논 문
60-1-14

비동기식 PRT 차량의 주행제어 알고리즘

An Algorithm for the Asynchronous PRT Vehicle Control System

정 상 기† · 정 락 교* · 김 백 현**
(Sang-Gi Chung · Rag-Kyo Jeong · Baek-Hyun Kim)

Abstract - A PRT vehicle's control method is presented in this paper. In the asynchronous vehicle control system, vehicles follow their leading vehicles. Leading vehicles are defined differently among the different types of track. The main topic of this paper is to present a method to define the leading vehicle among different types of track and the calculation algorithm of the safety length the following vehicle must maintain. Simulation program is developed using the algorithm and the results of the test run are presented. An asynchronous PRT vehicle control algorithm was presented by Szillat in the paper "A low level PRT Microsimulation, Dissertation, University of Bristol, 2001". But it is different from the algorithm in this paper. In the algorithm proposed by Markus, vehicles in the merging track are controlled synchronously, and its safety distance between the leading and the following car is evaluated after the establishment of the complicated future time-location table instead of simple equations proposed in this paper. .

Key Words : PRT, Simulation, Asynchronous control, APM

1. 서 론

PRT 시스템의 시격은 일반철도의 시격에 비하여 매우 짧은 것으로 알려지고 있다. 제한속도 100 km의 고속도로에서 앞차와의 최소거리를 100 m 이상으로 유지시킬 때의 시격은 3.6초가 된다. PRT는 전체시스템이 무인 자동으로 제어되므로 이 보다 더 시격을 줄일 수 있다. 한국철도기술연구원에서는 PRT 차량의 운행제어 성능 예측을 위한 시뮬레이터를 개발 중에 있다. 본 논문에서는 현재 개발 중인 시뮬레이터에 적용되고 있는 PRT 시스템의 운행제어 알고리즘에 대해 논의 한다. PRT 차량의 운행 제어는 크게 3 가지로 분류되어 왔다.[1] 맨 처음 나온 방식은 동기식 방법으로 트랙을 슬럿(slot, 일반철도의 블록에 해당함)으로 나누어 차량이 점유하고 있는 전 후의 슬럿에는 다른 차량이 진입할 수 없는 방식이다. 슬럿의 길이는 고정되지 않고 오직 슬럿을 통과하는 차량의 시간 만 고정된다. 이 시간은 시격의 함수이므로 시격이 고정된다. 즉 차량의 속도가 증가하면 슬럿의 길이가 길어지고 반대로 차량의 속도가 감소하면 슬럿의 길이가 짧아진다(그림 1 참조). 이 방식에서는 슬럿이 정해진 속도로 움직이고 차량은 슬럿의 한 가운데 포인트(point)에 위치하도록 계속 제어되면서 운행된다. 이 점이 동기식 제어 방식을 일명 포인트 펄로우어(point follower) 방식이라 부르는 이유이다. 이 방식에서는 차량이 역사에서 출발하여 목적지 역사에 도착할 때까지의 전 여정동안 차량이 점유하는 슬럿이 출발 전에 할당되지 않는 한, 그 차량은 출발역사에서 출발하지 못하도록 되어 있다. 이 방식은 선로의 활용에 비효율적이므로 유사 동기제어 방식이 고안되었다. 이 방식은 동기식 제어방식과 동일하나 차량이 출발역사에서 도착역까지의 시간에 따른 모든 슬럿을 사전에 할당 받지 않고 출발역사에서 출발한다. 운행도중 슬럿의 점유에 대해 다른 차량과 충돌이 일어날 경우 한 슬럿 전진 혹은 후퇴도록 운행제어 된다. 다음은 비동기식 차량 운행제어방식으로 우리 연구원에서 개발되는 운행제어 시뮬레이터에 적용되는 방식이다. 이 방식에서는 슬럿의 개념은 없고 일반철도에서의 디스턴스투고우(distance to go) 방식과 유사하게 선행차량과의 안전거리만 유지하면서 주행하게 된다. 선행차량의 위치, 속도 등을 항상 감시하면서 주행하므로 일명 카펄로우어(car follower) 방식이라고도 한다(그림 2 참조). 이 선행차량의 선정 및 선행차량과의 안전거리 확보 방식이 노선의 트랙 성격에 따라 달라질 수 있다. 향후 본 논문에서는 2개 이상 노선이 만나는 점 혹은 분리되는 점, 카브 등에 의해 제한속도가 달라지는 점, 또는 역사 등을 노드라 칭하고 노드와 노드 사이를 트랙이라고 부른다. 차량이 현재 주행하는 트랙이 합류(merge)하는 트랙, 분기(diverge)하는 트랙 혹은 합류나 분기를 하지 않는 일반트랙 중 어느 트랙이냐에 따라서 선행차량의 선정 및 선행차량과의 안전거리 확보 방식이 달라질 수 있으므로, 본 논문에서는 이 3가지 경우에 대한 운행제어 방식에 대해 논의하고자 한다.

† 교신저자, 정회원 : 한국철도기술연구원 수석연구원
E-mail : sgchung@krri.re.kr
* 정 회 원 : 한국철도기술연구원 책임연구원
** 정 회 원 : 한국철도기술연구원 선임연구원
접수일자 : 2010년 11월 8일
최종완료 : 2010년 12월 2일

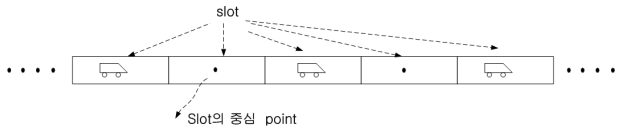

그림 1 동기식 운전방식(slot 중심에 차량이 위치함)
Fig. 1 Synchronous vehicle control(vehicles are located at the center of their slot)

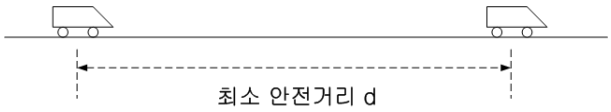

그림 2 비동기식 운전방식(car follower 방식)
Fig. 2 Asynchronous vehicle control(car follower control)

2. 본 론

2.1 안전 운행의 정의

비동기식 안전운행 제어 방식에서는 앞차와의 안전거리를 유지하며 운행하는 방식이므로 이 안전운행에 관한 명확한 정의가 필요하다. Szillat [2]에 의하면 안전운행에 관해 다음과 같이 정의하였고 개발되는 시뮬레이터의 안전운행 제어에서도 이 정의를 적용하였다. 즉 안전운행이란

① 정상운전 상태에서 앞차와의 안전거리를 충분히 유지하여 앞 차량이 최대 운전 파라메터(최대 감속도, 최대 저크)를 적용하여 감속하고 최대한의 데이터 통신 시간지연(latency)이 있을 경우에도 뒤 차량은 정상상태 운전이 허용하는 운전 파라메타를 적용하여 앞차와 충돌 없이 정지할 수 있어야 한다.

② 사고를 포함하는 비 정상 정지 시에도, 즉 앞 차량이 최대 사고 감속도로 정지할 경우에도, 앞차와의 안전거리가 충분히 유지되어, 뒤 차량은 최소 비상제동 감속도를 적용하여 또한 최대 시간의 기기반응 시간을 적용하여 앞차와 충돌 없이 정차할 수 있어야한다.

위에서 정의 된 안전거리 조건 ①을 만족시키는 안전거리를 그림 3을 참조하여 다음과 같이 수식을 유도할 수 있다. 그림 3에서, 첨자 1은 선행 차량과 관계되는 내용이고 첨자 2는 후행차량과 관계되는 내용을 의미한다. x, v, a, 및 j는 각각 차량의 위치, 속도, 가속도, 저크를 의미 한다.

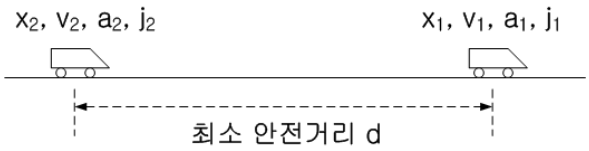

그림 3 비동기식 차량제어방식에서 안전거리
Fig. 3 Safety distance for asynchronous vehicle control

차량 1 즉 선행 차량이 가장 빨리 정차하는 방법은 일단 최대 저크를 이용하여 최대 감속도까지 감속하고 이 후부터는 최대 감속도로 속도가 0이 될 때까지 감속하는 것이다. 수식으로 표현하면 식(1) ~ 식(6)이 된다

$$t_{1j} = (a_1 - Dcc)/Jerk \tag{1}$$

$$y_{11} = v_1 \cdot t_j + \frac{1}{2}a_1 \cdot t_j^2 + \frac{1}{6}(-Jerk) \cdot t_j^3 \tag{2}$$

$$v_{1j} = v_1 + a_1 \cdot t_{1j} + \frac{1}{2}(-Jerk) \cdot t_{1j}^2 \tag{3}$$

$$t_{1d} = -\frac{v_{1j}}{Dcc} \tag{4}$$

$$y_{12} = v_{1j} \cdot t_{1d} + \frac{1}{2}(-Dcc) \cdot t_{1d}^2 \tag{5}$$

$$y_1 = y_{11} + y_{12} \tag{6}$$

여기서

t_{1j} : 1번 차량이 최대 jerk로 감속하여 최대 감속도에 이르는 시간

$Jerk$: 시스템에서 주어지는 최대 jerk

v_{1j} : 1번 차량이 최대 jerk로 감속하여 최대 감속도에 이를 때 1번 차량의 속도

t_{1d} : 1번 차량이 최대 감속도 이르러 jerk=0 가 된 후, 최대 감속도를 적용하여 속도=0 이 될 때까지 걸리는 시간

Dcc : 시스템에서 주어지는 최대 감속도(음의 값)

y_{11} : t_{1j} 시간에 1번 차량의 주행 거리

y_{12} : t_{1d} 시간에 1번 차량의 주행 거리

y_1 : 1번 차량이 감속하기 시작하여 정차할 때까지의 총 주행거리

2번 차량의 주행거리도 유사하게 식(7) ~ 식(13)로 표현된다. 식 (12)에서는 통신 지연 등을 고려한 시간 지연(latency)이 고려된 것이다. 즉 첫 번째 t_l 시간 동안은 v_2의 정속도로 진행한다.

$$t_{2j} = (a_2 - Dcc)/Jerk \tag{7}$$

$$y_{21} = v_2 \cdot t_{2j} + \frac{1}{2}a_2 \cdot t_{2j}^2 + \frac{1}{6}(-Jerk) \cdot t_{2j}^3 \tag{8}$$

$$v_{2j} = v_2 + a_2 \cdot t_{2j} + \frac{1}{2}(-Jerk) \cdot t_{2j}^2 \tag{9}$$

$$t_{2d} = -\frac{v_{2j}}{Dcc} \tag{10}$$

$$y_{22} = v_{2j} \cdot t_{2d} + \frac{1}{2}(-Dcc) \cdot t_{2d}^2 \tag{11}$$

$$y_{23} = v_2 \cdot t_l \tag{12}$$

$$y_2 = y_{21} + y_{22} + y_{23} \tag{13}$$

여기서

t_{2j} : 1번 차량이 최대 jerk로 감속하여 최대 감속도에 이르는 시간

$Jerk$: 시스템에서 주어지는 최대 jerk

v_{2j} : 1번 차량이 최대 jerk로 감속하여 최대 감속도에 이를 때 1번 차량의 속도

t_{2d} : 1번 차량이 최대 감속도 이르러 jerk=0 가 된 후, 최대 감속도를 적용하여 속도=0 이 될 때까지 걸리는 시간

t_l : 제어신호의 통신 지연시간 및 차량제어장치의 반응 지연시간

Dcc : 시스템에서 주어지는 최대 감속도(음의 값)

y_{21} : t_{1j} 시간에 2번 차량의 주행 거리

y_{22} : t_{1d} 시간에 2번 차량의 주행 거리

y_{23} : t_l 시간에 2번 차량의 주행 거리

y_2 : 2번 차량이 감속하기 시작하여 정차할 때까지의 총 주행거리

따라서 안전 주행을 위해서는 항상 식 (14)를 만족시키면서 운행하여야 한다. 앞으로 식(14)를 '안전운행조건_1'이라 부른다.

$$x_1 + y_1 > x_2 + y_2 \tag{14}$$

식 (14)는 정상운전 상태에서의 안전거리 확보이다. 다음은 비정상 상태에서의 즉 위의 안전조건 ②을 만족하는 안전거리를 분석하여 보자. 우선 주행 차량에 대해 jerk가 적용되지 않는다. jerk는 어디 까지나 승객의 승차감을 위한 제어 파라메타이다. 비상운전 시에는 적용되지 않는다. 따라서 1번 차량은 사고로 정지하고 2번 차량은 1번 차량의 사고를 인지한 후 비상제동장치를 사용하여 정지한다. 이를 수식으로 표현하면 식(15) ~ 식(16) 과 같고, 식 (18)이 모든 $t < t_s$에서 만족하여야 한다. 향 후 식 (18)을 '안전운행조건_2'라 부른다.

$$y_3 = v_1 \bullet t + \frac{1}{2} a_{mf} \bullet t^2 \tag{15}$$

$$y_4 = v_2 \bullet t_l + v_2 \bullet (t - t_l) + \frac{1}{2} d_{emer} \bullet (t - t_l)^2 \tag{16}$$

$$t_s = -\frac{v_2}{D_{emer}} + t_l \tag{17}$$

$$x_1 + y_3 > x_2 + y_4 \quad \text{for all} \quad t < t_s \tag{18}$$

여기서

a_{mf} : 최대사고 감속도(maximum failure deceleration ratio), 음의 값

d_{emer} : 최소 비상제동 감속도(minimum emergency deceleration ratio), 음의 값

y_3 : 1번 차량(여기서는 사고 차량이 됨)의 감속 후부터 정지 시까지의 이동 거리

y_4 : 1번 차량 감속 후부터 2번 차량의 주행거리

t_l : 1번 차량의 사고 후 2번 차량의 비상제동장치 작동 시까지의 지연 시간

t_s : 후행차량 즉 2번 차량이 속도 v_2에서 최소 비상제동 감속도(d_{emer})로 감속 시 속도가 0이 될 때까지 걸리는 시간

따라서 정상 모드 주행(오직 선두 차량 만 고려한 주행)에서는 식(14)와 식(18)을 만족하면서 운행하여야 한다

2.2 선행차량과 후행차량(leader- follower) 관계

PRT 안전운행제어 목적상 운행하는 모든 차량은 다음의 2가지 이유로 자신의 선행차량과 후행차량에 대한 명확한 정보를 갖고 있어야 한다. 첫째, 비동기식 운행제어 방식에서 모든 차량은 선행차량과의 안전거리 관계를 유지하면서 운행하여야 하는데 PRT 시스템은 정해진 노선이 없으므로 선행차량이 운행 도중 자주 변한다. 둘째, 실제 운행제어나 혹은 시뮬레이션의 진행에 있어서는 디스크리트(discrete) 한 시간 마다 차량의 제어 파라메타(위치, 속도, 가속도, 저크 등)를 계산하게 된다. 예를 들어 타이스텝이 1초인 경우 5초에 차량의 위치를 계산하였으면 다음에는 6초에 차량의 위치를 계산한다. 따라서 후행차량의 다음 1초간의 운행 방법을 결정하기 위해서는 선행차량이 다음 1초 동안에 어떻게 주행할 것인지에 대한 정보가 필요하다. 이러한 이유로 모든 차량의 운행제어 파라메타 계산은 항상 선행차량에 대한 계산이 선행된 후에 이루어 져야한다. 물론 선행차량이 없는 차량, 즉 맨 선두차량이거나 일정거리 앞까지 운행하고 있는 차량이 없는 경우는 예외가 될 것이다.

2.3 일반구간에서의 안전운행제어

일반 구간에서의 선행차량은 동일한 트랙 내에서 바로 앞서 주행하는 차량을 의미한다. 자기 차량이 트랙 내에서 제일 선두에서 주행 할 경우에는 다음 주행 예정 트랙의 맨 마지막 차량이 선행차량(leader)이 된다. 일반 구간에서 차량의 운행 제어는 안전조건_1 및 안전조건_2가 만족되도록 수행 되어야 한다.

2.4 병합구간에서의 안전운행제어

병합구간의 운행제어는 병합지점에서 일어날 수 있는 차량의 충돌이 일어나지 않도록 병합 구간에 있는 차량의 속도를 제어하는 것을 의미한다. 그림 4에 병합구간에서 운행하는 차량을 나타내었다. 일반 트랙이라면 ②, ③ 차량은 각각 ①, ② 차량과 또한 ⑤, ⑥ 차량은 각각 ④, ⑤ 차량과의 안전거리를 유지하면서 주행하면 된다. 그러나 이 경우 병합지점에 이르렀을 때 다른 트랙에서 접근하는 차량과 충돌할 수 있다. 이를 방지하기 위해서 운행제어 시스템은 병합구간에 있는 차량의 속도를 제어해야 한다. 동기제어(synchronous control)식 방식에서는 각 병합 트랙의 블록을 서로 일치시키고 다른 트랙에 있는 차량을 자신의 트랙으로 가상적으로 옮겨와 최소한 한 블록의 빈 블록을 사이에 두고 차량을 운전하는 방식으로 제어한다. 이 방식은 목적은 달성할 수 있지만 효율적이지는 못하다. 첫째, 그림 5의 예에서 ⑤번 차량의 속도가 ④번 차량의 속도에 비해 매우 빠르다면 ⑤ 번 차량이 ④ 번 차량보다 늦게 가기 위해서 일부러 속도를 감속하여야 하는데 이는 비효율적이다. 병합지점에 빨리 도착할 수 있는 차량부터 빨리 도착할 수 있도록 순서가 정해져야 한다. 또한 병합 구간의 양쪽 트랙이 동일한 속도로 진행한다는 것도 비효율적인 것이다. 본 시뮬레이터에서는 일반 승합차가 고속도로 램프에서 고속도로로 진입하는 방식을 도입하였다. 즉 상대편 트랙에서 주행하는 차량의 속도로서 병합지점에 도달하는 시각을 예측하고 자기차량의 병합지점 도착시각을 예측하여 비교함으로써 병합지점에 자신이 먼저 도착하도록(가속이 필요할 수도 있음) 운전할 것인지 혹은 상대 차량을 먼저 보내고 자신이 다음에 병합지점에 도달하도록(감속운전이 필요할 수도 있음) 운전할 것인지를 결정하는 것이다. 시뮬레이션에서도 병합구간에 있는 모든 차량의 병합지점 도착 예정시간을

계산하여 병합지점 도착 순서를 결정한다. 그림 4의 예에서 병합지점의 예상도착시각에 따라 병합지점 도착 순서가 ①, ②, ④, ③, ⑤, ⑥ 으로 정해지면 car ⑤의 경우 leader car는 ④가 되고 leader2 car는 ③이 된다. 즉 car ⑤는 car ④에 대해 안전운행법칙_1 및 안전운행법칙_2에 의해 주행하고 car ③ 에 대해서는 안전운행법칙_3에 의해 주행하여야 한다. 이 경우 car ⑤의 주행계산을 위해서는 car ④ 및 car ③의 주행계산이 선행되어야 한다.(leader- follower 관계) 안전주행 법칙_3이란 다음 ① ~ ③으로 표현된다.

① leader2의 병합지점 통과 예상시간을 계산한다(t1). 이 때 통과시간뿐 아니라 병합지점 통과 시의 예상속도(v2)도 함께 계산한다.

② 자기 차량이 앞으로 t1시간 동안 주행할 거리와 속도, 위치를 계산한다.

③ ① 및 ②의 결과 leader2 car 및 자기차량의 t1 시간 후 위치, 속도, 가속도가 계산된다. 이 자료로 t1 시간에서 "안전주행법칙 1" 및 "안전주행법칙 2"를 적용한다. 병합구간의 차량이 병합지점 통과 예상시간은 해당 트랙의 최고 허용속도, 최대허용 저크, 최대 가속도의 조건 하에서 가장 빠르게 간다는 가정 하에 계산된다. 결론적으로 병합구간을 운행하는 모든 차량은 안전 주행 법칙_1, 안전주행법칙_2 및 안전주행법칙_3을 준수하며 주행하여야 한다.

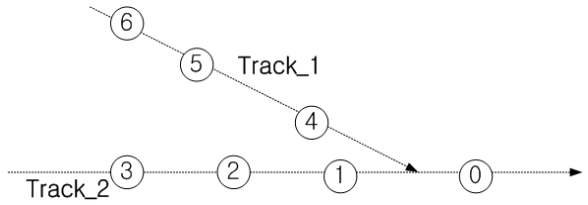

그림 4 병합구간 차량의 차량 주행(비동기식 제어)
Fig. 4 Asyncronous vehicle control at the merging point

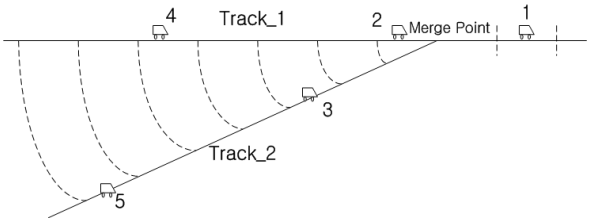

그림 5 병합구간 차량의 차량 주행(동기식 제어)
Fig. 5 Synchronous vehicle control at the merging point

2.5 분기(Diverge) 구간에서의 안전 운행

분기 구간에서 안전주행에 관한 특별한 법칙은 필요하지 않다. 다만 Leader-Follower 관계에 대해 다음과 같이 유의해야 한다.

아래 그림 6에서 track_C를 분기(diverge) 구간이라 부른다. 그림 7에서 차량 3은 트랙 C에 있다가 다음 타임 스텝에서 트랙 A로 주행한 경우이고, leader-follower 관계는 다음과 같이 되어 있을 것이다.

car[4].leader=3

다음 순간 leader-follower 관계는 car 4의 다음 트랙에 따라 2가지 경우가 생긴다.

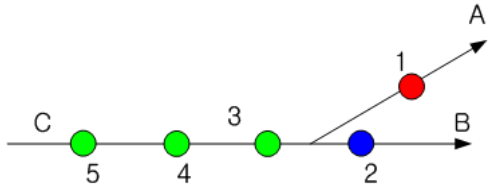

그림 6 분기구간 차량 (타임=n)
Fig. 6 Vehicle control at diverging point(time=n)

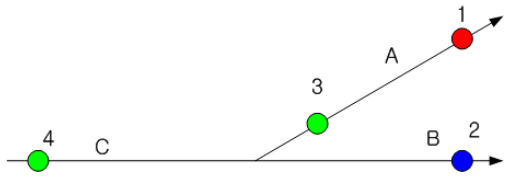

그림 7 분기구간 차량 (타임=n+1)
Fig. 7 Vehicle control at diverging point(time=n+1)

먼저 car 4의 다음 트랙이 track_A인 경우

car[4].leader=3

즉 아무 변화가 없다. 다음 car 4의 다음 트랙이 track_B인 경우에는

car[4].leader=2

즉 선행차량이 바뀌어야 한다. 일반적으로

① 분기구간을 주행하는 차량이 트랙에서 2번째 이면 leader(첫번째 차량) 차량과 다음 트랙을 비교한다.

② ①의 비교 결과 동일하면 일반 구간의 주행과 동일한 방법으로 주행한다.

③ ①의 비교 결과 다음 트랙이 서로 다르면 다음 트랙의 맨 마지막 차량이 leader_2 차량이 되고 leader 차량 및 leader_2 차량에 대하여 안전주행법칙_1 및 안전주행법칙_2를 준수하며 주행하여야 한다.

2.6 시뮬레이션 수행

본 알고리즘을 적용한 시뮬레이션 프로그램을 작성하여 시험하였다.

(1) 시험 노선 및 차량 제원

시험선은 그림 8과 같다. 시험선의 규모는 4 loop x 4 loop 트랙으로 구성하였다. 모든 트랙은 단방향으로 설정되어 있다. 1개 loop는 사방 약 500m이므로 전체 네트워크는 사방 2 km 정도이고 총 역사 수는 40개이다. 각 변의 중심에 역사를 위치하였으므로 승객은 최대 250m 이내에 역사에 접근할 수 있다. 시험차량의 제원은 최대운행 가속도 1.5 m/s/s, 최대운행 감속도 -1.25 m/s/s, 최대 허용 jerk 1.25 m/s/s/s, 최대 사고 감속도 -2.5 m/s/s, 최소 비상 감속도 -4.0 m/s/s, 차량 길이 2.5 m 로 가정하였다.

(2) 최소시격 및 병합구간 제어 검토

최소시격 및 병합구간 제어 검토를 위하여 역사 30에서 역사 68로 역사 84에서 역사 68로 각각 시간당 3600대의

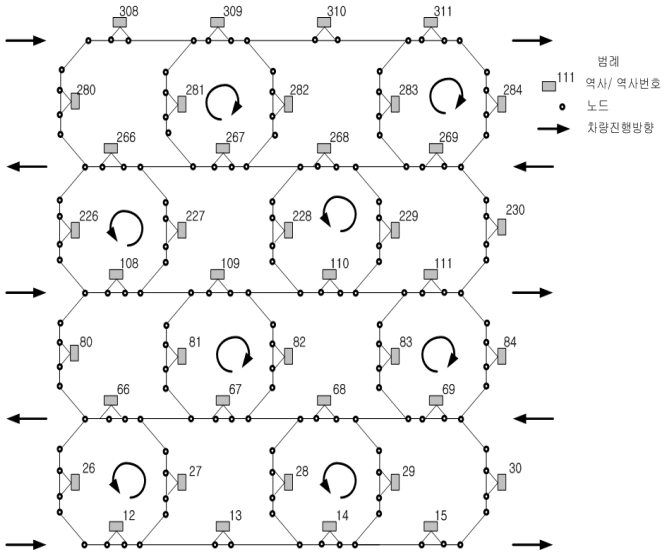

그림 8 시험선 트랙, 역사
Fig. 8 Track layout for simulation

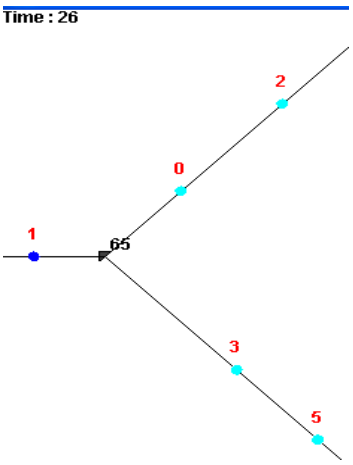

그림 9 시뮬레이션 결과(time=26
Fig. 9 Simulation result(time=26)

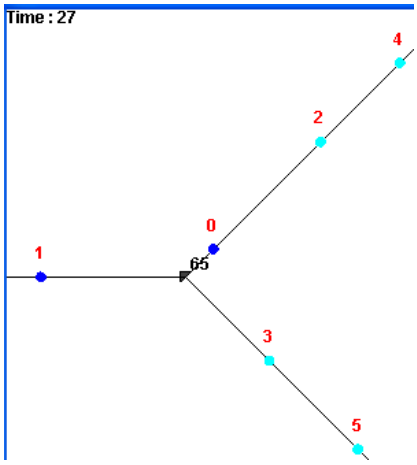

그림 10 시뮬레이션 결과(time=27)
Fig. 10 Simulation result(time=27)

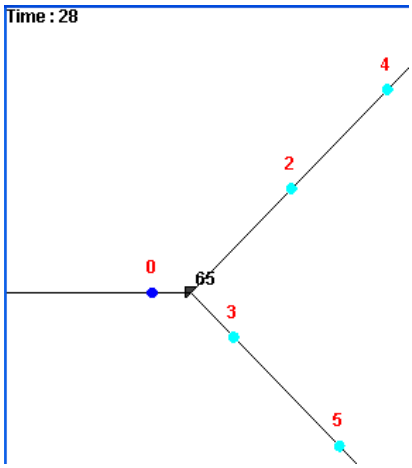

그림 11 시뮬레이션 결과(time=28)
Fig. 11 Simulation result(time=28)

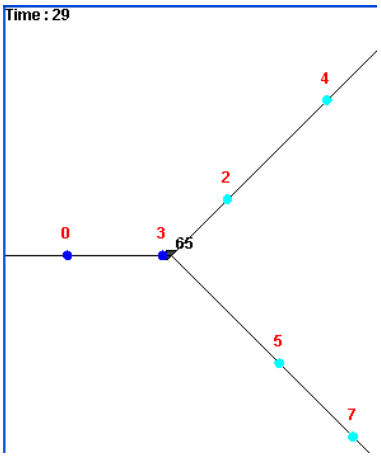

그림 12 시뮬레이션 결과(time=29)
Fig. 12 Simulation result(time=29)

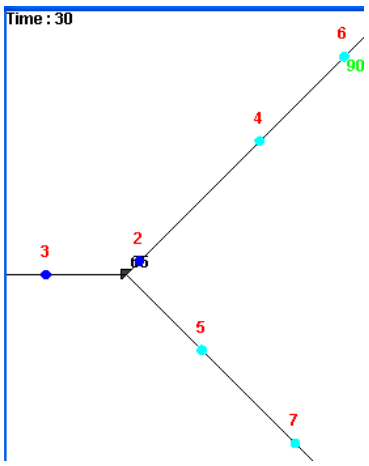

그림 13 시뮬레이션 결과(time=30)
Fig. 13 Simulation result(time=30)

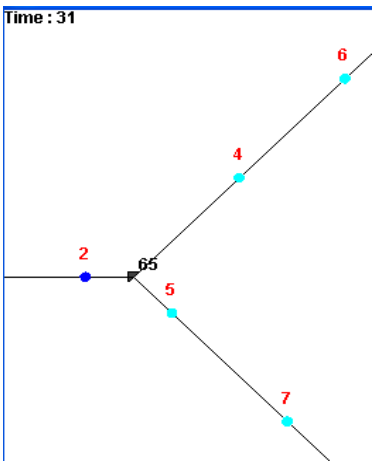

그림 14 시뮬레이션 결과(time=31)
Fig. 14 Simulation result(time=31)

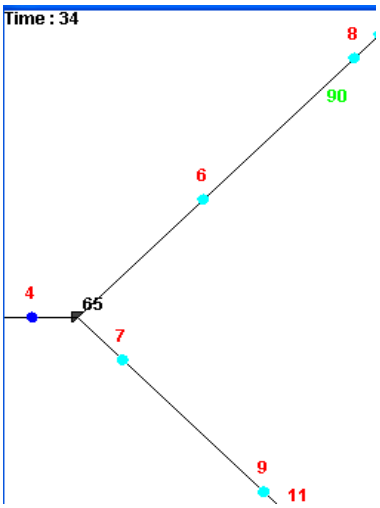

그림 15 시뮬레이션 결과(time=34)
Fig. 15 Simulation result(time=34)

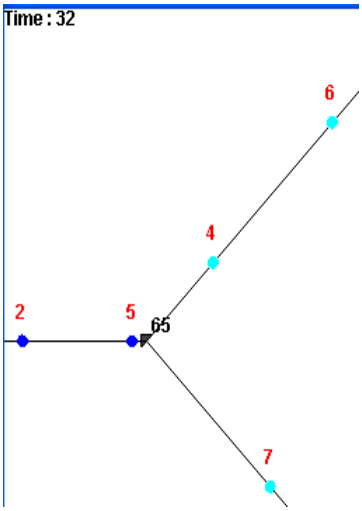

그림 16 시뮬레이션 결과(time=32)
Fig. 16 Simulation result(time=32)

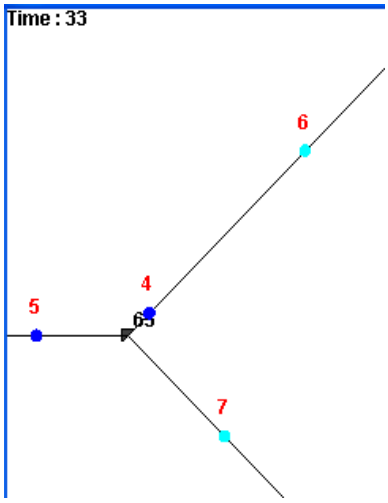

그림 17 시뮬레이션 결과(time=33)
Fig. 17 Simulation result(time=33)

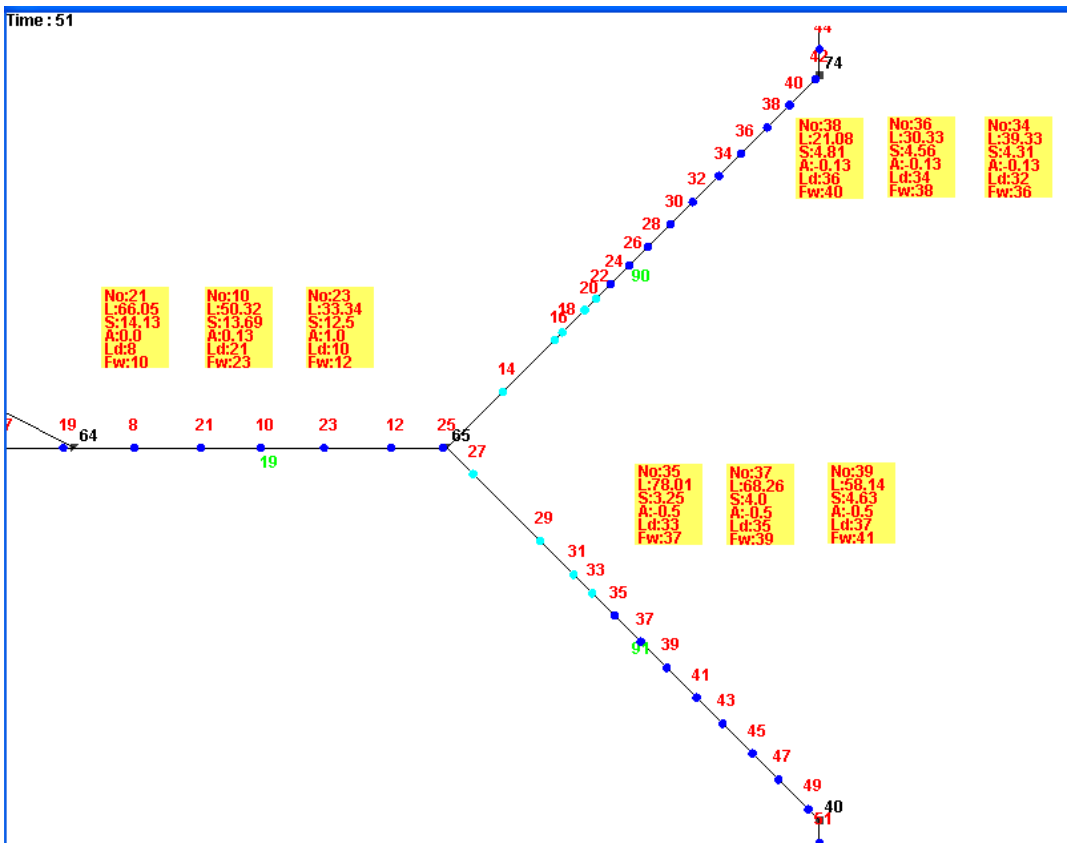

그림 18 시격 검토
Fig. 18 Headway investigation about simulation

차량을 배차하여 차량 주행모습을 검토하였다. 병합구간 제어는 시간의 진행에 따라 주행하는 모습을 관찰해야 하므로 그림 9~ 그림 17에 보여 주었다. 그림에서 작은 원 혹은 점은 차량이며 옆의 숫자는 차량번호(id)이다. 시격과 관련하여 병합구간을 통과한 후 (트랙 19) 시격은 입력 자료에서 요구한 대로 1초대 근방이다. 병합하는 2 트랙(트랙 90, 트랙 91)은 병합 후 시격 1.0 초를 유지하기 위해 초기 시격(1초)보다 지연되기 시작한다.(그림 18 및 표 1 참조)

표 1 병합구간에서 시격(최소시격 1초 경우)
Table 1 Headway at merging point
(minimum headway: 1 sec)

트랙 19				
차량	위치	속도	거리 - 차량길이	시격[초]
21	66.05	14.13		
10	50.32	13.69	13.23	1.0
23	33.34	12.5	14.48	1.16
트랙 90				
차량	위치	속도	거리 - 차량길이	시격
34	39.33	4.31		
36	30.33	4.56	6.50	1.43
38	21.08	4.81	6.75	1.40
트랙 91				
차량	위치	속도	거리 - 차량길이	시격
35	78.01	3.25		
37	68.26	4.0	7.25	1.81
39	58.14	4.63	7.62	1.65

3. 결 론

제시된 알고리즘을 이용하면 PRT 차량의 운행 제어는 충분히 가능하다. 본 알고리즘은 참고문헌 2에서 제시하는 알고리즘에 비해 훨씬 간단하며 효율적이다. 본 알고리즘은 또한 시뮬레이션뿐만 아니라 실제 차량 제어 시스템에서도 동일하게 이용될 수 있다. 현재 시뮬레이션 상에서는 보여지는 최소시격은 1초지만 이것은 시뮬레이션 타임스텝이 1초이기 때문이다. 타임스텝을 0.5초로 줄이면 0.5초의 시격 제어도 가능하다. 문제는 차량과 중앙 운행제어 센타 간의 통신의 신뢰도 및 차량 자체 제어의 신뢰도 이다. 연속적인 통신의 실패 및 차량제어의 실패를 몇 번 허용할 것인지가 시스템 최소 시격을 정하는 요소가 될 것이다.

감사의 글

본 연구는 국토해양부 교통체계효율화사업 “승객여정선택형 대중교통(PRT) 운영기술개발” 연구과제에 의하여 이루어진 연구로서, 관계부처에 감사 드립니다.

참 고 문 헌

[1] Jack H Irving, Harry Bernstein, C. L. Olson, Jon Buyan “Fundamentals of Personal Rapid Transit” Lexington Books, 1997

[2] Markus Theodor Szillat ″A low level PRT Microsimulation″ Dissertation, University of Bristol, 2001

저 자 소 개

정 상 기 (鄭 相 基)

1974년 2월 : 서울공대 전기공학과 졸 1980년 12월 : 미국 위스칸신주립대 전기공학과 석사, 2004년 2월 : 명지대학교 전기공학과 박사, 1997년 2월 ~ 한국철도기술연구원 수석연구원

정 락 교 (鄭 樂 敎)

1991년 2월 : 인하대학교 전기공학과(공학사), 1999년 8월 : 인하대학교 전기공학과(공학석사), 2005년 2월 : 인하대학교 전기공학과(공학박사), 1990년 12월 ~ 1994년 12월 : 한진중공업 사원, 1995년 1월 ~ 현재 : 한국철도기술연구원 열차제어통신연구실 책임연구원

김 백 현 (金 伯 鉉)

1994년 2월 : 인하대학교 전자공학과(공학사), 1996년 2월 : 인하대학교 전자공학과(공학석사), 2003년 2월 : 인하대학교 전자공학과(공학박사), 2003년 3월 ~ 현재 : 한국철도기술연구원 열차제어통신연구실 선임연구원