

에너지 효율적인 속도 프로파일 생성을 위한 Simulink 기반 열차 성능 시뮬레이션

논 문
59-10-17

A Train Performance Simulation using Simulink for Generating Energy-efficient Speed Profiles

장 문 호* · 한 문 섭**
(Moon-Ho Kang · Moon-Seob Han)

Abstract - In this research TPS (Train Performance Simulation) blocks are designed using Simulink and applied to generate speed profiles for energy-efficient train operation. With a train operation mode of maximum powering, coasting, and maximum braking, a braking point is calculated from forward-backward running profiles. Then, GA (Genetic Algorithm) is used to solve a running time constraint, and a coasting point is produced from the searching process of GA. With the braking point and the coasting point a speed profile is plotted. Train performance under a speed limit and gradient variations is simulated and resultant speed profiles are analyzed.

Key Words : Train performance simulation, Simulink, Energy-efficient train operation, Genetic algorithm, Forward-backward running profiles

1. 서 론

에너지 가격 상승과 함께 환경에 대한 관심이 고조됨에 따라 기존 열차 및 자동열차운전시스템 등에 새로운 열차운전 기법을 적용하여 에너지 효율을 최적화시키기 위한 다양한 시도들이 이루어지고 있다. 열차 에너지 최적화 기법들은 일반적으로 열차의 소비에너지와 열차주행시의 제반 조건들을 이용하여 목적함수를 설정하고 이를 최소화시킬 수 있는 최적의 열차 운전모드를 도출하는 형태를 취하는데, Pontryagin의 최대화 원리나 Lagrange 배수에 의한 최적화 과정을 통해 해석적으로 최적해를 계산하는 방식을 비롯하여 [1], [2] DP(Dynamic Programming), NLP(Nonlinear Programming) 등 수치해석적으로 해를 구하는 방식 [3], [4], 퍼지, GA(Genetic Algorithm) 등의 지능적 기법들을 이용하는 방식 [5], [6]들이 있다.

본 연구에서는 열차 운전시 에너지 효율을 향상시키기 위한 열차 속도 프로파일을 찾아내기 위하여, Simulink [7]를 이용하여 TPS(Train Performance Simulation) 프로그램을 제작하고, 전-후향 열차 속도 프로파일 계산법 [8], GA [5]를 적용하여 열차 성능 시뮬레이션을 행하고 결과를 분석하였다. 두 역간을 운행하는 열차의 운전 모드를 최대 역행, 타행, 최대 제동으로 간략화 시킨 후, 열차속도 프로파일을 구하기 위한 타행지점과 제동지점을 산출한다. 열차 제동지점은 궤도상에서 운행하는 열차의 전-후향 속도궤적을 구한

후 두 궤적의 교차지점으로부터 산출되고, 열차의 타행지점은 GA를 적용하여 역간의 운행시간 제한 조건을 만족하는 최적의 타행지점을 탐색하여 산출된다. 열차속도 프로파일과 함께 열차 주행시간과 열차소비에너지를 산출하고, 열차의 속도제한과 궤도의 구배 변동 등 선로조건을 다양하게 하여 시뮬레이션을 행하고 결과를 분석하였다.

2. 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.

$$T(v) - R(v, i, \gamma) - B(v) = M_c \frac{dv}{dt} \quad (1)$$

$$R(v, i, \gamma) = R_s(v) + R_r(v) + R_g(i) + R_c(\gamma) + R_t \quad (2)$$

$$R_r(v) = c_1 + c_2v + c_3v^2 \quad (c_1, c_2, c_3 > 0) \quad (3)$$

여기서, $T(v)$: 견인력[N], $B(v)$: 제동력[N], M_c : 열차유효질량[ton], v : 열차속도[m/s], t : 시간[s], t_f : 역간열차주행시간[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]

식(1)~(3)으로부터 시뮬레이션에 사용될 거리에 따른 열차의 주행속도, 시간, 소비에너지에 대한 이산치 식들을 구하면 다음과 같다 [9].

* 정 회 원 : 선문대 공대 정보통신공학과 부교수 공학박사
E-mail : mhkang@sunmoon.ac.kr

** 정 회 원 : 한국철도기술연구원 전기신호연구본부
전력연구팀 책임연구원

접수일자 : 2010년 7월 2일

최종완료 : 2010년 9월 16일

$$v_{k+1} = \sqrt{v_k^2 + 2a_k(s_{k+1} - s_k)} \quad (4)$$

$$t_{k+1} = t_k + \frac{s_{k+1} - s_k}{v_k} \quad (5)$$

$$E_{k+1} = E_k + (T(v_k) - B(v_k))(s_{k+1} - s_k) \quad (6)$$

$$a_k = \frac{F_k}{M_e} \quad (F_k = T(v_k) - R(v_k, i_k, \gamma_k) - B(v_k)) \quad (7)$$

$$\bar{v}_k = \frac{v_k + v_{k+1}}{2} \quad (8)$$

여기서, s :주행거리[m], a :가속도[m/s²], \bar{v} :평균속도[m/s], F :총견인력[N], E :소비에너지[kWh]

2.2 열차 속도 프로파일

열차의 운행모드를 그림 1에서 보이는 것과 같이 최대 역행, 타행, 최대 제동으로 설정하는 경우, 타행이 시작되는 지점을 조절하여 전체 운행구간 동안에 열차에서 소비되는 에너지를 최소화 시킬 수 있다. 이때, 열차의 속도 프로파일을 결정하기 위해서는 정해진 역간 거리(s_D)를 정해진 운전시간(t_D)에 운행하여야 하는 식(9)의 두 가지 제한 요소들을 모두 만족시키는 제동지점과 타행지점이 결정되어야 한다.

$$s = s_D, \quad t = t_D \quad (9)$$

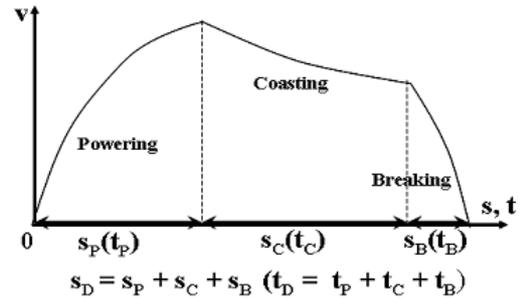
2.2.1 제동지점 결정

식(9)에서 첫 번째 제한 요소로서 열차가 목표역에 정확히 정지하기 위해서는, 정확한 제동 개시지점이 결정되어야 하는데, 이를 위해 역행과 타행에 의한 열차의 전방향 속도 프로파일을 구하고 제동에 의한 열차의 후방향 속도 프로파일을 구한 후, 두 프로파일의 교차점을 제동 지점으로 결정한다[8].

2.2.2 타행지점 결정

식(9)에서 두 번째 운전시간 제한요소를 만족시키기 위해 식(10)과 같이 간단한 적합도 함수(f)를 설정한 후, GA를 적용한다. 랜덤하게 도출되는 해집단(타행지점들)에 대해 TPS를 수행하면 각 타행 개시지점에 대한 역간 주행시간(t)이 산출된다. 이후, 주행시간을 식(10)에 대입하여 각 타행지점에 대한 적합도를 구하여 적합도가 큰 타행지점들이 선정되면, 이들에 대해 GA의 교배, 돌연변이를 적용하여 새로운 타행지점들을 도출한다. 이와 같은 과정을 정해진 탐색횟수에 도달할 때까지 반복한 후, 최종적인 타행지점을 결정한다[5].

$$f = K \left| \frac{t - t_D}{t_D} \right|, \quad (K < 0) \quad (10)$$



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

Fig. 1 Train operation modes

3. 시뮬레이션 프로그램

시뮬레이션 프로그램은 Simulink[7]를 이용하여 제작했으며, TPS 모듈, GA모듈 및 이들을 순차적으로 호출하여 수행시키는 스크립트 파일로 구성 된다. TPS 모듈은 크게 전향 속도 프로파일 산출블록, 후향 속도 프로파일 산출블록 및 속도 프로파일 통합블록으로 구성되고, GA모듈은 크게 Select 블록, Crossover 블록, Mutation 블록 등으로 구성된다[9].

3.1 전-후향 속도 프로파일 산출 블록

그림 2는 전-후향 속도프로파일 산출블록을 보이는 것으로, 전향 속도프로파일 산출블록(Forward (Powering/Coasting))에서 최대 역행과 타행을 통해 열차가 출발지로부터 목표지점을 향해 주행하는 동안의 속도프로파일을 산출하고, 후향 속도프로파일 산출블록(Backward(Breaking))은 최대 제동력으로 열차가 목표지점으로부터 출발지점을 향해 거꾸로 후향 주행하는 속도프로파일을 산출한다. Supervisor 블록은 순차적으로 전-후향 속도프로파일 산출블록을 동작시키고 두 속도프로파일의 교차점을 계산하여 제동 개시지점을 구한다.

그림 3은 전향 속도제한 산출블록의 내부 구성을 보인 것이다. 열차 속도에 대한 최대견인력이 저장된 look-up 테이블인 F_Traction블록으로부터 열차를 최대로 역행하기 위한 견인력이 얻어지고, F_resist블록에서는 주행저항이 계산되고 R_gradient블록에서 설정된 구배 데이터로부터 구배저항이 계산된다. 견인력과 저항들과의 차이값으로부터 Force2Accel블록에서 식(7)에 의해 가속도가 계산되고, Speed & Dist Calculation블록에서 식(4)를 이용하여 열차 속도가 계산된다. 열차속도는 Time Calculation블록에 인가되어 식(5)를 이용하여 주행 시간이 계산되고, 최종적으로 Energy 블록에서 식(6)에 의해 소비에너지가 계산되면 시뮬레이션의 한 사이클이 종료된다. 이후, 계산된 속도값을 이전 값으로 하여 위의 과정을 반복하고 열차 주행거리가 목표지점보다 커지게 되면 전향 속도제한 시뮬레이션을 중단한다. 한편 속도제한이 필요한 경우는, Speed_limit 블록에 의해 열차 속도에 대한 제동력이 저장된 F_Breaking블록과

F_Traction블록들 중에서 하나를 선택하여 속도제한을 이루도록 한다.

그림 4는 그림 3의 Speed & Distance 블록의 내부를 보이는 것으로 식(4)에 의해 가속도와 거리의 증가분을 이용하여 열차의 주행속도를 계산한다.

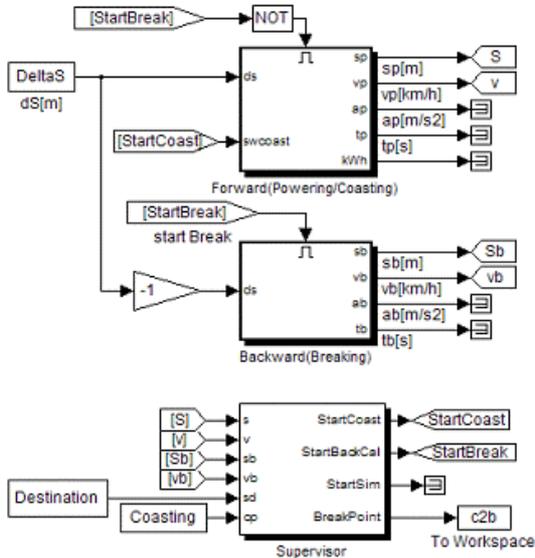


Fig. 2 Forward-backward speed profile calculation blocks

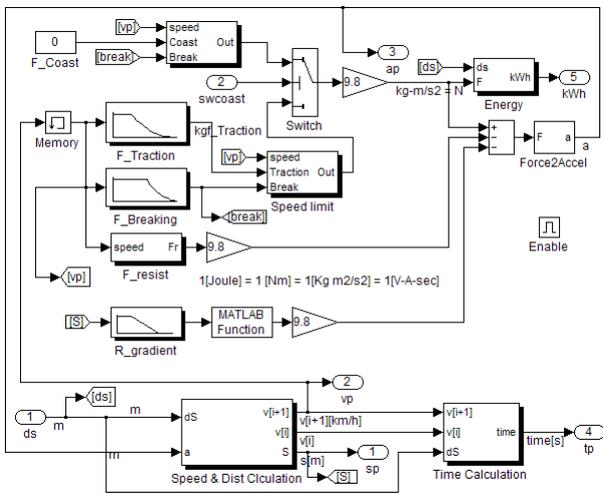


Fig. 3 Configuration of the Forward(Powering/Coasting) block of Fig. 2

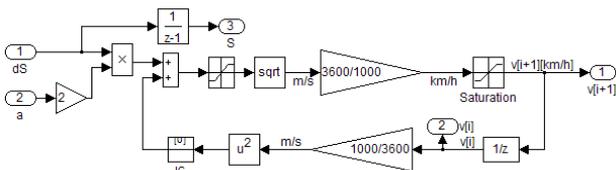


Fig. 4 Configuration of the Speed & Distance block

3.2 속도 프로파일 통합 TPS 블록

타행지점과 전-후향 속도 궤적의 교차점으로부터 제동지점이 결정되면 이들을 이용하여 목표지점까지 일정하게 주행 거리를 증가시키며 속도 궤적, 주행시간, 소비 에너지를 계산출하고 최종적으로 출력하는데, 이를 위한 Simulink 블록을 그림 5에 나타냈다. 그림에서 DeltaS, Coasting, Co2Br 은 각각 주행 거리 증가분, 타행 개시지점, 제동 개시지점을 나타낸다. 그림 6은 그림 5의 TPS 블록 내부 구성을 보이는 것으로, 열차가 초기 역행 후에 타행과 제동 상태로 운전 모드를 전환함에 따라 Mode control 블록에서 스위치를 전환하며 적절한 견인력을 열차에 제공하는 모습을 보인다. 각 구성 블록들은 전-후향 속도궤적 블록에서 사용된 블록들과 동일한 내부 구조를 가진다.

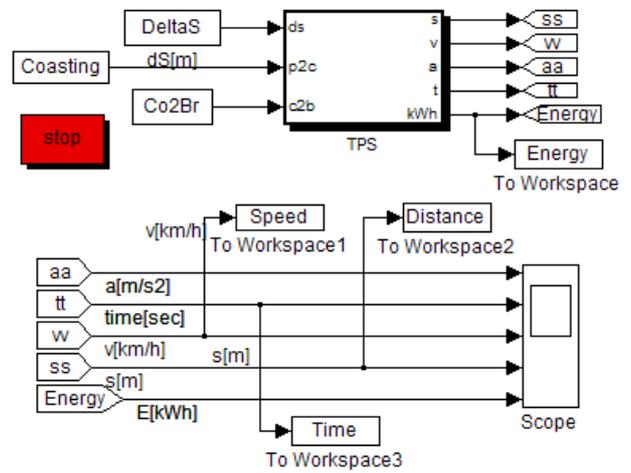


Fig. 5 Forward and backward speed profiles integrated TPS block

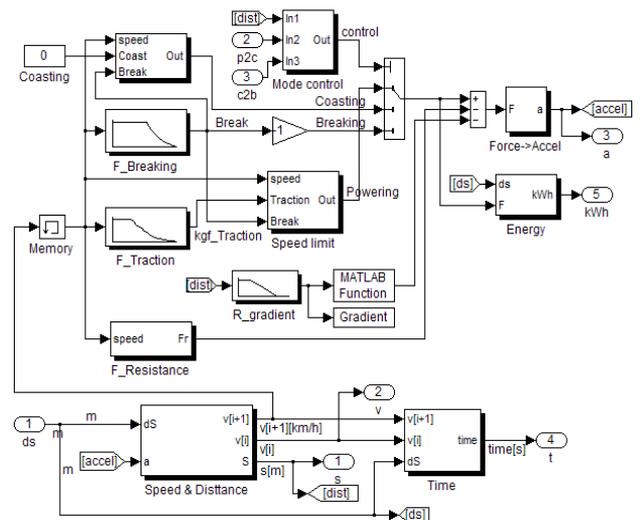


Fig. 6 TPS block configuration of Fig. 5

3.3 GA 블록

GA 기능들을 구현하기 위해, C-코드를 이용하여 필요한 GA관련 함수들을 작성한 후 Matlab의 Legacy C-코드 생성기를 이용하여 Simulink상에서 사용할 수 있는 c-sfunction 블록들을 GA 함수들로부터 생성하였다. 그림 7은 GA 블록을 보이는 것으로, 열차의 타행지점들이 랜덤하게 선정되면 이들에 대해 식(10)에 의해 적합도를 계산한 후, GApickchroms 블록에서 적합도 결과에 따라 부모와 자식염색체(타행지점) 들을 선택하여 출력하면, GAcrossover 블록에서 이를 받아 교배시킨 후 교배된 염색체들을 GAmutation 블록으로 출력하고, 블록에서는 염색체 돌연변이를 수행하여 최종적인 염색체들을 출력한다[9].

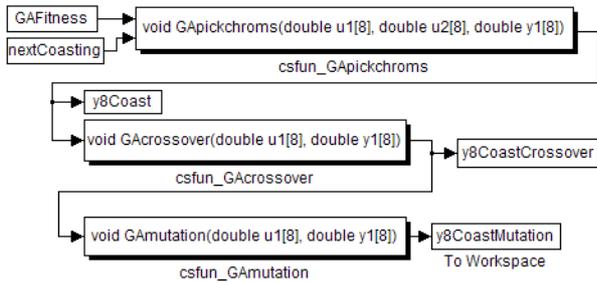


Fig. 7 GA Simulink block

그림 8은 TPS와 GA 블록을 이용한 시뮬레이션의 흐름도와 함께 각 단계에 대한 간략한 설명을 보인다.

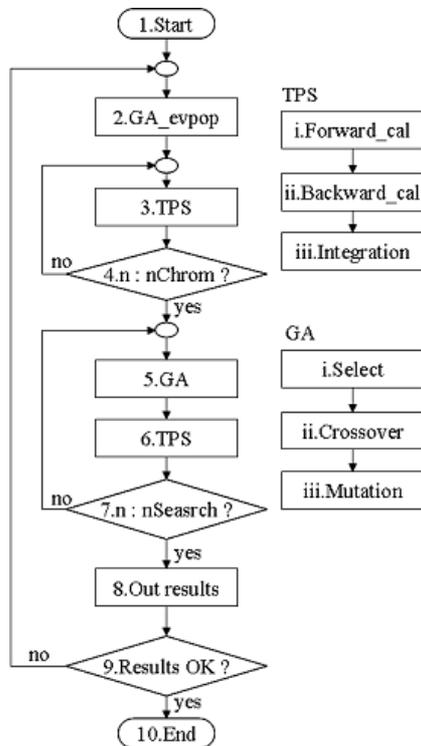


Fig. 8 Simulation flowchart

1. 시뮬레이션 목표값 및 파라미터 초기화
2. 초기 타행지점(염색체) 해집단 선정
3. TPS수행(전,후향속도프로파일 산출 및 통합, 타행지점들에 대한 적합도 함수값 산출)
4. 초기 해집단의 모든 멤버에 대해 TPS수행 완료
5. GA수행(염색체선택, 교배, 돌연변이)
6. TPS수행
7. 정해진 탐색횟수만큼 탐색 완료
8. 시뮬레이션결과 추출
9. 시뮬레이션 결과가 목표값을 만족하지 않으면 새로운 타행지점(염색체) 해집단 선정, 시뮬레이션 재개
10. 시뮬레이션 결과가 만족되면 결과 출력 후 종료.

4. 시뮬레이션 결과 고찰

표 1은 시뮬레이션에 사용된 TPS 파라미터, GA 파라미터들을 보인다. 열차운전모드는 최대역행, 타행, 최대제동으로 설정하고 열차편성은 전동차 5량, 부수차와 제어차가 5량인 총 10량으로 하였다. 타행 개시지점을 9비트로 코딩하여 기준지점으로부터 최대 511[m]까지 타행 개시지점을 설정할 수 있도록 하였다.

Table 1 Simulation Parameters

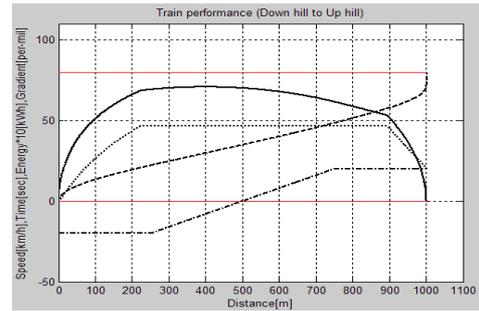
TPS 파라미터	
과천/분당선 인버터제어 전동차 4호선(1994)[10]	
열차편성	10량 (TcMM'TM'T1TMM'Tc) (M, M':구동차, Tc:제어차, T, T1:부수차)
열차중량(만차)	559[ton]
구동차 1량당 부하	111.8[ton]
구동차 1량당 전동기 수	4 Motors
구동차 1량당 최대견인력	11000[kgf/4-Motors]
구동차 1량당 최대제동력	9000[kgf/4-Motors]
구동차 1량당 주행저항(지하)	$(1.867+0.0359v+0.000745v^2) \times 111.8$ [kgf]
구배	-20 ~ 20 [%]
주행거리시스템 증가	1[m]
총 주행거리	1[km]
GA 파라미터	
해집단 염색체(타행개시점)수	8개
염색체 부호화	2진, 9비트
염색체 돌연변이	random, 최대 2%
총탐색횟수	10

그림 9는 목표주행거리를 1[km]로 하고 목표운전시간을 80[sec]로 설정한 경우의 열차 속도, 주행시간, 소비에너지를 보인다. 그림 9(a)는 -20[%]~20[%]로 하구배에서 상구배로 변동하는 상태로, 타행 개시지점은 222[m], 제동 개시지점은 900[m], 최대 소비에너지는 4.8[kWh]이고 주행시간은 80[sec]로서 목표운전시간을 만족함을 보인다. 그림 9(b)는 20[%]~-20[%]로 상구배에서 하구배로 변동하는 상태로, 타행 개시지점은 372[m], 제동 개시지점은 750[m], 최대 소비에너지는 7.9[kWh] 이고 주행시간은 81[sec]이다. 두 경우에 최대 소비에너지 차이는 크지만 제동시 에너지가 회생되어 최종 소비에너지는 약 2[kWh]로 거의 같음을 알 수 있다. 이는 초기 운전기간 동안 상구배인 그림 9(b)의 경우가 하구배인 그림 9(a)에 비해 상대적으로 가속기간은 길고 타행기간은 짧지만 제동기간이 길어져 회생에너지 양이 증가했기 때문이다.

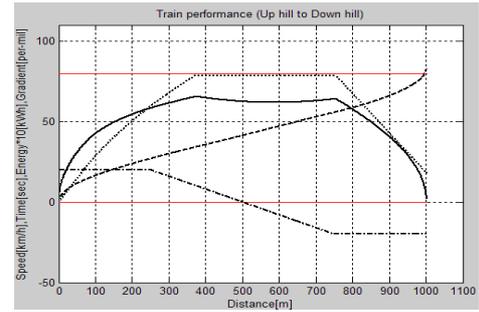
그림 10은 목표주행거리를 1[km]로 하고 목표운전시간을 100[sec]로 설정한 경우의 속도제한에 따른 열차성능을 보인다. 제한속도를 50[km/h]로 하여 역행기간 중에는 열차가 제한속도를 넘는 경우 제동을 인가하고 열차속도가 40[km/h]이하로 떨어지면 다시 역행하여 속도를 조절하였다. 또한, 타행기간 중에도 하구배인 경우에 열차가 제한속도를 넘을 수 있으므로 제한속도를 넘는 경우 제동을 인가하고 열차속도가 40[km/h]이하로 떨어지면 다시 타행하여 속도를 제한하였다. 그림 10(a)는 하구배에서 상구배로 변동하는 상태이고, 그림 10(b)는 상구배에서 하구배로 변동하는 상태로, 타행 개시지점은 각각 770[m]와 331[m]이다. 시뮬레이션 결과 구배 변동에 상관없이 속도제한이 잘 이루어지고 있고, 목표운전시간도 두 경우 80[sec]로 만족스러운 결과를 보인다.

그림 11은 -20[%]로 일정한 하구배 상태에서 제한속도를 50[km/h]로 하여 타행지점 변동에 따른 열차성능을 비교한 결과이다. (a), (b), (c) 각각의 타행개시지점은 319[m], 497[m], 782[m]로, 세 경우 모두 속도제한이 잘 이루어지고 있고 목표운전시간도 80[sec]를 만족한다. 한편, 제동시 회생에너지가 역행시 소비된 에너지 보다 증가하여 에너지가 열차로부터 공급되었음을 보인다. 총에너지 공급량은 약 4.5[kWh]로 세 경우가 비슷하나, 타행이 가장 일찍 시작된 (a)의 경우에 역행 기간 중 열차 운전모드 변경 횟수가 가장 적게 되어 (b)와 (c)의 경우에 비해 상대적으로 운전이 단순해짐을 알 수 있다.

그림 12는 20[%]~-20[%]로 상구배에서 하구배로 변동하는 상태에서 열차 성능을 나타냈다. (a)(case I)의 경우, 주행시간이 81[sec]이고, 타행개시 지점은 372[m], 최대 에너지 소모량은 7.9[kWh] 임을 보인다. (b)(case II)의 경우에는 주행시간이 81[sec]이고 타행개시 지점은 415[m], 최대 에너지 소모량은 8.3[kWh]로 case I의 경우보다 약간 증가하고, (c)(case III)의 경우에는 주행시간이 80[sec]이고, 타행개시 지점은 508[m], 최대 에너지 소모량은 9.7[kWh]로 case I의 경우보다 큰 폭으로 증가한 모습을 보인다. 에너지 회생을 고려하면 총 에너지 소비가 약 2[kWh]로서 세 경우 비슷한 결과를 보인다. 시뮬레이션 결과, 타행이 가장 빨리 시작되어 가장 오래 지속된 case I을 최적의 열차속도 프로파일로 선정할 수 있다. 표 2는 그림 11의 결과를 정리한 것이다.



(a)

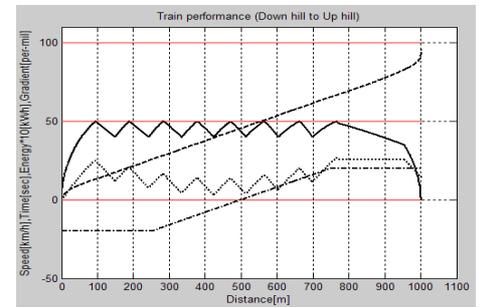


(b)

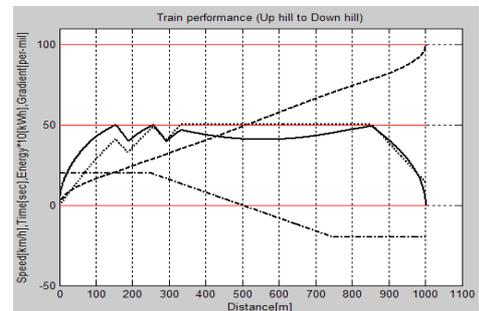
Fig. 9 Train performance

(- : speed - - : time ··· : energy - · - : gradient)

(gradient: -20~20[%](a) 20~-20[%](b), target distance:1[km] target time:80[sec])



(a)

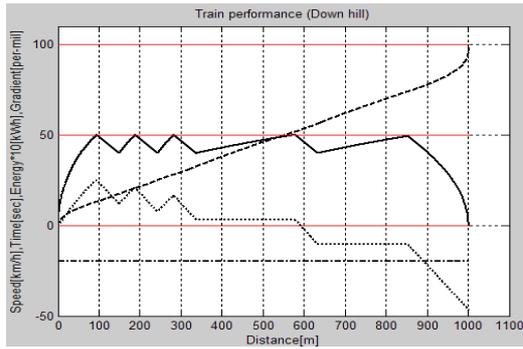


(b)

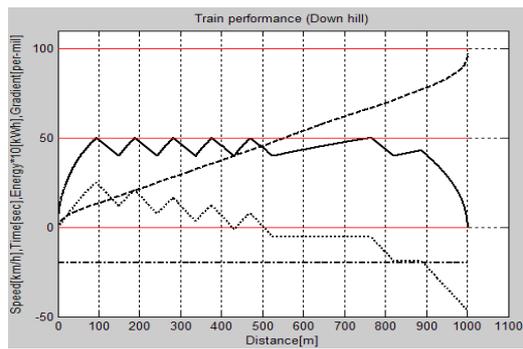
Fig. 10 Train performance with speed limit

(- : speed - - : time ··· : energy - · - : gradient)

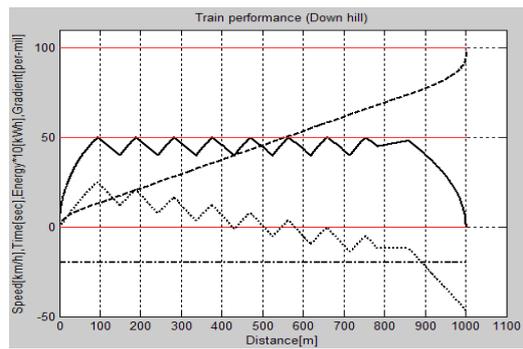
(gradient: -20~20[%](a) 20~-20[%](b), speed limit:50[km/h], target distance:1[km] target time:100[sec])



(a)

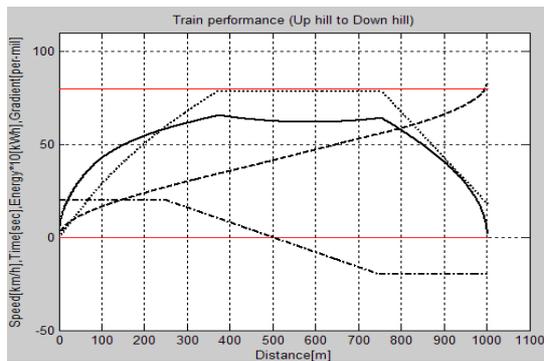


(b)

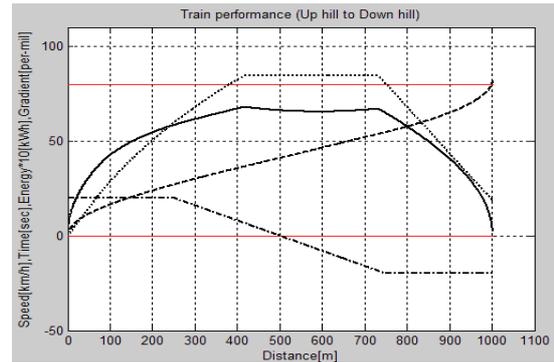


(c)

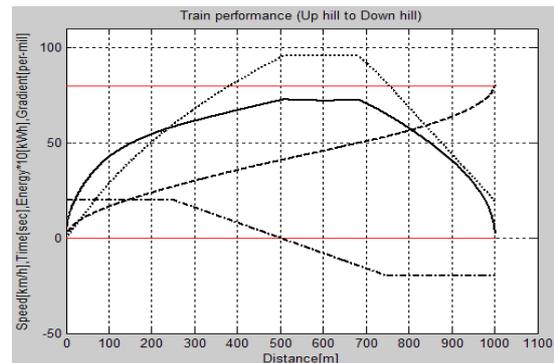
Fig. 11 Train performance with speed limit
 (—:speed - - :time ···:energy - · - :gradient)
 (gradient: -20[%], speed limit:50[km/h], target distance:1[km]
 target time:100[sec])



(a) case I



(b) case II



(c) case III

Fig. 12 Train performance
 (—:speed - - :time ···:energy - · - :gradient)
 (gradient: 20~-20 [%], target distance:1[km] target
 time:80[sec])

Table 2 Simulation results (case I, II, III)

case	주행 거리 [m]	주행 시간 [sec]	타행개시 지점 [m]	제동개 시지점 [m]	에너지소비 (회생) [kWh]	에너지소비 (비회생) [kWh]
I	1000	81	372	750	2.05	7.9
II	1000	81	415	725	2.08	8.3
III	1000	80	508	690	2.07	9.7

5. 결 론

본 연구에서는 Simulink를 이용하여 TPS 블록과 GA블록을 설계하고, 에너지 효율을 높이는 열차 속도 프로파일을 생성하는데 적용하였다. 역간의 열차 운전 모드를 최대 역행, 타행, 최대 제동으로 간략화 시키고, 선로의 구배 변동 및 속도제한 조건하에서 열차성능을 시뮬레이션하고 결과를 분석하였다. 시뮬레이션 결과 역간 주행거리와 주행시간 제한 조건을 동시에 만족시키는 속도 프로파일들을 구할 수 있었고, 이들 중에서 상대적으로 타행이 일찍 발생하는 속도 프로파일이 에너지 효율과 열차 운전의 단순성 면에서 장점이 있음을 확인하였고 향후, 자동열차운전 알고리즘 등에 활용될 수 있으리라 생각된다.

참 고 문 헌

- [1] P.G. Howlett and P.J. Pudney, Energy-Efficient Train Control, Springer, 1995.
- [2] Rongfang(Rachel) Liu a, Iakov M. Golovitcher, "Energy-efficient operation of rail vehicles", Transportation Research Part A, 37, pp. 917 - 932, 2003.
- [3] Yeo, C., Koseki, T., "Optimization of Running Profile of Train by Dynamic Programming", National Convention of IEEJ, pp. 85-86, 2002.
- [4] S.Y. Askey, Design and Evaluation of Decision Aids for Control of High-Speed Trains:Experiments and Model, Doctor's thesis, Massachusetts Institute of Technology, June, 1995.
- [5] K. K. Wong and T. K. Ho, "Dynamic coast control of train movement with genetic algorithm", International Journal of Systems Science, vol.35, no.13 - 14, pp. 835 - 846, 2004.
- [6] Hee-Soo Hwang, "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, pp. 792-801, 1998.
- [7] MathWorks, Simulink - Simulation and Model-Based Design.
- [8] Jyh-Cheng Jong and Sloan Chang, "Algorithms for Generating Train Speed Profile", Journal of Eastern Asia Society for Transportation Studies, no.6, pp.356-371, 2005.
- [9] Moonho Kang, Moonseob Han, "A GA-based algorithm for generating a train speed profile optimizing energy efficiency", Journal of the Korean Society for Railway, vol.12, no.6, pp. 878-886, 2009.
- [10] 철도청 (1994) 과천/분당선 인버터제어 전동차 정비지침서, pp. 146-151.

저 자 소 개



강 문 호 (康 文 浩)

1964년 7월 13일생. 1988년 고려대 전기공학과 졸업. 1995년 동 대학원 전기공학과 대학원 졸업(공학). 1995년-1997년 철도기술연구원 선임연구원 1997년-현재 선문대 정보통신공학과 부교수

Tel : 041-530-2339

Fax : 041-530-2981

E-mail : mhkang@sunmoon.ac.kr



한 문 섭 (漢 文 燮)

1964년 11월 26일생. 1987년 인하대 전기공학과 졸업. 1989년 동대학원 전기공학과 졸업(석사). 1989년-1995년 국방과학연구소. 1996년-현재 철도기술연구원 책임연구원

Tel : 031-460-5420

Fax : 031-460-5459

E-mail : mshan@krri.re.kr