

추종 제어 기법을 응용한 ATO 속도 및 정밀정차 제어 장치 개발

A development of speed and precision stop controller for ATO using servo-controller

박준영 ¹⁾	김용민 ²⁾	이태연 ³⁾	박재홍 ⁴⁾
Park, Jun-Young	Kim, Yong-Min	Lee, Tae-Yeon	Park, Jae-Hong
한성호 ⁵⁾	김원경 ⁶⁾	김동호 ⁷⁾	전병진 ⁸⁾
Han, Seong-Ho	Kim, Won-Kyong	Kim, Dong-Ho	Jone, Byoung-Jeen

ABSTRACT

This paper is mainly concerned with the development of the ATO(Automatic Train Operation) System. The ATO system is used for automatic or driverless operation of a train using various informations from TCMS, ATC, TWC. In this paper, the general architecture of the ATO system, implementation of ATO application software and the algorithm using servo-controller for automatic train speed restriction/regulation is presented.

1. 서론

최근 도시 철도 차량의 수요 증가와 운전 시력의 단축화, 철도 차량 기술의 진보에 따라 전동차의 자동 운전 및 무인 운전을 수행하기 위한 ATO (Automatic Train Operation) 장치가 필수적 요소가 되었다. ATO 장치는 외부 장치와의 통신을 통해 운행에 필요한 여러 가지 정보를 수집하고 이 정보를 바탕으로 전동차의 동작 상태(출발, 운행, 정지, 검사)를 결정하고 전동차를 자동/무인 운전하는 장치로서 운행의 안전성을 확보하고 승객의 편안함을 도모하며 운행 시간과 에너지를 절약하고자 하는 목적을 가진다.

ATO 장치는 자동/무인 운전을 수행하기 위해 그림 1에서 보듯이 여러 가지 외부 장치와 연관성을 가지고 동작한다. ATC(Automatic Train Control) 장치로부터 속도 지령을, TWC(Train Wayside Communication) 장치로부터 고정 속도 및 현재 역과 다음 역에 대한 정보와 같은 운행 구간 정보를, TCMS(Train Control and Monitoring System) 장치로부터 ATO 활성화/비활성화와 같은 ATO 제어 정보를 수신하고 이 정보를 내부에 저장된 트랙 데이터베이스로부터의 구배, 커브, PSM(Precision Stop Marker) 정보 등의 선로 정보와 함께 활용하여 제어 입력, 즉 추진력/제동력을 계산하고 이 값을 TCMS로 송신하여 속도 제어를 수행한다.

ATO 장치의 기능을 세분화하면 상태 결정 및 동작 제어, 목표 속도 제어, 정밀 정차 제어, 감시 기능으로 나누어 볼 수 있다. 상태 결정 및 동작 제어는 여러 가지 정보를 이용하여 ATO의 동작 상태(출발, 운행, 정지, 검사)를 결정하고 그 상태에서 실행해야 할 하위 동작들을 지시하는 기능이다. 목표 속도 제어는 전동차가 운행 중일 때 상황에 따라 목표 속도를 결정하고 이를 추종하도록 하는 것이며 정밀 정차 제어는 정지역의 정확한 위치에 전동차를 정차시키기 위한 제어를

1) 서울대학교 전기공학부 석사과정

4) 서울대학교 전기공학부 부교수

7) 우진산전(주)기술연구소 차장

2) 서울대학교 전기공학부 박사과정

5) 한국철도기술연구원 선임연구원

8) 우진산전(주)기술연구소 대리

3) 서울대학교 전기공학부 박사과정

6) 한국철도기술연구원 책임연구원

말한다. 감시 기능은 안전한 동작을 보장하기 위해 시스템 초기화와 검사, 실행 도중의 에러 상황에 대한 감시 및 기록을 지속적으로 수행하고 시각, 거리, 속도 등의 운행에 관한 정보를 기록하여 ATO 운행에 대한 분석을 할 수 있도록 해주는 기능이다.

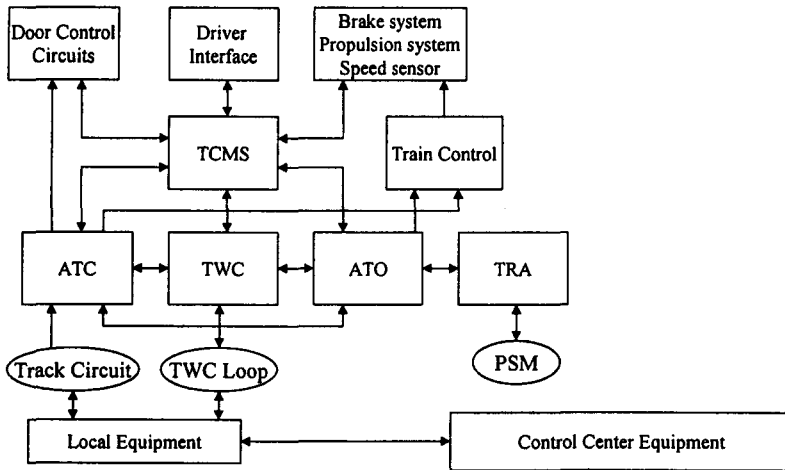


그림 1. ATO와 외부 장치와의 관계

본 논문에서는 추종 제어 기법을 응용한 ATO 속도 및 정밀정차 제어 장치의 개발에 대해 설명한다. 하드웨어로는 연산 및 제어, 통신 기능을 담당하는 CPU 보드와 전동차 운행에 필요한 정보를 수집하는 I/O 보드를 설계, 제작한다. 소프트웨어는 초기화 및 에러 검사를 담당하는 KRN(Kernel)부분과 외부 장치와의 통신을 담당하는 COM(Communication Handler)부분, 동작 상태 결정 및 목표 속도 제어와 정밀 정차 제어를 담당하는 Application 부분으로 나누고 각각을 세분화하여 구현한 후 통합한다. 특히, ATO 기능에 있어서 가장 핵심적인 부분인 속도 제어 알고리즘을 운행 시의 목표 속도 제어와 정지 시의 정밀 정차 제어의 경우로 나누어 제시한다. 개발된 ATO 장치를 시뮬레이션 프로그램과 연계하여 모의 운행 시험을 하고 그 결과를 분석함으로써 성능을 확인한다.

본 논문의 구성은 다음과 같다. 2절에서는 하드웨어의 역할과 개발 사양을 간략히 살펴본다. 3절에서는 소프트웨어의 세부 구조 및 구현을 살펴 본다. 특히, 핵심 부분인 속도 제어 알고리즘에 대해 자세히 설명한다. 4절에서는 개발한 ATO 장치의 모의 운행 시험 결과를 분석한다.

2. ATO 하드웨어의 구현

CPU 보드의 역할은 ATO 소프트웨어의 실행환경을 제공하여 제어 알고리즘을 실행시키는 것과 차상 및 외부 장치와의 통신을 통해 ATO 소프트웨어에 데이터를 전달하는 것, 운행 기록 및 에러 상황에 대한 기록을 주기적으로 메모리에 저장하는 것이다. CPU 보드의 사양은 Motorola MC68040 CPU, EPROM(512KB), Flash Memory(2MB), SRAM(8MB)으로 VMEbus 및 RS232C, RS485, 20mA current loop의 통신 규격을 지원하도록 설계, 제작하였다. 이들 통신 규격은 TCMS, ATC, TWC와의 데이터 교환에 사용된다. EPROM에는 RTOS가 탑재되며, Flash Memory에는 트랙 데이터베이스의 내용과 운행 중의 각종 정보(속도, 위치, 시간, PSM 검지 여부, 등)와 에러 정보가 저장된다. SRAM은 소프트웨어의 연산 과정에 사용된다.

I/O 보드의 역할은 PSM을 검지하는 것과 현재 전동차의 속도를 알려주는 Tachometer 입력을 읽는 것, CPU 보드로 수집한 데이터를 전송하는 것이다. I/O 보드의 사양은 Motorola MC68HC16Z1 마이크로 컨트롤러, Digital Input 8채널, Digital Output 8채널, Pulse Input 2채널이며 VMEbus를 통해 CPU 보드와 통신하도록 설계, 제작하였다. Digital Input 채널은 TRA(Trigger Receiver Assembly) 장치로부

터 PSM을 검지하여 CPU 보드로 전달하는데 쓰이고 Pulse Input은 Tachometer 입력을 읽는데 쓰인다.
3. ATO 소프트웨어의 구현

ATO 소프트웨어는 크게 KRN, COM, Application의 세 부분으로 구성된다. KRN은 ATO가 기동할 때의 처리와 시스템 검사, 에러 처리를 담당하는 부분이며, COM은 ATO 장치와 외부 장치(TCMS, ATC, TWC)사이의 통신을 담당하는 부분이고, Application은 동작 상태 결정 및 목표 속도 제어와 정밀 정차 제어를 주로 한 제어를 담당하는 부분으로서 ATO 동작 상태를 결정하는 ART(Artificial Driver) 부분, 운행 시의 목표 속도 제어 및 정밀 정차 제어를 위해 제어 입력을 계산하는 SPR(Speed Restriction) 부분, 계산된 제어 입력에 따라 추진력/제동력을 조절하는 REG(Speed Regulation) 부분, 트랙 데이터베이스를 제공하는 SRV(Service) 부분, PSM을 처리하는 PSM(PSM handler) 부분으로 구성된다.

이 절에서는 우선 소프트웨어의 Application 부분의 구현에 있어서 핵심적 부분인 속도 제어 알고리즘을 운행 시와 정밀 정차시의 두 가지 경우에 대해 자세히 제시한 후, 위의 소프트웨어 구성에 따라 각 부분을 모듈 별로 구현한다. 단, KRN의 경우 RTOS 자체에서 지원하는 기능이므로 별도의 구현 과정을 설명하지 않는다.

3.1 속도 제어 알고리즘

본 연구에서 설계한 전동차 속도 제어 알고리즘을 구현한 제어기는 그림 2와 같이 추종 입력 생성기와 추종 제어기의 두 가지 부분으로 나누어 구성되어 있다. 추종 입력 생성기는 운행 시에는 ATC 속도 지령과 TWC 고정 속도로부터 결정되는 속도 코드와 PSM에 관한 신호를 받아들여 가속 및 감속 시에 적합한 속도에 대한 추종입력을 생성하며 정차 시에는 PSM으로부터 오는 신호를 바탕으로 정지에 필요한 추종입력을 계산하여 추종제어기에 넘겨준다. 추종제어기는 추종 입력생성기로부터 오는 속도에 대한 추종입력과 실제 전동차의 속도간의 오차를 0으로 만들어줄 수 있도록 시스템에 투입되는 제어 입력값을 계산한다.

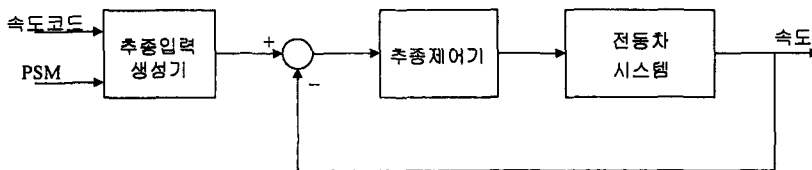


그림 2. 전동차속도제어기의 구조

3.1.1 추종 입력 생성기

운행 시 목표 속도를 계산하는 기준은 ATC로부터의 속도 지령과 TWC로부터의 고정 속도 및 운행 모드(정상 운전/회복 운전)이다. 목표 속도는 속도 지령과 고정 속도 중 가장 제한적인 값으로 선정된 후 운행 모드에 따라서 다시 일정한 값만큼 낮추어 결정된다.

목표 속도가 결정되면 현재 속도가 목표 속도의 패턴을 부드럽게 따라 가면서 속도 변화가 저크의 한계 이내에서 일어나도록 하는 추종입력, 즉 운행 속도 프로파일 $v(t)$ 를 다음 식 (1)과 같이 작성한다. 식 (1)은 가속구간을 4등분 한 후 첫 번째 구간에서는 일정한 양(+)의 저크를 주고 두 번째와 세 번째 구간에서는 저크를 0으로, 네 번째 구간에서는 일정한 음(-)의 저크를 주어 S자 형태의 부드러운 속도 변화를 만든다. 이 식에서 t_a 는 가속 구간의 시간폭으로서 $t_a = (5/6)v_{new} - v_{old} + 11$ 로 정하였으며, t_{off} 는 가속이 일어나는 시간을, v_{old} 는 변화하기 이전의 목표 속도를, v_{new} 는 변화한 이후의 목표 속도를 가리킨다.

$$v(t) = W(t-t_{off})v_{new} + (1-W(t-t_{off}))v_{old}$$

$$W(t) = \begin{cases} \frac{8}{3t_a^2}(t-\frac{1}{4}t_a-t_{off})^2 & t_{off} \leq t < t_{off} + \frac{1}{4}t_a \\ \frac{4}{3t_a}t_a(t-\frac{1}{4}t_a-t_{off}) + \frac{1}{6} & t_{off} + \frac{1}{4}t_a \leq t < t_{off} + \frac{3}{4}t_a \\ -\frac{8}{3t_a^2}(t-t_a-t_{off})^2 + 1 & t_{off} + \frac{3}{4}t_a \leq t < t_{off} + t_a \end{cases} \quad (1)$$

한편, 정차 시에는 운행 시의 속도 제어를 하던 알고리즘을 정밀 정차를 담당하는 알고리즘으로 대체한다. 정밀 정차를 담당하는 알고리즘은 정밀 정차를 위한 속도 프로파일을 다음과 같은 방식으로 만들어낸다.

우선, PSM1이 검출되면 식 (1)에 따라 전동차의 속도를 40 km/h까지 감속한 후 유지시킨다. PSM2를 검출하면 PSM4를 검출할 때까지 다음의 식 (2)에 따라 속도 프로파일 $v(t)$ 를 생성한다. 이 식에서 t_{PSM2} 는 PSM2를 검출한 시각, t_a 는 지정할 수 있는 감속구간이다.

$$v(t) = 1.5 W_{PSM23}(t) + 11.1(1 - W_{PSM23}(t))$$

$$W_{PSM23}(t) = \begin{cases} \frac{16}{5t_a^2}(t-\frac{1}{4}t_a-t_{PSM2})^2 & t_{PSM2} \leq t < t_{PSM2} + \frac{1}{4}t_a \\ \frac{8}{5t_a}t_a(t-\frac{1}{4}t_a-t_{PSM2}) - \frac{1}{5} & t_{PSM2} + \frac{1}{4}t_a \leq t < t_{PSM2} + \frac{1}{2}t_a \\ -\frac{8}{5t_a^2}(t-t_a-t_{PSM2})^2 + 1 & t_{PSM2} + \frac{1}{2}t_a \leq t < t_{PSM2} + t_a \end{cases} \quad (2)$$

PSM4는 정차위치로부터 3.5m 떨어져 있으며 이를 검출하면 다음 식 (3)과 같이 정차 위치까지 속도를 직선적으로 감소시킨다.

$$v(t) = -0.33*(t-t_{PSM4}) + 1.5 \quad (3)$$

이 식에서 식의 단위는 m/sec이며 t_{PSM4} 는 PSM4가 검출된 시각이고 0.33은 뒤에서 보여질 모의 운행 시험에서 선정한 값이다.

3.1.2 추종 제어기

추종 제어기는 전동차의 속도가 추종 입력 생성기에서 생성한 속도 프로파일을 추종하도록 제어하는 부분이다. 그림 3은 추종 제어기의 구조를 나타낸 것이다. A, B가 포함된 블록은 대상 시스템인 전동차의 동특성을, A_c, B_c 가 포함된 블록은 보상기의 동특성을 나타낸다. 추종 제어기는 추종하고자 하는 입력신호의 Laplace 변환 결과 중 추종오차를 발생시킬 수 있는 요소를 입력신호의 모델로서 보상기에 포함시키고 입력신호의 모델이 포함된 확장 시스템을 안정화시킬 수 있도록 페루프 시스템을 구성함으로써 추종 성능을 보장한다. 이 때, 보상기의 역할은 추종입력인 $r(t)$ 와 시스템의 출력 $y(t)$ 의 오차인 $e(t)$ 를 입력으로 받고 feed forward 이득 K_c 를 통해 시스템에 필요한 제어 입력을 생성하는 것이다. K는 대상 시스템에 대한 feedback gain이다.

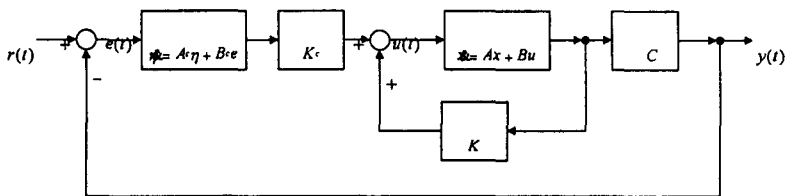


그림 3. 추종제어기의 구조

그림 3을 보면 대상시스템과 보상기가 직렬 연결된 것으로 볼 수 있으며 대상시스템에 투입되는 제어입력은 시스템의 상태변수와 보상기의 출력을 각각 K, K_c 의 게인이득으로 변화시켜 생성되는 것으로 볼 수 있다. 두 시스템을 직렬로 연결한 시스템은 다음과 같다.

$$A_a = \begin{bmatrix} A & 0 \\ B_c C & A_c \end{bmatrix}, \quad B_a = \begin{bmatrix} B \\ B_c D \end{bmatrix}, \quad C_a = [C \ 0] \quad (4)$$

그림 3에서 K_c 역시 보상기의 상태변수에 대한 상태제환으로 생각할 수 있으므로 이 시스템을 안정화시키는 제환이득을 다음과 같이 연결된 시스템의 상태제환 이득으로 합쳐서 생각할 수 있다.

$$K_a = [K \ K_c] \quad (5)$$

그림 3의 추종 제어기를 ATO 장치에 적용하는 과정은 다음과 같다. 우선 연속 시간 형태로 주어지는 대상 시스템의 동특성과 보상기의 동특성을 이산 시간 형태로 변환해야 한다. 왜냐하면 ATO 장치는 컴퓨터를 이용하여 구현되므로 일정 시간 간격으로 측정신호를 샘플링하여 이용하게 되고 대상 시스템에 가해지는 입력 역시 컴퓨터의 계산 주기에 따라 이산 시간 신호의 형태로 구해지기 때문이다. 연속 시간 시스템의 이산화 기법은 널리 알려져 있으며 참고 문헌[2]를 참조하기 바란다.

식 (1), (2), (3)에서 볼 수 있듯이 추종 입력 속도에는 시간에 대한 1차, 2차 함수 성분이 포함되어 있기 때문에 추종제어기 내부에 이들에 대한 정보를 포함시켜야 한다. 추종 입력 속도에 대해 Laplace 변환을 하면 다음과 같다.

$$L[\dot{t}] = \frac{1}{s^2}, \quad L[\dot{t}^2] = \frac{1}{s^3} \quad (6)$$

이 식을 보면 추종입력은 원점에 극점을 갖고 있는 신호이며 최대 3개까지의 중복된 극점을 갖고 있음을 알 수 있다. 특히, $\frac{1}{s^3}$ 에서의 s^3 이 가장 많은 극점이므로 이를 기준으로 하여 보상기의 동특성을 선정하였으며 이를 이산화하여 다음의 식 (7)을 얻는다.

$$A_{ad} = \begin{bmatrix} 1 & T_s & \frac{T_s^2}{2} \\ 0 & 1 & T_s \\ 0 & 0 & 1 \end{bmatrix}, \quad B_{ad} = \begin{bmatrix} \frac{T_s^3}{6} \\ \frac{T_s^2}{2} \\ T_s \end{bmatrix} \quad (7)$$

한편, 대상 시스템인 전동차는 다음 식 (8)과 같이 점질량으로 모델링된다.

$$\frac{d}{dt} v(t) = -rv(t) + \frac{u(t)}{M} \quad (8)$$

여기에서 $v(t)$ 는 전동차의 속도, r 은 저항계수, M 은 전동차의 질량, $u(t)$ 는 전동차에 가해지는 힘을 나타낸다. 이 모델을 이산화하면 다음의 식 (9)를 얻는다.

$$A_d = e^{-rT} \quad B_d = \frac{1}{Mr}(1 - e^{-rT}) \quad (9)$$

이제, 대상 시스템과 보상기가 합쳐진 4차의 시스템에 대한 상태제환이득을 지정해야 한다. 이를 위해 우선 최적제어방법인 LQR을 이용하여 최적의 극점의 위치를 선정한 다음 그 위치를 기본으로 하여 극점의 위치를 변경하는 방법을 택하였다. 이 과정에서 LQR만을 적용할 경우 전체 시스템의 overshoot가 크게 나타나는 경향을 보여 가속 및 감속 시에 출렁거림으로 인해 목표속도로 안정화되는데 걸리는 시간이 크게 나타나는 단점이 발생하였다. 따라서 큰 overshoot를 만들어 내는 극점의 위치를 보완하여 전체적인 overshoot를 줄이는 방향으로 극점을 0.6982, 0.9682, 0.92+0.0332i, 0.92-0.0332i 으로 선정하였다. 이 때의 상태제환이득은 다음과 같다.

$$K = 1.9742 \text{e}6 \quad K_c = [2.3041 \text{e}6 \ 6.3462 \text{e}6 \ 5.4786 \text{e}6]$$

3.2 Communication Handler의 구현

COM은 직렬 통신을 담당하는 I/O handler와 I/O handler를 통해 데이터가 담긴 메시지를 외부 장치와 송수신하는 TCMS handler, ATC handler, TWC handler로 구성된다. I/O handler는 TCMS handler, ATC handler, TWC handler와 각각 짝을 이루어 하나씩의 독립된 task로 존재하며 50 msec 주기로 각각의 외부 장치와 통신을 수행한다.

I/O handler는 통신 채널을 초기화하고, 메시지의 송신, 수신을 담당하여 외부 시스템이 송신한 메시지를 수신하여 메시지가 올바른 것인지 검사한 후 받은 메시지를 외부 시스템에 해당하는 메시지 처리부(TCMS handler, ATC handler, TWC handler)로 보내준다. 또, 메시지 처리부(TCMS handler, ATC handler, TWC handler)로부터의 메시지를 외부 시스템으로 전송한다. 또, I/O handler는 TRA interface 기능을 수행하여 PSM을 검출하면 ATO application 내부의 PSM handler에 알려준다.

TCMS handler, ATC handler, TWC handler는 TCMS가 송신한 메시지를 정해진 프로토콜에 맞추어 해독하여 그 내용을 바탕으로 ATO application이 이용하는 정보를 갱신한다. 또, 각각 ATO application이 TCMS, ATC, TWC로 보고해야 할 사항을 정해진 프로토콜에 따른 메시지로 만들어 I/O handler로 보내어 전송하게 한다.

ATO와 TCMS, ATC, TWC 사이에서 주고 받는 데이터는 다음 그림 4와 같다.

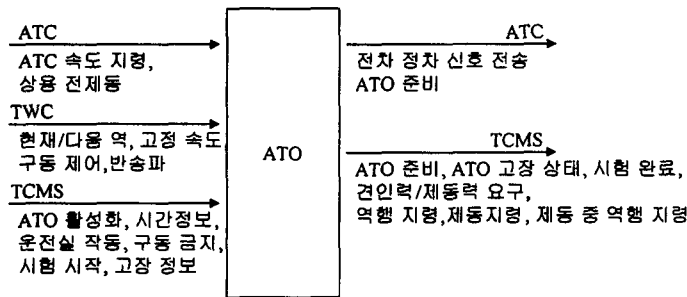


그림 4. ATO와 외부 장치간의 송수신 데이터

3.3 Application의 구현

application은 ART, SPR, REG, SRV, PSM으로 구성된다. 이들은 각각 RTOS 상에서 독립된 task로 존재하며 서로 유기적으로 필요한 정보를 주고 받는다.

ART는 외부 장치로부터 수집한 여러 가지 정보를 바탕으로 ATO의 동작 상태를 결정하고 동작 상태에 따라 지정된 동작을 하기 위해 ATO application의 다른 모듈들의 task를 제어한다. ATO 동작 상태들 사이에는 상태 천이 조건이 존재하고 운행에 관련된 정보와 외부 장치와의 통신을 통해 수집한 정보에 의해 특정한 상태 천이 조건이 만족될 경우에만 상태의 천이가 일어나도록 제어된다.

SPR은 운행 또는 정밀 정차를 위한 목표 속도 프로파일을 산출한다. 목표 속도 프로파일이 결정되면 현재 속도를 목표 속도로 맞추기 위해 필요한 제어입력(견인력/제동력)을 계산하여 REG로 넘겨준다. SPR의 속도 제어 알고리즘은 운행 시와 정밀 정차 시의 알고리즘 2개로 나뉘며 3절의 내용을 구현한 것이다.

REG는 SPR이 계산하여 전달해 주는 제어입력(견인력/제동력)을 실제의 전동기의 능력을 고려하여 변환한 후 TCMS로 넘겨주는 역할을 한다. 이런 변환이 필요한 이유는 전동차의 속도에 따라 전동기가 발생시킬 수 있는 최대 견인력/제동력이 달라지기 때문이다. REG는 SPR이 전달해 준 견인력/제동력이 그 시점에서의 속도에서의 최대 견인력 또는 최대 제동력에 대해 얼마만큼의 비효율을 계산하여 TCMS로 전송한다.

SRV는 트랙 데이터베이스를 접근하여 전동차가 운행하는 트랙에 대한 여러 가지 정보를 SPR에 전달해주는 역할을 한다. 트랙 데이터베이스에는 route 번호, 구간 거리, 출발역 번호, 출발역 TWC loop 번호, 정지역 번호, 정지역 TWC loop 번호, 출발 위치, 정지 위치, PSM의 위치, 구배의 위치와 구배 정도, 커브의 위치와 커브 정도 등 출발역과 정지역에 대한 정보가 수록되어 있다.

PSM은 PSM이 검출되면 검출된 PSM 정보를 SPR에 알려주고 PSM 위치와 연결 순서가 올바른

지 검사한다. 또, TRA의 정상적인 동작 여부를 검사한다.

4. 시뮬레이션을 통한 모의 운행 시험

개발된 ATO 장치의 성능을 검증하기 위해서 시뮬레이션 프로그램을 이용하여 모의 운행 시험을 실시하였다. 그림 5는 모의 운행 시험 환경을 나타낸 것이다. ATO Train Simulator는 PC 상에서 실행되어 전동차 모델을 모사하며, TCMS, TWC, ATC 등의 외부 장치를 흉내내어 ATO 장치와 여러 가지 데이터를 주고 받도록 작성한 소프트웨어이다.

CPU 보드의 역할은 ATO 소프트웨어의 실행환경을 제공하여 제어 알고리즘을 실행시키는 것과 차상 및 외부 장치와의 통신을 통해 ATO 소프트웨어에 데이터를 전달하는 것, 운행 기록 및 에러 상황에 대한 기록을 주기적으로 메모리에 저장하는 것이다. I/O 보드의 역할은 PSM을 검지하는 것과 현재 전동차의 속도를 알려주는 Tachometer 입력을 읽는 것, CPU 보드로 수집한 데이터를 전송하는 것이다. Tachometer 입력은 Pulse Signal 형태로 주어져야 하므로 본 모의 시험 환경에서는 Train Simulator가 계산한 현재 전동차의 속도를 National Instrument 사의 AT-MIO16E1 보드를 이용한 Pulse Signal Generator를 사용하여 Pulse Signal로 변환한 후 I/O 보드가 읽도록 하였다.

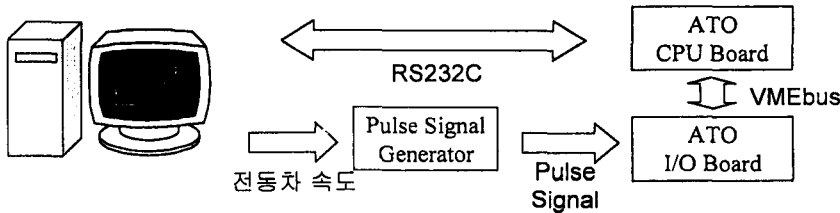


그림 5. Train Simulator를 이용한 모의 운행 시험 환경

모의 운행 시험은 CPU 보드 상에 탑재된 ATO 소프트웨어를 실행시켜 Train Simulator로 제어 입력을 계산하여 전달하면 Train Simulator는 그 결과인 전동차 속도를 I/O 보드로 보내고 이 값이 다시 CPU 보드의 ATO 소프트웨어로 반영되는 구조로 진행되었다.

그림 6, 7, 8, 9는 서울시 지하철 8호선의 트랙데이터베이스를 이용하여 전동차 하중이 200 ton이라는 가정 하에 구간 번호 61번을 목표 속도 60 km/h로 모의 운행한 결과를 나타낸 것이다. 61번 구간은 총 운행 길이가 850.03m이며 PSM1, PSM2, PSM3, PSM4의 위치는 출발 위치로부터 각각 304.03m, 741.53m, 829.03m, 846.53m 떨어진 곳에 있다. 그리고 총 3개의 구배 변화 지점이 포함되며, 커브 변화 지점은 포함되어 있지 않은 노선이다. 그림 6은 전동차의 속도를, 그림 7은 전동차의 상대 위치를, 그림 8은 전동차를 움직이기 위해 인가한 제어 입력을, 그림 9는 전동차의 저크를 각각 시간축에 따라 나타낸 것이다.

그림 6을 살펴보면, 운행을 시작하지 약 20초 후에 목표 속도인 60 km/h를 추종한다. 40초 후에 PSM1이 검지되면서 시속 40 km/h 까지 속도를 낮춘다. 약 70초 후에 PSM2를 검지하면서 다시 속도가 감소하며 약 90초 후에 정지한다. 그림 7을 살펴보면 정지 후 정밀 정차 에러가 약 -2.1 cm로서 일반적인 최대 허용치인 30 cm 이내를 충분히 만족함을 확인할 수 있다. 또, 그림 9를 살펴보면 저크가 본 연구에서 지정된 한계값인 최대 가속도 3.0km/h/s와 최대 감속도 3.5km/h/s 내에 포함되어 있음을 알 수 있다.

이러한 방식으로 지하철 8호선의 54번 ~ 65번 구간에 대한 모의 운행 시험을 실시하고 정밀 정차 에러 및 저크가 목표 사양을 충분히 만족시킴을 확인하였다.

5. 결론

본 논문에서는 추종 제어 기법을 응용한 ATO 속도 및 정밀정차 제어 장치의 개발을 살펴보았다. 하드웨어로는 제어 및 연산, 외부 장치와의 통신을 지원하는 CPU 보드와 전동차 상태에 대한 정보를 수집하는 I/O 보드를 설계, 제작하였다. 소프트웨어는 초기화 및 에러 검사를 담당하는 KRN 부분과 외부 장치와의 통신을 담당하는 COM 부분, 동작 상태 결정 및 목표 속도 제어와 정밀 정차 제어를 담당하는 Application 부분으로 나누고 각각을 세분화하여 구현한 후 통합하였다. 특히, Application 부분에서 가장 핵심적인 요소인 속도 제어 알고리즘을 운행 시의 목표 속도 제어와 정지 시의 정밀 정차 제어의 경우로 나누어 제시하고 적용하였다. 최종적으로 개발된 ATO 장치를 서울시 지하철 8호선의 트랙 데이터베이스를 가지고 시뮬레이션 프로그램과 연계하여 모의 운행 시험을 하고 그 결과를 분석하였다. 모의 운행 시험 결과 속도 제어 알고리즘의 설계대로 전동차의 속도가 제어되었으며 정밀 정차 에러와 저크 등의 설계 사양을 충분히 만족시키는 것을 확인하였다. 개발한 ATO 장치는 실제의 전동차 차량에 탑재되어 다른 외부 장치와 함께 동작 테스트를 거치게 될 것이다.

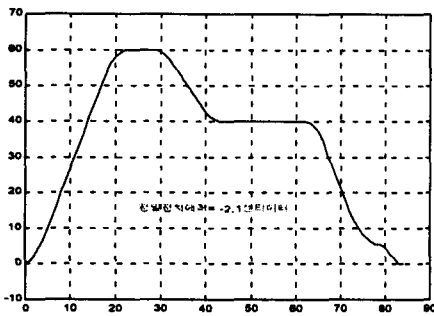


그림 6. 구간 번호 61에 대한 모의 운행 결과 시간(초) 대 속도(km/h) 그래프

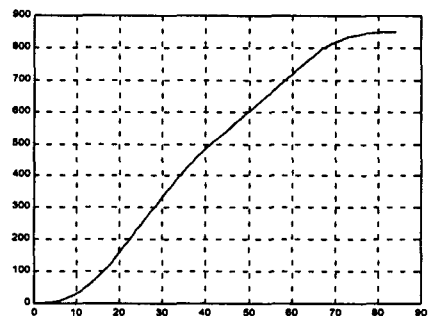


그림 7. 61번 구간에 대한 모의 운행 결과 시간(초) 대 거리(m) 그래프

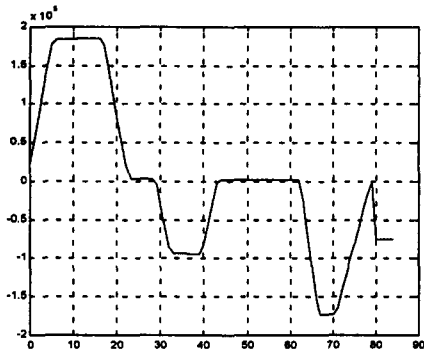


그림 8. 61번 구간에 대한 모의 운행 결과 시간(초) 대 제어입력(N) 그래프

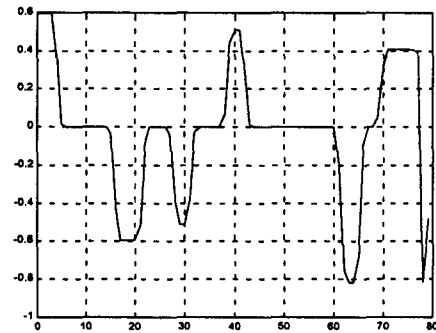


그림 9. 61번 구간에 대한 모의 운행 결과 시간(초) 대 저크(m/s³) 그래프

참고문헌

1. 우진산전 부설 기술연구소(1998), "종합제어장치 연구개발 보고서-ATO 분야"
2. Chi-Tsong. Chen(1984), "Linear System Theory and Design", Saunders College Publishing, pp.559-560
3. 김중환, 강리택(1998), "ATO 장치의 자동 열차 속도 조절 알고리즘에 관한 연구"