리는 통신장치가 주요부분을 구성하며, 이는 컴퓨터와 제어기 사이의 양방향 통신기능을 담당한다.

지금까지 개발된 장치도 CID의 특성에 따라 LSU-CID, Mini-CID 및 REL-CID 등으로 명명되고 있다. 지금까지 많은 CID가 개발되었으나 컴퓨터와의 통신방식에 따라 크게 직렬포트를 사용하는 방식과 I/O Interface Board를 사용하는 방식으로 구분할 수 있다.

각종 제어기의 종류에 따라 기계적인 특성 및 통신 방식에 있어 상이한 면이 있으므로 CID가 조금씩 다르게 제작되었으나, 기본적인 구조는 다음과 같다.

Louisiana 대학의 LSU-CID는 RS-232C를 이용하여 자료를 교환하는 시스템으로써 노트북 컴퓨터에 연결되도록 구성되었다. 따라서, 인터페이스 보드에 비해 자료의 전송속도는 떨어지나 휴대가 간편하여 현장용용성이 우수하다는 장점이 있다. 반면, TTI의 REL-CID는 I/O Interface Board를 이용하며, 실험실에서 TCP/IP 통신망을 이용하여 자료를 교환하는 구성체계이다. 이는 자료의 전송속도가 빠르며, 네트워크단위 신호체계의 평가가 가능하나, 대규모의 컴퓨터 시스템과 통신망을 필요로 한다.

두 시스템이 통신방식에 있어서는 상이한 면이 있으나, 기본적인 기능은 교통량의 변화에 대응하는 제어기 및 신호제어 알고리즘의 효율성을 평가하는 것으로 동일하다고 할 수 있다.

교통량의 발생을 위한 마이크로 시물레이션 프로그램

〈표 1〉 REL-CID와 LSU-CID의 비교

구 분	REL-CID	LSU-CID
개발기관	Texas A&M Univ.	Louisiana State Univ.
개발연도	1995	1998
자료교환	I/O Interface Board	Serial Port
자료 교환속도	상대적으로 빠름	상대적으로 느림
적용장소	실험실	현장
요구 컴퓨터	네트워크급	PC급
시물레이션 모형	CORSIM, TexSIM, VISSIM, Integration	
장점	통신속도가 빠르고, 호환성이 우수	연결이 쉽고, 휴대가 용이

은 기본적으로 CORSIM을 탑재하고 있다. CORSIM의 Run-Time Extension 기능이 검지와와 현시 정보간의 통신을 담당하게 된다. 그러나, 필요한 경우 TexSIM이나 VISSIM, Integration 등도 이용할 수 있도록 하고 있다.

지금까지 CID가 개발된 신호제어기에는 NEMA TS1 및 TS2, Eagle사의 2070 및 2070N, 각종 Ramp Metering 및 입체교차로에 이용되는 제어기 등이 있다. 이들 CID는 현재 제어기의 현장 설치전 내부시험, 실시간 제어알고리즘 개발 및 평가, 학생과 실무자를 위한 교육용으로 이용되고 있으며, 대중교통우선처리 알고리즘이나 Ramp Metering 제어기의 평가, 건널목 시물레이션 등 실시간 감응제어 알고리즘 개발분야로 그 적용영역을 확대하고 있다.

HILSS의 개발로 시물레이션에 현실감을 부여하였으며, 에뮬레이션이 불가능한 제어기의 기능에 대한 에뮬레이션이 가능하게 되었다. 따라서, 현장에 제어기를 설치하기 전에 광범위한 Bench-Testing을 수행할 수 있게 되었다. 그러나, 미국 개발시스템은 제어기의 종류에 따라 CID를 달리 개발함으로써 통일성이 결여되고 이용에 있어 범용성이 부족하다는 한계가 있다.

### 2. 유럽사례

유럽에서는 광역 실시간 교통신호제어전략을 개발하기 위하여 기존의 시물레이션 모형에 SCOOT, SCATS, SPOT 등의 제어시스템을 연결하기 위한 목적으로 시

〈표 2〉 유럽의 HILSS 구현가능 시물레이션 모형

시물레이션 모형	적용 UTCS	개발국
NEMIS	SPOT, SCOOT	이탈리아
AIMSUN2	SCOOT	스페인
HUTSIM	SPOT	핀란드
SIGSIM	SCOOT	영국
VISSIM	SCATS	독일
DRACULA	BALANCE	영국

작하였다. 미국이 HILSS라는 명칭을 부여하고 있는 반면 유럽에서는 기존 소프트웨어 기반 시물레이션 모형의 발전된 형태로 인식하여 별도의 명칭을 부여하지 않고 있다.

유럽의 개발시스템은 시물레이션 모형과 실제 시스템의 연결시 특별히 고안한 하드웨어 장비가 없고 대신 FEP(Front-End-Processing) 컴퓨터를 사용하고 있다는 것이 가장 큰 특징이라고 할 수 있다.

현재까지 HILSS의 기능이 구현 가능한 시물레이션 모형은 NEMIS를 비롯 AIMSUN2, DRACULA 등 다수가 있는 것으로 알려지고 있다(ITS, 2000).

그중 적용사례가 비교적 구체적으로 발표된 NEMIS를 보면 DRIVEII 프로젝트의 도시교통제어전략에 대한 실내 및 현장 실험틀의 개발을 목적으로 시작되었다(Biora, 1995; ITS, 2000). 도시교통제어시스템의 현장 설치전에 전략을 점검하고 파라메타를 조정할 뿐만 아니라 효율성을 시험하기 위하여 외부 현장시스템과의 인터페이스를 지원하고 있다.

현재 유럽의 다른 시물레이션 모형과 마찬가지로 TCP/IP에 기반한 표준 프로토콜을 이용하여 LAN/WAN의 접속이 가능하며, NEMIS와 외부 시스템이 연결된 네트워크간의 정보교환을 통하여 대규모 교통망의 신호체계, 대중교통 우선처리, 가변정보제공, 동적 경로 안내 등 다양한 ITS의 기능을 평가하고 있다.

### 3. HILSS의 방식별 분류

본 연구를 포함하여 기존 개발시스템은 시물레이션 모형을 탑재한 운영프로그램, 인터페이스 장치 그리고, 실제 시스템과의 자료교환을 담당하는 통신모듈 등의 3가지가 주요 구성요소이다. 각각 구현방식에 따라 차이가 있을 수 있으나, 본 연구에서는 첫째, Target 시스템과의 통신방식, 둘째, Target 시스템의

평가 범위에 따라 기존시스템을 기능별로 구분하고자 한다.

#### 1) Target 시스템과의 통신방식에 의한 구분

시물레이션 프로그램과 실제 Target 시스템과의 통신에는 다음과 같이 High Level Interface 방식과 Low Level Interface 방식으로 구분할 수 있다.

High Level Interface 방식은 하드웨어 장비가 필요 없이 소프트웨어에 의한 인터페이스이며, 주로 인터넷이나 LAN/WAN을 이용한 범용의 TCP/IP 디지털 인터페이스 통신방식이다. 또한, 개별 시스템에 대응하기 위하여 자체 개발한 인터페이스 프로그램을 이용한 RS-232C 디지털 인터페이스 통신도 활용하고 있다.

Low Level Interface 방식은 특별히 고안된 하드웨어 장치를 이용한 인터페이스로 아날로그 인터페이스 통신방식이다. 이는 Target 시스템의 하드웨어 및 소프트웨어에 부가적인 개조 없이 최하위 입출력 단자를 통하여 직접 접속하는 것으로 차량의 검지신호 및 실제 현장 기기의 출력신호를 아날로그 값으로 송수신 한다.

High Level Interface 방식은 Target 시스템의 소프트웨어에 통신방식을 일치시켜야 하므로 호환성이 낮으며, 대상 시스템의 하드웨어에 대한 검사기능이 취약하다는 단점이 있으나, 광역망 평가시 장비의 추가 없이도 운영이 가능하다.

반면 Low Level Interface 방식은 대상 시스템의 최하위 입출력 단자에 직접 접속함으로써 범용의 호환성이 우수하고 하드웨어에 대한 검사가 가능하다는 장점이 있으나, 별도의 인터페이스 장비를 필요로 하

〈표 3〉 통신 인터페이스 방식에 의한 HILSS의 구분

	High Level Interface 방식	Low Level Interface 방식
특징	<ul style="list-style-type: none"> <li>· 별도의 장비 불필요</li> <li>· 대상시스템 H/W 검사 기능 취약</li> <li>· 광역 network 평가시 장비추가 불필요</li> <li>· 범용의 호환성 낮음</li> <li>· 네트워크 컴퓨터를 이용하여 Off-Road 적용성 우수</li> </ul>	<ul style="list-style-type: none"> <li>· 별도의 장비필요</li> <li>· 대상시스템의 H/W검사 기능 구비</li> <li>· 광역 network 평가시 장비추가 필요</li> <li>· 범용의 호환성 우수</li> <li>· PC급 컴퓨터를 이용하여 On&amp;Off-Road 모두 적용</li> </ul>
개발 사례	<ul style="list-style-type: none"> <li>· 유럽 대부분의 시스템</li> <li>· 미국은 두 방식의 중간형태</li> </ul>	<ul style="list-style-type: none"> <li>· 본 연구의 개발시스템</li> </ul>

고 광역망의 평가시 추가적인 장비가 요구된다.

본 연구에서는 하드웨어의 검사 및 호환성 유지를 위하여 Low Level Interface 방식으로 개발하였다. 현재 유럽대부분의 시스템은 High Level Interface 방식이며, 미국은 두 방식의 중간형태를 취하고 있다.

2) 대상시스템의 평가범위에 의한 구분

시뮬레이터가 Target 시스템을 평가하는 범위에 따라서는 전략적 제어수준 평가와 전술적 제어수준 평가의 두 가지 방식으로 구분할 수 있다.

전략적 제어수준 평가 방식은 네트워크 단위의 전략적 제어효율성을 평가하는 것으로 탑재되는 교통류 시뮬레이션 모형은 Macroscopic 혹은 Mesoscopic 시뮬레이션만으로 적용 가능하다. 따라서, 통신 인터페이스 방식은 High Level Interface로 충분하다.

전술적 제어수준 평가 방식은 교차로 단위의 전술적 제어 효율성을 평가하는 것으로 반드시 Microscopic 교통류 시뮬레이션 모형의 탑재가 필요하다. 통신 인터페이스 방식은 Low Level Interface가 적합하며, High Level Interface로도 가능하나 이 경우 Low Level Interface와 같이 대상시스템에 별도의 통신 모듈이 추가되어야 한다. 전술적 제어수준 평가시 대상 교차로의 수가 증가할 경우 인터페이스 장비로 인한 물리적인 용량의 한계가 있을 수 있다.

<표 4> 평가범위에 의한 HILSS의 구분

	전략적 제어수준 평가 방식	전술적 제어수준 평가 방식
특징	<ul style="list-style-type: none"> <li>· 네트워크 단위 제어전략 평가</li> <li>· 평가를 위해 Macroscopic 혹은 Mesoscopic 모형탑재</li> <li>· High Level Interface 주적용</li> </ul>	<ul style="list-style-type: none"> <li>· 교차로 단위 제어전술 평가</li> <li>· 평가를 위해 Microscopic 모형탑재</li> <li>· Low Level Interface 적합</li> <li>· 광역망 평가시 물리적 용량의 한계</li> </ul>
개발 사례	<ul style="list-style-type: none"> <li>· 유럽 대부분의 시스템은 전략적 제어수준 평가에 적합</li> <li>· 미국의 개발방식은 전략적 제어수준의 평가에 적합하며 일부 전술적 제어수준의 평가도 가능</li> <li>· 본 연구의 개발시스템은 전술적 제어수준 평가를 위해 개발되었으며, 전략적 제어수준 평가기능은 향후 개발예정</li> </ul>	

III. 시뮬레이션시스템의 개발 및 활용방안

1. 개발시스템의 구성 및 기능

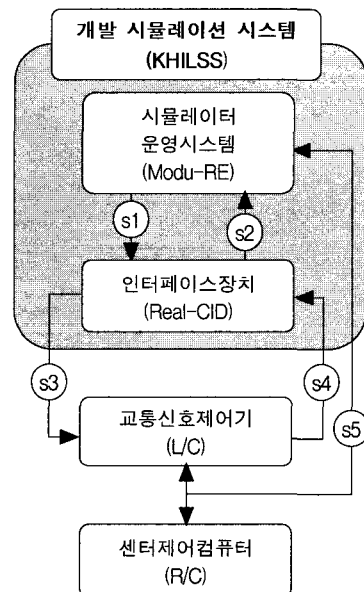
본 연구의 시뮬레이션 시스템은 크게 시뮬레이터 운영시스템(Modu-RE<sup>2)</sup>)과 인터페이스 장치(Real-CID<sup>3)</sup>)의 두 부분으로 구성된다.

Modu-RE는 교통류 시뮬레이션 모형을 탑재하고 인터페이스 장치를 통해 입출력된 교통신호제어시스템의 각종 신호를 기록, 해독, 분석 및 평가하는 소프트웨어이다. MS-Windows 환경에서 Borland C++ builder 5.0 기반으로 개발되었으며, 800MHz이상의 CPU와 RAM 128Mb이상의 PC급 컴퓨터에서 구동이 가능하다.

Modu-RE는 환경설정 모듈, 에뮬레이션 모듈, 검사 및 평가 모듈로 구성된다.

환경설정 모듈은 교통신호제어시스템과의 연결에 필요한 통신설정, 교통신호제어시스템의 출력정보 검출을 위한 설정, 차량검지신호 에뮬레이션을 위한 설정 등 각종 시뮬레이션 환경을 설정한다.

에뮬레이션 모듈은 실제 교통신호제어시스템과의



주) s1 : 차량검지출력신호(Digital), s2 : 신호등기신호(Digital)  
 s3 : 차량검지신호(Analog), s4 : 신호등기출력신호(Analog)  
 s5 : 제어기상황, 검지기정보(RS-232C)

<그림 3> 개발 시뮬레이션 시스템의 구성

2) Microscopic Out-door environment Digitizing Unit - Realtime Evaluator.

3) Real - Controller Interface Device.

접속을 통하여 에뮬레이션을 수행한다.

검사 및 평가 모듈은 환경설정 및 에뮬레이션을 수행 후 교통신호제어시스템의 성능검사 및 기능성을 평가한다.

Modu-RE는 인터페이스 장치를 통하여 운영자에 의해 설정된 교통상황(비포화, 포화, 과포화)에 적합하도록 가상의 검지기 신호를 생성하여 제어기에 공급하고<sup>4)</sup>, 이에 대응하여 신호제어기에서 각 신호등(보행등 적색·녹색, 차량등 적색·황색·좌회전·녹색)으로 출력한 신호를 검출한다.

이 밖에 제어기의 수동조작 스위치의 입력과 같은 외부 I/O 에뮬레이션 모듈에서는 제어기의 제어상태를 변경·운영할 수 있는 기능, 신호제어기가 센터시스템으로 전송하는 제어기의 동작상태, 운영현시정보, 검지기의 가공정보 등을 통신단자에서 Capture함으로써 Modu-RE에서 분석한 데이터와 신호제어기에서 생성하여 센터제어컴퓨터로 전송하는 데이터의 비교를 통하여 제어기의 기능의 적합성을 검사하고 고장여부 까지 판단할 수 있다.

아울러, 신호제어전략 개발지원 기능으로 교통상황(비포화, 포화, 과포화 등)에 따른 검지기 신호 자동 생성 기능을 이용하여 현장상황과 유사한 다양한 교통환경을 모사하여 신호제어 알고리즘 가동에 필요한 입력데이터를 제공하고, 알고리즘 수행결과를 신호등기출력으로 나타나기 때문에 이의 분석과 제어기능의 정상수행 여부 등을 통하여 알고리즘의 정확성 및 효율성 등을 평가할 수 있다.

개발 시뮬레이션 시스템의 중요한 하드웨어 장치인 Real-CID는 Modu-RE에서 생성한 디지털 형태의 차량 검지기 및 제어기의 외부 입력신호를 아날로그로 변조하여 제어기의 T/F(Terminal Facility)단자 입력단에 제공하고, 신호제어기로부터 출력되는 아날로그 형태의 신호등기 신호를 디지털로 변조하여 Modu-RE에 전송하는 동작을 반복한다.

**2. 교통신호제어시스템 평가원리**

시뮬레이션 시스템의 기능은 기본적으로 정형화된

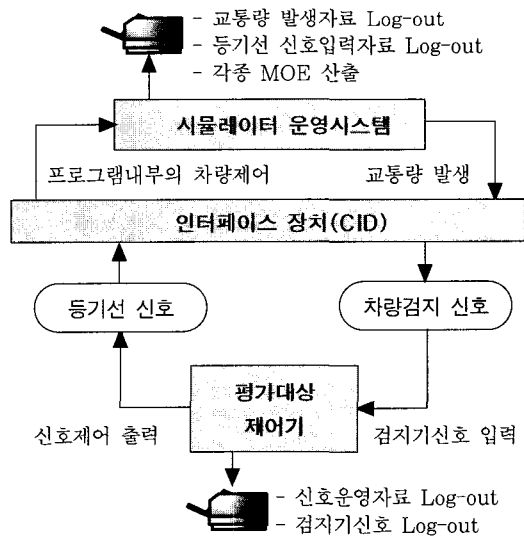
입의 교통상황을 생성하고 이에 따른 교통신호제어기의 반응 즉, 교통상황 지표 산출의 정확성과 신호제어의 정확성 및 효율성을 평가하는 것이다.

이를 보다 상세하게 설명하면 교통신호제어기와 연결된 시뮬레이터 인터페이스 장치에서는 차량검지기 신호 즉, 점유, 비점유 값에 따른 검지기 혹은 제어기 외부 입력신호 레벨을 인위적으로 발생하여 제어기의 T/F(Terminal Facility)단자 입력단에 제공한다. 그리고, 신호제어기로부터 출력되는 신호등기 신호를 A/D(Analog/Digital)로 변환하여 시뮬레이터 운영시스템에 공급하는 동작을 반복한다.

이 과정에서 시뮬레이터 운영 시스템으로부터 제공된 검지기신호는 제어기에서 점유/비점유로 변환되고 신호제어기의 경우 포화도, 점유율, 속도 등과 같은 교통상황 자료를 산출하게 된다.

본 연구의 개발 시뮬레이션 시스템은 이러한 검지기 신호 생성 과정에서 송출된 기준 값을 알고 있기 때문에 제어기에서 정확한 교통상황 변수가 산출되는지 여부를 평가할 수 있게 된다.

또한, 이러한 검지기 정보에 기초하여 신호제어기의 감응제어와 같은 신호제어기 자체 기능인 제어 알



〈그림 4〉 시뮬레이션 시스템의 작동 개념도

4) 기본적으로 차량검지기 신호를 에뮬레이트 하기 위해서는 먼저, 차량도착간격(inter-arrival time)에 따라 차두시간을 결정한 후 점유시간과 비점유시간을 결정한다. 따라서 결국 운영자가 선택하는 설정 값들은 차두시간과 점유시간, 비점유시간의 결정에 필요한 값들이다. 본 개발에서는 이러한 방법으로 ①고정된 점유/비점유 값에 의하여 반복 생성하는 방법, ②랜덤확률분포에 의한 방법, ③교통류 모형에 의한 방법 등 3가지 방법으로 제공하도록 하였으며, 각 방법에 필요한 기본적인 파라미터는 모사하고자 하는 교통상황(비포화, 포화, 과포화 등)에 따라 운영자에 의해 입력하게 된다.

고리즘(전술적 제어)에 따라 신호시간을 조정하게 되며, 결과는 센터 신호제어시스템에 전송되게 된다.

시뮬레이터 인터페이스 장치에 연결되는 컴퓨터에 성능검사를 위한 소프트웨어 및 효율성 평가를 위한 미시적 교통류 시뮬레이션 모형을 탑재하여 제어기의 기능작동여부 뿐만 아니라 교통신호제어시스템의 각종 제어모드별 운영상태까지도 평가할 수 있다.

#### N. 개발시스템 신뢰도 및 적용사례

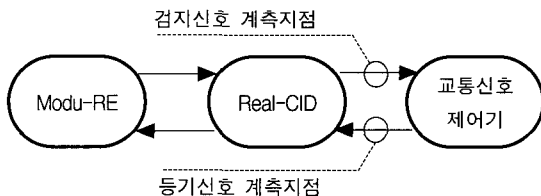
##### 1. 개발시스템의 신뢰도 평가

###### 1) 신뢰도 평가 목적 및 방법

신뢰도 평가는 개발시스템의 신호발생 및 검지능력에 대한 정확도 평가로서 입출력 신호 즉, Modu-RE의 검지신호출력 및 등기신호검출 정확도에 대한 평가를 목적으로 한다. 신호 입출력 정확성은 개발시스템의 가장 기본적이고 핵심적인 기능으로 시뮬레이션 시스템과 제어기간에는 차량검지신호와 등기신호만이 인터페이스 되고 있으므로 두 신호의 정확도 평가만으로 신뢰도의 평가는 충분하다고 할 수 있다.

개발 시뮬레이션 시스템은 교통신호제어기를 비롯한 루프검지기 기반의 ITS의 각 시스템에 대하여 기능의 정확성 평가뿐만 아니라 하드웨어적인 성능검사 기능을 구비하고 있으므로 시뮬레이션 시스템 자체에 대한 높은 신뢰도 수준의 정확도가 요구된다.

신뢰도 평가 방법은 <그림 5>와 같이 시뮬레이션 시스템 인터페이스 장치인 Real-CID와 교통신호제어기 사이의 두 접점에 정밀 신호계측장비를 각각 연결하고 여기서 계측된 신호변화 값과 Modu-RE의 검사 및 평가모듈에서 계측한 값을 비교분석 하였다.



<그림 5> 개발 시스템 신뢰도 실험방법

##### 2) 검지신호 출력 정확도 평가

<그림 6>은 시뮬레이터에서 출력한 차량 검지에 대한 디지털 신호 값과 실제로 출력되어 신호제어기에 입력되는 아날로그 신호의 실측값에 대한 추세를 나타낸 것이다. Modu-RE의 에뮬레이션 모듈에서 출력한 값을 기준으로 할 때 약간의 차이는 있으나, 동일한 추세로 검지신호가 발생되고 있음을 알 수 있다.

<그림 7>은 개별 신호의 회귀선 분포를 나타내는데 점유시간의 경우 Modu-RE의 기준값 보다 약간 긴 시간으로 출력되고 있으며, 비점유시간은 약간 짧은 시간으로 출력되고 있다.

그러나, 점유시간과 비점유시간의 합 즉, 차두시간의 추세 및 회귀선 분포는 점유시간과 비점유시간의 오차가 상쇄됨으로써 전체적인 오차가 감소하고 있다.

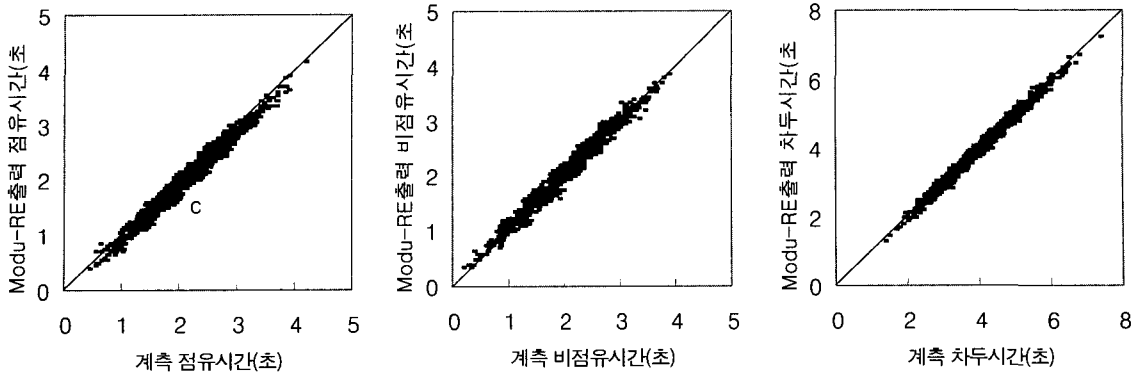
검지신호의 발생에 있어서 점유시간은 5.2275%, 비점유시간은 6.0444%, 그리고, 차두시간은 1.9453%의 오차율<sup>5)</sup>을 기록하고 있다.

비록 오차율의 수치상 다소 높다고 할 수는 있으나, 출력 값과 계측 값의 차이가 점유시간 0.076199초, 비점유시간 0.075261초, 차두시간 0.000938초로 낮게



<그림 6> 점유/비점유/차두시간 신호출력 추세

5) 오차율(%) =  $\frac{1}{n} \sum_{i=1}^n \frac{|f_i - f_d|}{f_i} \times 100$



〈그림 7〉 점유/비점유/차두시간 회귀선 분포

〈표 5〉 검지신호출력 정확도 평가결과

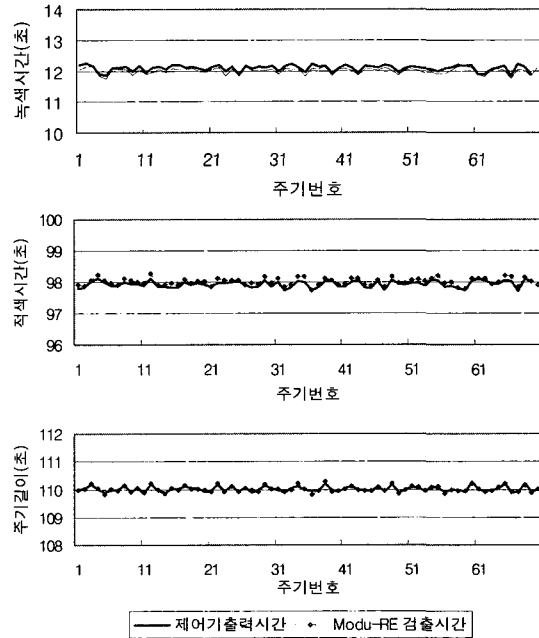
자료구분		평균값(초)	오차(초)	오차율(%)	비고
점유 시간	출력값	2.009782	0.076199	5.2275	N= 1.973
	계측값	2.085981			
비점유 시간	출력값	2.014318	0.075261	6.0444	
	계측값	1.939057			
차두 시간	출력값	4.024100	0.000938	1.9453	
	계측값	4.025038			

나타나 전체적으로 오차 80msec 이하의 높은 성능을 보이고 있다.

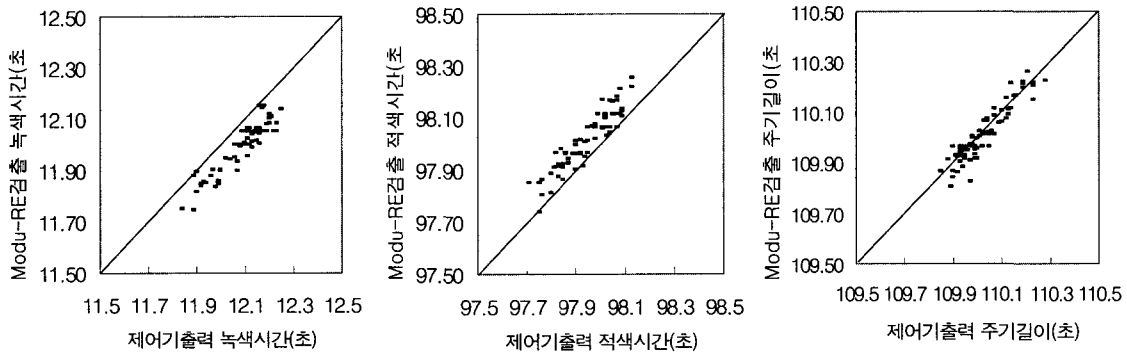
이러한 오차는 Modu-RE에서 디지털로 출력된 신호가 인터페이스 장치인 Real-CID를 거쳐 아날로그 신호로 변조된 과정에서 필연적으로 발생하는 오차이다.

3) 등기신호 검출 정확도 평가

다음은 교통신호제어기에서 출력된 등기신호 출력을 개발 시뮬레이션 시스템이 검출한 것으로 〈그림 8〉 및 〈그림 9〉에서 보는 바와 같이 녹색신호의 경우 실제



〈그림 8〉 녹색/적색/주기길이 신호검출 추세



〈그림 9〉 녹색/적색/주기길이 회귀선 분포



〈표 6〉 등기신호검출 정확도 평가결과

자료구분		평균값(초)	오차(초)	오차율(%)	비고
녹색 신호	검출값	12.000571	0.080920	0.6880	N=70
	계측값	12.081491			
적색 신호	검출값	98.014400	0.080487	0.0831	
	계측값	97.933913			
주기 길이	검출값	110.014971	0.001841	0.0300	
	계측값	110.016812			

보다 다소 짧게 검출하고 있으며, 적색신호의 경우 반대로 조금 길게 검출하고 있다.

주기길이로서 설정한 녹색신호시간과 적색신호시간의 합은 각 현시가 오차를 상쇄함으로써 정확한 추세 및 값을 계측하고 있다.

시험 결과를 수치로 보면 녹색신호의 검출은 0.6880%, 적색신호시간의 검출은 0.0831%, 주기길이는 0.0300%의 오차율을 나타내고 있어 평균 오차율 1% 미만의 정확한 계측을 수행하고 있음이 증명되었다.

**2. 교통신호제어시스템에의 적용사례**

본 연구에서는 국내의 신신호제어기에 개발 시뮬레이션 시스템을 적용하여 호환성 여부 및 제반 기능의 정상작동 여부에 대한 성능시험을 실시하였다.

호환성 시험은 검지신호출력, 등기신호검출, 수동 조작패널 입출력, 센터제어컴퓨터 통신캡처, Modu-RE의 제반기능 등 총 다섯 가지 부문에 대하여 적합성 여부를 시험하였다.

제어기의 성능시험은 신호제어기의 입출력 정확도 및 절차, 포화도 및 포화교통류를 산출, 과점유 판단, 감응제어기능, 스피백 제어기능 등에 대하여 다양한 평가를 실시하였다.

본 연구에서는 신신호제어기만을 대상으로 하고 있으나, 개발 시뮬레이터는 일반신호기를 포함한 모든 신호제어시스템에 적용이 가능하다.<sup>6)</sup>

**1) 개발 시뮬레이션 시스템의 호환성**

호환성 시험은 국내 6개 사의 신신호제어기를 대상

〈표 7〉 개발 시뮬레이션 시스템 호환성 시험결과

시험구분	호환 여부	비고
검지신호출력	적격	Real-CID에 별도의 장비 없이 직접 연결 가능
등기신호검출	적격	
수동조작패널 입출력 에뮬레이션	적격	제어기에 따라 DC 12V가 on, 0V가 off 혹은 0V가 on, 12V가 off로 상이하나 Modu-RE에서 초기 환경설정 가능
센터제어컴퓨터 통신 캡처	적격	Modu-RE에 통신규약을 사전 설정 가능
Modu-RE의 제반기능 시험	적격	특이사항 없이 호환

으로 하였으며, 시험결과 별도의 추가장비 없이 신호 제어기의 입출력 단자에 직접 연결하여 Modu-RE의 제반기능이 정상적으로 작동하였고 〈표 7〉은 시험결과를 종합한 것이다.

수동조작 패널의 입출력 에뮬레이션에 대한 실험결과 신호제어기에 따라 직류(DC) 12V가 on, 0V가 off로 설정되거나, 반대로 0V가 on, 12V가 off로 설정되어 상이한 면이 있으나, 하드웨어의 별도 추가 없이 Modu-RE에서 사전에 환경을 설정하도록 하고 있어 호환이 가능하다.

센터제어컴퓨터의 통신캡처 기능 역시 현재는 서울 신신호의 통신규약을 기정값으로 설정하고 있으나 신신호제어기에 따라 통신규약이 상이한 경우 Modu-RE의 사전 환경설정으로 호환 가능하다.

개발 시뮬레이션 시스템의 운영시스템인 Modu-RE의 각종 기능은 프로그램의 수정 없이 완벽한 호환성을 유지하였고, 작동에 오류가 발생하지 않았다.

**2) 신호제어기 성능검사 사례**

현행 교통신호제어기의 성능검사는 주로 국가공인 시험기관인 한국기계연구원에서 하고 있으나 성능검사 항목이 외관검사, 기능검사, 환경시험, 진동시험, 충격시험, 절연시험 등 하드웨어적인 시험에 그치고 있으며, 제어의 효율성을 고려한 소프트웨어적인 기능에 있어서는 검사방법 및 기준 등이 미흡한 상태이다.

국내에서 신신호시스템용 제어기가 개발되어 설치 운영 중에 있음에도 불구하고 신신호시스템의 각종

6) 일반신호제어기는 주로 국도 상의 독립된 교차로나 지방 소도시와 같이 센터와는 별개로 제어가 이루어지는 교차로에 설치 운영되는 제어기로 현재 TOD 제어 중심으로 운영되고 있지만, 자체 기능으로 NEMA에서 규정하고 있는 검지기 기반의 감응제어 기능을 보유하고 있어 기능상의 정확성 평가는 물론 신호등기 출력의 정확도 면에서도 신신호제어기뿐 아니라 전자신호제어기 일반신호제어기의 기능도 평가가 가능하다.

기능에 대한 성능검사 방법이 없어 기존 전자교통신호제어기 검사기준에 의거 신신호제어기를 시험·검사하고 있는 실정이다.

본 연구에서는 이러한 문제점을 개선하기 위하여 교통관리의 효율성 평가와 아울러 제어시스템의 성능검사를 수행할 수 있도록 시뮬레이션 시스템을 개발한 것이다.

개발 시뮬레이션 시스템을 이용한 제어기의 성능검사는 Modu-RE에 검사 및 평가라는 단일 모듈로 구성되어 있으며, 환경설정에 이어 모듈구성시 전과정을 자동으로 수행 후 최종결과를 출력하도록 개발되었다.

다음은 국내 K사의 신신호제어기 성능검사 사례이며, 일반사항, ON-LINE 기능시험, 제어모드별 시험, 검지기 정보처리 시험 등 총 25개 기능시험 중 가장 기본적인 5가지 항목에 대하여 부록에 제시된 시험결과 log-out 자료를 예로 설명한다.

기능시험과 관련하여 한가지 주목해야 할 사항은 등기출력정확도에 대한 NEMA(1989)의 기준 즉, 1주기의 오차가 100msec 이내이어야 한다는 기준 외에는 공인된 검사기준이 없다는 것이다. 따라서, 본 연구에서는 부득이 대상제어기의 적부판단 결과가 아닌 시험결과와 값을 제시하였다.

#### (1) 등기출력 정확도

[부록 1]은 4현시의 황색시간 3초를 가진 주기 110초의 신호등기출력을 검출한 것으로 현시별 최고 과부족시간은 각각 녹색시간 160msec, 황색시간은 75msec, 보행시간은 200msec, 보행점멸시간은 20msec, 보행전시간은 60msec이고, 총 주기는 25msec가 짧게 제공된 것으로 분석되었다. <표 7>의 등기출력 정확도 결과는 [부록 1]의 Ph# 1~4 중 오차가 가장 큰 값만을 선별한 것이다.

#### (2) 현시출력 정확도

[부록 2]는 신호현시 출력결과에 대해 현장 신호제어기가 센터시스템에 보고한 내용과 시뮬레이션 시스템이 검출한 등기출력 신호를 비교하여 각 현시의 출력 정확도를 분석한 결과이다.

내용을 보면 L/C는 현장제어기가 센터제어컴퓨터로 보고한 내용을, SIM은 시뮬레이션 시스템에서 검출한 등기출력신호를, 그리고 CHA는 SIM과 L/C값의 차이

를 의미하는 것으로 각 현시에 따라 -180~+200msec의 과부족 오차를 보이고 있다.

#### (3) 검지기 정보처리 정확도

[부록 3]은 신호제어기의 차량 검지정보처리에 대한 분석결과로서 각각 현장 신호제어기(LC)와 시뮬레이션 시스템(SIM)의 점유율(\*\_OCC), 비점유율(\*\_NOCC), 교통량(\*\_VOL), 포화교통류율(\*\_SFR), 포화비점유시간(\*\_tVal), 포화도(\*\_DS) 등을 의미한다.

대상 신호제어기의 검지기 중 DET#1의 검지결과를 보면, LC\_OCC와 LC\_NOCC를 각각 11.00과 19.00으로 계산하여 인식하고 있으나, 이에 대한 시뮬레이션 시스템의 계측값인 SIM\_OCC 및 SIM\_NOCC는 각각 10.57, 19.43으로 차이가 발생하고 있음을 알 수 있다.

이러한 오차는 신호제어기가 아날로그 형태의 차량 검지신호를 디지털 신호로 변환하는 과정에서 불가피하게 발생하는 오차이다.

그러나 교통량, 포화교통류율 등은 시뮬레이션 시스템에서 출력한 값과 동일하게 계산하고 있어 정확한 검지정보처리가 이루어지고 있다.

#### (4) 감응제어기능

[부록 4]는 최소녹색시간(Min green) 5초, 최대녹색시간(Max green) 20초를 내장하고 있는 신호제어기의 감응제어기능에 대한 실험결과이다.

교통량이 한 대도 없는 상황(No call), 계속적으로 차량이 도착하는 상황(Continuous call) 및 임의적으로 도착하는 상황(Random call) 등의 3가지 조건에 대하여 시험한 결과 각각 5초, 20초, 12초로 등기출력(Used green)이 계측되어 감응제어기능이 정상적으로 작동하고 있음이 확인되었다.

#### (5) 스피백 제어기능

[부록 5]는 스피백 제어기능에 대한 시험 결과로써 차량의 증가로 교차로가 막히는 상황과 스피백이 해소되는 상황을 인위적으로 연출한 것이다.

스피백 발생 시에는 시뮬레이션 시스템에 최소녹색시간 7초, 그리고 스피백이 해소되는 상황에서는 주어진 녹색시간 27초가 등기 출력된 것으로 계측되어 스피백 제어기능이 정상적으로 제대로 작동하고 있음이 확인되었다.

〈표 8〉 신호제어기 성능검사결과

검사종류	항목	기준값 (초)	시험값 (초)	비고
등기출력 정확도	주기	110.000	109.975	Ph#1~4 중 최고오차값
	녹색시간	28.000	27.840	
	황색시간	3.000	3.075	
	보행자녹색	20.000	19.800	
	보행자점멸	5.000	5.020	
	보행전시간	2.000	2.060	
현시출력 정확도	주기	110.000	110.070	기준값과 시험값의 차이는 'CHA'로 제시됨
	Phase1	15.000	15.200	
	Phase2	33.000	33.049	
	Phase3	31.000	31.001	
	Phase4	31.000	30.820	
	Phase5	15.000	15.200	
	Phase6	33.000	33.049	
	Phase7	31.000	31.046	
	Phase8	31.000	30.840	
검지기 정보처리 정확도	점유	11.00	10.57	DET#1의 시험값
	비점유	19.00	19.43	
	교통량	7	7	
	포화교통류율	2200	2200	
	포화비점유시간	0.95	0.95	
	포화도	58	59	
감응제어 기능	No call	5	5	정상작동
	Continuous	20	20	
	Random	5~20	12	
스필백 제어기능	스필백 발생	7	7	정상작동
	스필백 해소	27	27	

V. 결론 및 향후 연구과제

본 연구에서는 ITS의 각종 시스템에 대한 효율성 평가를 목표로 교통신호제어시스템의 성능평가를 위한 시뮬레이션 시스템을 개발하였다. 이는 종래의 소프트웨어 기반 시뮬레이션 프로그램에 인터페이스 장치를 통해 실제시스템의 하드웨어를 직접 연결하여 실시간으로 교통관리시스템의 기능성을 평가할 수 있는 시스템이다.

교통신호제어시스템을 대상으로 개발 시뮬레이션 시스템의 성능을 시험한바 정확도 및 안정성 면에서 우수하고, 별도의 추가장비 없이도 모든 신호제어기에 호환성을 확보하는 것으로 평가되었다.

본 연구는 세계적으로 초기단계에 있는 기술을 독

자적으로 개발하였으며, 특히 기존 미국과 유럽 개발 사례의 취약점인 범용성 부족 문제의 해결을 위하여 최단말 입출력 신호를 아날로그화 함으로써 기능적인 측면에서 제어의 효율성뿐만 아니라 하드웨어 자체의 성능검사까지도 수행이 가능하다는데 의의가 있다.

본 연구에서는 현재 시뮬레이션 시스템의 하드웨어 및 운영소프트웨어를 완성하여 교통신호제어기의 기능검사를 수행하고 있으나, 신호제어시스템의 각종 제어방법별 운영효율성 평가나 ITS 부시스템의 제어 알고리즘에 대한 적용성 평가를 위해서는 운영소프트웨어 내에 보다 정교하게 도로의 교통상황을 재현할 수 있는 미시적 교통류 재현 모형의 탑재가 추가적으로 필요하다.

시뮬레이션 시스템에 탑재될 교통류 재현모형에 요구되는 주요기능은 다음과 같다.

- 교통상황, 도로조건 및 관계상황 등 제반 네트웍상황뿐만 아니라 차량의 행태를 최대한 현실에 가깝게 묘사하여야 한다.
- 모형 내부에 검지기를 설치하고 현실의 검지기와 동일한 자료의 출력이 가능하여야 한다.
- 모형과 연결된 실제 제어기에서 출력된 제어내용을 입력받아 모형내부의 차량을 제어할 수 있어야 한다.
- 모형내부의 CPU time과 Clock time의 동기가 가능하여야 한다. 즉, 현실의 1초가 시뮬레이션 상황에서도 1초로 동일하게 인식되어야 한다.
- 효율성을 평가할 수 있는 다양한 MOE를 산출하여야 한다.
- Off-Line으로 추가적인 분석을 위한 입출력 자료의 저장이 필요하다.

본 연구의 시뮬레이션 시스템 개발로 교통관리시스템의 운영효율을 과학적이고 객관적인 방법에 의해 평가함으로써 문제점을 정확히 파악하고 개선 내용 및 범위의 조기파악이 가능하게 되었다. 따라서, 고장이나 오류 발생시 응급처리 및 복구시간의 단축으로 효율적인 유지관리 및 비용을 절감할 수 있을 것이다.

또한, ITS의 신기술이나 신제품의 도입시 예상되는 문제점을 미리 예측할 수 있고 효율성의 사전측정이 가능하여 과학적인 정책결정에 도움이 되고 시행착오를 줄일 수 있어 예산의 절감이 기대된다.

산업체에서는 시스템 개발시 오작동 및 기능상의

문제점에 대한 사전 진단이 강화되고, 알고리즘 수정 능력이 증대되어 민간부분의 개발비용 및 기간단축을 도모할 수 있을 것이다.

연구와 관련하여 유사한 시스템의 개발 혹은 관련 연구시 기술지원 및 교류로 개발의 활성화를 촉진할 것으로 기대된다.

향후 교통류 재현 모형의 탑재를 위한 연구가 완료 되면 ITS 시스템에 대한 효율성 평가도구로 확장사용이 가능하여 시스템 개발, 도입 시스템 선정, 설치 후 시스템의 조기 안정화 및 유지관리 등 각 단계에서 검증수단으로 유용하게 활용될 것이다.

## 참고문헌

1. 교통개발연구원(1993), "교통신호제어시스템 기술 개발, 시뮬레이터 개발보고서".
  2. 도로교통안전관리공단(1999), "교통신호제어시스템 성능검사 및 평가 시뮬레이터 개발-1단계(요구분석 및 시스템 설계)", 교통과학연구원, No.99-08-360.
  3. 도로교통안전관리공단(2000), "교통신호제어시스템 성능검사 및 평가 시뮬레이터 개발-2단계(제작 및 사례적용)", 교통과학연구원, No.00-03-373.
  4. 서울지방경찰청·도로교통안전관리공단(1996), "서울특별시 첨단교통신호제어 시스템 표준화 제정(교통신호제어기 부문)".
  5. 이영인(1994), 서울시 신신호 교통제어시스템, 대한교통학회지 제12권 제2호, pp.65~74.
  6. 장덕명·김영남(1994), 원형검지기과 기존 검지기의 비교 분석에 관한 연구, 대한교통학회지 제12권 제4호, pp.35~52.
  7. Black G. Hansen, Peter T. Martin, and H. Joseph Perrin, Jr, "SCOOT Real-Time Adaptive Control in CORSIM Simulation Environment", In Transportation Research Record 1727, TRB, National Research Council, Washington, D.C., 2000, pp.27~31.
  8. David E. Lucas, Pitu B. Mirchandani, and K. Larry Head, "Remote Simulation to Evaluate Real-Time Traffic Control Strategies", In Transportation Research Record 1727, TRB, National Research Council, Washington, D.C., 2000, pp.95~100.
  9. F Biora, K A Fox, P Franklin, M Hallworth, A D May and F O Montgomery(1995), "A Best Practice Manual For Innovative UTC Schemes", DRIVE II Project V2016 : PRIMAVERA, Deliverable No. 15.
  10. Federal Highway Administration(1998), "Traffic Software Integrated System Version 4.2 User's Guide".
  11. ITE(1990), Traffic Detector Handbook, Institute of Transportation Engineers, Washington, D.C.
  12. ITS(2000), SMARTTEST : Final Report for Publication, Institute for Transport Studies, University of Leeds, Leeds, U.K.  
<http://www.its.leeds.ac.uk/smartest/>
  13. NEMA(1989), NEMA Standards Publication No. TS-1, "Traffic Control Systems", National Electrical Manufacturers Association, Washington, D.C.
  14. Qi Yang, Haris N. Koutsopoulos, and Moshe E. Ben-Akiva(2000), "Simulation Laboratory for Evaluating Dynamic Traffic Management Systems", In Transportation Research Record 1710, TRB, National Research Council, Washington, D.C., pp.122~130.
  15. Roelof J. Emgelbrecht, Christopher M. Poe, Kelvin N. Balke(1999), "Development of a Distributed Hardware-In-The-Loop Simulation System for Transportation Networks", In Presentation and Review 78th Annual Meeting of the Transportation Research Board.
  16. Roelof J. Emgelbrecht(1999), "Hardware-In-The-Loop Simulation", on the web.  
<http://translink.tamu.edu>
- ♣ 주 작 성 자 : 정준하  
 ♣ 논문투고일 : 2001. 5. 12  
 논문심사일 : 2001. 6. 19 (1차)  
 2001. 7. 13 (2차)  
 2001. 8. 27 (3차)  
 2001. 9. 6 (4차)  
 심사판정일 : 2001. 9. 6

**[부록] 교통신호제어기 성능검사 출력자료**

**[부록 1] 등기출력 정확도 시험 log-out 일부**

Cont.ID: 1, Cycle#: 3, CycleLength: 109.975, Eoc\_Time : 16:10:18

*** Ph#	Min.Grn	Green	Yellow	Ped.Solid	Ped.Flash	Bef.Ped
1	5	12.125	<u>3.075</u>	0.000	0.000	0.000
2	5	30.047	3.002	20.025	4.987	2.039
3	5	28.001	3.000	0.000	0.000	0.000
4	5	<u>27.840</u>	2.980	<u>19.800</u>	<u>5.020</u>	<u>2.060</u>

**[부록 2] 현시출력 정확도 시험 log-out 일부**

Cycle#: 4, Time: 16:10:10

구분	Cycle	Phase1	Phase2	Phase3	Phase4	Phase5	Phase6	Phase7	Phase8
L/C	110.000	15.000	33.000	31.000	31.000	15.000	33.000	31.000	31.000
SIM	110.070	15.200	33.049	31.001	30.820	15.200	33.049	31.046	30.840
CHA	0.070	0.200	0.049	0.001	-0.180	0.200	0.049	0.046	-0.160

**[부록 3] 검지기 정보처리 정확도 시험 log-out 일부**

DET#	GRN	LC_OCC	SIM_OCC	LC_NOCC	SIM_NOCC	LC_VOL	SIM_VOL	LC_SFR	SIM_SFR	LC_tVal	SIM_tVal	LC_DS	SIM_DS
1	27	11.00	10.57	19.00	19.43	7	7	2200	2200	0.95	0.95	58	59
2	27	15.00	14.64	15.00	15.36	10	10	2200	2200	0.95	0.95	81	82
3	27	11.00	10.51	19.00	19.49	10	10	2200	2200	0.95	0.95	68	68
4	27	13.00	12.68	17.00	17.32	7	7	2200	2200	0.95	0.95	65	65
5	27	16.00	15.76	14.00	14.24	9	9	2200	2200	0.95	0.95	81	82
6	27	11.00	10.51	19.00	19.49	7	7	2200	2200	0.95	0.95	58	59
7	27	10.00	9.69	20.00	20.31	9	9	2200	2200	0.95	0.95	61	62
8	27	17.00	17.22	13.00	12.78	8	8	2200	2200	0.95	0.95	82	82

**[부록 4] 감응제어기능 시험 log-out 일부**

Gap actuation control test start . . .

\*\*\* Case 1: No vehicle call(1) at Time: 15:39:22 \*\*\*

Phase#: 1, Min green: 5, Max Green: 20, Used green: 5

\*\*\* Case 2: Continuous vehicle call(1) at Time: 15:41:12 \*\*\*L

Phase#: 1, Min green: 5, Max Green: 20, Used green: 20

\*\*\* Case 3: Random vehicle call(1) at Time: 15:43:02 \*\*\*

Phase#: 1, Min green: 5, Max Green: 20, Used green: 12

Gap actuation control test end . . .

**[부록 5] 스프릴백 제어기능 시험 log-out 일부**

Spill-Back control test start . . .

\*\*\* Case 1: Spill-back enabled (1) at Time: 19:57:22 \*\*\*

Phase#: 1, Min green: 7, Max Green: 27, Used green: 7

\*\*\* Case 2: Spill-back disabled (1) at Time: 19:59:42 \*\*\*

Phase#: 1, Min green: 7, Max Green: 27, Used green: 27

Spill-Back control test end . . .