한국철도학회논문집
제12권 제6호 2009년 ▌pp. 878-886

에너지 최적의 열차 속도 궤적 생성을 위한 GA 기반 알고리즘

A GA-Based Algorithm for Generating a Train Speed Profile Optimizing Energy Efficiency

강문호† · 한문섭*
Moonho Kang · Moonseob Han

Abstract This paper proposes an optimal algorithm for generating a train speed profile giving optimal energy efficiency based on GA (Genetic Algorithm) and shows its effectiveness with simulations. After simplifying the train operation mode to a maximum traction, a coasting and a maximum breaking, adjusting the coasting point to minimize the train consuming energy is the basic scheme. Satisfying the two constraints, running distance and running time between two stations, a coasting point is determined by GA with a fitness function consisting of a target running time. Simulation results have shown that multiple coasting points could exist satisfying both of the two constraints. After figuring out consumed energies according to the multiple coasting points, an optimal train speed profile with a coasting point giving the smallest consumed energy has been selected. Simulation blocks for the train performance simulation and GA have been designed with the Simulink.

Keywords : GA (Genetic Algorithm), optimal energy efficiency, train speed profile, coasting

요 지 본 논문에서는 열차 운전시 최적의 에너지 효율을 얻기 위해서 GA(Genetic Algorithm)를 이용하여 열차 속도 프로파일을 생성하는 최적 알고리즘을 제시하고 시뮬레이션을 통해 유효성을 보였다. 역간의 열차 운전 모드를 최대 역행, 타행, 최대 제동으로 간략화 시키고 타행지점을 조절하여 열차 운전시 소비되는 에너지를 최소화 시키는 방식을 기본으로 하여, 정해진 거리의 두 역간을 정해진 운전시간 내에 도달하기 위하여 목표 운전시간을 이용하여 적합도(Fitness) 함수를 설정한 후, GA 알고리즘을 적용하여 역간 거리와 목표 운전시간의 두 제한요소를 모두 만족시키는 타행 지점들을 결정하였다. 시뮬레이션 결과 두 제한 요소들을 만족하는 타행 지점이 여러 개가 존재함을 확인하였고, 각 타행지점들에 따른 소비 에너지를 도출하여 에너지 소비량이 가장 적은 타행지점을 선정하여 최종적인 열차 속도 프로파일을 결정하였다. 시뮬레이션을 위해 Simulink를 이용하여 열차성능 시뮬레이션 블록들과 GA 블록들을 설계하였다.

주 요 어 : 유전알고리즘, 최적 에너지 효율, 열차 속도 프로파일, 타행

1. 서 론

철도는 단위 수송량당 에너지 소모량과 이산화탄소 배출량이 타 교통수단에 비해 현저히 적은 친환경 교통수단으로서 최근에 철도노선에 대한 전철화 구간이 양적 질적으로 크게 증가하고 있다. 이에 따라 전기에너지 사용도 계속 증가할 것으로 예상되어 철도에서의 효율적인 에너지 사용이 요구된다. 기존의 철도의 에너지 저감은 크게 차체와 내부장치를 비롯한 차량설계분야, 대체에너지 사용, 열차 운전 방식의 개선 등 철도의 하드웨어와 소프트웨어 전 분야에 걸쳐 이루어지고 있다. 이들 중에서 하드웨어 관련분야는 에너지 저감 효과가 크지만 일반적으로 오랜 시간이 소요되고 새로운 설비가 요구되어 비용이 상승하며 기존의 시스템에는 적용이 어려운 반면에, 소프트웨어적인 방법들은 비교적 적용이 수월하고 적은 비용으로 기존의 철도에도 사용이 가능한 장점을 지닌다. 특히, 에너지 가격 상승과 함께 환경에 대한 관심이 고조됨에 따라 기존 열차 및 자동 열차 주행 시스템 등에 새로운 열차운전 기법을 적용

† 책임저자 : 정회원, 선문대학교 정보통신공학과 부교수
E-mail : mhkang64@gmail.com, mhkang@sunmoon.ac.kr
TEL : (041)530-2339 FAX : (041)530-2309
* 정회원, 한국철도기술연구원(KRRI) 전철전력연구실 책임연구원

하여 에너지 효율을 최적화하기 위한 다양한 시도들이 이루어지고 있다[1].

열차 에너지 효율을 최적화하기 위한 기법들은 일반적으로 열차소비에너지와 열차주행시의 제반조건들을 이용하여 목적함수를 설정하고 이를 최소화시킬 수 있는 최적의 열차 운전모드를 도출하는 형태를 취하는데, Pontryagin의 최대화원리나 Lagrange 배수에 의한 최적화 과정을 통해 해석적으로 최적해를 계산하는 방식을 비롯하여[2,3] DP (Dynamic Programming), NLP(Nonlinear Programming) 등 수치해석적으로 해를 구하는 방식[4], 퍼지, GA(Genetic Algorithm) 등의 지능적 기법들을 이용하는 방식[5,6]들이 있다.

본 논문에서는 열차 운전시 최적의 에너지 효율을 얻기 위한 열차 속도 프로파일을 찾아내기 위하여 GA(Genetic Algorithm)를 적용하고 시뮬레이션을 통해 유효성을 보였다. 역간을 운행하는 열차의 운전 모드를 최대 역행, 타행, 최대 제동으로 간략화 시킨 후, 타행지점을 조절함으로서 주행기간동안에 열차에서 소비되는 에너지를 최소화 시키는 방식을 기본 알고리즘으로 한다. 장거리 열차 운전시에는 타행과 가속이 반복되어 사용되지만, 운전구간이 짧은 지하철 구간 등에서는 일반적으로 가속, 타행, 제동의 단순한 주행패턴의 운전이 주로 이루어지고 있고, 이론적으로도 가속, 타행, 제동으로 최대 효율을 얻을 수 있기 때문에[2] 짧은 구간 내에서는 가능하다면 열차의 운전 모드를 가속, 타행, 최대 제동으로 간략화 시키는 방식이 효율적인 것으로 판단된다.

본 논문에서는 일차적으로 GA의 적용 가능성을 검토함을 주목적으로 하여 1km의 짧은 역간 주행 거리를 상정하고, 역간을 정해진 운전시간 이내에 도달하기 위하여 목표 운전시간을 이용한 적합도(Fitness) 함수를 설정한 후, GA를 적용하여 역간 거리와 목표 운전시간의 두 제한요소를 모두 만족시키는 타행 지점들을 산출하였다. 시뮬레이션 결과 두 제한 요소를 만족하는 타행 지점이 주행구간동안 여러 개가 존재함을 확인하였고, 각 타행지점들에 따른 소비 에너지를 도출하여 에너지 소비량을 최소로 하는 타행 지점을 선정하여 최종적인 열차 속도 프로파일을 결정하였다. 시뮬레이션을 위해 Simulink[7]를 이용하여 열차성능 시뮬레이션 블록들과 GA 블록들을 설계하였다.

2. 열차 성능 시뮬레이션(TPS) 모델

2.1 열차운동 방정식

선로를 주행하는 열차의 운동방정식을 나타내면 식 (1)과 같다. 식 (2)는 식 (1)에 포함된 열차저항을 구성하는 성분들을 보이고, 식 (3)은 이들 중에서 주행저항을 나타낸다[8]. 식 (4)는 열차가 주행하는 동안 열차에서 소비된 총에너지를 보이는 것으로 제동시는 에너지가 회생됨을 가정하였다. 열차의 운전모드는 견인력과 제동력의 인가상태에 따라 역행, 타행, 제동모드로 분류되고 이를 표로 나타내면 표 1과 같다.

$$T(v) - R(v,i,r) - B(v) = M_e \frac{dv}{dt} \tag{1}$$

$$R(v,i,r) = R_s(v) + R_r(v) + R_g(i) + R_c(\gamma) + R_t \tag{2}$$

$$R_r(v) = c_1 + c_2 v + c_3 v^2 \quad (c_1, c_2, c_3 > 0) \tag{3}$$

$$E(t) = \int_0^{t_f} (T(v) - B(v))v(t)dt \tag{4}$$

$T(v)$:견인력[N], $B(v)$:제동력[N], $E(t)$:에너지[kWh]
$R(\cdot)$:열차저항[N], M_e:열차유효질량[ton]
v:열차속도[m/s], t:시간[s], t_f:열차주행시간[s]
i:구배[‰], γ:곡선반경[m], c_1, c_2, c_3:주행저항 계수
$R_s(v)$:출발저항[N], $R_r(v)$:주행저항[N]
$R_g(i)$:구배저항[N], $R_c(\gamma)$:곡선저항[N], R_t:터널저항[N]

Table 1. Train operation modes

열차 운전모드	견인력($T(v)$)	제동력($B(v)$)	에너지($E(t)$)
역행	+	0	증가
타행	0	0	불변
제동	0	+	감소

열차의 이산치 다이내믹을 구하기 위해 열차 주행거리를 독립변수로 하여 열차 주행시간과 속도를 표현하면 식 (5), (6)과 같다. 식 (7)은 식 (5), (6)에서의 속도와 가속도 평균값을 나타내고, 식 (8)은 i번째 스텝에서의 열차 가속도를 나타내고 식 (1)로부터 구해진다.

$$t = \int \frac{ds}{v} = \lim_{\Delta s_i \to 0} \sum_i \frac{\Delta s_i}{\bar{v}_i} \tag{5}$$

$$v = \int \frac{ads}{v} = \lim_{\Delta s_i \to 0} \sum_i \frac{\bar{a}_i \Delta s_i}{\bar{v}_i} \tag{6}$$

$$\bar{v}_i = \frac{v_i + v_{i+1}}{2}, \quad \bar{a}_i = \frac{a_i + a_{i+1}}{2} \tag{7}$$

$$a_i = \frac{F_i}{M_e} \quad (F = T(v) - R(v,i,r) - B(v)) \tag{8}$$

s:거리[m], Δs:거리증분[m], a:가속도[m/s^2]
$\bar{v}$:평균속도[m/s], $\bar{a}$:평균가속도[m/s^2], F:총견인력[N]

$\Delta s_i (= s_{i+1} - s_i)$가 충분히 작다면 식 (5)로부터 열차주행시간에 대한 이산식은 다음과 같이 표현되고

$$t_{i+1} = t_i + \frac{s_{i+1} - s_i}{\overline{v_i}} \tag{9}$$

Δs_i 동안 열차의 가속도도 일정하다고 할 수 있으므로 $\overline{a_i} = a_i$ 가 되므로, 식 (6)으로부터 식 (10)과 같이 열차주행 속도에 대한 이산식을 구할 수 있다.

$$v_{i+1} = v_i + \frac{a_i(s_{i+1} - s_i)}{\overline{v_i}} = v_i + \frac{2a_i(s_{i+1} - s_i)}{v_{i+1} + v_i}$$
$$\Rightarrow \quad v_{i+1} = \sqrt{v_i^2 + 2a_i(s_{i+1} - s_i)} \tag{10}$$

한편, 소비 에너지에 관한 이산치 식은 식 (4)로부터 다음과 같이 구해진다.

$$E_{i+1} = E_i + (T_i(v) - B_i(v))(s_{i+1} - s_i) \tag{11}$$

본 논문에서는 식 (9)~(11)을 기본 식들로 하여 열차 동특성 이산치 모델을 구성하고 주행거리에 따른 열차성능에 대한 시뮬레이션을 행하였다.

2.2 열차 주행 에너지 최적 알고리즘

2.2.1 에너지 최적 속도 프로파일

지하철구간과 같이 거리가 짧은 역간을 운행하는 열차의 경우 운행모드는 대략적으로 최대 역행, 타행, 최대 제동으로 간략화 시킬 수 있고, 이 경우 열차의 에너지 소비도 최소가 된다[2]. 이에 따라 본 논문에서는 열차의 운행모드를 Fig. 1에서 보이는 것과 같이 최대 역행, 타행, 최대 제동으로 간략화 시킨 후, 타행이 시작되는 지점을 조절하여 전체 운행구간 동안에 열차에서 소비되는 에너지를 최소화 시키는 방식을 채택하였다. 이때, 열차의 속도 프로파일이 정해진 역간 거리(s_D)를 정해진 운전시간(t_D)에 운행하여야 하는 다음의 두 가지 제한 요소들을 만족시켜야 한다.

$$s = s_D,\ t = t_D \ (s_D\text{:역간 거리},\ t_D\text{:역간 목표시격}) \tag{12}$$

2.2.1.1제동점 결정

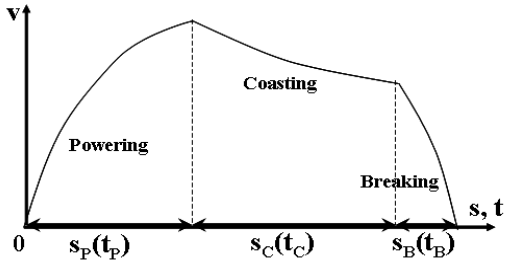
식 (12)에서 첫 번째 제한 요소로서 열차가 목표역에 정확히 정지하기 위해서는, 정확한 제동 개시지점이 결정되어야 하는데, 이를 위해 역행과 타행에 대한 시뮬레이션을 통해 먼저 열차의 전방향 속도궤적을 구한 다음, 제동에 대한 시뮬레이션을 통해 열차의 후방향 속도 궤적을 계산하고, 두 궤적의 교차점을 제동 시작지점으로 결정한다[9].

2.2.1.2타행점 결정

식 (12)에서 두 번째 제한 요소인 열차가 정해진 시간 내에 목표지점에 도착하기 위해서, 열차의 실제 주행시간(t)과 목표 주행시간(t_D)과의 차이를 이용하여 2.2.2절의 식 (13)과 같이 적합도(Fitness) 함수를 설정하고 이를 최소로 하는 타행 개시지점을 찾기 위하여 GA(Genetic Algorithm)를 적용한다[10]. GA적용에 관한 상세한 내용은 2.2.2절에서 설명한다.

2.2.1.3 최적 프로파일 결정

한편 열차 주행 모드에 따라서, 동일한 주행구간 동안에 두 가지 제한 요소를 만족하는 타행점과 제동점, 즉 속도 프로파일이 여러 개 존재할 수 있으므로, 여러 차례 GA를 반복 수행하여 속도 프로파일들을 구하고 이들 중에서 열차 소비에너지를 최소로 하는 타행점과 제동점을 가지는 속도 프로파일을 최종적인 열차 속도 프로파일로 결정한다.



$s_D = s_P + s_C + s_B$ $(t_D = t_P + t_C + t_B)$

s_D:역간거리, s_P:역행거리, s_C:타행거리, s_B:제동거리
t_D:역간목표시격, t_P:역행시간, t_C:타행시간, t_B:제동시간

Fig. 1. Train operation modes

2.2.2 GA(Genetic Algorithm)

본 연구에서 최적 타행점을 구하기 위해 사용되는 GA의 각 구성요소들을 간략하게 나타내면 다음과 같다.

Coding: 2진수를 이용하여 타행지점을 부호화.

Population: 타행지점들의 모집단으로부터 랜덤하게 초기의 해집단을 생성.

Evaluation: TPS(Train Performance Simulation)를 통해 해집단의 각 멤버들에 대한 적합도 값을 산출.

Selection: 해집단의 멤버들 중에서 적합도 값이 가장 큰 두 멤버를 부모 멤버로 선택.

Crossover: 부모의 유전자 비트 일부를 랜덤하게 선택, 교배시켜 새로운 해집합을 생성.

Mutation: 국지 최적해에 빠지지 않도록 해집단의 비트 일부분을 랜덤하게 변화시킴.

앞 절에서 설명한 바와 같이 열차 운행시 역간 열차 주행

시간(t)이 목표 주행시간(t_D)과 같아져야 하는 시간제한 요소를 만족시키기 위해 식(13)과 같이 적합도 함수(f)를 설정하는데, 해집단(타행 개시지점)의 각 멤버를 이용하여 TPS를 수행하면 각 타행 개시지점에 대하여 주행거리에 따른 주행시간(t)이 산출되고 이로부터 f를 계산할 수 있다. 한편, 식 (13)에서 t_D값은 해당 운전구간에 대한 실제적인 운전상황을 고려하여 설정하는 것이 타당하지만 본 연구에서는 열차 다이내믹과 선로 조건을 실제와 달리 단순화한 모델을 사용하였기 때문에 이 모델에 대한 열차 성능 시뮬레이션 결과(Fig. 11)를 토대로 하여 적당한 t_D값을 설정한다.

해집단의 모든 멤버들에 대해 f가 구해지면, 이들 중에서 f가 큰 상위 2개의 타행 개시지점을 부모로 하여 교배와 돌연변이를 통해 새로운 해집단을 생성한다. 이때, 부모 염색체는 보존시킴으로서 진화과정을 단순화시키고 해집단의 수렴속도가 빨라지도록 한다. 이후 새로운 해집단을 이용하여, 정해진 탐색횟수에 도달할 때 까지 위의 과정을 반복하여 최종적인 해집단을 생성한다.

$$f = K\left|\frac{t-t_D}{t_D}\right| \ (K<0) \tag{13}$$

f:적합도함수(fitness function), K:상수

끝으로, 열차 주행 모드에 따라 동일한 주행구간 내에 해집단이 여러 개 존재할 수 있으므로, 여러 차례 GA를 반복 수행하여 해집단을 구하고(4장의 Case I, II, III) 이들 중에서 소비 에너지를 최소로 하는 해집단에 의한 속도 프로파일을 최종적인 열차 속도 프로파일로 결정한다.

3. 시뮬레이션 프로그램 설계

본 연구에서는 Simulink[7]의 그래픽 툴박스들을 이용하여 윈도우상에서 전체 시스템 블록들을 객체 지향적으로 설계함으로서 시스템의 설계와 변경 및 해석이 용이하도록 하였다. 시뮬레이션 프로그램은 크게 전후향 속도궤적 산출블록, 통합 TPS(Train Performance Simulation)블록, GA 블록으로 구성되고 각각의 동작을 설명하면 다음과 같다.

3.1 전-후향 속도궤적 산출 블록

Fig. 2는 Simulink로 구성한 전-후향 속도궤적 산출 블록을 보이는 것으로, 최대 역행과 타행을 통해 열차가 출발지로부터 목표지점을 향해 주행하는 동안의 속도궤적을 산출하기 위한 전향 속도궤적 산출블록(Forward(Powering/Coasting)), 최대 제동력으로 열차가 목표지점으로부터 출발지점을 향해 거꾸로 후향 주행하는 속도궤적을 산출하기 위한 후향 속도궤적 산출블록(Backrward(Breaking)), 순차적으로 전-후향 속도궤적 산출블록을 동작시키고 두 속도궤적의 교차점을 계산하여 제동 개시지점을 구하는 Supervisor 블록으로 구성된다.

전향 속도궤적 산출블록에서는 식 (9)-(11)을 이용하여 일정하게 주행 거리를 증가시키며(DeltaS: Δs) 거리에 따른 열차주행시간, 속도, 가속도, 소비에너지들을 구하고, 후향 속도궤적 산출블록에서는 거리증가를 음으로 하여(-DeltaS) 후향 궤적을 산출한다. 한편, Δs를 크게 할수록 시뮬레이션 속도는 빨라지나 너무 크게 증가시키면 시뮬레이션 결과에 오차가 크게 수반될 수 있고, 반대로 Δs_i를 작게 하면 시뮬레이션 정밀도는 증가하나 수행시간이 증가하므로 시뮬레이션 시간과 정밀도를 고려하여 적절한 Δs_i 값을 선정한다.

Fig. 3은 전향 속도궤적 산출블록의 내부 구성을 보인 것이다. 시뮬레이션이 시작되어 열차의 속도-최대견인력 관계가 look-up 테이블 형태로 저장되어진 F_Traction블록으로부터 열차가 정지한 초기상태 ($v_0=0$)에서 열차를 최대로 역행하기 위한 견인력이 얻어지면, F_resist블록에서 계산되는 주행저항과 견인력과의 차이가 구해진 후, Force2Accel블록에서 식 (8)에 의해 초기 가속도가 계산되고, 식 (10)을 이용하여 Speed & Dist Calculation블록에서 다음 스텝에서의 열차 속도가 계산된다. 열차속도는 Time Calculation블록에 인가되어 식 (9)를 이용하여 다음 스텝에서의 주행 시간이 계산되고, 최종적으로 Energy 블록에서 소비에너지가 계산되면 시뮬레이션의 한 사이클이 종료된다. 이후, 새로운 속도값을 이용하여 위의 과정을 반복하고 열차 주행거리가 목표지점보다 커지게 되면 전향 속도궤적 시뮬레이션을 중단한다. 시뮬레이션 도중에 산출되는 열차 위치와 속도 데이터는 Fig. 2의 Supervisor블록에 의해 저장된다.

Fig. 4는 후향 속도궤적 산출블록의 내부 구성을 보인 것으로, 열차의 초기위치를 목표지점으로 설정하고 거리증가를 음으로 하여(-DeltaS) 출발지점으로의 후향 궤적을 산출한다. 시뮬레이션이 시작되면 열차의 속도-최대제동력 관계를 저장한 look-up 테이블인 F_Breaking블록으로부터 열차가 정지한 초기상태 ($v_0=0$)에서 열차의 최대 제동력이 얻어지고, 이로부터 순차적으로 감속도, 속도, 시간 등을 계산한다. 한 사이클이 종료되면, 새로운 속도값을 이용하여 위의 과정을 반복한다.

Supervisor블록은 후향 속도궤적 시뮬레이션 동안 거리별 열차 속도가 계산되면 이 값을 전향 속도궤적 시뮬레이션시 저장해둔 거리별 열차 속도와 매스텝 비교하여, 후향

속도가 전향 속도보다 증가하게 되면 후향 속도궤적 시뮬레이션을 중단하고 이때의 열차위치를 저장하여 제동개시지점으로 설정한다[9].

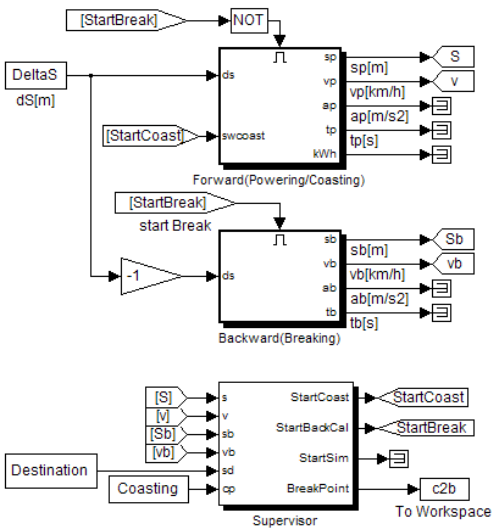


Fig. 2. Forward-backward speed profile calculation blocks

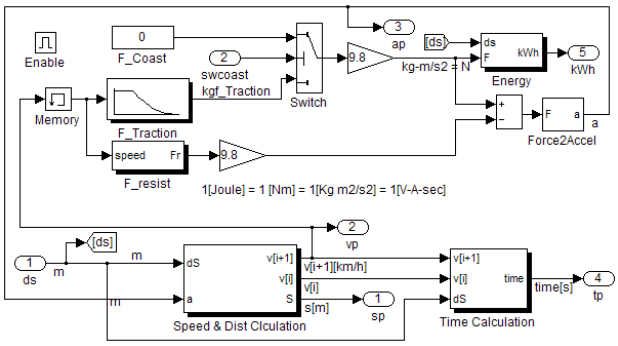


Fig. 3. Configuration of the Forward(Powering/Coasting) block of Fig. 2

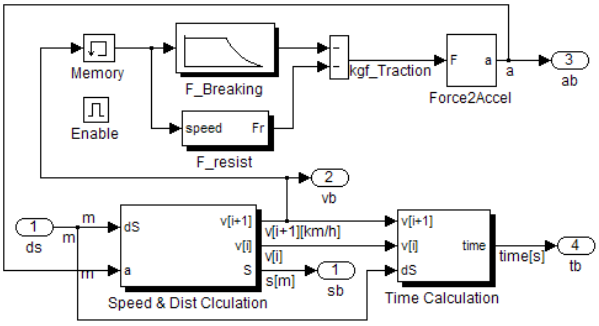


Fig. 4. Configuration of the Backward(Breaking) block of Fig. 2

3.2 통합 TPS 블록

타행지점과 전-후향 속도 궤적의 교차점으로부터 제동지점이 결정되면 이들을 이용하여 목표지점까지 일정하게 주행 거리를 증가시키며 속도 궤적, 주행시간, 소비 에너지 등을 재산출하고 최종적으로 출력하는데, 이를 위한 Simulink 블록을 Fig. 5에 나타냈다. 그림에서 DeltaS, Coasting, Co2Br 은 각각 주행 거리 증가분, 타행 개시지점, 제동 개시지점을 나타낸다. Fig. 6은 Fig. 5의 TPS 블록 내부 구성을 보이는 것으로 견인력 테이블과 제동력 테이블이 모두 포함되고, 각 구성 블록들은 전-후향 속도궤적 블록에서 사용된 블록들과 동일한 내부 구조를 가진다. Fig. 7은 Fig. 6의 Speed & Distance 블록의 내부를 보이는 것으로 식 (10)에 의해 가속도와 거리의 증가분을 이용하여 열차의 주행속도를 계산하는 블록이다. Fig. 8은 전체 TPS 흐름도를 보이는 것으로 흐름도의 각 단계를 Fig. 2와 5에서의 Simulink 블록들과 연관시켜 표시하였다.

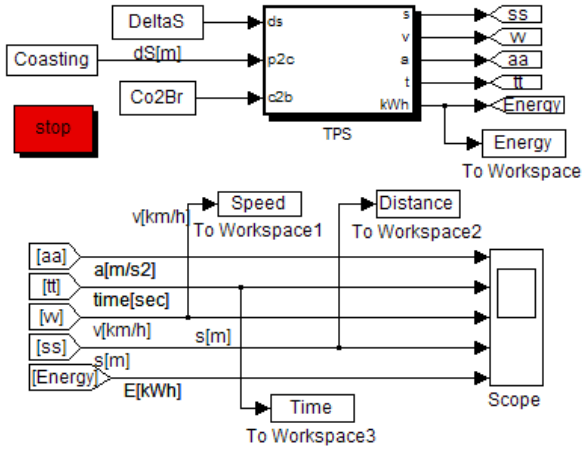


Fig. 5. Forward and backward speed profiles integrated TPS block

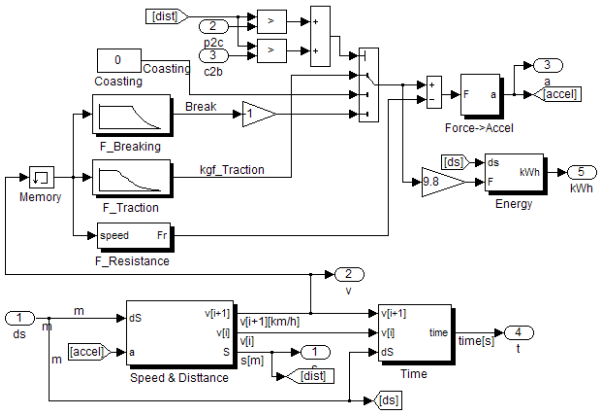


Fig. 6. TPS block's configuration in Fig. 5

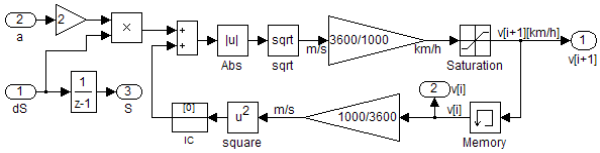


Fig. 7. Configuration of the Speed & Distance block

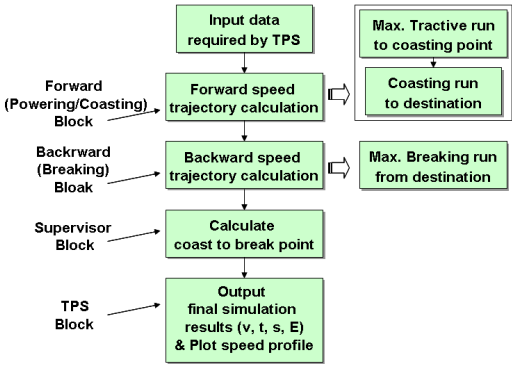


Fig. 8. Total TPS flowchart

3.3 GA 블록

2.2.2절에서 설명된 GA 기능들을 구현하기 위해 먼저 C-코드를 이용하여 필요한 함수들을 작성한 후, Matlab의 Legacy C-코드 생성툴을 이용하여 C-코드 GA 함수들로부터 Simulink상에서 사용할 수 있는 c-sfunction 블록들을 생성하였다[11]. Fig. 9는 이들을 서로 연계하여 구성한 GA 블록을 보이는 것으로, 열차의 타행지점들로 구성된 모집단으로부터 랜덤하게 초기 타행지점들을 선정하고, GApickchroms 블록에서 이들을 이용하여 TPS를 행하며 식 (13)의 적합도를 계산한 후 계산 결과에 따라 부모와 자식 염색체(타행지점) 들을 선택하여 출력하면, GAcrossover 블록에서 이를 입력받아 랜덤하게 교배시킨 후 교배된 염색체들을 GAmutation 블록으로 출력하고, GAmutation 블록에서는 랜덤하게 염색체 돌연변이를 수행하여 최종적인 염색체들을 출력한다. Fig. 10은 GA에 대한 흐름도를 나타냈다.

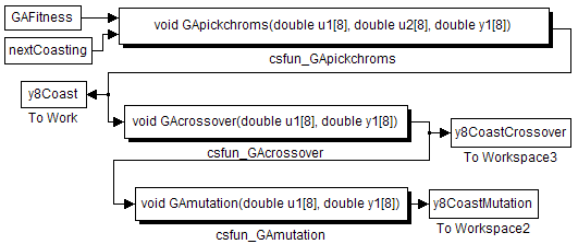


Fig. 9. GA block

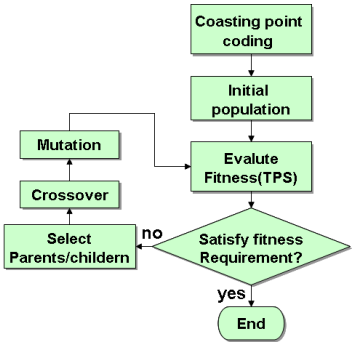


Fig. 10. GA flowchart

4. 시뮬레이션 결과 고찰

Table 2는 시뮬레이션에 사용된 TPS 파라미터와 GA 파라미터들을 보인다. 열차운전모드는 최대역행, 타행, 최대제동의 3단계로 설정하고 열차저항은 주행저항만을 고려하여 간략화하였다. 타행 개시지점을 9비트로 코딩하는 경우, 열차의 총 주행거리를 1[km]로 설정할 때 최소단위를 1[m]로 하여 511[m]까지의 타행 개시지점을 설정할 수 있다. 해집단은 512개의 모집단 멤버들 중에서 랜덤하게 8개를 선정한다. 해집단 크기는 시뮬레이션 수행속도와 최종적인 해의 신뢰도에 영향을 미치므로 이를 고려하여 반복 시뮬레이션을 결과를 토대로 최종적으로 8로 결정하였다. 한편, 8개의 해집단 멤버들 중에서 적합도가 큰 순서로 2개의 부모 염색체를 선정하고 염색체의 9개의 비트들 중에서 랜덤하게 하나를 선택한 후 선택된 비트를 중심으로 하여 부모 염색체를 교배한다. 돌연변이율에 대해서는, 확률을 2%로 하여 해집단 비트들 중에서 하나의 비트를 랜덤하게 선택하여 돌연변이 시킨다.

Table 2. Simulation Parameters

TPS 파라미터	
과천/분당선 인버터제어 전동차 4호선(1994)[12]	
열차편성	10량 (TcMM'TM'T1TMM'Tc) (M,M':구동차, Tc:제어차, T,T1:부수차)
열차중량(만차)	559[ton]
구동차 1량당 부하	111.8[ton]
구동차 1량당 전동기 수	4 Motors
구동차 1량당 최대견인력	11000[kgf/4-Motors]
구동차 1량당 최대제동력	9000[kgf/4-Motors]
구동차 1량당 행저항(지하)	$(1.867+0.0359v+0.000745v^2)$ ×111.8[kgf]
주행거리 스텝 증가(Δs_i)	4[m]
GA 파라미터	
해집단 염색체 (타행 개시지점) 수	8개
염색체 부호화	2진, 9비트
적합도 함수	$f=-1000\ \lvert(t-t_D)/t_D\rvert$
염색체 돌연변이	random, 최대 2%
총탐색횟수	10

시뮬레이션은 2단계로 수행하였다. 먼저 열차의 주행성능을 파악하기 위해 GA를 사용하지 않고 타행 시작지점을 임의의 스텝으로 증가시키면서 열차 주행거리에 따른 속도, 시간 및 열차 에너지 소모량을 계산하였다. Fig. 11(a)는 타행 시작지점을 100[m]에서 500[m]까지 50[m] 간격으로 증가시키며 각 타행지점에 따르는 열차의 거리-속도 성능을 보이는 그림으로 타행 시작 지점이 증가함에 따라 최대가속 구간과 최대제동 구간은 증가하고, 타행구간은 감소하는 모습을 보인다. 타행개시 지점을 100[m]로 하는 경우 48[km/h]의 속도

에서 최대가속에서 타행모드로 진입하고 37[km/h]에서 타행모드에서 최대제동모드로 전환하여 목표지점 1[km]에서 정지하는 모습을 보인다. 타행개시 지점을 최대 500[m]로 하는 경우에는 79[km/h]까지 가속된 후 타행모드로 진입하고 75[km/h]에서 타행모드에서 최대제동모드로 전환된 후 목표지점에서 정지하는 모습을 보인다. Fig. 11(b)와 (c)는 동일하게 타행 시작지점을 증가시키며 각 타행지점에 따르는 열차의 거리-시간을 보이는 그림으로 타행 시점이 증가함에 따라 1[km]를 주행하는데 걸리는 시간도 증가함을 알 수 있다. Fig. 11(c)는 Fig. 11(b)를 목표지점 부근에서 확대한 것으로 타행개시 지점을 100[m]로 하는 경우 주행시간이 95[sec]로 가장 길고, 타행개시 지점을 500[m]로 하는 경우에는 68[sec] 로서 가장 짧아짐을 알 수 있다. Fig. 11(d)는 각 타행지점에 대해서 회생 제동을 가정하여 열차에서 소비된 에너지를 산출한 것으로, 타행 개시점이 증가할수록 최대가속구간이 증가되어 소비에너지도 증가하는 모습을 보여 타행개시 지점을 100[m]로 하는 경우 약 1.3[kWh]가 소비되고 타행개시 지점을 500[m]로 하는 경우에는 약 2.1[kWh]의 에너지가 소비되는 결과를 보인다.

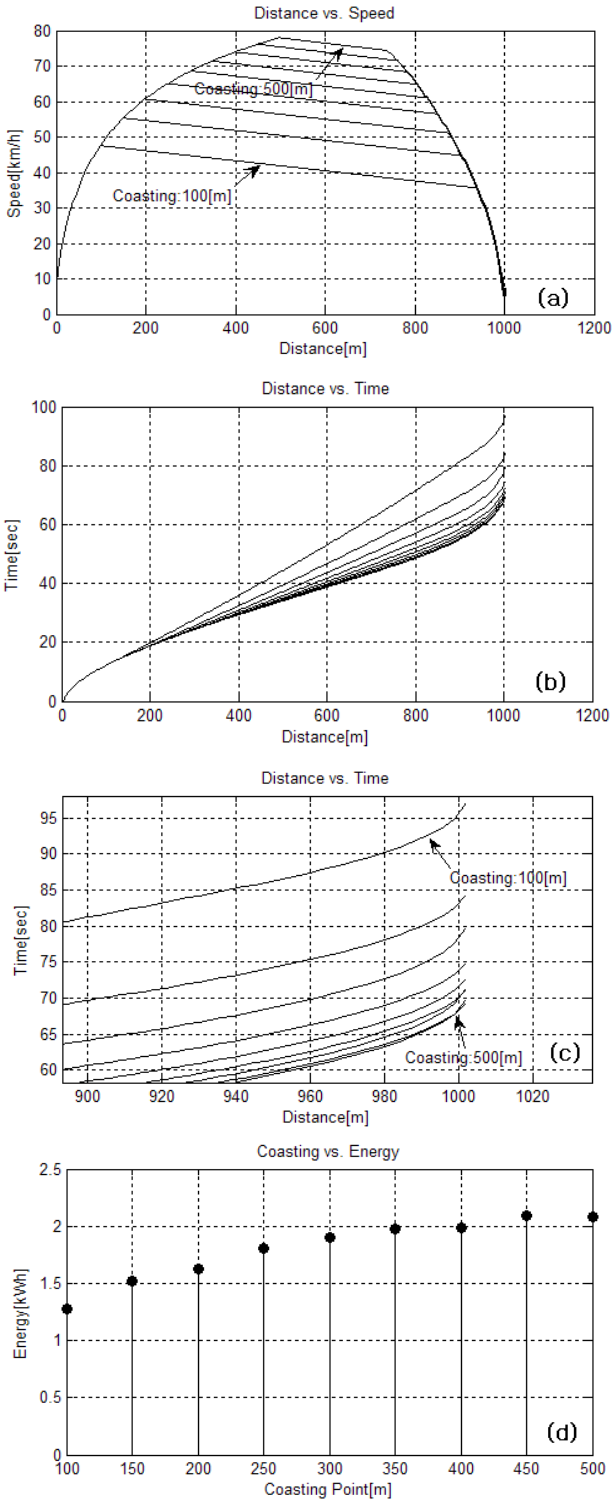


Fig. 11. Train performance according to the step variations of coasting starting points (target distance: 1[km], coasting points: 100[m]~500[m])

Fig. 12-14는 GA를 적용한 세 가지 경우에 대한 열차주행성능을 보인다. 목표주행거리를 1[km]로 하고 목표운전시간을 70[sec]로 하여 GA의 적합도 함수값이 수렴한 시점에서의 주행거리에 따른 주행속도와 시간 및 소비에너지를 나타냈다. Fig. 12(case I)의 경우, 1[km] 주행시 주행시간이 70.34[sec]로 목표주행시간을 1초 이내에서 만족하고 있으며 타행개시 지점은 447[m], 전체 에너지 소모량은 2.0517[kWh] 임을 보인다. Fig. 13(case II)의 경우에는 1[km] 주행시 주행시간이 70.59[sec]로 역시 목표주행시간을 1초 이내로 만족하고, 타행개시 지점은 363[m], 전체 에너지 소모량은 2.0064[kWh]로 case I의 경우보다 에너지 소모량이 약간 줄어든 결과를 보이는데 타행개시점이 약 90[m] 줄어들었기 때문으로 판단되어진다. Fig. 14(case III)의 경우에는 1[km] 주행시 주행시간이 70.1[sec]로 목표주행시간을 1초 이내로 만족하고, 타행개시 지점은 511[m], 전체 에너지 소모량은 2.08694[kWh]로 세 가지 경우들 중에서 에너지 소모량이 가장 크게 나타나는데 타행개시점이 가장 멀리 떨어져 있기 때문으로 판단되어진다. 한편, 제동시 에너지 회생을 고려하지 않고 가속시의 에너지 소비만을 고려한다면 case I, II, III 각각의 경우 소비된 에너지가 7.8[kWh], 6.9[kWh], 8.2[kWh]로서 차이가 더욱 벌어짐을 알 수 있다. 결과적으로 세 가지 경우 모두 목표운전거리와 목표운전시간을 동시에 만족하는 열차주행성능 결과를 얻을 수 있었고, 소비에너지가 가장 적게 되는 case II를 최적의 열차속도궤적으로 선정할 수 있을 것으로 생각된다. Table 3은 case I, II, III에 대한 시뮬레이션 결과를 정리한 것이다. Table 4는 case I, II, III에 대한 적합도 함수값과 염색체값(타행 개시지점)을 나타낸 표로서 탐색횟수가 2인 경우부터 적합도값과 염색체가 수렴하고 있음

을 보인다. 돌연변이율이 너무 작게 설정되었기 때문에 실제 돌연변이는 발생하지 않은 것으로 판단되나 적합도 함수가 만족할 만한 값으로 수렴되었기 때문에 돌연변이율에 별도의 수정을 가하지 않았다.

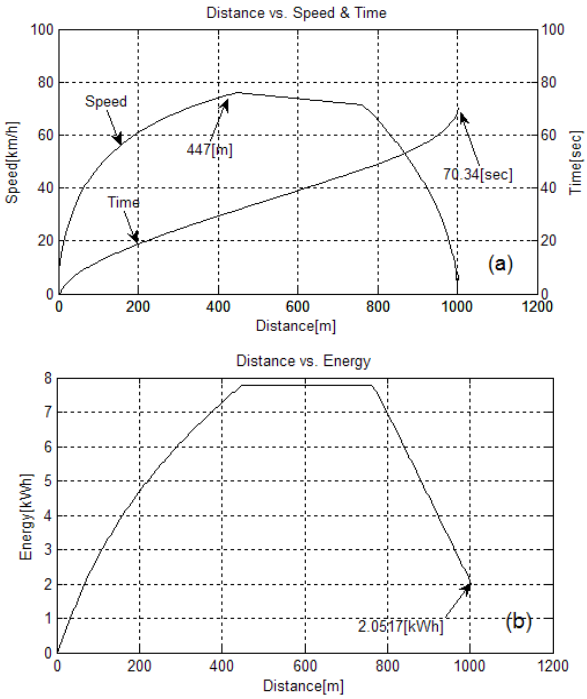


Fig. 12. Train performance target distance: 1[km], target time 70 sec], GA case I)

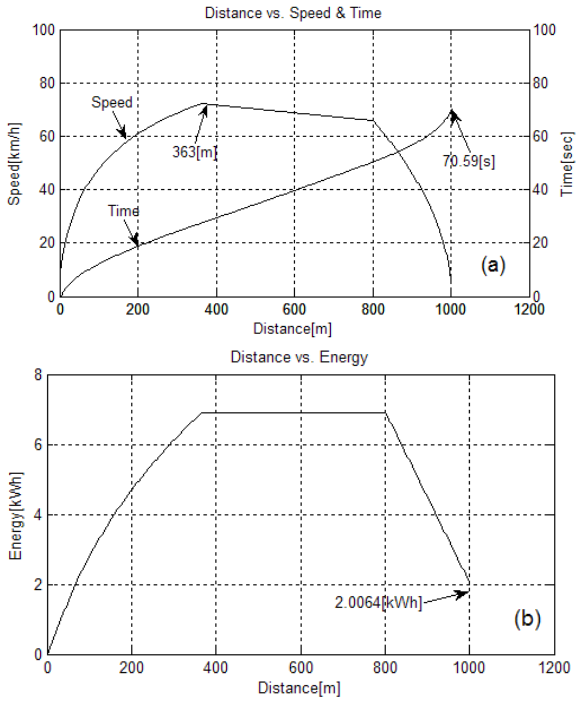


Fig. 13. Train performance (target distance: 1[km], target time 70 [sec], GA case II)

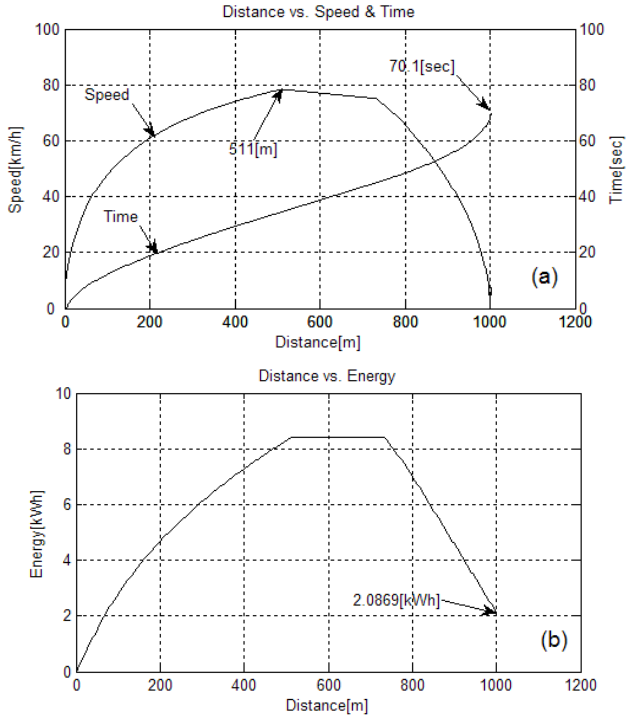


Fig. 14. Train performance (target distance: 1[km], target time 70 [sec], GA case III)

Table 3. Simulation results for GA case I, II, III

case	주행 거리 [m]	주행 시간 [sec]	타행개시 지점[m]	제동개시 지점[m]	에너지소비 (회생) [kWh]	에너지소비 (비회생) [kWh]
I	1000	70.34	447	760	2.0517	7.8
II	1000	70.59	363	800	2.0064	6.9
III	1000	70.10	511	730	2.0869	8.2

Table 4. Fitness values and chromosomes (coasting points) for GA case I, II, III (n : the number of searching times)

case	n	Fitness values	Chromosomes (Coasting points)
I	0	[-63.7 -88.0 -4.9 -2218.9 -122.4 -1257.5 -11.6 -14.3]	[261 248 447 27 211 41 375 428]
	1	[-4.9 -11.6 -11.6 -11.6 -11.6 -4.9 -4.9 -4.9]	[447 375 375 375 375 447 447 447]
	2	[-4.9 -4.9 -4.9 -4.9 -4.9 -4.9 -4.9 -4.9]	[447 447 447 447 447 447 447 447]
II	0	[-173.2 -75.3 -1397.0 -425.6 -173.2 -256.1 -17.8 -329.0]	[166 266 37 95 167 137 491 116]
	1	[-17.8 -75.3 -8.4 -8.4 -8.4 -23.1 -23.1 -23.1]	[491 266 363 363 363 394 394 394]
	2	[-8.4 -8.4 -8.4 -8.4 -8.4 -8.4 -8.4 -8.4]	[363 363 363 363 363 363 363 363]
III	0	[-679.7 -4.0 -25.1 -9.2 -18.9 -1590.0 -1.1 -32491.6]	[64 397 327 454 465 34 511 11]
	1	[-1.1 -4.0 -1.1 -1.1 -1.1 -4.0 -4.0 -4.0]	[511 397 511 511 511 397 397 397]
	2	[-1.1 -1.1 -1.1 -1.1 -1.1 -1.1 -1.1 -1.1]	[511 511 511 511 511 511 511 511]

5. 결 론

본 논문에서는 열차 운전시 운행거리와 운행시간의 두 가지 제한 조건을 동시에 만족하면서 소비에너지를 최소로 하기 위한 열차 속도 프로파일을 찾아내기 위하여 전후향 열차 속도 궤적을 산출하고, GA(Genetic Algorithm)를 적용하였다. 역간을 운행하는 열차의 운전 모드를 최대 역행, 타행, 최대 제동으로 간략화 시킨 후, 타행지점을 조절함으로서 주행기간동안에 열차에서 소비되는 에너지를 최소화 시키는 방식을 기본 알고리즘으로 하였다. 역행과 타행에 대한 전방향 속도궤적과 제동에 대한 후방향 속도 궤적을 계산하여 양자의 교차점인 제동 시작지점을 계산하여 정해진 역간거리를 주행하기 위한 속도 프로파일을 산출하였다. 또한, 두 역간을 정해진 운전시간 이내에 도달하기 위하여 목표 운전시간을 이용한 적합도 함수를 설정하고 GA 알고리즘을 적용하여 목표 운전시간을 만족시키는 타행 지점들과 이에 따른 소비에너지들을 산출하였다. 시뮬레이션 결과, 이들 타행 지점들 중에서 에너지 소비를 최소로 하는 타행지점을 포함하는 속도 프로파일을 선정하면 최적의 결과를 얻을 수 있을 것으로 기대된다. 향후에는 열차 다이내믹에 선로 구배, 곡선 등 다양한 선로조건들을 첨가하여 신뢰성을 높이기 위한 연구, 열차의 승차감과 안전운행 등을 고려하여, 운전 노치의 변경을 통한 열차의 역행 및 제동 모드 변경에 따른 열차 최적 속도궤적에 관한 연구, 열차 제어에의 GA적용에 있어서 GA 파라미터들의 선정 근거에 관한 더욱 상세한 연구들을 진행할 것이다.

참 고 문 헌

1. Deutsche Bahn AG, Energy efficiency strategies for rolling stock and train operation.
2. P. G. Howlett and P. J. Pudney(1995), *Energy-Efficient Train Control*, Springer.
3. Rongfang(Rachel) Liu a, Iakov M. Golovitcher(2003), “Energy-efficient operation of rail vehicles,” Transportation Research Part A, Vol. 37, pp.917-932.
4. C. Yeo, and T. Koseki(2002), “Optimization of running profile of train by dynamic programming,” National Convention of IEEJ, pp.85-86.
5. K. K. Wong and T. K. Ho(2004), “Dynamic coast control of train movement with genetic algorithm,” International Journal of Systems Science, Vol. 35, No. 13-14, pp.835-846.
6. Hee-Soo Hwang(1998), “Control strategy for optimal compromise between trip time and energy consumption in a high-speed railway,” IEEE Trans. on Systemss, Man, and Cybernetics-Part A: Systems and Humans, Vol. 28, No. 6.
7. MathWorks, Simulink - Simulation and Model-Based Design.
8. W. J. Davis, Jr.(1926), *The Tractive Resistance of Electric Locomotives and Cars*, General Electric Review.
9. Jyh-Cheng Jong and Sloan Chang(2005), “Algorithms for generating train speed profile,” Journal of Eastern Asia Society for Transportation Studies, Vol. 6, pp.356-371.
10. K. F., Man, K. S. Tang, S. Kwong, and W. A. Halang(1997), *Genetic Algorithms for Control and Signal Processing*, Springer Verlag.
11. MathWorks, Simulink 7 Writing S-Functions.
12. 철도청(1994), 과천/분당성 인버터제어 전동차 정비지침서, pp. 146-151.

접수일(2009년 7월 6일), 수정일(2009년 9월 16일),
게재확정일(2009년 12월 11일)