미시적 시뮬레이터를 이용한 교통대응 신호시스템의 운영효과 분석

李 榮 寅*

〈目 次〉

Ⅰ. 서론
1. 연구의 목적
2. 연구의 범위 및 방법
Ⅱ. 기존 시뮬레이터의 고찰
1. TRAF-NETSIM
2. VISSIM
3. PARAMICS
4. TRANSIMS
Ⅲ. 미시적 시뮬레이터의 개발
1. 시뮬레이터 개요
2. 시뮬레이터의 구조 및 탑재 알고리즘
3. 시뮬레이터의 진행과정
Ⅳ. 시뮬레이터의 신뢰성 분석 및 제어전략 평가
1. 시뮬레이터의 신뢰성 분석
2. 독립교차로에서의 신호제어전략 평가
3. 네트워크에서의 신호제어전략 평가
Ⅴ. 결론

Ⅰ. 서 론

1. 연구 목적

도시 가로망의 교통안전과 소통을 증진시킬 수 있는 주요한 방안 중의 하나는 교통 신호제어전략을 교통수요에 대응하여 수립하는 것이다. 최적화된 신호제어전략은 교차로의 교통용량 증대와 서비스수준 향상, 그리고 교통사고 감소 등에 크게 기여할 수 있다. 국내에서는 도심 교통혼잡을 완화시킬 수 있는 방안으로서 교통대응 신호시스템인 COSMOS를 비롯하여 다양한 신호제어기법들이 개발되고 있다. 그러나 다양한 신호제어기법들의 운영효과를 현장에서 직접 측정하기에는 수많은 제약들이 존재한다. 시시각각으로 변화하는 다양한 현장 교통상황에 개별 교통신호 제어전략이 어떻게 작용할 지를 판단하기 위해서는 많은 시간과 노력이 필요하며, 경우에 따라서는 정확한 효과분석이 불가능할 수도 있다. 미시적 시뮬레이션 모형을 이용한 운영효과 분석은 이러한 현장분석의 제약을 완화하는 대안으로 인식되고 있다. 기존의 미시적 시뮬레이션 모형은 정주기 제어의 운영효과는 적절하게 분석할 수 있다. 그러나 기존 모형은 현

* 서울대학교 환경대학원 환경계획학과 교통관리전공 부교수
** 본 연구는 서울대학교 2003년도 신임교수 정착연구의 연구비 지원으로 수행되었음.

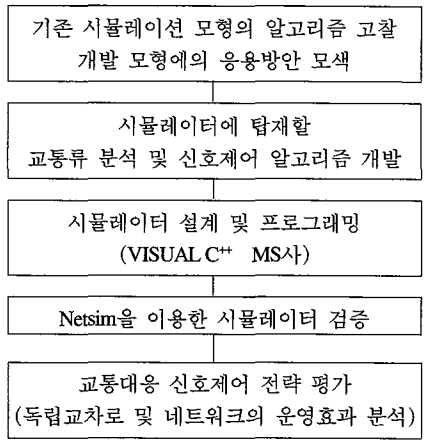
장 교통상황에 따라 신호시간을 최적화하는 최적 신호시간 산출기능이 결여되어, 신호주기별로 신호시간이 변화하는 교통대응 제어전략의 효과 분석에는 한계가 있다.

본 연구의 목적은 신호시간 최적화 기능을 수행 할 수 있는 미시적 시뮬레이터를 개발하고, 이를 이용하여 현재 서울에서 운영되고 있는 실시간 신호제어 시스템(COSMOS)의 각종 신호전략을 평가하는 데 있다.

2. 연구의 범위 및 절차

본 연구에서 개발된 미시적 시뮬레이터는 신호교차로를 포함하는 네트워크에서 주어진 신호제어 전략에 따라 개별 차량단위로 교통상황을 변화시키며 시뮬레이션을 수행한다. 네트워크에서 차량의 진행상황은 시뮬레이터의 애니메이션을 통해 시각적으로 확인이 가능하며, 시뮬레이션 모형은 개별 차량 단위로 수집되는 교통정보를 통하여 교차로의 효과척도를 산출하는 기능을 수행한다.

현 단계의 모형은 미시적 시뮬레이터의 초기 모형으로서, 다양한 운전자의 성향을 반영할 수 있는 기능이 제한적이며, 분석 네트워크도 직교 교차로로 구성된 장방형의 가로망으로 국한되어 있다. 그러나 본 모형은 차량의 발생이나 진행 등 차량의 행태를 결정하는 핵심적인 부분에서는 기존 모형과 동일한 형태로 제작되었으므로 동일 교통상황에서의 차량진행 상황은 통계적으로 유의한 범위 내에서 동일하게 분석할 수 있다. 또한 직교 교차로로 구성된 가로망 역시 일반적인 신호제어전략의 평가 목적으로는 충분한 것으로 판단된다. 본 연구의 진행과정은 〈그림 1〉과 같다.



〈그림 1〉 연구 진행과정

Ⅱ 기존 미시적 시뮬레이터의 고찰

1. TRAF-NETSIM(Network Simulation Tool)

NETSIM은 네트워크 내를 주행하는 차량의 운행상태를 매 초 단위로 명료하게 모형화 한다. 모형의 기본적인 접근 방법은 교통상황을 각 시간 간격별로 재현 및 응용하는 것이다. NETSIM의 주요기능들은 포화 교통류 상태 하의 복합 네트워크 모의시험, 실시간 수집 자료에 바탕을 둔 동적인 교통류제어, 정주기제어 및 감응제어 전략 평가, 신호조건 및 교통여건의 변화에 따른 보정계수의 산출 등을 포함하고 있다. 이 모형은 네트워크 내로 진입한 각 차량들의 위치와 시간을 계속하여 추적한다. 차량 추종모형은 정상시에는 일반적인 차량 추종 행태를 따른다. 그러나 교통상황이 혼잡한 경우에는 선행차량이 최대 감속도($3m/s^2$)로 정지할 경우, 뒷 차량은 선행 차량과 충돌하지 않는다는 조건 하에 선행차량을 추종한다는 가정으로 진행된다.

NETSIM은 정주기 운영 전략과 녹색시간을 연장하는 감응제어(actuated-control) 전략의 운영효과를 분석할 수 있다. 그러나 교차로의 신호주기와 녹색시간을 신호주기별로 제어하는 교통대응(adaptive) 전략의 운영효과를 분석하는 데는 한계가 있다. TSIS version 4.02에서는 CORSIM Runtime Extension 기능을 추가하여 RT-TRACS의 교통대응 신호제어전략을 시뮬레이션 할 수 있도록 하고 있다. 그러나 이 기능은 현재 RT-TRACS의 신호제어 전략 평가에 국한되어 있으며, RT-TRACS 이 외의 교통대응(adaptive) 제어 전략을 다양하게 평가할 수 있도록 연구가 진행되고 있다.

2. VISSIM

VISSIM은 독일의 PTV사에 의해 개발된 미시적 시뮬레이션 모형이다. VISSIM은 스코틀랜드에서 개발된 PARAMICS와 함께 Animation 기능이 뛰어난 미시적 시뮬레이션 모형으로 분류되고 있다. 이 모형의 대표적 기능은 다음과 같다.

- 교통류분석(지점별 속도, 구간별 속도 및 통행시간, 혼잡교통류 분석) 기능
- O-D에 의한 분석 및 구간 교통량에 의한 분석
- 다양한 차종구분 및 운전자 행태구분 가능: 대상차종구분 및 위반 운전자 반영
- 2D 및 3D Presentation 기능

VISSIM에서 뒷 차량의 차량추종은 psycho-physical model(Wiedemann 1974)에 의하여, 차선 변경은 rule-based algorithm에 의해 이루어진다. 또한 VISSIM은 승용차, 대중교통, 보행자에 대한 시뮬레이션과 신호등가로, 고속도로에 적용이 가능하며, VMS 효과분석, Toll Gate 상황 분석, 그리고 동적 통행배정(Dynamic Assignment)도 수행할 수 있다.

3. PARAMICS

PARAMICS 모형은 크게 Modeller, Analyzer, Processor의 세 모듈로 구성되어 있다. Paramics Modeller는 교통상황의 모델링 모듈로서 모형 구축기능, 교통 시뮬레이션(3-D 시각화 지원) 기능, 통계 데이터의 출력 기능을 수행한다. Modeller는 다음 교통 상황들을 미시적으로 분석할 수 있다.

- 혼합형태의 도시 및 일반도로 네트워크 분석
- 로터리(교차점), 대중교통, 주차장 시설, 유고상황 분석
- 화물전용차선 분석, 버스 전용차선 등의 다인승 버스 전용차선 분석 등

Paramics Processor는 시뮬레이션 환경구축을 위한 도구로 네트워크 시뮬레이션을 배치모드(batch mode)에서 구동할 때 사용된다. Processor는 그래픽 사용자 인터페이스를 제공하므로 시뮬레이션 파라미터를 손쉽게 세팅할 수 있고, 다양한 통계 데이터를 선택하여 출력할 수 있으며, 네트워크에 유입된 차량의 구성인자(attributes)를 구분하여 서로 다른 시뮬레이션을 동일 네트워크 상에서 구동할 수 있다. Paramics는 여러 세트의 시뮬레이션을 구동 할 경우 해당 모형 내에서 변화추이를 파악할 때 유용하게 사용된다.

4. TRANSIMS(Cellular Automata Theory)

Los Alamos National Laboratory에서 개발된 TRANSIMS는 cellular automata theory와 컴퓨터의 이진(binary)구조를 응용하여 대규모(large-scale) 네트워크에서 교통과 관계된 모든 통행자의 미시적 모의실험을 수행하는 모형이다. 이 모형은 기본적으로 차량추종 및 차로변경을 포함한 비보호 좌회전, 대기행렬, 용량저하현상(breakdown) 등의 표현까지 세밀한 수준으로 대규모 네트워크 분석에 효과적인 능력을 가진 모의실험 모형으로 분류되고 있다. 이 모형에서는 분석 네트워크를 모두 일정한 간격의 기본 가정값을 가진 셀(cell)들로 나누어 개별 차량들을 이동시킨다. 셀의 이동은 속도값에 의해 결정되는데 이는 한 모의실험 클럭(clock)안에서 전방 셀 내의 차량유무를 검색하고 전이확률(stochastic transition)을 고려하여 결정된다.

TRANSIMS에서는 GM 5세대 모형을 근간으로 추종모형을 사용하고 있지만 일정 셀의 폭으로 규정된 이산적 공간이동(discrete space)으로 정확한 차량위치 추적에는 한계가 있다. 또한 차로변경을 포함한 차량전이에 관련된 규칙(rule)들이 운전자 행태를 기초로 한 경험적인 확률값의 사용으로 정밀한 분석을 위한 분석적인(analytic) 체계성이 결여되어 있다.

Ⅲ. 미시적 시뮬레이터의 개발

1. 시뮬레이터 개요

본 시뮬레이터는 신호운영 전략을 평가하고 그 효과를 측정, 분석함으로써 신호 운영 전략의 개선 등에 활용하기 위한 목적으로 개발되었다. 신호 운영전략의 평가를 위해서는 최소한 신호 시간의 단위, 즉 1초 단위 또는 그보다 더 작은 단위로 시뮬레이션을 진행하는 것이 유리하다. 본 연구에서는 수행속도 보다는 보다 정밀한 현실 묘사를 위하여 중, 소규모의 네트워크(교차로 20-30개 정도)를 0.1초 시간 단위에서 시뮬레이션 할 수 있는 미시적 시뮬레이터를 개발하였다. 교통류 시뮬레이터인 본 모형의 기능(모듈)은 다음과 같다.

- 차량진행 기능: 차량추종, 차로변경(선택적, 의무적)
- 신호운영 기능: 정주기식(입력자료 필요), 교통대응식(점유율에 따른 주기결정 및 현시배분)
- 검지자료 산출 기능: 지점검지기를 통한 교통량, 점유율 산출
- MOE 산출기능: 여행시간, 교통량, 정지지체 산출
- 에니메이션 기능(시뮬레이션과 동시에 진행)
- 임의의 네트워크 구축 기능(교차로 20-30개 수준의 격자형 네트워크)

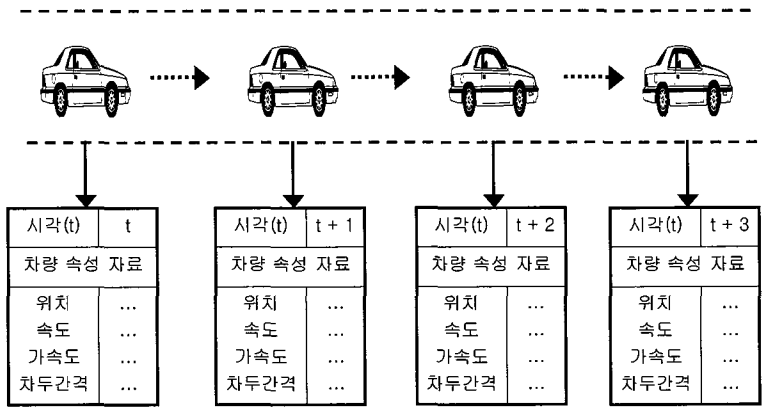
2. 시뮬레이터 구조 및 탑재 알고리즘

본 시뮬레이터는 객체 지향적 제작기법을 통하여 개발되었다. 이 기법은 구현하고자 하는 환경의 구성요소들을 동질적인 특징과 기능을 수행하는 객체들로 구분하고, 이러한 객체들이 특정한 사건(Event)이 발생할 때마다 이에 반응하며 시뮬레이션을 진행시키는 방식이다. 이러한 제작 기법은 교통과 같은 복잡한 현실을 묘사하는데 있어서 가장 유리한 방법 중의 하나이다. 본 시뮬레이터는 객체 지향적 프로그램 Language 중의 하나인 Microsoft社의 Visual C++를 이용하여 Window 2000 환경에서 실행 가능하도록 제작되었다.

1) 미시적 차량진행 방식

본 시뮬레이터는 Discrete-Event Simulator이다. 이 방식은 분석대상 즉 어떠한 System의 변화하는 상황을 시간에 따라 전개시키는 것으로 전체 시뮬레이션 시간을 이산적인 일정간격의 시간단위로 나누어 매 Time step 마다 System의 상태를 표현하는 변수(차량의 위치, 신호등의 등화 등)들을 동시에 갱신시키는 것이다.

본 시뮬레이터에서 가장 중요한 Event는 차량이 도로 상을 주행하는 것(차량의 위치정보 갱신)과 시간에 따라 신호 현시가 바뀌도록 하는 것이다. 본 시뮬레이터에서 채택한 시간적 간격(Δt: time)에 따른 차량진행 방식을 그림으로 나타내면 〈그림 2〉와 같다. 이를 차량의 주행 과



〈그림 2〉 시간적 간격에 따른 차량 진행 방식

정에 맞추어 설명하면, 매 time step(0.1초) 마다 네트워크 상의 모든 차량에 대하여 각 차량의 앞 차량과의 거리 및 해당 차량의 속도를 고려하여 감/가속도를 산출하고, 이를 통해서 다음 time step에서의 차량의 위치를 계산하여 차량의 위치정보를 갱신하게 된다.

2) 차량행태 관련 알고리즘

(1) 차량발생 모형

Time Based 시뮬레이션 모형에서 매 시간 간격마다 발생되는 Event 중 하나는 진입링크에서 차량의 발생(네트워크 진입)여부를 확인하는 것이다. 본 시뮬레이터에서는 음지수 분포를 사용하여 시간당 교통량에 따라 개별차량의 발생 차두시간 간격을 결정하였다.

음지수 분포는 일정 시간간격(t: headway) 동안 차량이 한 대도 발생되지 않는 확률 값을 산출하는 확률분포함수이다. 그러나 차량의 차두시간 간격을 얻기 위해서는 음지수 함수의 역함수, 즉 차두시간 간격의 함수가 필요하기 때문에 이를 얻기 위해서 확률분포 함수식 양변에 log를 취하여 정리하면 다음과 같다. 여기에서, $f(t)$는 확률분포 함수 값이므로 0에서 1 사이의 값을 갖는다. 그런데 프로그램 상에서 산출되는 난수(random number)는 기본적으로 두 값 사이의 값을 균일하게 발생시키는 것이므로 프로그램 상에서 0과 1 사이의 값을 반환하도록 지정하여 $f(t)$의 값으로 사용하면 된다. 지금까지의 과정을 정리하면 개별차량의 차두시간 산출식은 다음과 같이 정리된다.

$$f(t) = 1 - e^{-\frac{(t-c)}{T-c}}$$

$$\log(1 - f(t)) = -\frac{(t-c)}{T-c}$$

$$t = c - (T-c) \times \log(1 - f(t))$$

여기에서, t: 개별차량 차두시간, c: 최소차두시간, T: 평균차두시간

(2) 차량추종모형

본 시뮬레이터에서 채택한 차량추종 모형은 시간 t에서의 후행차량의 반응은 $t-\Delta t$에서의 선행차량에 의해서 주어지는 후행차량의 민감도와 자극의 강도에 의존한다는 개념의 GM 5th 모형을 채택하였다.

response = f(sensitivity, stimuli)

$$\ddot{x}_{n+1}(t+\Delta t) = \frac{a_{l,m}[\dot{x}_{n+1}(t+\Delta t)]^m}{[x_n(t)-x_{n+1}(t)]^l}[\dot{x}_n(t)-\dot{x}_{n+1}(t)]$$

여기에서, response: 후행차량의 가/감속도

sensitivity: 민감도

stimuli: 상대속도

$\ddot{x}_{n+1}(t+\Delta t)$: $t+\Delta t$에서의 $n+1$번째 차량의 가속도

(3) 차로변경 모형

본 모형의 차로 변경과정은 〈그림 3〉과 같다. 그림에서 차선변경을 시도하는 차량은 차선변경을 시작하기 전 시점인 시각 t에서 자신이 차선변경을 시행한 시점(시각 $t+T$)에 목적차선에서 새롭게 발생되는 추종관계에 의한 간격 1과 간격 2를 예측한다. 예측 결과, 이 간격들이 모두 안전을 기준으로 한 거리를 만족하는 경우 실제로 $t+T$ 시점에 차선변경을 실행한다. 여기서 안전을 기준으로 한 간격 1은 목적차선의 선행 차량이 시각 $t+T$에서 급 감속을 하여 정지하는 경우 차선을 변경한 차량이 이를 보고 따라서 감속을 했을 때 충돌하지 않는 거리를 의미한다.

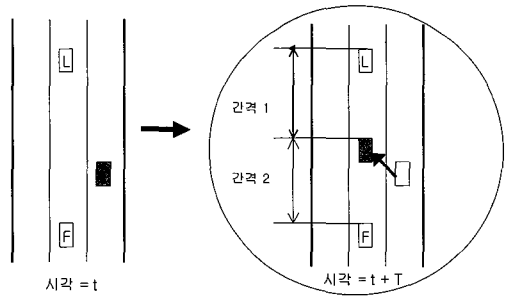
$$\text{간격 } 1 \geq T\dot{x}'_{n+1}(t) + \frac{{x'_{n+1}}^2(t+T)}{2x''_{n+1}(t+T)} + L - \frac{{x'_n}^2(t)}{2x''_n(t)}$$

여기에서, x_n: 목적차선의 lead 차량

x_{n+1}: 차로변경을 시도하는 차량

$\ddot{x}_{n+1}(t+T)$: $t+T$에서의 $n+1$번째 차량의 가속도

선택적 차선변경의 경우 NETSIM에 적용되고 있는 "Intolerable Speed"라는 개념을 이용하여 개별 차량의 속도가 Intolerable Speed 보다 높은 경우에는 선택적 차선변경을 시도하지 않도록 되어 있다. 물론 이러한 속도의 결정에는 운전자의 개별적인 성향이 반영되도록 되어있다.



〈그림 3〉 차로변경 과정

$V_i = V_f \times (0.7 \times DAF)$

V_i = Intolerable Speed

(본모형: $DAF = 1$, NETSIM의 초기 값)

3) 신호제어 알고리즘

(1) 정주기식 신호제어(TOD)

본 시뮬레이터에서 정주기식 신호제어는 사전에 입력된(pre-timed) 주기, 현시길이, 옵셋 길이를 사용한다. 따라서 대상 네트워크 내의 교차로 중에서 정주기식 신호제어를 운영하고자 하는 교차로의 경우 미리 해당 교차로에 적용될 신호변수를 입력해 주어야 한다. 그리고 이러한 신호변수 값들은 시뮬레이션이 진행되는 동안 동일하게 반복적으로 적용된다.

(2) 교통대응 신호제어(COSMOS)

본 시뮬레이터에서 적용된 교통대응 신호시스템의 주요 알고리즘은 중요교차로(CI), 준중요교차로(SCI), 비중요교차로(MI), 그리고 운영 교차로 그룹(Sub Area) 단위로 적용되어 진다. 각 단위별 적용 알고리즘은 〈표 1〉과 같다.

〈표 1〉 교차로별 적용 알고리즘

중요교차로(CI: Critical Intersection)	준중요교차로(SCI: Sub Critical Intersection)
- 포화도(DS) 산출 알고리즘 - 주기산출 알고리즘 - 포화도비를 이용한 현시분할 알고리즘 - 좌회전 감응제어 알고리즘 - 옵셋결정 알고리즘	- 포화도(DS) 산출 알고리즘 - 포화도비를 이용한 현시분할 알고리즘 - 좌회전 감응제어 알고리즘 - 옵셋전이 알고리즘
비중요교차로(MI: Minor Intersection)	**운영 교차로 그룹(SA: Sub Area)**
- Pattern Selection 기능 - 포화도비를 이용한 현시분할 알고리즘 - 옵셋전이 알고리즘	- Sub Area 결합분리 알고리즘

〈표 2〉 검지기의 자료산출 과정

단계	정보산출 과정
1 단계(초기화)	검지기가 설치될 차로, 차로에서의 위치 및 검지기 길이 정보 입력
2 단계	매 time step(0.1초)마다 각 차로를 주행하는 차량들의 위치정보 검색하여 검지기 위치와 비교하고 검지가 상을 주행하는 차량의 존재 여부 파악하여 검지기의 on/off 상태를 판정 및 저장
3 단계	해당차로의 현시 종료시점에서 현시동안의 검지기의 on/off 상태정보를 이용하여 교통량 및 점유율 산출

3) 검지기 관련 알고리즘

본 시뮬레이터의 주요 개발 목적인 대응 신호제어 전략의 평가를 위해서는 신호시스템 운영에 필요한 교통정보(교통량, 점유율 등)를 수집하는 검지기 시스템이 필수적인 사항이다. 본 시뮬레이터에서는 Loop 검지기를 차로별로 설치하여 소통자료를 수집할 수 있도록 하였다. 교통대응 신호시간은 시뮬레이터의 검지기에서 산출되는 data를 통해서 신호주기별로 계산하고, 이를 시뮬레이션 분석시 적용하도록 설계되었다. 본 시뮬레이터에서 검지기가 정보를 산출하는 과정은 〈표 2〉와 같다.

3. 시뮬레이터의 진행과정

본 시뮬레이터의 구조는 네트워크 모듈을 중심으로 시뮬레이션에 필요한 주요 모듈들이 유기적으로 연결되어 있다. 시뮬레이터의 모듈간의 연결 관계는 〈그림 4〉와 같다. 위와 같은 연결관계 속에서 본 시뮬레이터는 0.1초의 단위시간 마다 다음과 같은 과정을 반복 수행하게 된다.

1) 네트워크 정보 갱신 과정

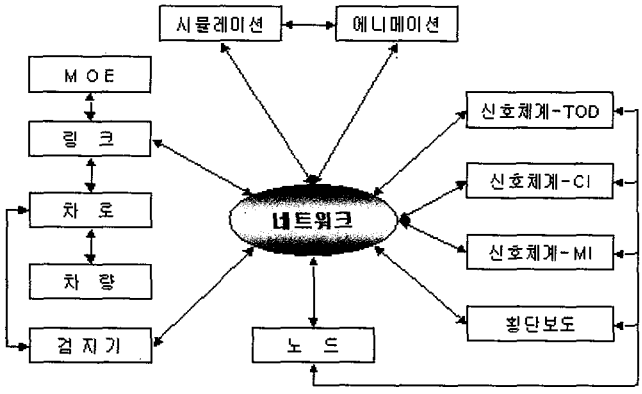
(1) 차량정보 갱신

차량의 주행 상태는 선두차량, 앞 차량을 추종 중인 차량, 차로변경을 진행하고 있는 차량, 그리고 교차로내부에서 주행 중인 차량으로 구분되며, 주행상태별 차량정보의 처리 절차는 〈표 3〉과 같다.

(2) 검지기 정보 갱신

시뮬레이터에서는 교차로의 각 차로별로 검지기를 설치할 수 있다. 시뮬레이션 수행시 네트워크의 소통자료는 네트워크 내의 모든 검지기를 대상으로 수집된다. 교차로의 소통자료는 시뮬레이션 수행시 개별차량의 궤적정보를 이용하여 검지 영역내의 ON/OFF 상황을 판단하고, 검지기의 점유시간과 비 점유시간을 산출한다. 개별 검지기별 통과 교통량, 지점속도, 그리고 점유율은 루프검지기의 계산방식과 동일한 방법으로 산출한다.

시뮬레이션 수행과정	• Network 갱신(update/0.1sec) • 신호현시 갱신(update/sec) • Graphic/Animation 갱신(update/0.1sec) • MOE 산출(사용자가 지정한 시간단위)



〈그림 4〉 주요 구성 모듈간의 관계

〈표 3〉 차량의 주행상태 별 차량정보의 처리 절차

차량 구분	차량정보의 처리 절차
case 1 (선두차량)	• 전방 신호등에 반응하여 가감속 결정 • 차로변경 여부 판단
case 2 (앞차량을 추종중인 차량)	• 차량추종 모델에 의하여 가감속 결정 • 차로변경 여부 판단
case 3 (차로변경을 진행하고 있는 차량)	• 차량추종 모델에 의하여 가감속 결정 • 차로 변경중의 이동되는 위치 결정
case 4 (교차로 내에서 진행 중인 차량)	• 전방향별로 회전각도/속도 결정 • 차로 i 내의 모든 차량 위치정보 갱신

2) 신호시간 정보 갱신과정

대상 네트워크의 모든 교차로는 각각의 신호체계를 가지고 있다. 본 시뮬레이터에서는 교차로를 중요교차로(CI), 준중요교차로(SCI), 비중요교차로(MI), 그리고 횡단보도의 4개 형태의 교차로로 구분한다. 각각의 교차로는 해당 현시결정 방법에 따라 연결된 모든 접근로에 대하여 녹색, 적색, 황색의 신호를 연결된 접근로에 대하여 결정하게 된다.

3) 화면표출 및 애니메이션 정보 갱신과정

대상 네트워크내에서 주행 중인 모든 차량은 자신의 현재 위치에 따라 0.1초마다 그래픽상의 *x*좌표와 *y*좌표를 갱신하게 된다. 이러한 정보를 통하여 화면표출 및 애니메이션을 처리하는 모듈에서는 0.1 초 단위로 모니터 상의 차량들의 위치를 갱신해주게 된다. 이를 빠른 시간에 반복하면 차량이 주행하는 것처럼 화면에 표출되어 진다.

4) 효과척도(MOE) 산출 과정

기본적으로 모든 차량들은 하나의 링크에 진입하는 순간의 시각과 링크를 벗어나는 시점의 시각을 저장하고, 그리고 링크 내에서 정지하고 있었던 시간의 합을 계산하게 된다. 이 정보들을 통하여 링크 여행시간, 주행시간, 여행속도, 정지지체 등을 산출하게 된다.

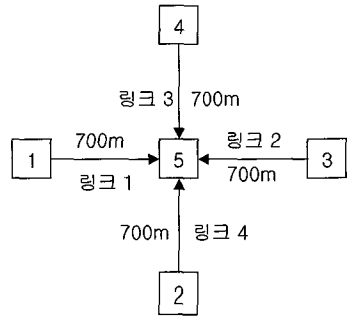
Ⅳ. 시뮬레이터의 신뢰성 분석 및 제어전략 평가

1. 시뮬레이터의 신뢰성 분석

본 연구에서는 시뮬레이터의 운영효과를 평가하기 위하여 상용 프로그램인 NETSIM을 이용하였다. 본 분석은 각 교차로에 연결된 4개의 유입 링크에 15분 간격으로 교통량을 조절하여 서로 다른 교통량 조건(교통량이 점점 증가하는 경우, 감소하는 경우 등)을 설정하고 각 링크별로 산출되는 효과측도(주행시간 및 정지지체 등)를 본 시뮬레이터와 NETSIM을 비교하여 본 시뮬레이터의 성능 및 산출 효과척도의 신뢰성을 평가하였다. 신뢰성 평가를 위한 평가대상 네트워크의 구조와 입력 조건은 〈그림 5〉와 〈표 4〉와 같다.

1) 신뢰성 분석결과 1: 링크별 평균주행속도 및 정지지체 비교 결과

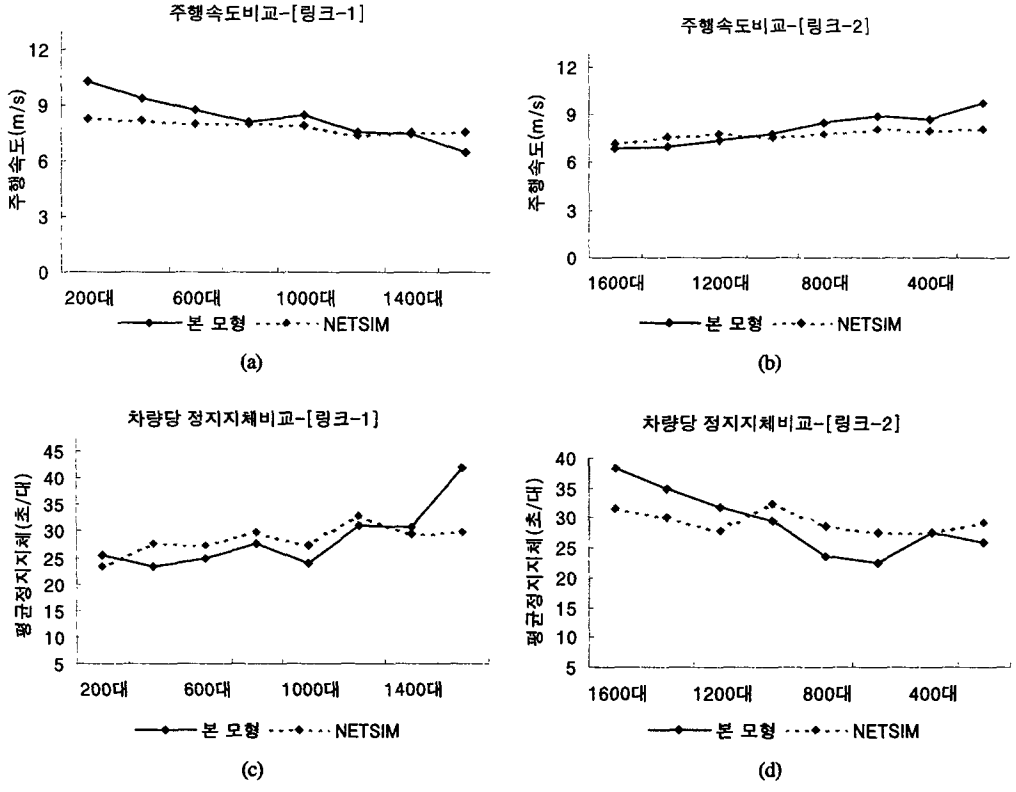
링크별 평균 주행속도(m/s)와 차량당 평균 정지지체(초/대)에 대한 NETSIM과 본 시뮬레이터의 산출결과는 〈그림 6〉과 같다. 분석결과, NETSIM과 제안 모형에서 산출한 링크별 평균주



〈그림 5〉 평가네트워크 구조

〈표 4〉 입력 교통량 조건

총 평가시간	7200초 (2시간)			
신호체계	TOD(4현시)			
주기	녹색시간			
120초	20	40	20	40
	입력교통량(대/시)			
시간대	링크 1	링크 2	링크 3	링크 4
0:00-0:15	200	1600	1600	200
0:15-0:30	400	1400	1400	400
0:30-0:45	600	1200	1200	600
0:45-1:00	800	1000	1000	800
1:00-1:15	1000	800	800	1000
1:15-1:30	1200	600	1000	800
1:30-1:45	1400	400	1200	600
1:45-2:00	1600	200	1400	400



〈그림 6〉 링크별 주행 속도 및 평균 정지지체 분포 비교결과

행속도의 분포는 입력교통량 조건에 따라 약간의 차이를 보이고 있으나, 링크 1과 2 모두 전체적으로 입력 교통량이 증가함에 따라 주행속도는 감소하는 경향을 나타내고 있다 (그림 a와 b 참조). 제안 모형과 NETSIM에서 산출된 차량당 정지지체는 링크 1, 2 모두 입력 교통량이 증가함에 따라 증가하는 경향을 나타내고 있다(그림 c와 d 참조).

2) 신뢰성 분석 결과 2: 링크별 평균주행속도 및 정지지체 비교 결과에 대한 Paired t-test 검정 결과(α = 0.05)

제안 모형의 산출 결과의 신뢰성은 Paired t-test로 검정하였다. 검정 결과, 제안모형의 결과로 산출된 차량 당 평균 정지지체와 주행속도는 유의수준 0.05에서 NETSIM의 산출결과와 동일하다는 결론을 내릴 수 있었다. 평가 결과에 대한 통계적 비교 결과는 〈표 5〉와 같다.

〈표 5〉 Paired t-test 결과

	주행속도(km/시)		평균 정지지체(초/대)	
	본모형	NETSIM	본모형	NETSIM
링크-1	8.27	7.86	28.66	28.41
t-통계량	1.2603		0.1352	
기각역 (양측, α = 0.05)	2.3646		2.3646	
링크-2	8.10	7.76	29.25	29.36
t-통계량	1.2221		-0.0626	
기각역 (양측, α = 0.05)	2.3646		2.3646	
링크-3	7.59	7.67	30.58	29.33
t-통계량	-0.3387		0.4726	
기각역 (양측, α = 0.05)	2.3646		2.3646	
링크-4	8.63	8.07	26.41	28.10
t-통계량	3.5537		-1.8904	
기각역 (양측, α = 0.05)	2.3646		2.3646	

〈표 6〉 적용알고리즘

구분	항목
신신호 시스템의 적용 알고리즘	- 포화도(DS) 산출 알고리즘 - 주기산출 알고리즘 - 포화도비를 이용한 현시분할 알고리즘 - 좌회전 감응제어 알고리즘

2. 독립교차로에서의 신호제어전략 평가

1) 평가 목적 및 대상

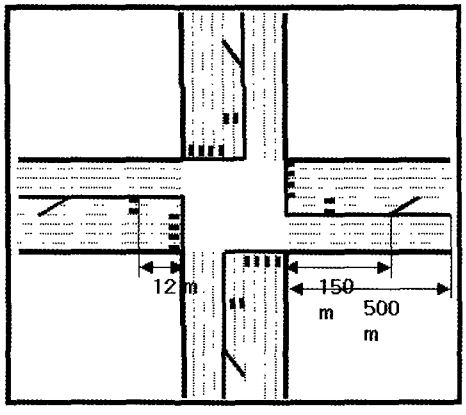
독립교차로의 신호제어전략은 중요 교차로에서 운영되는 신호제어 알고리즘의 운영결과를 대상으로 평가하였다. 신호제어 전략은 정주기 제어(TOD) 전략과 교통 대응제어 전략을 대상으로 비교분석하였다. 본 분석의 정주기 신호시간은 TRANSYT-7F를 이용하여 산출하였으며, 교통대응 신호시간은 시뮬레이터에 탑재된 검지기를 통하여 수집되는 소통자료를 토대로 교통대응 알고리즘에 의하여 주기별로 자동 산출하도록 하였다. 본 분석에 적용한 교통대응 신호제어 알고리즘은 다음과 같다.

2) 평가환경 및 조건

본 실험에서는 각 링크별로 서로 다르게 변화하는 교통량 조건을 입력하고 전체적인 교통량 상황을 점차 증가시켜가면서 신호제어의 효과를 측정하였다.

3) 평가결과

본 실험의 수행한 결과 대응제어와 감응제어를 실시한 효과는 〈표 7〉과 〈그림 8, 9〉에서 보

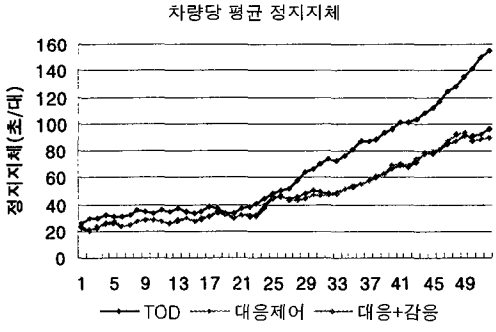


링크 방향		동	서	남	북
교통상황	시간	교통량			
비포화 시 (1800대/시)	-0:15	1500	1800	1400	1600
	-0:30	1700	1800	1500	1700
	-0:45	1900	1800	1800	1800
근포화 시	-1:00	2100	1800	2500	1900
	-1:15	2300	2500	2600	2100
	-1:30	2500	2500	2700	2300
과포화 시 (2500대/시)	-1:45	2600	2500	2500	2600
	-2:00	2600	2500	2200	3000

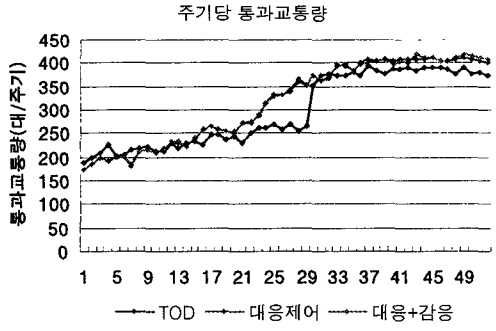
〈그림 7〉 독립교차로 실험 네트워크 및 입력자료

〈표 7〉 평가 결과

전략별 / 효과척도 / 교통상황	TOD		대응제어		대응+감응	
	통과교통량 (대/15분)	정지지체 (초/대)	통과교통량 (대/15분)	정지지체 (초/대)	통과교통량 (대/15분)	정지지체 (초/대)
비포화	220	33.2	217	21.7	268	26.4
근포화	295	55.4	333	42.6	332	41.6
과포화	384	119	405	73.5	405	74.7



〈그림 8〉 차량당 평균정지지체 비교



〈그림 9〉 주기당 통과교통량 비교

는 바와 같이 비포화시와 근포화시에 보다 높게 나타나는 것으로 판단된다. 그러나 과포화시 시뮬레이터의 의무적 차로변경의 실패율이 높아져 실제로 좌회전 차량수가 크게 감소하는 현상이 발생하였다. 따라서 이에 대한 추가적인 연구가 필요할 것으로 판단된다.

3. 네트워크에서의 신호제어전략 평가

1) 평가 목적 및 대상

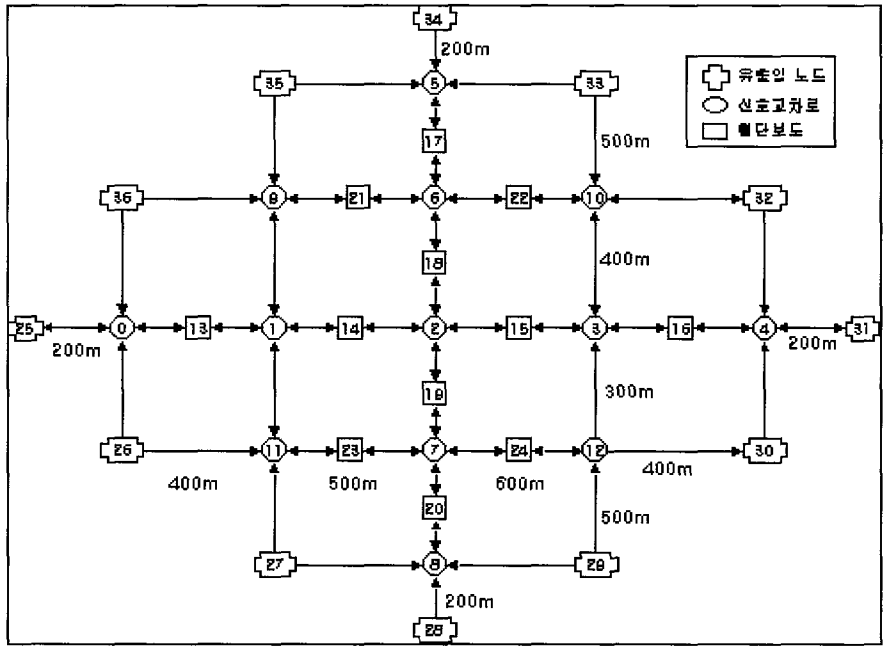
네트워크의 신호 운영효과는 교통축의 연동방향에 의해 크게 좌우된다. 교통 대응제어 전략에서 비중요교차로의 신호시간과 교통축의 연동방향은 중요교차로의 소통상황을 중심으로 결정된다. 즉, 중요교차로의 신호시간은 실시간 시뮬레이션 자료에 의하여 신호주기별로 계산되며, 비중요교차로의 신호시간 및 연동방향, 연동값은 중요교차로의 소통상황을 토대로 패턴을 선택하여 신호를 운영한다. 네트워크 분석에 적용한 적용알고리즘 및 효과측도는 다음과 같다.

2) 평가환경 및 조건

실험 네트워크는 〈그림 10〉과 같이 13개의 신호교차로와 12개의 횡단보도, 12개의 유출입 노드로 구성하였으며, 입력교통량 조건은 〈표 9〉와 같다.

〈표 8〉 적용알고리즘 및 효과 척도

효과척도	적용 알고리즘
- 교차로 통과교통량 - 차량당 평균 정지지체	- MI의 Pattern Selection 기능 - 옵셋결정 알고리즘 - 옵셋전이 알고리즘



〈그림 10〉 TOD 및 대응제어 실험네트워크

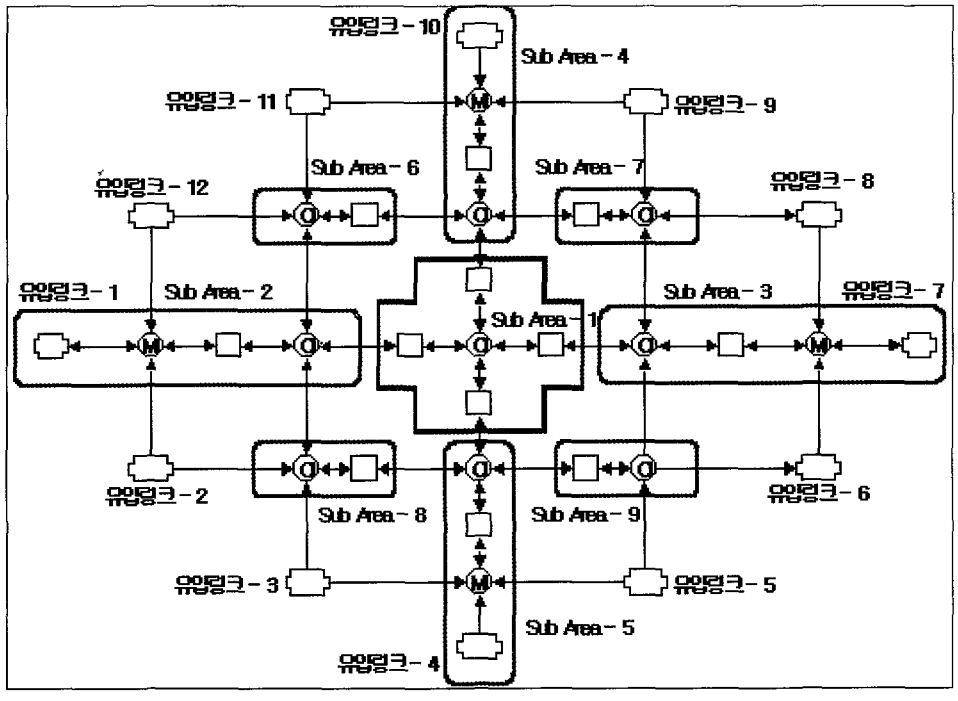
정주기 제어와 대응제어의 운영효과는 동일 실험네트워크에서 동일 교통량조건으로 평가하였다. 이때 정주기 제어의 경우는 모든 교차로가 정주기 방식으로 운영되며, 대응제어는 〈그림 11〉과 같이 교차로 운영 그룹(SA)을 구성하여 운영하였다. 대응제어는 9개의 운영 그룹으로 이루어지며, 9개의 중요교차로(CI) 및 4개의 비중요교차로(MI)로 구성하였다. 중요교차로(CI)는 실시간으로 수집되는 검지기 데이터를 이용하여 신호시간 및 옵셋 패턴을 결정하여 운영된다.

2) 평가결과

정주기 제어와 대응제어 비교 평가는 정지지체를 이용하여 실시하였으며, 13개 교차로중 1~3번 교차로의 주기당 통과교통량 및 차량당 정지지체는 〈표 10〉과 〈그림 12-17〉과 같다. 분석결과 모든 교차로의 총 통과교통량은 정주기 제어(TOD) 보다 대응제어의 경우가 2.6% 증가하

〈표 9〉 유입링크별 입력 교통량 조건

시간 \ 유입방향	동	서	남	북
0:00-0:15	1600	1600	800	800
0:15-0:30	1600	1600	800	800
0:30-0:45	1600	1600	800	800
0:45-1:00	800	800	1600	1600
1:00-1:15	800	800	1600	1600
1:15-1:30	800	800	1600	1600
1:30-1:45	800	800	1600	1600
1:45-2:00	800	800	1600	1600
회전비율	직진: 0.9, 좌회전: 0.1	직진: 0.9, 좌회전: 0.1	직진: 0.9, 좌회전: 0.1	직진: 0.9, 좌회전: 0.1

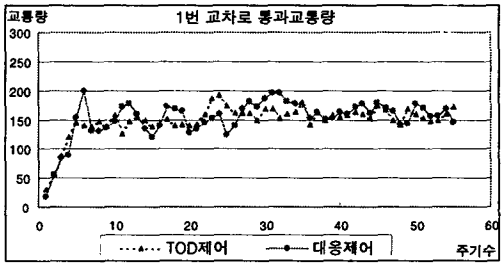


〈그림 11〉 대응제어의 SA 구성

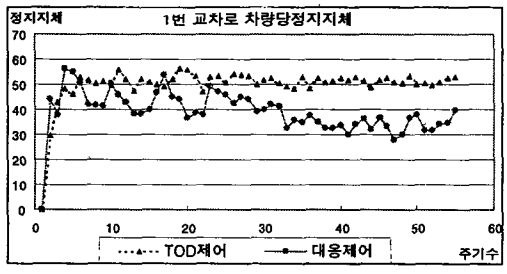
여 큰 차이가 없는 것으로 나타났다. 차량당 정지지체의 경우는 4번과 5번 교차로에서 각각 2.2%, 9.0%의 증가를 보였으나 나머지 교차로에서는 크게 감소하였음을 볼 수 있다. 이러한 결과는 정주기 제어가 미리 설정된 신호계획에 의해 운영됨에 따라 실험 후반부의 교통량 변화

〈표 10〉 교차로별 통과교통량 및 정지지체

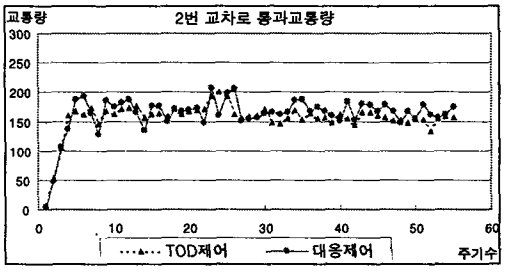
교차로 번호	총통과교통량(대)			차량당정지지체(초)		
	TOD	대응	증감율(%)	TOD	대응	증감율(%)
0	8970	8866	-1.2	32.54	31.11	-4.4
1	8221	8412	2.3	50.97	39.49	-22.5
2	8588	8947	4.2	48.28	36.12	-25.2
3	7185	7280	1.3	50.59	41.78	-17.4
4	8895	8863	-0.4	29.54	30.19	2.2
5	9196	9617	4.6	30.09	32.81	9.0
6	7896	8272	4.8	37.24	30.71	-17.9
7	7855	8224	4.7	67.76	33.77	-50.2
8	9296	9588	3.1	51.88	38.15	-26.5
9	7517	7867	4.7	81.17	35.51	-56.3
10	6645	6728	1.2	84.91	35.31	-58.4
11	7450	7669	2.9	79.93	35.80	-55.2
12	6368	6505	2.2	62.39	40.37	-35.3
계	104082	106838	2.6	52.90	35.29	-33.3



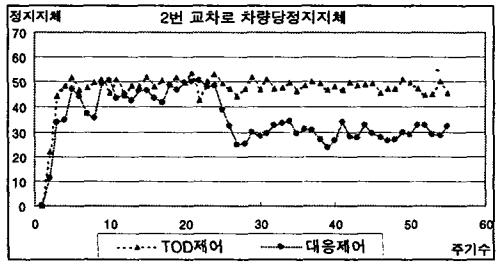
〈그림 12〉 1번교차로의 주기당 통과교통량



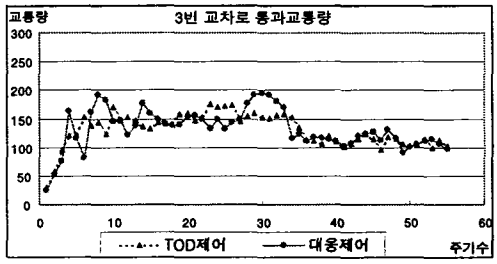
〈그림 13〉 1번 교차로의 차량당정지지체



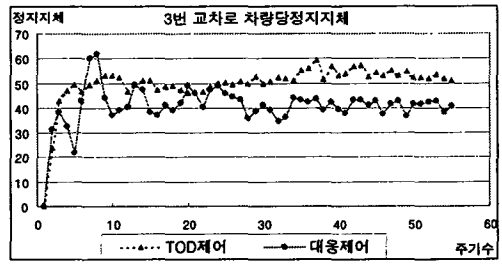
〈그림 14〉 2번 교차로의 주기당 통과교통량



〈그림 15〉 2번 교차로의 차량당정지지체



〈그림 16〉 3번 교차로의 주기당 통과교통량



〈그림 17〉 3번 교차로의 차량당정지지체

에 대응하지 못하여 현시배분 및 연동값의 선택이 어긋남에 따라 대응제어보다 지체가 크게 증가 한 것으로 판단된다.

Ⅴ. 결 론

본 연구에서는 다양한 신호제어 전략을 평가할 수 있는 미시적 시뮬레이터를 개발하고, 이를 이용하여 교통대응 신호 시스템(COSMOS)의 신호전략 들을 평가하였다.

본 연구를 통하여 개발된 시뮬레이터는 차량발생, 차량추종, 차로변경 및 선두차량의 신호반응 등의 기본적인 차량의 행태에 대해 현실적인 묘사가 가능하다. 개별차량의 초기 속도 및 가/감속도는 현장 상황의 운전자 특성을 반영하기 위해서 정규분포에 의해 산출되는 값을 사용하였다. 시뮬레이션 과정에서 네트워크의 노드-링크 체계는 교차로로 지정된 각 노드들을 사용자가 원하는 타입의 신호체계로 지정하여 다양한 신호전략을 모의실험 할 수 있도록 하였고, 15분 간격의 유입교통량을 입력하여 현실적으로 변화하는 교통상태를 묘사할 수 있도록 하였다.

교통대응 신호체계의 제어전략은 본 연구에서 개발된 시뮬레이터를 이용하여 평가하였다. 평가 결과, 교통대응 신호제어 전략의 운영 효과가 통과교통량 및 평균 지체시간의 측면에서 정주기 제어전략에 비하여 우월한 것으로 분석되었다. 이는 교통대응 또는 교통감응 제어전략이 교통수요에 부응한 신호시간을 산출할 수 있기 때문인 것으로 판단된다.

본 연구에서 개발된 시뮬레이터는 초기 수준의 모형으로서 다양한 현장상황을 모사하기에는 한계가 있다. 따라서 시뮬레이터의 현장 적용성을 높이기 위해서는 개별 운전자의 주행행태를 모형에 반영하기 위한 연구가 지속적으로 수행 되어야 할 것이다. 또한 교통대응 신호시스템의 경우에도 현장 교통상황을 고려한 검지정보의 처리 알고리즘, 교차로 기하구조 및 서로 다른 개별 차량 행태 문제, 통신 상의 문제, 신호 시스템 장애 발생시의 대응 방안 등에 대한 기능 보완이 지속적으로 이루어 져야 할 것이다.

참고문헌

1. 도철웅, "교통공학 원론(상)", 청문각
2. 이영인, 한동희. "첨단 신호시스템의 신호제어전략 평가를 위한 미시적 시뮬레이터의 개발," 대한 교통학회지 제21권 제2호(통권67호). pp. 83-94, 대한교통학회, 2003. 4.
3. 이영인, 김이래. "교통대응 신호제어전략의 평가를 위한 미시적 시뮬레이터의 원형 개발." 대한교통학회지 제19권 제6호(통권58호). pp. 143-160, 대한교통학회, 2001. 12.
4. 안계형. "과포화 교통상태에의 적용을 위한 시뮬레이션 모형 개발", 한국시뮬레이션학회지, 1997.
5. 이영인. "도시교통류 미시적 시뮬레이션 모형의 개발", 서울시립대 도시과학연구원, 1999.
6. Averill M.Law & W.David Kelton, SIMULATION MODELING AND ANALYSIS , McGraw-Hill, Inc. 1991.
7. Aycin, Benekohal, 2000, "Analysis of Stability and Performance of Car-following Models in Congested Traffic", TRB 79th Annual Meeting.
8. Edited by Dietrich Stauffer, Kai Nagel, Jorg Esser, and Marcus Rickert, 1999, "Large-scale traffic simulation planning" to appear in: Annual Review of Computational Physics Ⅶ, World Scientific Publishing Company.
9. C. J. Messer et al, "Traffic Flow Theory", TRB SPECIAL REPORT 165.
10. M Rickert, K Nagel, M Schreckenberg, A Latour, physica A. Copyright by Elsevier. "Two lane traffic simulations using cellular automata".
11. Edited by Dietrich Stauffer, Kai Nagel, Jorg Esser, and Marcus Rickert, 1999, "Large-scale traffic simulation planning" to appear in: Annual Review of Computational Physics Ⅶ, World Scientific Publishing Company.