

신호제어장치 시제품의 성능평가를 위한 현장에뮬레이션 모듈 설계

황종규, 이종우, 정의진  
한국철도기술연구원 철도신호통신연구팀

A Design of Track Builder for Railway Signalling Test-bench

Jong-Gyu Hwang, Jong-Woo Lee, Eui-Jin Joung  
Korea Railroad Research Institute(KRRI)

**Abstract** - The railway signaling system consists of computerized vital devices. Therefore it is important to validate the required functions of developed signaling system. To verify the condition and functions of signaling functions, the plentiful laboratory test is required. To achieve this, the emulation S/W for signaling test-bench is needed. The object-oriented concept for trackside signaling equipments emulation is described in this paper.

1. 서 론

철도신호제어시스템은 열차운행의 안전성을 보장하는 최종적인 장치로 신호시스템의 설계 및 제작에 있어서 많은 노력들이 필요로 하게 된다. 즉, 새로운 신호시스템을 설계 및 개발하여 실제 적용에 앞서 충분히 시험을 거쳐서 안전성을 확보하여야 하지만 실제의 선로를 포함한 현장 신호장치들을 모두 갖추고 시제품들을 시험하기는 거의 불가능하다.

따라서 실험실 레벨에서 신호제어장치들의 시제품들에 대한 시험을 원활히 할 수 있도록 현장에 에뮬레이션 하는 별도의 장치를 필요로 한다[1][2]. 이 현장에뮬레이터에는 열차가 주행할 수 있는 선로, 궤도회로, 신호기, 선로전환기를 포함한 현장 신호장치들을 포함하여야 하며, 또한 신호장치들은 열차의 운행에 따른 보안장치이므로 열차에 대한 모델링을 필요로 한다.

이때 실제 궤도회로처럼 이 선로선형에서 진로 등의 정보를 가지고 신호시스템의 각종 로직에 따라서 열차의 운행이 시뮬레이션될 수 있어야 한다. 즉, 물리적인 선로선형을 소프트웨어로 표현하여 적절한 시뮬레이션이 될 수 있도록 하여야 한다.

본 논문에서는 이러한 현장신호장치들을 소프트웨어로 모델링하여 열차가 주행할 수 있도록 하는 현장 에뮬레이터에 대해 설명하였다. 그리고 이를 위해 일반 그래프 이론을 확장한 Double Vertex 그래프 알고리즘을 이용한 철도 선로선형의 모델링 방법을 연구하였다.

2. 현장 신호장치 모델링

본 연구에서는 현장 신호장치들을 소프트웨어로 모델링 하기 위해서 객체지향(Object-Oriented) 기법을 적용하였다. 이 방법은 오늘날 많이 사용되는 소프트웨어 설계방법으로 크게 다음과 같은 특성을 갖는다(3)(4).

- Object & Class : 클래스는 Object를 설명하는 것으로 Object들이 가질 기능 등을 멤버함수 및 변수 형태로 설명되어진다. 동일한 클래스명을 갖는 Object들은 동일한 내부 기능들을 가지게 되며 프로그램 내부에서 생성된 Object를 클래스의 인스턴스(Instance)라 한다.

- 상속성 : 상속성은 클래스들 사이의 관계를 설명한 것으로 임의의 한 클래스의 구현이 이미 존재하는 다른 클래스를 기반으로 하고 있음을 나타낸다. 특유한 기능을 갖는 클래스를 구현할 경우 이미 구현되어 있는 클래스를 상속받아 기능을 추가하면 된다. 이러한 상속성을 활용함으로써 새로운 클래스의 구현시 기존의 클래스와 다른 내용만 구현하면 될 수 있다. 본 논문에서는 신호기의 경우 CSignal이라는 클래스를 구현하였고, 이를 상속받아 고속전철용 마커를 나타내는 CMarker를 구현했고, 또 이 클래스를 상속받아 CKSignal이라는 국철용 신호기를 구현할 수 있도록 했다.

신호제어장치의 시제품들에 대해 실험실 레벨에서 시험을 위한 현장신호장치들의 에뮬레이션 모듈의 구현을 위해서 앞서 설명한 OO기법을 사용하였다. 이에 따라 선로선형을 생성하는 Track Builder 클래스, 열차에 에뮬레이션 하는 Train 클래스, 시제품들과 통신을 위한 Communication 클래스, 결과의 모니터링 및 분석을 위한 Monitoring 클래스로 구성한다. 또한 생성된 선로의 구매, 커브, 제한속도 등의 입력을 외부의 데이터베이스를 사용하였다. 물론 이 외부의 데이터베이스의 사용을 위한 데이터베이스용 클래스가 별도로 존재하게 된다. 이 데이터베이스용 클래스에서는 데이터베이스용 파일의 필드값이 프로그램 내부에서 사용될 변수들로 변경된다. 즉 이 클래스는 생성된 선로가 실제와 같도록 매칭시켜 주는 거리별 구매, 커브, 특정구간의 제한속도 등의 정보를 가지게 된다.

현장 에뮬레이션 모듈을 위한 클래스의 구성은 그림 1과 같다. 여기서 시스템 클록은 시뮬레이션이 수행되는 간격을 의미하는 것으로 임의로 설정된 시간에 따라 시뮬레이션이 수행될 수 있도록 하였다. 이 설정시간에 따라 실제 운행시분을 계산할 수 있다.

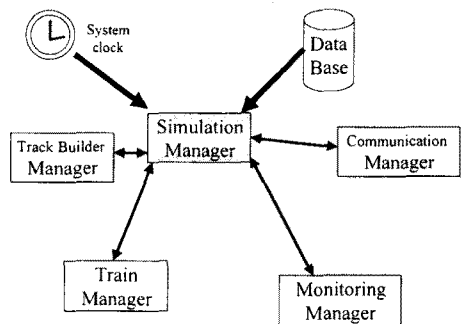


그림 1. 현장 에뮬레이터의 구성

그림 1에서 Simulation Manager라는 클래스는 시뮬레이션을 전체적으로 수행하는 클래스로 다른 클래스

들을 사용하여 시뮬레이션이 수행된다. (본 프로그램에서는 CView 클래스)

이들 클래스 중 현장 신호장치들을 모델링하는 Track Builder 모듈의 구성은 다음과 같다. 즉 현장 신호장치들인 궤도회로, 선로전환기, 신호기 그리고 열차운행을 위한 진로 등이 클래스로 구현되어져 있다.

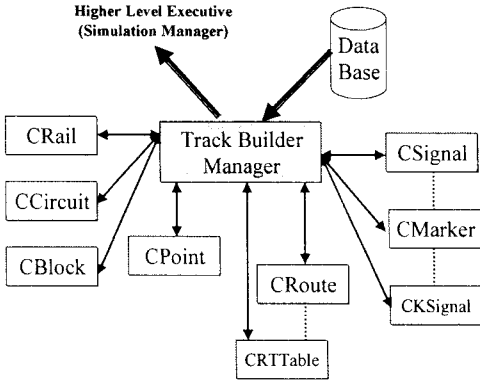


그림 2. Track Builder Model Class Hierarchy

이중 신호기는 앞에서 설명한 것처럼 상속성을 활용하여 구현하였다. 그리고 선로선행의 기본이 되는 CRail 클래스는 선로를 위한 기본 단위로 실제로는 존재하지 않는 구성요소이나 본 프로그램에서는 선로선행의 생성을 위해 구현한 것이다. 이 CRail Object들이 여러 개가 묶여져서 하나의 궤도회로 Object(CCircuit class)가 된다. 다음은 구현된 이 클래스의 일부를 나타낸 것이다. 여기에서 사용된 CRail 클래스와 CCircuit 클래스의 포인터형은 이들 객체들을 서로 링크시켜 열차가 주행할 수 있도록 하기 위한 것으로 자세한 알고리즘은 다음절에 나타내었다.

```
class CCircuit : public CObject
{
// public variables list
..... // 궤도회로 상태변수, 입력된 커브, 구배 및 제한속도

CRail* prev vertex rail[2], next vertex rail[2];
CCircuit* prev, *next, *prev side, *next side;

// public functions list
.....
}; //END CCircuit class
```

그림3은 앞에서 설명한 모듈에 의해 생성된 선로선행을 바탕으로 열차가 주행할 수 있도록 하는 모듈이다. 이 모듈은 Motor 클래스와 Speed Control 클래스로 구성되어지는데, 본 연구에서는 신호시스템 시뮬레이션 용이므로 프로그램의 단순화를 위해 Motor 클래스는 외부의 테이블에 의해 속도에 대한 출력 견인력 등을 바탕으로 간략히 구현하였으며 일반적으로 TPS 프로그램에서 구현되는 전류, 전압, 소비전력 등은 생략하였다.

### 3. 궤도회로 모델링

#### 3.1 궤도선행 모델링

궤도선행은 철도의 선로 특성을 나타내는 부분이다. 이 궤도선행에는 궤도회로, 분기기 등 다양한 구성요소들이 존재한다. 또한 시뮬레이션을 위해서는 이러한 각각의 선행들이 서로 연결되어 전체적인 선로선행을 이루어야 한다.

궤도의 궤도회로에 있어서 대부분 분기기가 있어서 이에 따라 진로가 달라지므로 이에 대한 고려가 반드시 있어야 한다. 일반적인 방법에 의해서는 이러한 분기기와

진로의 개념을 모델링 하기가 어렵다.

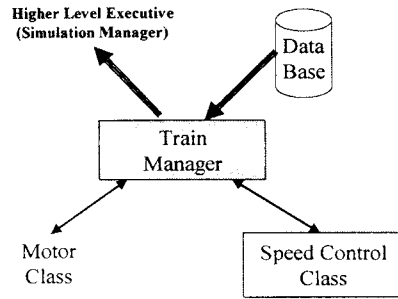


그림 3. Train Manager Model Class Hierarchy

그림 4(a)는 분기기가 포함된 간단한 선로선행을 나타낸 것이다. 이 선로선행에는 다음과 같은 진로들이 가능하다.

- X → A → B → Z     · Z → C → A → X
- X → A → C → Z     · Z → B → A → X

이러한 진로들의 구성은 가능하지만 분기기의 특성상 'Z → B → A → C → Z'와 같은 진로는 설정될 수가 없으므로 이러한 진로들이 배제된 상태로 선로선행이 모델링 되어야 한다. (a)와 같은 실지의 궤도회로 선행은 그래프 이론에 의해 (b)와 같이 모델링 하거나, (c)와 같이 방향성을 갖는 간단한 방법으로 모델링 되어지지만 앞에서 설명한 바와 같은 진로들을 배제시킬 수 있는 모델은 아니다. 따라서 철도신호시스템의 궤도회로 선행, 특히 분기기가 포함된 경우는 새로운 방법에 의한 모델링을 필요로 한다.

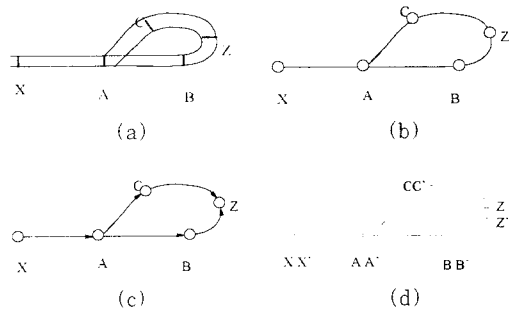


그림 4. 선로선행의 모델링

#### 3.2 Double Vertex Graph 알고리즘(5)~(7)

그림 4의 (a), (b) 및 (c)의 궤도회로 선행 모델링은 분기기가 포함된 진로를 나타내기에는 적절치 못한 방법들이다. 본 연구에서는 이러한 분기기의 특성에 따른 경로의 표현이 가능하도록 선로선행을 (d)와 같이 모델링 하는 방법을 사용하였다.

기존의 그래프 이론에서는 각각 하나의 vertex와 하나의 노드로 구성되어져 있으며, 이러한 그래프 이론을 바탕으로 하여 궤도회로 선행을 모델링 할 경우 분기기 등에 대한 정확한 표현이 불가능하여 철도신호시스템의 각종 기능들의 시뮬레이션이 어렵다. 본 연구에서는 하나의 노드에 두 개의 vertex를 가진 'Double-Vortex Graph' 알고리즘을 사용하여 궤도회로 선로선행에 ego 한 모델링을 하였다.

따라서 경로를 찾을 경우 'vertex-vertex-node-vertex-vertex-node-...'와 같은 조합에 의해 가능해질 수 있으며 이에 대한 선행이 (d)와 같다. 이러한 이

중 vertex에 의한 선로선형의 표현에 있어서 열차가 주행할 수 있는 진로는 다음과 같이 나타낼 수 있다.

- XX' → AA' → BB' → Z
- XX' → AA' → CC' → Z'
- Z'Z → B'B → A'A → X'
- ZZ' → C'C → A'A → X'
- CC' → Z'Z → B'
- BB' → ZZ' → C'

여기에서 "BB' → A' → C'"와 같은 경로는 존재하지 않게 된다. 따라서 이러한 이중 vertex들과 노드들을 서로 링크시킴으로써 경로들을 표현할 수 있고, 또한 분기기의 특성을 나타낼 수 있다. 이러한 그래프 구조를 바탕으로 소프트웨어 상으로 현장의 궤도회로를 모델링하고, 또한 이들을 서로 링크시켜 열차가 주행할 수 있도록 에뮬레이션 하는 알고리즘을 설명한다.

### 3.3 궤도회로 링크 알고리즘

철도신호시스템의 시뮬레이션을 수행하기 위해서는 소프트웨어로서 우선적으로 선로의 선형에 대한 모델링이 이루어져야 한다. 이러한 궤도회로의 모델링을 위해서 본 연구에서는 앞서 설명한 이중 vertex 그래프 구조를 사용하였다.

구현한 소프트웨어에서는 궤도회로를 구성하는 가장 기본단위인 레일을 클래스로 구성하였다. 이들 레일들이 몇 개가 묶여져서 하나의 궤도회로 클래스가 된다.

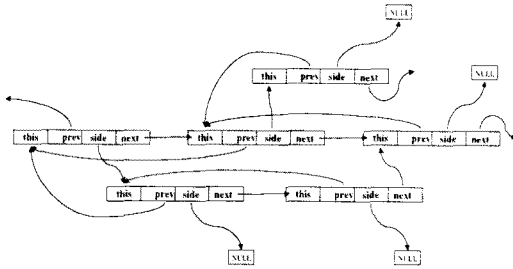


그림 5. 궤도회로의 연결원리

각 회로의 형식에 따라 달라지지만 기본적으로 오른쪽에 위치한 회로는 'next' 포인터로 연결되고, 왼쪽은 'prev' 포인터로 연결되고 측면에 위치하게 되면 'side' 포인터로 연결되게 된다. 이처럼 연결할 레일의 정보들을 포인터 정보로 가짐으로써 서로 링크되게 되는데, 이러한 과정이 편집된 모든 레일 인스턴스에서 수행됨으로써 전체적으로 선로가 링크되게 되고, 이로써 열차가 이 선로 위를 따라 주행시뮬레이션을 할 수 있는 바탕이 되게 된다. 이렇게 연결된 레일 인스턴스들은 몇 개씩 하나의 궤도회로 객체로 묶이게 된다.

한 궤도회로에 소속된 회로의 포인터는 각 궤도회로의 맨 앞에 위치하는 회로가 Vertex\_rail[0], 맨 뒤에 위치하는 회로가 Vertex[1] 그리고 측면 회로가 존재하는 경우 Vertex\_rail[2]에 저장된다. 이렇게 함으로써 Vertex[ ]의 배열에 들어있는 포인터에 의해 시뮬레이션이 될 수 있도록 한다. 시뮬레이션 시 각 궤도회로의 내부에서는 레일의 포인터로 열차가 진행하지만 다음 궤도회로로 넘어갈 때는 Vertex 구조에 의해 정해진 다음 궤도회로로 진행하게 된다. 이렇게 해서 다음 궤도회로로 연결되어지면 또 다시 회로의 포인터로 연결되는 과정을 반복 수행하게 된다. 즉 이러한 알고리즘에 의해 시뮬레이션 시 열차의 진행이 이루어지도록 하였다.

궤도회로가 연결되는 구조를 살펴보면 그림 3(a)처럼 각각 별개의 궤도회로로 구성되어 있다가 하나의 선로 프로파일이 링크시킴이 되면 (b)처럼 각각 궤도회로의 맨 앞과 맨 뒤, 그리고 측면 회로의 포인터와 연결되는 점들은 같은 번호를 가지

게 된다. 이렇게 하여 임의로 편집된 선로 프로파일이 열차가 달릴 수 있는 실제 선로가 되게 된다.

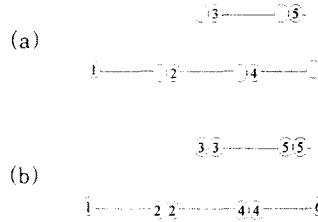


그림 6. Vertex 구조에 의한 궤도회로의 연결

## 4. 결론

본 논문에서는 신호제어장치 시제품들의 실험실 레벨에서의 테스트를 위한 현장신호장치 에뮬레이션 모듈의 객체지향기법 관점에서 설명하였다. 이들을 OOP로 구현함으로써 향후 설계를 진행하면서 추가되거나 변경되는 사항을 쉽게 반영할 수 있도록 하였다. 그리고 철도 신호시스템의 시뮬레이션을 위한 소프트웨어에 있어서, 철도만이 갖는 특성을 가지면서 선로의 선형을 표시할 수 있는 Double Vertex 그래프 구조에 대해 설명하였다. 이 알고리즘의 적용으로 철도선로의 특징 중의 하나인 분기기의 특성을 나타낼 수 있고 또한 열차의 안전에 중용한 영향을 미치고, 시뮬레이션 시 가장 중요한 열차의 주행진로의 표현 등도 이 알고리즘의 적용에 따라 쉽게 구현될 수 있다.

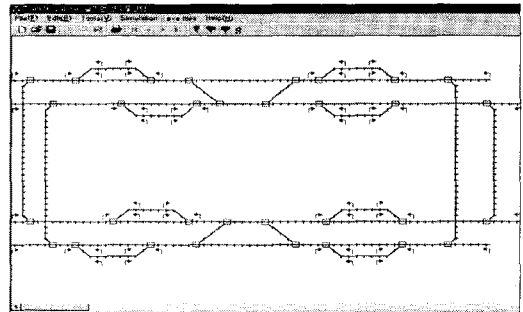


그림 7. 제안된 알고리즘의 테스트용 선로 프로파일

### [참고 문헌]

- [1] 稱毛弘苗, et al., "次世代運轉制システムのシ室内實驗", RTRI Report Vol 5, No. 1, pp. 48-55, 1991.
- [2] <http://www.azurenet.com/>
- [3] L.K. Siu and C.J. Goodman, "An Object-oriented Concept for Simulation of Railway Signalling and Train Movements", Proceedings of Computers in Railways, pp. 545-556, 1992.
- [4] W. Weits, "Simulation of Railway Traffic Control", Proceedings of IFORS'97, pp.461-468, 1997.
- [5] 황중규 외, "Double Vertex 그래프에 의한 궤도회로 토폴로지의 생성", 2000년도 대한전기학회 추계학술대회 논문집, pp. 389-391, 2000.
- [6] M. Montigel, "Formal Representation of Track Topologies by Double Vertex Graphs", Proceedings of Computers in Railways, pp. 359-370, 1998.
- [7] S. Axler, et al., "Modern Graph Theory", Springer, 1998.