

승객 유동을 고려한 DEVS 기반 자기부상열차 운행 시물레이션

차무현^{1†} · 이재경¹ · 백진기²

An Operation Simulation of MAGLEV using DEVS Formalism Considering Traffic Wave

Moo Hyun Cha · Jai Kyung Lee · Jin Gi Beak

ABSTRACT

The MAGLEV (Magnetically Levitated Vehicle) system, which is under commercialization as a new transportation system in Korea, is operated by means of unmanned automatic control system. Therefore the plan of train operation should be carefully established and validated in advance. In general, when making the train operation plan, the statistically predicted traffic data is used. However, traffic wave can occur when real train service is operated, and the demand-driven simulation technology is required to review train operation plans and service qualities considering traffic wave. This paper presents a method and model to simulate the MAGLEV's operation considering continuous demand changes. For this purpose, we employed the discrete event model which is suitable for modeling the behavior of railway passenger transportation, and modeled the system hierarchically using DEVS (Discrete Event System Specification) formalism. In addition, through the implementation and experiment using DEVSim++ simulation environment, we tested the feasibility of the proposed model and it is also verified that our demand-driven simulation technology could be used for the prior review of the train operation plans and strategies.

Key words : Discrete Event Simulation, Train Operation Simulation, Traffic Wave, DEVS

요약

새로운 교통수단으로서 실용화가 진행 중인 자기부상열차 시스템은 무인자동운전으로 운영되며, 사전에 면밀한 열차운행 계획의 수립과 검증이 요구된다. 일반적으로 열차운행 계획을 수립할 때는, 노선계획에 따른 통계화된 수송수요 예측자료를 활용하지만, 실제 열차 운행 시에는 수송 파동이 생길 수 있으며, 이를 고려하여 열차 운행 계획과 서비스 품질을 사전에 검토할 수 있는 수요기반 시물레이션 기술이 요구된다. 본 연구에서는 연속적인 수요파동을 반영할 수 있는 자기부상열차 운행 시물레이션 방법 및 모델을 제시하였다. 이를 위해, 열차 운행에 따른 승객 수송과정을 가장 효과적으로 모사할 수 있는 이산사건 모델을 적용하였으며, DEVS 형식론을 이용하여 시스템을 체계적으로 모델링 하였다. 또한, DEVSim++ 시물레이션 환경을 이용한 구현과 실험을 통해, 제한한 모델에 대한 구현 가능성을 검증하였고, 이러한 수요기반 시물레이션을 이용하면 운행계획과 전략에 대한 효과적인 사전 검토가 가능함을 확인하였다.

주요어 : 이산사건 시물레이션, 열차운행 시물레이션, 수송 파동, DEVS

1. 서론

접수일(2011년 3월 15일), 심사일(1차 : 2011년 8월 12일),

게재 확정일(2011년 8월 12일)

¹⁾ 한국기계연구원 시스템신뢰성연구실

²⁾ 한국기계연구원 도시형자기부상열차 실용화 사업단

주 저 자 : 차무현

교신저자 : 차무현

E-mail : mhcha@kimm.re.kr

자기부상열차(MAGLEV : Magnetically Levitated Vehicle)는 자석의 힘으로 부상하고 가이드웨이 상에서 자기장에 의해 일정한 공극을 유지한 채 선형전동기로 추진하는 새로운 개념의 교통수단이다. 우리나라는 최근 도시형 자기부상열차 실용화 사업을 통해, 무인자동운전이 가능한 시

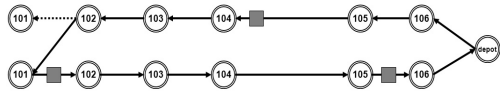
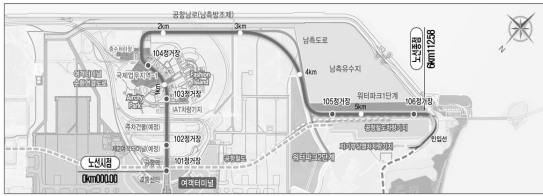


그림 1. 도시형 자기부상열차 시범노선 계획(인천국제공항)

속 110 km/h급의 자기부상열차를 개발하고, 인천국제공항 내에 시범노선을 건설하여 상업운전을 계획하고 있다^{1,2)}. 시범노선은 그림 1과 같이 교통센터, 여객터미널, 국제업무지구, 용유복합역 등을 경유하여 총 6.1 km 연장의 복선으로 건설되며, 6개의 역사가 설치될 계획이다.

자기부상열차는 기본적으로 무인 자동운전 방식으로 운영되기 때문에, 사전에 면밀한 열차운행 계획의 수립과 검증이 요구된다. 일반적으로 열차운행 계획을 수립할 때는, 열차 운행 스케줄을 시간에 따른 열차 이동거리로 도표화한 DIA(Train Diagram의 약칭)가 주로 사용된다. DIA 설계 시에는, 노선계획에 따른 수송수요 예측자료와, 선로와 차량 제원을 기반으로 동적 성능을 분석하는 열차 성능시뮬레이션(TPS, Train Performance Simulation) 자료를 바탕으로 상세한 열차 운행 스케줄을 수립하게 된다. 이때, 수송수요는 1일 또는 혼합 시간대의 평균 등에 의해 통계화된 추정치를 이용하게 된다.

한편, 실제 열차 운행 시에는 수송수요에 대한 시간적, 시기적 파동이 있을 수 있으며, 이러한 수송 파동(Traffic Wave)은 계절, 요일, 주변 행사 등 다양한 요인에 의해 발생할 수 있다. 따라서, 통계화된 수요를 바탕으로 개발된 열차운행 계획에 대해, 실제계와 흡사한 연속적인 수송 파동을 반영하여, DIA 설계의 타당성을 검토하고, 열차 지연 또는 승객 대기시간 등의 수송 서비스에 대한 품질을 예측하는 수요기반 시뮬레이션 기술이 요구된다.

관련 연구를 살펴보면, 해외의 경우 OpenTrack¹⁰⁾, Bahn¹¹⁾, RailSys¹²⁾ 등과 같이, 철도 시스템을 구성하고 운행 스케줄 및 운영전략을 시뮬레이션 할 수 있는 다양한 툴들이 소개 되었다. 하지만 변동적인 수요파동이 시뮬레이션 과정에 직접 포함 되거나, 또는 시뮬레이션 결과를 평가하는데 사용되지 않았다¹³⁾. ITMS(Interstation Train Movement Simulation)는 선행열차와 후속열차의

간격제어를 수행하는 다양한 신호 시스템 전략에 대해 평가하고 이를 최적화하기 위해 개발된 열차운행 시뮬레이터이다¹⁴⁾. 이산사건 모델링과 객체지향 프로그래밍 등을 사용하여 개발되었지만, 위와 마찬가지로 연속적인 수요 파동은 고려되지 않았다. Pablo 등¹³⁾은 칠레 산티아고 지하철의 운영효율 증가를 위해, 승객 수요를 고려한 시뮬레이션을 수행하고, 제어 시스템 등의 운영전략을 평가하였다. 이 논문에서는 변화하는 승객 도착율에 따른 시스템 행동을 해석적으로 예측하기는 매우 어렵기 때문에, 열차 도착과 승객 탑승 등 주요 사건을 이벤트로 표현하고 이들 사건 사이의 거동을 시뮬레이션 하는 이벤트기반 동적 시뮬레이터(event-driven dynamic simulator)를 객체지향 언어를 사용하여 구현하였다. 하지만, 실제로 가속도를 포함하여 연속 모델로 표현되어야 하는 열차운행 모델의 경우, 항상 일정한 속력을 가지는 모델로 표현되었다. 또한, Paolucci 등¹⁵⁾은 복잡한 지하철 시스템을 모델링하기 위해 객체지향 방법론을 이산사건 모델링과 통합하고자 하였으며, 이를 이용하여 지하철 운행 스케줄과 신호제어 전략을 생성하는 시뮬레이터를 개발하였다. 이 연구는 일정한 승객 발생률을 기반으로 운행 스케줄을 생성하는 데 초점이 맞춰져 있으며, 시간적인 승객 수요의 변화는 고려되지 않았다.

국내의 경우, 경부선을 대상으로 열차운영계획의 수립과 시설투자를 위한 사전검토를 수행하기 위해, 다중열차주행 시뮬레이터가 개발되었으며³⁾, 이를 위해 다양한 시뮬레이션 모델을 검토하고, 인프라, 차량, 신호기, 시각표 등과 같은 철도 시스템 자원에 대한 모델을 제시하였다⁴⁾. 특히, TPS와 같이 운동방정식으로부터 열차 주행을 충실히 모의하기 위해서는 연속형(continuous) 모델을 사용하며, 열차의 도착과 출발 등 이벤트의 발생과 상태 천이만으로 열차 주행을 모의하기 위해서는 이산형(discrete) 모델이 필요함을 제시하였다. 하지만, 앞선 연구와 마찬가지로 수요 파동이 시뮬레이션에 포함되거나, 시뮬레이터 결과를 평가하기 위한 방법으로 이용되지 않았다. 이 밖에도, 도시철도 무인운영 시뮬레이터 설계방안⁵⁾, 경량전철의 무인운전 및 제어 시스템⁶⁾ 등과 같이 무인운전과 관련된 시뮬레이션 방법들이 연구되어 왔으나, 아직 초기 연구단계로서 승객 유동이 고려된 이산사건 모델링 및 시뮬레이션 기법의 적용은 이루어지지 않았다.

본 연구에서는, 무인자동운전으로 운영될 예정인 자기부상열차 시스템에 대해, 사전에 정밀한 운행계획의 수립과 검토를 위해서, 연속적인 수요파동을 고려한 열차운행 시뮬레이션 방법 및 모델을 제시하였다. 시스템 모델의

경우, 열차 운행에 따른 승객 수송과정을 가장 효과적으로 모사할 수 있는 이산사건 모델을 사용하였으며, 모델링 방법은 복잡한 시스템 모델에 대해 체계적인 수학적 형식론을 제공하고 객체지향적 프로그래밍이 가능한 DEVS 형식론¹⁷⁾을 적용하였다. 또한, 제시한 모델을 바탕으로 시범노선에 대한 열차 운전 시격, 정차 시간, 승객 대기 시간 등을 DEVSIM++¹⁸⁾환경을 이용해 시뮬레이션 하였고, 그 결과를 바탕으로 제안한 시뮬레이션 방법이 기존의 열차 운행 계획과 전략을 검토하는데 효과적으로 이용될 수 있음을 확인할 수 있었다.

2. 시스템 개요

본 연구에서는 서론에서 기술된 바와 같이, 총 6개의 역사(상행과 하행)로 이루어진 자기부상열차 시범노선에 대해, 수요예측에 기반하여 각 역사 별로 연속적인 수요 파동을 생성하고, 기존에 설계된 열차운행계획(DIA)를 바탕으로 총 4량의 열차를 운행하며, 각 열차의 역사 정차와 승객 도착 및 승하차를 시뮬레이션하고, 각각의 사건 발생 시간을 수집하고 통계하고자 한다.

그림 2에서는 열차의 이동과 승하차의 관계를 개념적으로 도식화하였고, 본 시스템에서의 중요 모델은 차량(Train), 역사(Station) 및 승객(Passenger)이 될 수 있음을 확인할 수 있다. 차량의 경우, 신호제어에 의해 지정된 궤도상에서 운행하며, 역사 도착, 출발과 같은 이산 사건이 발생하게 된다. 또한, 그림 3과 같이 열차가 도착하면, 출입문이 열리고 승객 하차와 승차가 이루어지며, 지정된

대기시간 이후에 출발하게 된다. 이러한 승하차 과정 역시 이산 사건으로 모델링 될 수 있다. 그리고, 각 역사에 승객이 진입하는 사건과 열차를 대기하는 과정은 랜덤 프로세스와 큐잉으로 모델링 될 수 있으며, 이러한 철도의 운행과 승객 승하차를 포함하는 시스템 모델링 방법은 이산사건 모델이 매우 적합하다¹⁶⁾.

그림 4에서는 전체 시뮬레이션 시스템의 개요를 다이어그램으로 나타내었다. 먼저 지정된 운행 정보에 기반하

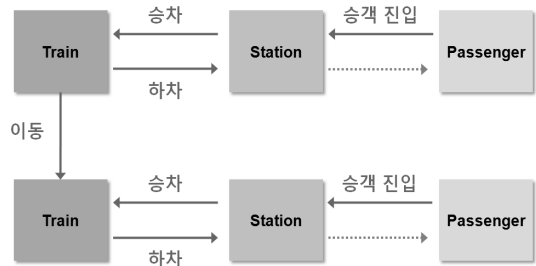


그림 2. 열차 이동에 따른 승객 승하차 개념 모델

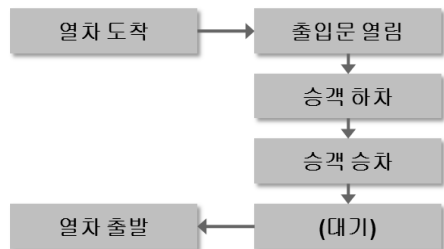


그림 3. 열차 도착에 의한 승객 승하차 과정

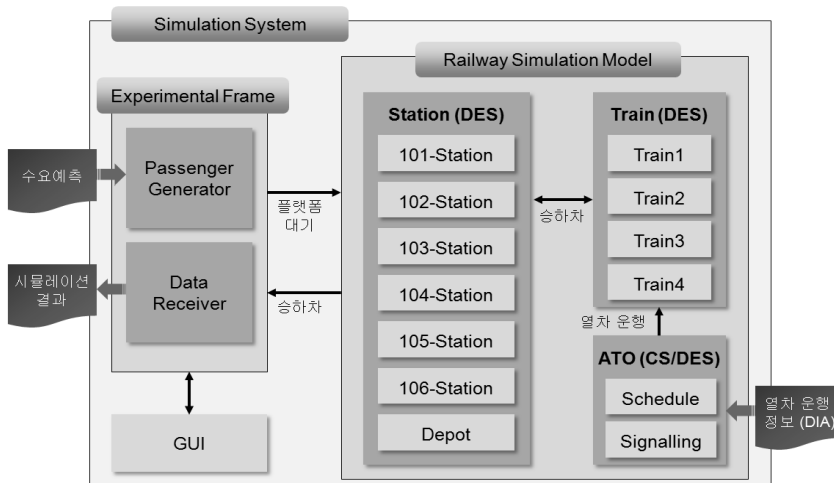


그림 4. 전체 시뮬레이션 시스템의 개요

여 무인자동운행을 수행하고, 역사 별 승객 승하차를 시뮬레이션 하는 모델을 Railway Simulation Model로 정의하였다. 이 모델에는 스케줄링과 신호제어를 담당하는 ATO(Automatic Train Operation)와 승객 승하차와 관련된 메시지를 송수신 하는 Train 및 Station이 포함된다. Train 및 Station의 동작은, 모두 이산 사건 입출력에 따른 내부 상태의 변화로 표현될 수 있기 때문에 DES(Discrete Event System) 모델이 적합하며, ATO의 경우, 이론적으로 CS(Continuous System)로 모델링 할 수 있으나, 실제 시스템 구현에서는 일정한 주기로 열차 위치를 갱신하는 DES 모델링이 가능하다.

또한, 수요예측에 기반하여 랜덤 프로세스로 진행되는 승객 생성 모델링을 위해 Passenger Generator가 필요하며, 열차의 운행과 승객 승하차와 관련된 모든 데이터를 수집하고 통계하기 위한 Data Receiver가 필요하다. 이들은 앞선 Railway Simulation Model에 대해, 다양한 입력을 생성하고 그 출력을 분석할 수 있는 실험 장치(Experimental Frame) 모델로 추상화 할 수 있다. 또한, 이는 전체 시뮬레이션 과정과 데이터를 모니터링 할 수 있는 GUI 시스템과 연동할 수 있다.

3. DEVS 기반 시스템 모델링

Zeigler^[17]에 의해 제안된 DEVS(Discrete Event System Specification) 형식론은 이산사건 시스템을 모듈 별로 나누고, 이를 계층적으로 모델링 할 수 있는 수학적인 틀을 제공한다. DEVS 형식론은 집합이론에 기반하여 시스템의 상태방정식을 표현하며, 원자 모델과 결합 모델을 사용하여 구조화된 모델링이 가능하다. 시스템의 동작은 각 원자 모델들이 서로 사건을 주고 받으며 시간에 따른 각각의 상태전이 과정으로 표현되며, 결합 모델은 각 구성요소 모델간의 사건 전달을 담당한다. DEVS 형식론에 의해 원자 모델은 다음과 같이 3개의 집합과 4개의 함수로 표현되며, 시간이 진행됨에 따라 내부상태가 변화하거나 외부의 입력에 의해 내부 상태가 바뀌는 이산사건 시스템의 동작을 체계적인 의미론으로 기술한다.

- $M = \langle S, X, Y, \delta_{int}, \delta_{ext}, \lambda, ta \rangle$
- S : 일련의 상태 집합
- X : 입력 사건의 집합
- Y : 출력 사건의 집합
- δ_{int} : 내부 상태전이 함수 ($S \rightarrow S$)

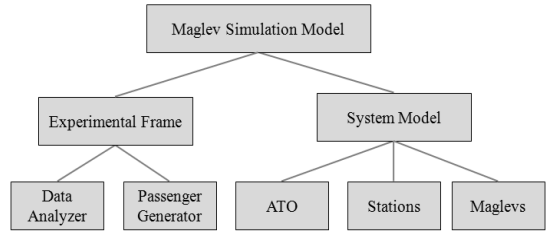


그림 5. System Decomposition Tree

δ_{ext} : 외부 상태전이 함수 ($Q \times X \rightarrow Q$)

λ : 출력함수 ($Q \rightarrow Y$)

ta : 시간 진행 함수 ($S \rightarrow Real$)

$Q = \{(s,e) | s \in S, \text{ and } 0 \leq e \leq ta(s)\}$

: 시간이 포함된 M의 상태

본 연구에서는 DEVS형식론을 이용하여 이산사건 시스템으로 표현되는 자기부상열차 운행 및 승하차 시스템을 모델링 하고자 한다. 이를 위해 먼저 시스템 DT (decomposition tree)를 그림 5와 같이 구성하였다. 전체 시스템 모델은 101~106역사(상행6, 하행5, 총 11역사) 및 각 역사의 승객 발생기, 그리고 총 4편성의 열차로 구성된다. 그리고, 이 구조로부터 DEVS로 표현되는 원자 모델과 결합 모델을 정의하고, 이들의 구조와 메시지 교환 관계를 정의하였다. 그림 6은 본 시스템 모델을 DEVS 형식론으로 표현하기 위한 전체 다이어그램을 보여준다. 전체 시스템 모델은 결합 모델인 CCplMaglevSimulation이며, CCplExpFrame과 CCplSystemModel로서 구성된다. 또한, 그 하위에는, 같은 성격을 가지는 원자 모델들을 하나의 결합모델로 표현한 CCplGenerators, CCplStations, CCplMaglevs에 의해 구조화된다.

먼저 자기부상열차를 표현하는 원자 모델인 CAtmMaglev는 운행중(Driving), 출입문 열림(Open), 승객하차(GetOff), 승객승차(GetOn), 대기(Idle), 출입문 닫음(Close) 등 6개의 시스템 상태로 표현될 수 있다. 운행중 상태에서는 매 주기마다 업데이트되는 ATO로부터 현재 위치를 갱신하게 되며, 역사에 도착하게 되면 도착 출력(Arr)을 생성하고, 시스템 상태를 Open으로 변경한다. 출입문이 열리면, 승객 하차 출력(Out)을 생성하고, 모든 승객이 하차한 후 새로운 승객(In)을 입력 받는다. 모든 승객이 승차한 후, 대기과정을 거쳐 출입문을 닫고 출발하게 되면, 출발 출력(Dep)을 생성한다. 이때, 모든 하차가 이루어진 시점에서, 역사에게 승객 승차가 가능하다는 신호(Ready)를 전송하게 되고, 역사는 이 시점부터 승객을 방출하게 된다.

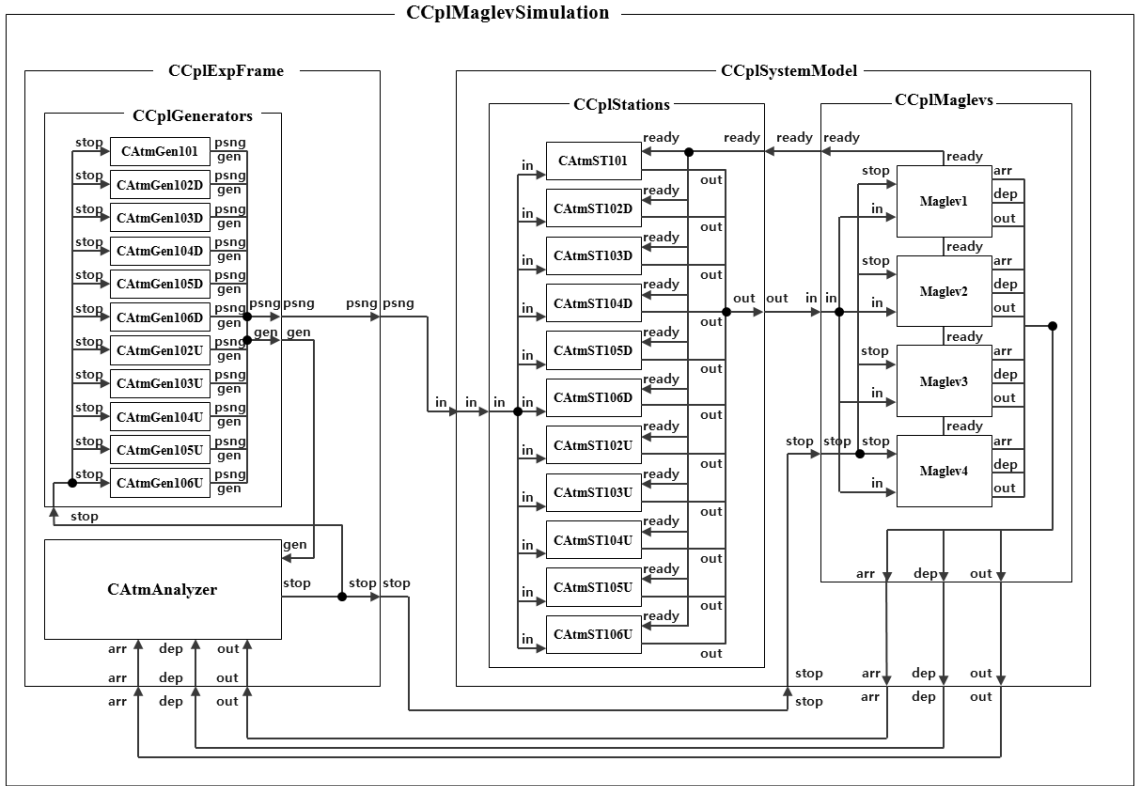


그림 6. DEVS 모델링을 위한 시스템 다이어그램

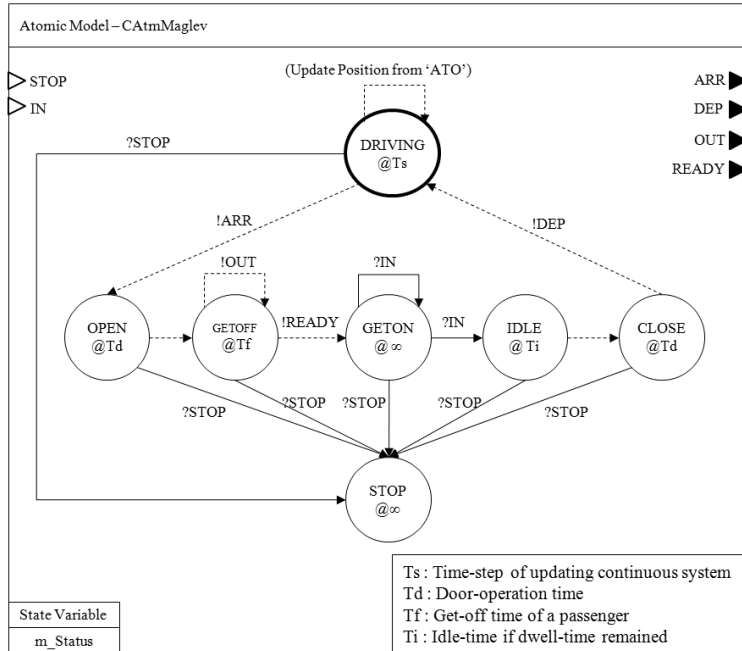


그림 7. 자기부상열차 원자 모델의 DEVS 다이어그램

그림 7에서는 이러한 승객 승하차 동작을 표현하는 DEVS 다이어그램을 나타내었다. 각 시스템의 상태는 시간 진행 파라미터와 함께 원기호로 표현되고, 내부 상태 천이는 점선, 외부 상태 천이는 실선으로 나타내었다. 외부 상태 천이를 유발하는 외부입력은 물음표 기호(?), 내부 상태 천이의 결과인 내부출력은 느낌표 기호(!)로서 표현된다. 이러한 시스템 동작을 바탕으로 아래와 같이 DEVS 형식론을 기술하였다.

$$\mathbf{CAtmMaglev} = \langle \mathbf{S}, \mathbf{X}, \mathbf{Y}, \delta_{\text{int}}, \delta_{\text{ext}}, \lambda, \mathbf{ta} \rangle$$

$$\mathbf{X} = \{\text{stop, in}\}$$

$$\mathbf{Y} = \{\text{arr, dep, out, ready}\}$$

$$\mathbf{S} = \{\text{DRIVING, OPEN, GETOFF, GETON, IDLE, CLOSE, STOP}\}$$

$$\delta_{\text{ext}} : \mathbf{Q} \times \mathbf{X} \rightarrow \mathbf{Q}, \mathbf{Q} = \{ (s,e) \mid s \in \mathbf{S} \text{ and } 0 \leq e \leq \mathbf{ta}(s) \}$$

$$\delta_{\text{ext}} ((\text{DRIVING}, t), \text{stop}) = (\text{STOP}, 0)$$

$$\delta_{\text{ext}} ((\text{OPEN}, t), \text{stop}) = (\text{STOP}, 0)$$

$$\delta_{\text{ext}} ((\text{GETOFF}, t), \text{stop}) = (\text{STOP}, 0)$$

$$\delta_{\text{ext}} ((\text{GETON}, t), \text{stop}) = (\text{STOP}, 0)$$

$$\delta_{\text{ext}} ((\text{IDLE}, t), \text{stop}) = (\text{STOP}, 0)$$

$$\delta_{\text{ext}} ((\text{CLOSE}, t), \text{stop}) = (\text{STOP}, 0)$$

$$\delta_{\text{ext}} ((\text{GETON}, t), \text{in}) = (\text{GETON}, 0) \text{ (if not last passenger of station)}$$

$$\delta_{\text{ext}} ((\text{GETON}, t), \text{in}) = (\text{IDLE}, 0) \text{ (if last passenger of station)}$$

$$\delta_{\text{int}} : \mathbf{Q} \rightarrow \mathbf{Q}$$

$$\delta_{\text{int}} (\text{DRIVING}, T_s) = (\text{DRIVING}, 0) \text{ (} T_s \text{ : time-step of updating continuous system)}$$

$$\delta_{\text{int}} (\text{DRIVING}, T_s) = (\text{OPEN}, 0) \text{ (if maglev arrived at a station)}$$

$$\delta_{\text{int}} (\text{OPEN}, T_d) = (\text{GETOFF}, 0) \text{ (} T_d \text{ : door-operation time)}$$

$$\delta_{\text{int}} (\text{GETOFF}, T_f) = (\text{GETOFF}, 0) \text{ (} T_f \text{ : get-off time of a passenger)}$$

$$\delta_{\text{int}} (\text{GETOFF}, T_f) = (\text{GETON}, 0) \text{ (if last passenger of maglev)}$$

$$\delta_{\text{int}} (\text{IDLE}, T_i) = (\text{CLOSE}, 0) \text{ (} T_i \text{ : idle-time if dwell-time remained)}$$

$$\delta_{\text{int}} (\text{CLOSE}, T_d) = (\text{DRIVING}, 0) \text{ (} T_d \text{ : door-operation time)}$$

$$\lambda : \mathbf{Q} \rightarrow \mathbf{Y}$$

$$\lambda (\text{DRIVING}, T_s) = \text{arr (if arrived)}$$

$$\lambda (\text{GETOFF}, T_f) = \text{out (if passengers in maglev)}$$

$$\lambda (\text{GETOFF}, T_f) = \text{ready (if no more passengers in maglev)}$$

$$\lambda (\text{CLOSE}, T_d) = \text{dep}$$

$$\mathbf{ta} : \mathbf{S} \rightarrow \mathbf{R}$$

$$\mathbf{ta}(\mathbf{S}) = \infty, \mathbf{S} = \{\text{GETON, STOP}\}$$

$$\mathbf{ta}(\mathbf{S}) = T_d, \mathbf{S} = \{\text{OPEN, CLOSE}\}$$

$$\mathbf{ta}(\text{GETOFF}) = T_f$$

$$\mathbf{ta}(\text{IDLE}) = T_i$$

역사 원자 모델인 CAtmStation의 경우, 승객의 입장과 플랫폼상의 승객 대기를 전형적인 큐잉모델로서 표현할 수 있다. 즉, 역사는 승객 생성기와 이를 수송하는 열차 사이의 버퍼역할을 하게 되며, FIFO(First In First Out) 동작을 표현하는 큐(Queue)로서 구현할 수 있다. 또한, 열차로부터 Ready 입력을 받으면, 대기 상태(Wait)에서 탑승 상태(Board)로 바뀌게 되며, 탑승출력(Out)이 발생한다. 모든 승객이 탑승하면, 다시 대기상태로 복귀하며, 승객 생성기로부터 출력되는 승객 도착 입력(In)은 대기 상태 및 탑승상태 모두 가능하도록 설계하였다. 역사에서의 승객 승하차 관련 DEVS 형식론은 아래와 같으며, 그림 8에서는 이를 DEVS 다이어그램으로 나타내었다.

$$\mathbf{CAtmStation} = \langle \mathbf{S}, \mathbf{X}, \mathbf{Y}, \delta_{\text{int}}, \delta_{\text{ext}}, \lambda, \mathbf{ta} \rangle$$

$$\mathbf{X} = \{\text{in, ready}\}$$

$$\mathbf{Y} = \{\text{out}\}$$

$$\mathbf{S} = \{\text{WAIT, BOARD}\}$$

$$\delta_{\text{ext}} : \mathbf{Q} \times \mathbf{X} \rightarrow \mathbf{Q}, \mathbf{Q} = \{ (s,e) \mid s \in \mathbf{S} \text{ and } 0 \leq e \leq \mathbf{ta}(s) \}$$

$$\delta_{\text{ext}} ((\text{WAIT}, t), \text{in}) = (\text{WAIT}, 0)$$

$$\delta_{\text{ext}} ((\text{WAIT}, t), \text{ready}) = (\text{BOARD}, 0)$$

$$\delta_{\text{ext}} ((\text{BOARD}, t), \text{in}) = (\text{BOARD}, 0) \text{ (continue existing internal transition)}$$

$$\delta_{\text{int}} : \mathbf{Q} \rightarrow \mathbf{Q}$$

$$\delta_{\text{int}} (\text{BOARD}, T_b) = (\text{BOARD}, 0) \text{ (} T_b \text{ : boarding time of a passenger)}$$

$$\delta_{\text{int}} (\text{BOARD}, T_b) = (\text{WAIT}, 0) \text{ (if no more passengers in station)}$$

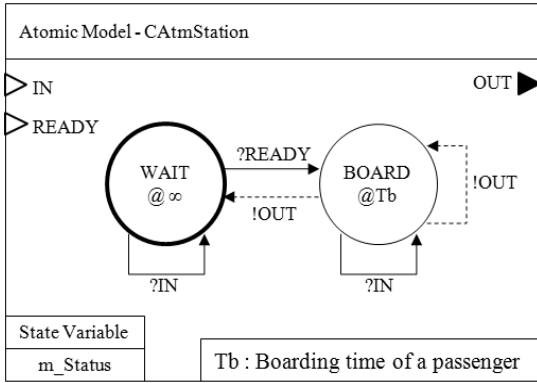


그림 8. 역사 원자 모델의 DEVS 다이어그램

$\lambda : Q \rightarrow Y$
 $\lambda(\text{BOARD}, T_b) = \text{out}$
 $ta : S \rightarrow R$
 $ta(\text{WAIT}) = \infty$
 $ta(\text{BOARD}) = T_b$

열차에 탑승한 승객과 정거장에 대기하고 있는 승객은 모두 배열 형식의 데이터 구조로 각 모델의 인스턴스에 저장된다. 승객의 승하차 시뮬레이션 과정에서 중요한 데이터로 고려될 수 있는 승객 변수는, CAtmMaglev 모델 및 CAtmStation 모델에서 정의한 시스템 상태 천이 과정에 의해 종속적으로 변경되며, 그 데이터를 얻거나 변경하기 위한 특별한 시간 진행 이벤트를 필요로 하지 않기 때문에, 각 모델의 내부 변수로 정의하여 사용하였다.

Experimental Frame의 승객 생성기(CAtmGenerator)의 경우, 활성 상태(Active)일 때, 랜덤 프로세스에 의해 승객을 생성하며, 그 출력을 역사로 전송하거나, 분석기(CAtmAnalyzer)로 전송한다. 또한, 분석기로부터 중지 명령(Stop)을 받으면, 대기 상태(Stop)로 천이한다. 분석기는 CCplSystemModel에서 생성된 다양한 열차운행 출력들을 수집하며, 생성기로부터 승객생성 정보도 수집한다. 또한, 시뮬레이션 종료조건을 체크하고 필요할 경우, 종료 메시지(Stop)를 출력하여 전체 시뮬레이션의 종료를 담당한다. DEVS 형식론은 앞선 경우와 마찬가지로 기술이 가능하며, 그림 9와 10에서는 이들에 대한 DEVS 다이어그램을 나타내었다.

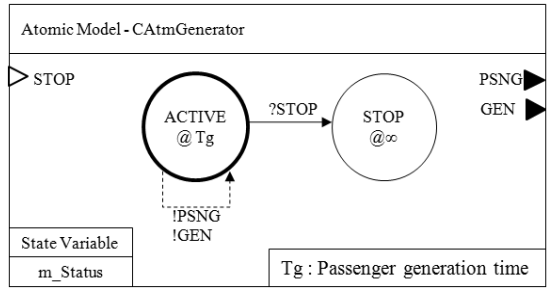


그림 9. 승객 생성기 원자 모델의 DEVS 다이어그램

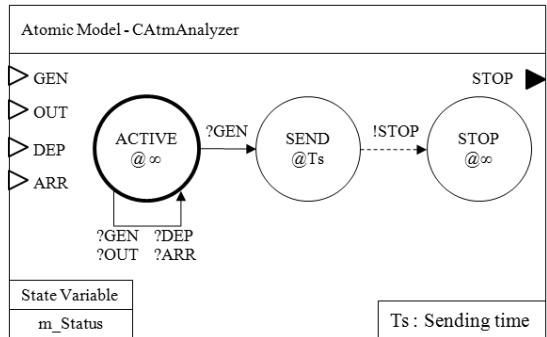


그림 10. 분석기 원자 모델의 DEVS 다이어그램

4. 시스템 구현

4.1 구현환경 및 GUI

DEVS 형식론으로 모델링된 시스템을 풀기 위해, DEVSIm++시뮬레이션 환경을 사용하였다. DEVSIm++은 계층적 스케줄링 알고리즘에 의해, 사건 발생 순서에 따라 시스템 정보(상태정보)를 갱신하며 전체 시뮬레이션을 진행하는 구현 환경과 엔진으로서, 객체지향언어인 C++로 구현된다^[8]. 컴파일러는 Visual C++ 2008을 사용하였다. 그림 11은 본 연구에서 구현된 시뮬레이션 프로그램의 GUI이며 MFC를 사용하여 분석기에서 모니터링되는 다양한 정보를 그래픽 또는 텍스트로 출력하였다. 상단에는 자기부상열차의 운행과 승하차 승객 변화를 직관적으로 파악할 수 있도록 데이터를 그래픽화 하였으며, 좌측 리스트 박스에는 승하차 시뮬레이션과 관련된 메시지 및 로그정보가 나타나고, 오른쪽 리스트 박스에는 DEVSIm++ 엔진의 로그정보가 나타나도록 하였다. 또한, 시뮬레이션 비율 설정과 시작, 종료 등의 시뮬레이션 제어를 수행할 수 있도록 명령 버튼을 구현하였다.

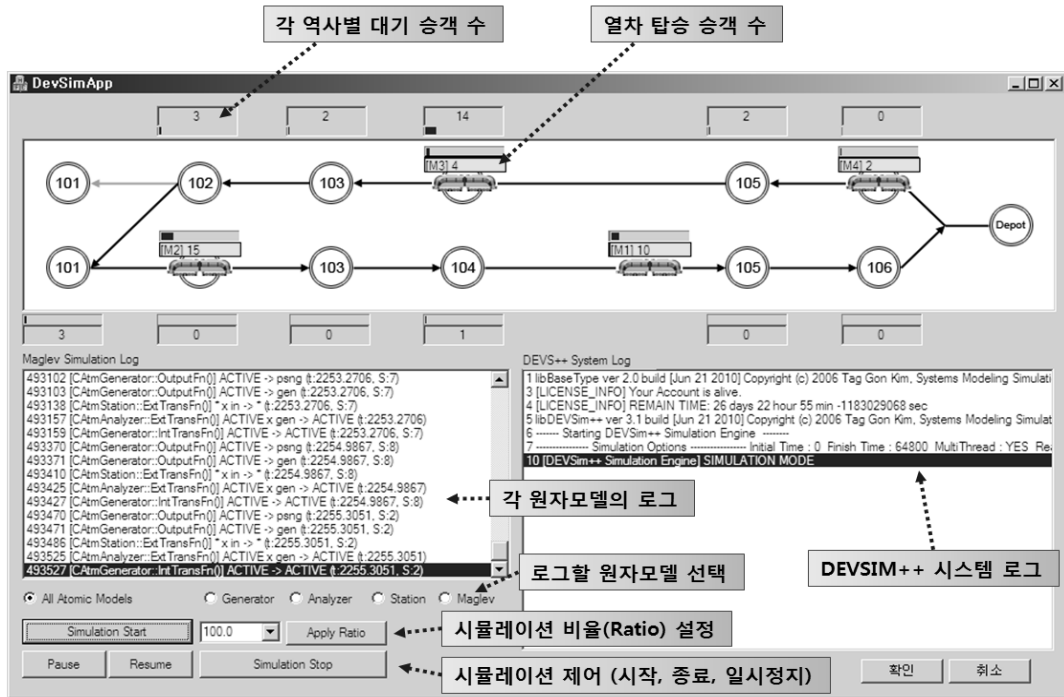


그림 11. 시뮬레이션 프로그램의 GUI

4.2 열차 운행 제어

본 연구에서는 차량 성능 시뮬레이션(TPS)결과에 의해 설계된 차량 운행 계획(DIA)을 이용하여 차량의 연속적인 움직임을 시뮬레이션 하였다. 즉, DIA로부터 알 수 있는 ‘각 역사간 거리’, ‘각 역사간 표준 주행 시간’, ‘각 역사간 최고 속도’를 바탕으로, 일정한 가속도로 가속, 감속, 등속 운동을 반복하는 단순화된 열차 주행 모델을 생성하였다. 실제로는 노선의 구배와 커브에 의해 동적인 주행이 이루어지지만, 본 연구의 경우, 승객 승하차 시뮬레이션이 주 목적이므로, 특정 이벤트 시간을 얻기 위한 운동 모델을 적용하는 것도 타당하다. 그림 12에서는 각 역사 사이의 운행 거리를 시간에 따라 구할 수 있는 연속 방정식 도출 과정을 나타내었으며, 이 방정식을 이용하여 매 갱신주기마다 열차의 위치를 획득할 수 있다.

또한, 응급 상황이 아닌 경우, 열차 제어에 있어 가장 중요한 것이 선행열차와의 거리유지이다. 즉, 승객 수요파동으로 인해 선행열차의 운행지연이 발생할 수 있고, 사고를 방지하기 위한 신호제어가 발생하며, 그 여파는 후속열차에게 전파될 수 있다. 일반적인 무인운전 시스템에서는 앞차와의 거리를 실시간 체크하며 속도를 조절하여 일정 거리를 유지하는 이동 폐쇄 방식이 사용된다. 본 연

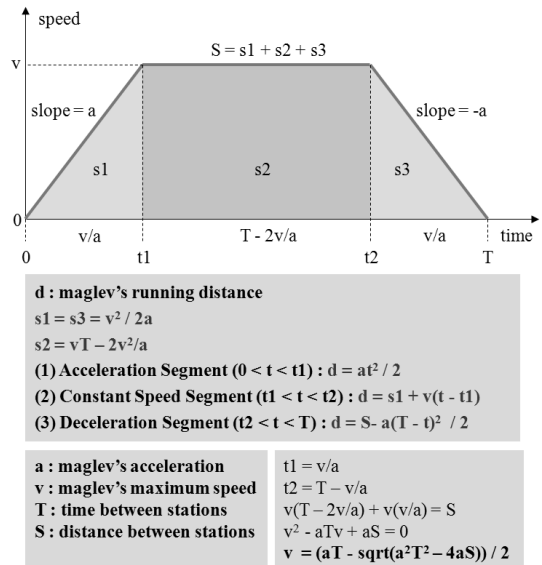


그림 12. 열차 주행 연속 방정식

구에서는 선행 열차와의 거리가 정해진 간격(1 km)보다 짧을 경우, 열차 운영을 잠시 멈추는 방식으로 구현하였다. 거리 유지를 위한 일시적인 가감속의 경우, 승객의 승

하차와는 무관하며 전체적인 시뮬레이션 결과에는 영향을 미치지 않기 때문이다.

4.3 수송 파동 생성 및 승하차 시간

본 연구에 있어 가장 중요한 고려사항은, 수송파동에 의한 승객의 유동적인 변화를 시뮬레이션에 반영하는 것이다. 본 연구에서는 자기부상열차 실용화 노선의 교통영향평가에 수록된, 역사별 1일 평균 수요예측 데이터와, 가장 혼잡한 시간대(침두시)의 시간당 승객 예측 데이터를 활용하였다. 또한 상기 자료에는 주변 도로의 교통량 분석 결과도 포함되어 있으며, 이를 바탕으로 침두시와 비침두시, 준침두시 시간대를 결정하였다. 표 1에서는 이를 이용하여 생성된 각 시간대별 승객 생성 속도를 나타내고 있다. 침두와 비침두 시간대는 각각 침두시 예측데이터와 1일 예측 데이터를 그대로 활용하였으며, 준침두의 경우 이들의 평균으로 가정하여 계산하였다. 승객 생성 속도는 일반적으로 포아송 분포를 따르며, 승객당 생성 시간은 Exponential 분포를 따르므로, 이에 따라 랜덤 프로세스를 적용하였다. 또한, 열차에 탑승한 승객의 하차 위치 결정도 필요하다. 본 연구에서는 표 1에서 예측된 승객 생성 비율을 역사 이용률로 가정하고, 현재 탑승객의 하차율은 각 역사의 이용률을 따르도록 설계하였다.

또한, 열차의 정차 시간에 가장 많은 영향을 끼치는 것

은 승객의 승하차 시간이라 할 수 있다. 출입문의 개폐 시간은 미리 지정되지만, 승객 별 승하차 시간은 미리 예측하기 힘든 입력 데이터이다. 본 연구에서는 서울시 지하철 2호선의 승하차 시간분석 데이터^[9]를 바탕으로, 1인당 승차 시간 평균은 0.88초, 하차 시간 평균은 0.76초로 설정하였다. 이 논문에서는 승하차 승객의 수가 많을수록 승하차 시간도 길어지며, 다양한 조건을 고려한 시간 예측 모형도 제시되었지만, 본 연구에서는 입력 데이터의 간략화를 위해 이를 고려하지는 않았다. 또한, 현재의 시범 노선 운행 전략은 표준 정차시간(20초)을 준수하는 것이므로, 승객이 적을 경우에도 그 시간 동안 대기하도록 하였다.

5. 실험 및 결과

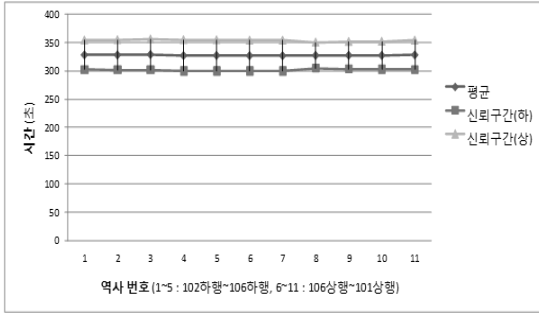
시뮬레이션 시간은 열차의 하루 영업시간(오전 5시~오후 11시)인 총 18시간(64800초)으로 설정하였고, 시뮬레이션은 총 7회 수행하였다. 또한, 열차 주행을 담당하는 연속 시스템은 시뮬레이션 시간을 입력으로 받아 진행되며, 시간 간격은 0.01초로 설정하였다. 시뮬레이션 비율의 경우, GUI에서 차량의 진행과 승하차 상황 모니터링 가능하도록 최대 100으로 설정하였다.

수송파동을 고려한 시뮬레이션을 통해, 열차의 정상 운영을 평가하고 해당 노선의 서비스 품질을 검토하기 위해, 운전 시격(열차가 운행되는 간격), 열차 정차 시간 및 승객 대기 시간을 시뮬레이션 결과로서 측정하였다. 운전 시격은 역사 별 도착, 출발 정보를 바탕으로 계산되며, 운영 계획상의 운전 시격과 비교검토를 할 수 있다. 정차 시간과 승객 대기시간은 열차 이용객이 많거나 선행열차 지연에 의한 신호제어 등에 의해 변동될 수 있으며, 운영 계획상의 스케줄과 비교하여 탑승객 또는 대기 승객의 불편을 정량화 할 수 있다. 그림 13은 각 역사 별로 측정된 데이터로부터 운전 시격(a), 정차 시간 평균(b), 승객당 대기 시간(c)을 95% 신뢰도로 추정하여 그래프로 나타내었다. 마지막 역사인 106역사의 경우, 하행선 정거장(5번)과 상행선 정거장(6번)이 분리되어 있으며, 따라서 5번 정거장에서는 승객의 하차만 발생하기 때문에, 승객 당 대기시간 그래프(c)에는 해당 데이터가 표현되지 않았다.

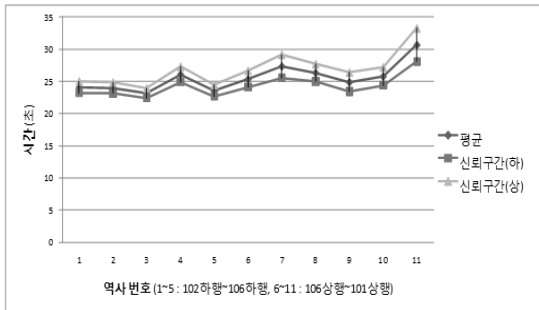
운전 시격 결과의 경우, 초기 운영 계획상의 5분에 비해, 약간 지연된 시격(5.45분)을 나타내었으며, 이는 승객이 많을 경우 표준 정차시간보다 오래 정차하게 되며, 승객이 적을 경우에는 반드시 표준 정차시간(20초)을 지키

표 1. 시간대별 승객 생성 속도

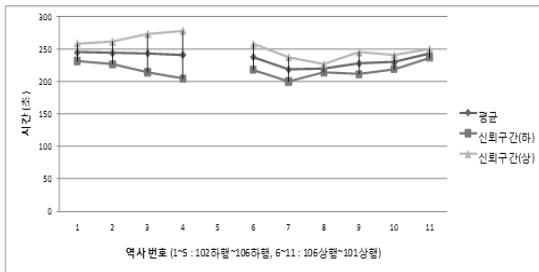
역사 번호	비침두시		침두시		준침두시
	승차율 (명/일)	GenTime (초/명)	승차율 (명/시)	GenTime (초/명)	GenTime (초/명)
하행선					
101	11,906	5.44	1,869	1.92	3.68
102D	2,940	22.04	462	7.79	14.91
103D	1,914	33.85	300	12.00	22.92
104D	644	100.62	101	35.64	68.13
105D	423	153.19	66	54.54	103.86
상행선					
106U	1,461	44.35	229	15.72	30.03
105U	1,492	43.43	234	15.38	29.40
104U	8,953	7.23	1,406	2.56	4.89
103U	1,557	41.61	244	14.75	28.18
102U	3,866	16.76	607	5.93	11.34
합계	35,156		5,518		



(a) 역사 별 운전 시력



(b) 역사 별 정차 시간



(c) 역사 별 승객 대기 시간

그림 13. 시뮬레이션 결과 및 통계

도록 운행 전략을 설정했기 때문인 것으로 판단된다. 열차의 평균 정차 시간은 25.6초로 시뮬레이션 되었으며, 이도 역시 위의 운행 전략으로 인해 승객 승하차 시간 평균이 표준 정차시간을 초과했기 때문인 것으로 판단된다. 역사 별 승객 대기시간의 경우, 평균적으로 3.9분 정도가 예측되었으며, 전체적으로 하행 선로의 대기시간이 상행 보다 큰 것으로 나타났다. 이는 기본적으로 101역사(교통센터, 상행과 하행이 교차되는 곳)에 수송수요가 많고, 상행과 하행 승객이 동시에 승하차함으로써, 101역사에서 열차 지연이 다수 발생한 결과라 예측할 수 있다.

결과적으로 본 연구의 모델링 및 시뮬레이션의 결과는 운행계획과 전략을 검토하는데 이용될 수 있음을 확인할

수 있었다. 하지만, 다수의 가정과 추정치를 기반으로 실험이 이루어 졌기 때문에, 현재의 결과로서는, 구현 가능성의 검증과 참고자료의 정도로 활용될 수 있을 것이다. 추후에는, 표준 정차 시간의 조정 또는 해제, 이동 폐쇄 거리 조정 등 운행 전략을 수정하고 다시 시뮬레이션 해봄으로써, 승객의 서비스 품질을 높일 수 있는 방안을 검토할 수 있을 것이다. 또한, 승객 유동은 단순한 수요 예측 추정치를 바탕으로 시뮬레이션 되었지만, 실제 차량이 영업 운행된 후, 실제 승객량과 승하차 시간을 반영하면 좀 더 정확한 시뮬레이션 결과를 얻을 수 있으며, 이를 통해 열차 운행 계획의 효과적인 최적화도 가능할 것이라 기대된다.

6. 결론

본 연구에서는 새로운 교통수단으로서 실용화가 진행 중인 자기부상열차 시스템에 대해, 승객의 유동적인 변화를 반영할 수 있는 열차 운행 시뮬레이션 프로그램을 개발하였다. 미리 설계된 열차 운행 계획과 각 역사 별 수요 예측 데이터를 활용하고, 운행 전략에 관한 여러 가지 가정(이동폐쇄, 정차시간, 승객 승하차 시간 등)을 기반으로, 실제 열차가 운행되기 이전에 열차 운행 계획과 전략을 검토할 수 있는, 이산사건 기반의 모델링 및 시뮬레이션 방법을 제안하였다. 또한, DEVS 형식론과 DEVSsim++ 시뮬레이션 환경을 이용한 구현과 실험을 통해, 그 구현 가능성과 유용성을 검증하였다. 본 연구의 시스템 모델링 방법 및 시뮬레이션 과정은 자기부상열차뿐 아니라, 무인 자동운전으로 운영되는 다양한 철도교통수단에 대해서도 적용될 수 있을 것이며, 앞으로는 좀 더 현실에 가까운 모델 또는 데이터를 적용하거나, 실제 데이터를 활용할 수 있는 연구를 기대한다.

감사의 글

본 연구는 KAIST 전기 및 전자공학과 ‘EE612: Discrete Event Systems Modeling Simulation’ 교과목의 일부로 수행된 연구이며, DEVS 및 DEVSsim++과 관련한 이론 및 구현환경을 제공해 주신 김탁근 교수님께 감사드립니다. 또한 이 연구는 국토해양부 미래도시철도기술 개발사업의 ‘도시형 자기부상열차 실용화 사업’의 지원을 받아 수행 되었으며, 국토해양부 및 한국건설교통기술평가원 관계자 여러분께 감사드립니다.

참고 문헌

1. 박도영, 신병천, 유문환, 김동성, “도시형 자기부상열차 실용화 현황,” 대한전기학회 전기의세계, 56(5), pp. 14-22, 2007년 5월.
2. 신병천, 김달식, 박도영, 백수현, 강홍식, “도시형자기부상열차 실용화 사업현황 및 시범노선 계획 소개,” 대한기계학회 추계학술대회발표집, Industrial Session-1, 2007년 10월.
3. 김동희, 김영훈, “고속선을 고려한 다중열차주행 시뮬레이터 개발,” 한국철도학회 춘계학술대회발표집, pp. 58-65, 2001년 5월.
4. 김동희, 김성호, 오석문, “다중열차 시뮬레이션을 위한 철도시스템 모델,” 한국철도학회 논문집, 4(2), pp. 58-65, 2001년 6월.
5. 오세찬, 김길동, 박성혁, 이장무, 이한민, “차세대도시철도 시스템 무인운행시뮬레이터 기본설계방안 연구,” 한국철도학회 추계학술대회발표집, pp. 33-38, 2006년 11월.
6. 이창형, 이종우, “ATO 자동운전 기반의 무인운전 경전철 TPS에 관한 연구,” 한국철도학회 추계학술대회발표집, pp. 1132-1137, 2008년 11월.
7. 김탁곤, “모델링 시뮬레이션 공학,” 한국정보과학회지, 25(11), pp. 5-15, 2007년 11월.
8. 안명수, 박성봉, 김탁곤, “DEVSsim++: 의미론에 기반한 이산사건 시스템의 객체지향 모델링 및 시뮬레이션 환경,” 한국정보과학회논문지, 21(9), pp. 1652-1664, 1994년 9월.
9. 오석문, “서울시 혼잡 지하철역의 승하차 시간 분석,” 한국철도학회 춘계학술대회발표집, pp. 42-49, 2005년 5월.
10. A. Nash and D. Hurlimann, “Railroad simulation using OpenTrack,” Computers in Railways IX, pp. 45-54, May 2004.
11. JB BAHN : Simulation of railway and streetcar networks, URL: http://www.jbss.de/hpg_eng.htm
12. A. Radtke and D. Hauptmann, “Automated planning of timetables in large railway networks using a microscopic data basis and railway simulation technique,” Computers in Railways IX, pp. 615-625, May 2004.
13. Pablo Grube, et al., “An event-driven simulator for multi-line metro systems and its application to Santiago de Chile metropolitan rail network,” Simulation Modelling Practice and Theory, vol. 19, no. 1, pp. 393-405, Jan. 2011.
14. CS Chang, et al., “Development of train movement simulator for analysis and optimisation of railway signalling systems,” International Conference on Developments in Mass Transit Systems, pp. 243-248, April 1998.
15. Massimo Paolucci and Raffaele Pesenti, “An object-oriented approach to discrete-event simulation applied to underground railway systems,” Transactions of the Society for Modeling and Simulation International, vol. 72, no. 6, pp. 372-383, June 1999.
16. R. John Hill and Theresa K. Yates, “Modelling railway block signalling systems using discrete-event simulation,” ASME/IEEE Joint Railroad Conference, pp. 1-9, April 1992.
17. B.P. Zeigler, H. Praehofer and T.G. Kim, Theory of modeling and simulation, 2nd Ed, Academic Press, Jan. 2000.



차 무 현 (mhcha@kimm.re.kr)

2000 포항공과대학교 기계공학과 학사
2006 KAIST 기계공학과 석사
2006~현재 한국기계연구원 선임연구원

관심분야 : 모델링&시뮬레이션, 가상현실 시뮬레이터



이 재 경 (jkleece@kimm.re.kr)

1995 아주대학교 컴퓨터공학과 학사
1997 아주대학교 컴퓨터공학과 석사
2008 충남대학교 컴퓨터공학과 공학박사
1997~현재 한국기계연구원 선임연구원

관심분야 : 가상현실 시뮬레이터, 인간-기계 상호 작용



백 진 기 (jgbeak@kimm.re.kr)

1992 경희대학교 토목공학과 학사
1993 한국철도기술공사 궤도설계부
2004 (주)동명기술공단 철도부
2007~현재 한국기계연구원 도시형자기부상열차 실용화 사업단

관심분야 : 철도계획