

TCS(Traction Control System)을 위한 실시간 시뮬레이터 개발에 관한 연구

A Study on Development of Real-Time Simulator for Electric Traction Control System

김태운¹ · 천세영² · 양순용^{1*}

Tae Un Kim, Seyoung Cheon and Soon Young Yang

Received: 02 Aug. 2019, Accepted: 23 Aug. 2019

Key Words : Traction Control System(트랙션 컨트롤 시스템), Engine Intervention Traction Control(엔진 간섭 트랙션 컨트롤), Brake Intervention Traction Control(제동 간섭 트랙션 컨트롤), Real-Time Simulation(실시간 시뮬레이션), Software In the Loop Simulation(소프트웨어 루프 시뮬레이션)

Abstract: The automotive market has recently been investing much time and costs in improving existing technologies such as ABS (Anti-lock Braking System) and TCS (Traction Control System) and developing new technologies. Additionally, various methods have been applied and developed to reduce this. Among them, the development method using the simulation has been mainly used and developed. In this paper, we have studied a method to develop SILS (Software In the Loop Simulation) for TCS which can test various environment variables under the same conditions. We modeled hardware (vehicle engine and ABS module) and software (control logic) of TCS using MATLAB/Simulink and Carsim. Simulation was performed on the climate, road surface, driving course, etc. to verify the TCS logic. By using SILS to develop TCS control logic and controller, it is possible to verify before production and reduce the development period, manpower and investment costs.

기호 설명

S : Slip Ratio

t_1 : Dead Time Between Air Intake and Explosion

t_2 : Lag Time of Intake System

T_e : Engine Torque

T_s : Target Engine Torque

Th_{na} : Wheel Deceleration Threshold

V_{Fmax} : Maximum front Wheel Speed

V_{ref} : Vehicle Reference Speed

w_{f1}, w_{f2} : Wheel Axle Angular Speed[rpm]

λ_{lim} : Wheel Slip Ratio Threshold

1. 서 론

근래의 자동차 시장은 수많은 편의 기능과 안전기술의 각축장으로 변모하고 있다. 자동차 회사들은 ABS, TCS와 같은 기존의 기술을 향상시키거나, 새로운 기술을 개발하기 위하여 많은 시간과 비용을 투자하고 있다. 또한, 그와 함께 투자 시간과 비용을 줄이기 위해 다양한 방법을 시도하고 발전시켜왔다. 그중 시뮬레이션을 이용한 개발 방법이 주로 이용되고 발전해왔다. 수학적 모델링을 통해 수치적 해석을 수행

* Corresponding author: soonyy@ulsan.ac.kr

1 Department of Mechanical Engineering, University of Ulsan, Ulsan 44610, Korea

2 Brake Control Engineering Team, Hyundai Mobis, Yongin-si, Gyeonggi-do 16891, Korea

Copyright © 2019, KSFC

This is an Open-Access article distributed under the terms of the Creative Commons Attribution Non-Commercial License(<http://creativecommons.org/licenses/by-nc/3.0>) which permits unrestricted non-commercial use, distribution, and reproduction in any medium, provided the original work is properly cited.

하던 초기 시뮬레이션은 비선형성을 지닌 하드웨어의 모델링에서 한계를 나타냈다. 이를 보완하기 위해 모델링이 어려운 하드웨어를 실험을 통해 얻은 결과로 대체하는 기법과 실물 그대로 시뮬레이션에 연동시키는 HILS(Hardware-In-the-Loop-Simulation) 기법으로 발전하였다.¹⁾ 하지만, HILS 역시 다양한 시스템 및 환경을 요구하는 현재의 개발프로세스에 MCU, 하드웨어를 개발해야 하는 단점을 가지고 있어 모두 적용하기 어렵다. HILS를 구성하기 전 MCU를 컴퓨터상의 소프트웨어로 대체하고 차량의 모델을 검증하는 SILS(Software In the Loop Simulation) 기법을 개발하여 차량 개발프로세스에 적용하는 연구가 진행되고 있다.^{2,4)}

본 논문에서는 MATLAB/Simulink를 이용하여 TCS의 차량 엔진, ABS 모듈 등의 하드웨어와 제어 소프트웨어를 모델링하였다. 또한, 차량의 주행 조건을 모델링하기 위해 Carsim을 이용하여 차량, 기후, 노면, 주행코스 등의 다양한 환경변수를 모델링하여 동일한 조건에서 반복 테스트가 가능하도록 하였다. 이를 통해 TCS 제어 로직 및 제어기 개발에 이용하기 위해 생산 전 검증이 가능한 TCS용 실시간 시뮬레이터를 개발하였다.

2. TCS 제어 개요

차량이 발진 또는 가속, 등반 시 구동륜과 노면 마찰력이 충분히 확보되지 못하면 구동륜이 헛돌아 차량이 구동, 가속되지 않는 현상이 발생한다. 이렇게 구동륜이 헛도는 현상을 방지하기 위해 구동륜의 속도를 제어함으로써 마찰력을 확보하는 시스템을 TCS(Traction Control System)라 한다. 노면과의 마찰력을 확보하려는 근본적인 목표에서는 ABS(Anti-Lock Brake System)와 유사하지만, ABS와 TCS의 차이점으로 ABS는 감속 상황에서 제동력 확보가 목적이고 TCS는 가속 상황에서 구동력 및 선회 안정성을 유지해 차량이 차선을 이탈하는 것을 방지하는 것이 목적이다.⁵⁾ TCS의 기능은 크게 미끄러지기 쉬운 노면에서 가속성 및 선회 안정성을 확보하여 주는 슬립제어 기능과 선회 가속 시에 구동력을 제어하여 조향 성능을 향상시키는 트레이스 제어 기능으로 구분 할 수 있다.

2.1 슬립제어

슬립제어란 후륜 센서에서 얻어지는 차속과 전륜 센서에서 얻어지는 구동륜속도를 비교하여 구동차량

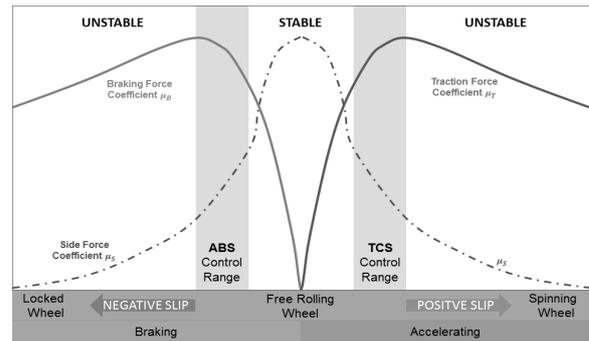


Fig. 1 Slip control region of ABS / TCS

의 슬립율이 적정 수준에서 벗어나지 않게 엔진의 출력 및 구동륜의 제동을 제어하는 것이다. 이는 차량이 주행할 때에는 타이어에 생기는 구동력과 횡력이 직진과 선회 시 각 상황에서 최고효율을 가질 수 있도록 슬립율을 제어한다. Fig. 1은 TCS와 ABS의 슬립 제어 영역에 대한 것으로 TCS의 제어 영역은 최대 접지력을 확보할 수 있는 영역이며 보통 슬립율이 20~30% 정도이다. 선회 시에는 이보다 낮은 10~20% 정도의 슬립율을 확보한다.

2.2 트레이스 제어

트레이스 제어는 조향 휠과 가속 페달의 출력값으로부터 운전자의 선회와 가속 의지를 판단한다. 이때 비구동륜의 좌, 우측 속도차를 검출하여 구동력을 제어하여 안정된 선회를 가능하게 한다. 후륜 좌우 센서에서 측정된 차륜 속도 차이를 통한 선회 반경과 평균치로부터 차체 속도를 연산하고 두 값을 통해 구해진 횡력이 기준값을 초과할 때 구동력을 제어하게 된다. 조향각 센서와 가속페달 센서를 통해 운전자의 선회의지와 가속 의지를 판단하여 가속 페달을 밟은 상태에서도 적절한 조향이 가능하도록 한다. 본 논문에서는 조향 휠이나 가속 페달 센서를 사용하지 않고 차륜의 속도만을 이용하여 TCS 제어 로직을 구현하였다.

3. TCS 제어 로직

TCS의 제어 방법은 크게 2가지 방식으로 나눌 수 있다. 첫 번째는 엔진 토크 등 출력을 제한하거나, T/M의 구동력을 제어하는 ETCS 방식으로 안정적이지만 상대적으로 응답성이 느린 특징이 있다. 두 번째는 ABS 모듈을 이용하여 직접 제동력을 작용하는 BTCS 방식으로 응답성은 빠르나 브레이크를 사용하기 때문에 제어하기 까다롭다는 특징이 있다. 두 가

지 방식을 상황에 따라 단독 또는 병합하여 TCS 제어에 사용한다.

3.1 BTCS (Brake Traction Control System)

BTCS는 차량의 구동축인 2개의 전륜의 슬립 값을 판단기준으로 하여, 차량의 속도와 양쪽 전륜의 슬립 값 3개의 항목을 비교하여 전륜에서 모두 슬립이 일어나는 경우와 전륜 중 한쪽에서 슬립이 크게 일어나는 경우로 나누어 Yaw Moment를 보상하는 Flag와 각 구동륜의BTCS Flag를 활성화한다. 이 Yaw Moment Flag를 기준으로 전륜 좌, 우의 BTCS Flag를 활성화하게 되면 구동축인 2개의 전륜 슬립값과 BTCS Flag 값을 판단기준으로 하여 각 브레이크의 가압, 유지, 감압 동작을 수행하여 슬립을 줄인다.⁶⁾

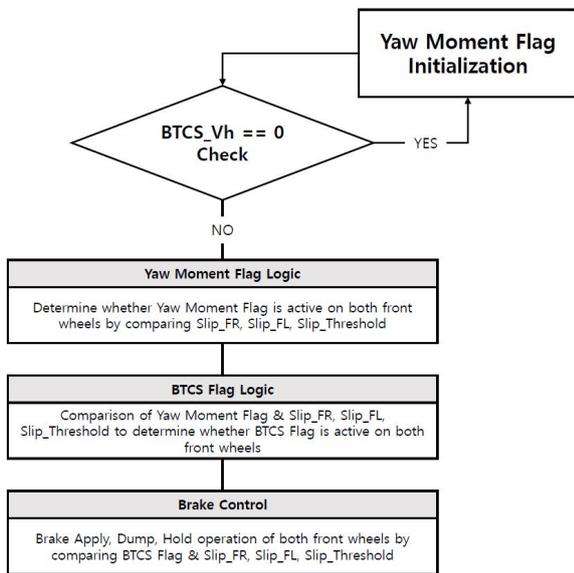


Fig. 2 BTCS Control Logic

3.2 ETCS(Engine Traction Control System)

ETCS는 차량의 속도와 구동륜의 속도차를 측정하여 두 속도차가 일정 수준 이상이 될 경우 PI 제어를 통해 엔진 토크를 줄임으로써 슬립을 최소화한다. 제어 로직은 엔진 모델로부터 현재 토크를 입력받아 구동력 확보를 위해 계산된 공칭 엔진 토크를 출력한다. 이를 CAN 통신을 통해 ECU로 전달하며 ECU는 입력받은 토크로 엔진을 제어하게 된다. ETCS에서 사용되는 엔진 모델은 간단히 엔진전달함수로 구성할 수 있으며 식 (1)과 같다.

$$T_e = e^{-st_1} \frac{1}{1 + st_2} T_s \quad (1)$$

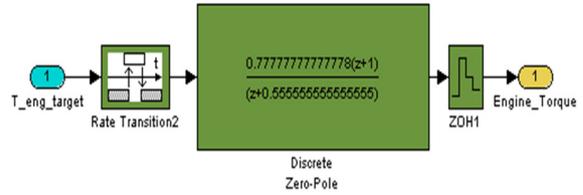


Fig. 3 Engine Transfer Function Model

3.2.1 구동력 제어

전륜구동 차량이 미끄러운 노면에서 구동할 경우 전륜의 속도가 항상 후륜보다 크게 된다. 이때 TCS는 전륜 중에 속도가 더 큰 구동륜을 제어하여 슬립율을 적정 수준으로 유지한다. 이때, 슬립율(S)는 현재 차륜 속도와 기준 차속의 차를 기준 차속으로 나눈 값이며, 식 (2)와 같이 표현된다.

$$S = \frac{V_{wheel} - V_{ref}}{V_{ref}} \quad (2)$$

식 (2)는 식 (3)과 같이 표현 가능하며,

$$V_{wheel} = (1 + S) V_{ref} \quad (3)$$

식 (3)과 식 (4)를 이용하여 타겟 슬립율(λ^*), 타겟 차륜 속도(V^*), 기준 차속(V_{ref})을 결정한다. 이때, 슬립값만으로 제어를 수행할 경우 저속 시 차륜 속도의 경계점이 지나치게 작아지는 문제와 고속 시 차륜 속도 경계점의 절댓값이 커지는 문제점이 발생한다. 이를 해결하기 위해서 차속에 따라 차륜의 슬립 허용치를 다르게 적용해야 한다. 슬립 허용치의 계산을 간단히 하기 위하여 식 (4)와 같이 전륜의 속도에서 현재 차속을 뺀 값을 속도차(D_V)로 정의하여 계산한다. 차량의 차속에 따른 허용 속도차는 Fig. 4의 맵핑

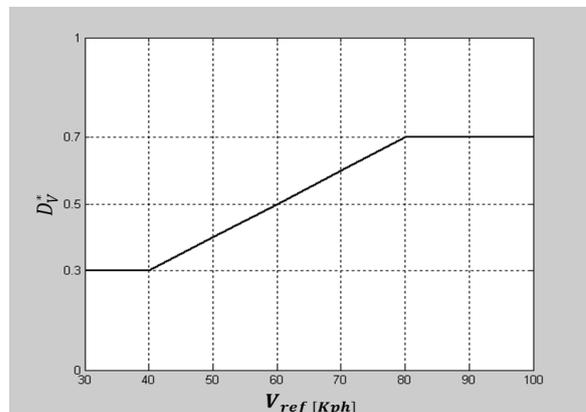


Fig. 4 D_V Target Mapping Curve

기울기를 적용하여 식 (5)와 같이 타겟 속도차를 결정한다. 이 식 (4)의 값이 식 (5)의 타겟 범위를 만족하도록 제어한다.

$$D_V = V_{Fmax} - V_{ref} \quad (4)$$

$$D_V^* = f(V_{ref}) \quad (5)$$

3.2.2 구동력 제어의 작동 및 Flag 설정

구동력 제어 Flag는 Fig. 5와 같은 조건으로 설정된다. ABS 로직이 작동된 경우는 사용자가 브레이크 페달을 작동한 경우로 간주 할 수 있다. 이때 TPS 값을 0으로 볼 수 있으며 추가적인 엔진 제어, ETCS 제어가 필요하지 않다는 뜻이다.

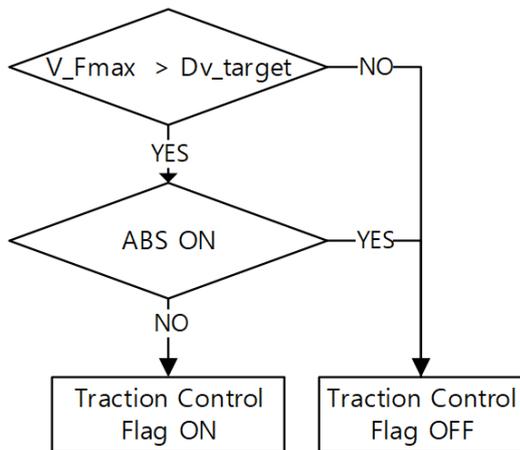


Fig. 5 ETCS Control Logic

구동력 제어 Flag가 ON 되면 구동력 제어가 시작되며, 이후 전륜, 후륜 속도차(D_V)가 목표치(D_V^*) 이 내에서 일정 시간 동안 지속해서 유지되었을 경우만 구동력 제어를 멈추게 된다. 이는 제어의 안정성을 보장하기 위해 사용되는 로직으로 순간적인 안정, 불안정이 반복되는 조건에서 유용하다. 이때 유지되는 시간은 튜닝에 의해 결정된다.

3.2.3 구동력 제어를 위한 PI 제어기

앞서 결정된 D_V^* 를 이용하여 타겟 슬립율(λ^*)을 식 (6) 과 같이 정의한다.

$$\lambda_w^* = \frac{D_{V_{target}}}{V_{Fmax} + 1} \quad (6)$$

분모에 더해진 “1” 은 V_{Fmax} 가 0에 가까운 매우 작은 값일 때 발생하는 수치적인 오류를 방지하기 위한 값이다. 계산된 타겟 슬립율을 식 (3)에 대입하여 식 (7)과 같이 타겟 차속을 계산한다.

$$V_w^* = (1 + \lambda_w^*) V_{ref} \quad (7)$$

계산된 목표 차속(V_w^*)을 차륜속도 V_{Fmax} 와 비교하여 제어를 수행한다. 즉 타겟 차속과 V_{Fmax} 의 차이를 오차 값으로 하는 PI 제어기를 설계하여 엔진 토크제어를 수행한다.

$$T_{engine}^* = \left[K_p + \frac{K_i}{s} \right] (V_w^* - V_{Fmax}) \quad (8)$$

3.2.4 트레이스 제어

본 논문의 TCS에서 트레이스 제어는 횡가속도 센서, Yaw Rate 센서, 조향각 센서에 의한 입력 요소를 고려하지 않았으며 4개의 차륜에서 입력되는 속도만으로 트레이스 제어를 수행하는 방식으로 개발하였다. 트레이스 제어에 필요한 Yaw Rate는 각 차륜의 속도를 이용하여 추정하였다. 측면 슬립이 없고 차륜 슬립에 의해서만 Yaw Moment가 생긴다고 가정하면 TCS가 작동하는 조건에서는 비구동륜인 후륜의 속도를 통해 추정된 Yaw Rate를 차량의 Yaw Rate 혹은 차체의 운동 기준으로 판단하여 전륜 속도를 통해 추정된 Yaw Rate와 비교하여 제어를 수행한다. 전륜, 후륜 각각의 Yaw Rate와 횡가속도를 추정하면 식 (9)와 식 (10)과 같이 표현 할 수 있다.

$$\begin{aligned} r_r &= (V_{OutRear} - V_{InRear})/t_{wheel} \\ r_f &= (V_{OutFront} - V_{InFront})/t_{wheel} \end{aligned} \quad (9)$$

$$\begin{aligned} A_{yr} &= r_r V_{ref} \\ A_{yf} &= r_f V_{ref} \end{aligned} \quad (10)$$

두 횡가속도의 차이가 Yaw Moment의 에러 값이 된다. 이러한 에러를 보상해 주는 토크를 계산하여 제어 신호로 출력한다.

$$T_{error} = \left[k_{pe} + \frac{K_{ie}}{s} \right] \cdot C \cdot (|A_{yf}| - K_{TCGain}|A_{yr}|) \quad (11)$$

4. 시스템 및 시뮬레이션 구성

TCS는 기존 ABS 장착 차량의 ABS 제어기 및 시스템을 동일하게 사용하여 BTCS 제어 로직을 수행한다. 또한, 엔진을 제어하며 ETCS를 제어 로직을 수행한다. TCS 로직을 처리하는 TCS 모듈과 차량 시스템 구성 개념도는 Fig. 6과 같다.

본 논문에서는 BTCS 제어 로직을 구현하기 위한 제동시스템을 참고문헌의 ABS 시스템에서 참고하여 TCS 제어기 시스템에 응용하였다.⁷⁾ 또한, 실제 제어기 구축과정에서 각 제어시스템에서 공동으로 사용할 수 있는 파라미터는 공유하여 사용함으로써 제어기의 프로세스 부하를 줄였다. 특히, TCS에서 제어상태를 판단하게 되는 각 차륜의 슬립 값은 ABS에서의 계산된 차륜 슬립 결과 값을 사용하였다. 시뮬레이션의 구성은 Fig. 7과 같다. 차량 동역학 모델은 Carsim으로 트랜스미션 모델, TCS 로직, 브레이크 로직, 브레이크 유압 모델을 가지는 ABS 모듈을 Matlab/Simulink로 모델링하였다. 차량 모델은 Carsim을 이용⁸⁾하여 C-Segment Data를 바탕으로 모델링 하였으며, 엔진 모델은 시뮬레이션 처리속도 및 부하를 줄이기

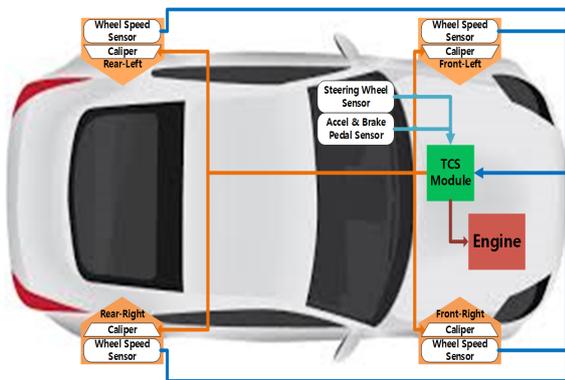


Fig. 6 Configuration of TCS System

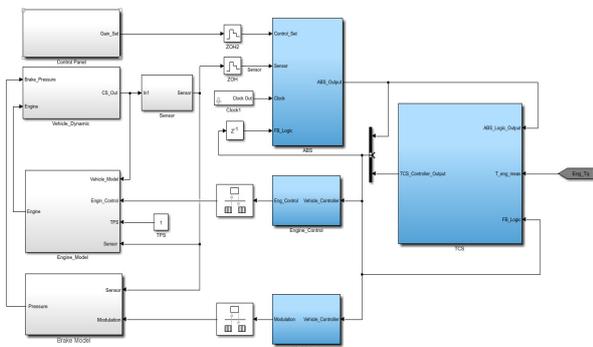


Fig. 7 TCS Simulation Simulink Model with ABS Logic

위하여 Simulink에 엔진 토크 맵 데이터를 배치하여 모델링 하였다.

시뮬레이션 모델을 구성하여 TCS 로직이 작동하는 조건과 그 조건을 만족하기 위한 차량의 상태를 시뮬레이션 상에서 Table 1과 같이 설정하였다.

Table 1 TCS Logic Operation Condition

	TCS Operation Table	Simulation Setting
1	Acceleration intention	TPS funcion
2	TCS Switch ON	ON
3	Gear : Not (P or N)	Drive Gear State
4	Brake Not-Operation	BLS Signal Off
5	ABS Not-Operation	Off Logic

TCS Logic Model에는 사용되는 입출력 변수와 그 변수의 처리는 다음과 같이 설정하였다.

V_{ref} 는 차속으로 ABS Module의 Observer 출력값을 이용하고 차륜속 각속도는 Carsim 출력을 센서값으로 다시 입력하였다. Gear Ratio는 Carsim 출력을 TPS Signal은 Simulink에서 TPS 함수를 생성하여 입력하였다. BTCS는 ABS모듈을 이용하여 기능 구현하고 ETCS는 Throttle, 연료분사방식, 점화 시기 등의 직접적 제어 모사를 하지 않고 엔진 ECU가 존재한다고 가정하고 타겟 토크를 명령하여 엔진 출력이 변한다고 가정하였다.

5. Simulation 결과

노면 마찰력, 언덕 구배 등의 환경 변수의 변동과 TCS 제어 여부에 따른 차량의 거동을 분석하기 위해 실시간 TCS 시뮬레이션을 수행하였다. 결과로 설계한 TCS 제어기가 효과적으로 작동하여 차량의 구동특성이 다음과 같이 향상됨을 확인하였다. 첫 번째로 마찰력이 작은 노면조건($\mu = 0.35$)에서 차량의 구동륜인 전륜과 후륜의 속도차가 일정 범위 내에서 유지되며, 0.15g의 가속 성능을 확보하였다. 두 번째로 결빙에 의해 좌우 노면의 마찰력이 달라지는 언덕 조건($\mu_1 = 0.85, \mu_2 = 0.35, 15\% \text{ Hill}$)에서는 Yaw Rate $\pm 1.2 \text{ deg/s}$ 이하를 만족하며 등판 가능성을 확인하였다. 마지막으로 원형 도로에서의 고속 주행 조건에서 TCS를 사용하지 않을 때 목표 경로를 5m 이상 벗어나는 경우가 발생하지만, TCS 제어 시 목표 경로를 1m 내외의 차이로 주행하여 트레이스 제어가 유효하게 작동하고 있음을 확인하였다. 각 조건에 대한 시뮬레이

선의 상세 결과 및 분석을 아래와 같이 나타내었다.

5.1 젖은 노면 (Low mu)

초기 속도가 0kph, Full Throttle, $\mu = 0.35$ 의 조건으로 차량이 주행했을 때, 낮은 μ 값에 의해 구동륜인 전륜에서 슬립이 발생하였다. TCS 기능을 사용하지 않은 경우, 구동륜과 실제 차속이 30kph 이상의 차이를 보였으나 TCS 기능을 사용하여 최대 10kph의 범위 내에서 구동륜의 슬립을 성공적으로 제어하였다.

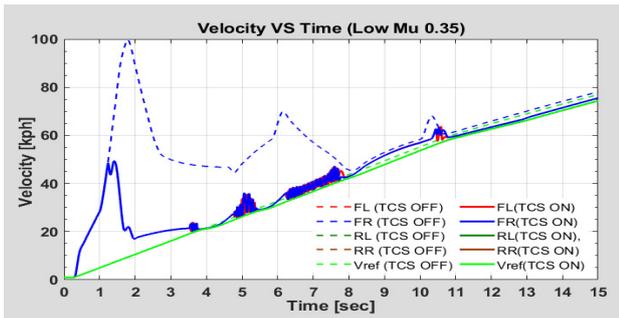


Fig. 8 Velocity Curves: 0.35 Mu (TCS On /Off)

TCS 중 ETCS가 작동하여 Fig. 9와 같이 엔진 토크를 제어하여 구동륜의 슬립을 제어하여 구동력 손실을 방지하였다. 또한, 차량이 50kph까지 가속하는데 약 0.15g의 가속 성능을 확보하였다.

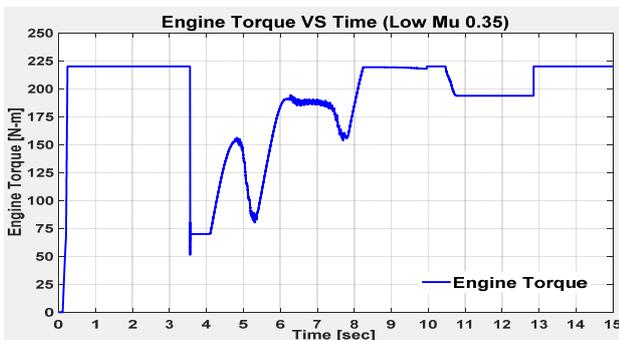


Fig. 9 Engine Torque Curves: 0.35 Mu (Split)

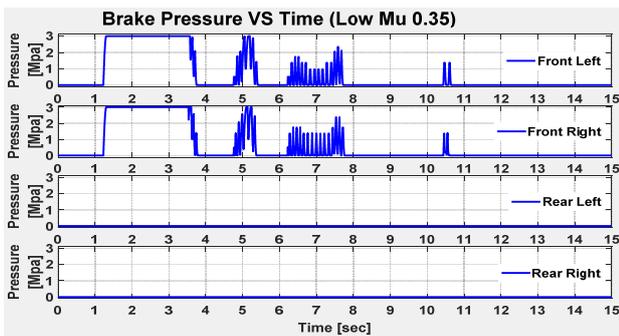


Fig. 10 Brake Pressure Curves: 0.35 Mu

TCS는 양륜의 제동압력을 동시에 제어하지 않고 한쪽씩 번갈아 가며 제어를 수행하게 된다. 그 결과 Fig. 10과 같이 간격을 두고 양륜의 브레이크 압력 차가 발생하여 양 구동륜의 속도차가 생기는 모습을 Fig. 8에서 확인 할 수 있다.

이번 시뮬레이션에서는 60kph 이하 구간에서 ETCS와 BTCS를 동시에 사용하였으며 그 이상 구간에서는 ETCS를 단독으로 사용하도록 설정하였다. Fig. 10과 같이 설정에 의해 10.5초 지점에서 전륜의 속도차가 벌어져도 브레이크 압력이 상승하지 않는다.

5.2 결빙구간이 존재하는 언덕 (Split mu)

초기 속도가 5kph, Full Throttle, $\mu_1 = 0.85$, $\mu_2 = 0.35$, Split Condition after 5m, Switching Every 50m의 조건으로 15% 경사로를 주행하다가 좌우 노면이 다른 마찰력을 가지는 노면에 진입하였을 때 TCS 기능을 사용하지 않은 경우 Yaw Rate가 최대 2.4 deg/s 이상을 넘어갔으나 TCS 기능을 사용한 차량은 Yaw rate ± 1.2 deg/s 조건을 만족하며 언덕 구간을 주행하였다. TCS 미사용 시 구동륜 중 한 쪽이 결빙구간($\mu = 0.35$)에 진입하게 되면 Fig. 11과 같이 급격히 차량의 바퀴가 헛돌게 되어 Fig. 12의 4초 부근에서 볼 수 있듯이 Yaw rate가 순간적으로 2.4deg/s를 초과하게 된다.

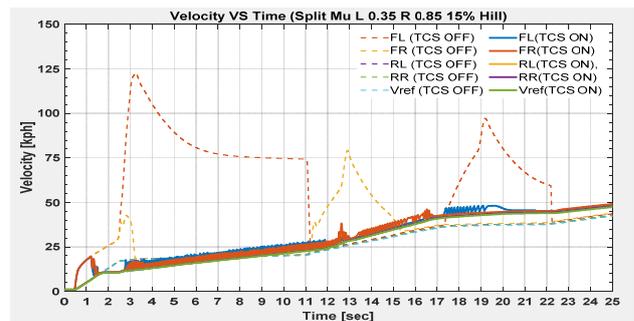


Fig. 11 Velocity Curves: Split Mu (TCS On /Off)

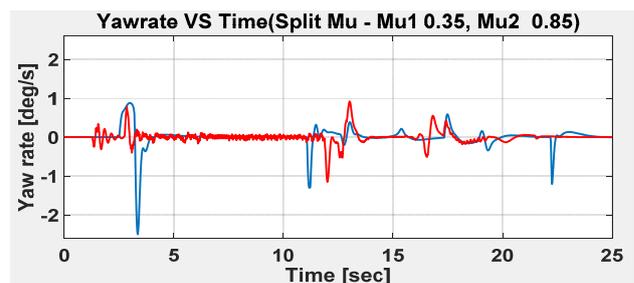


Fig. 12 Yaw rate Curves: Split Mu (TCS On/Off)

TCS는 헛도는 구동륜을 제어하기 위하여 BTCS, ETCS를 적절히 조합하여 구동륜의 슬립을 제어한다. Fig. 13과 같이 결빙구간에 진입할 때 헛도는 구동륜이 바뀌어 제동압력도 그에 따라 변하는 것을 확인할 수 있다.

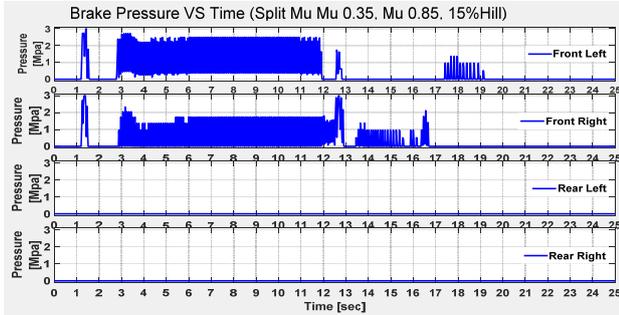


Fig. 13 Brake Pressure: Split Mu (TCS On/Off)

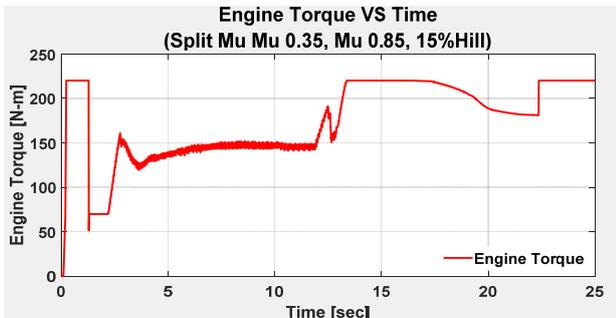


Fig. 14 Engine Torque: Split Mu

5.3 원형도로 (Circle Road)

회전반경 400ft, 초기 속도 100kph, Full Throttle, $\mu = 0.85$ 의 조건으로 400ft의 원형 도로를 100kph 이상의 속도로 주행하며 트레이스 제어를 사용할 때와 사용하지 않는 경우를 시뮬레이션하여 비교 분석하였다.

시뮬레이션 결과 Fig. 17과 같이 TCS의 트레이스 제어를 사용하지 않는 경우에는, 기준 경로와 차량의 실제 경로가 크게는 5m 이상 차이가 발생하는 데 반해 트레이스 제어를 사용하였을 때에는 기준 경로를 오차 범위 $\pm 1m$ 에서 잘 추종하는 것을 확인할 수 있다. 하지만, 본 논문에서 개발한 TCS의 트레이스 제어는 앞서 언급한 바와 같이 횡가속도 센서, 조향각 센서에 의한 입력 요소를 고려하지 않고 차량의 속도를 이용한 Yaw Rate 추정 결과만을 이용하기 때문에 Fig. 18과 같이 정확한 Circle Road 제어에 어려움이 있다.

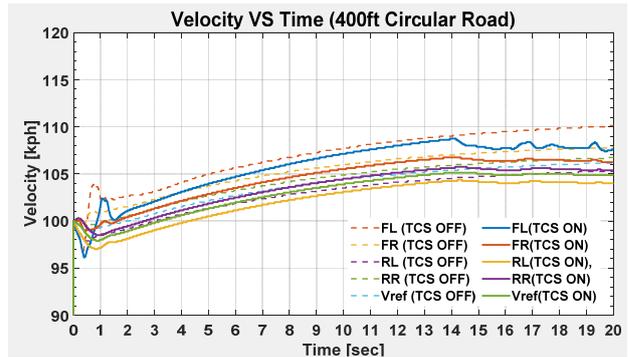


Fig. 15 Velocity Curves: Circular Road (TCS On /Off)

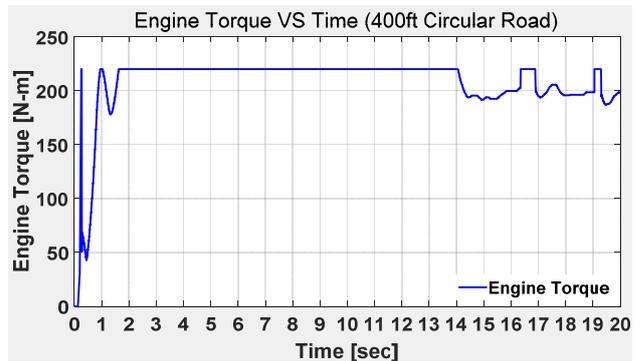


Fig. 16 Engine Torque Curves: Circular Road

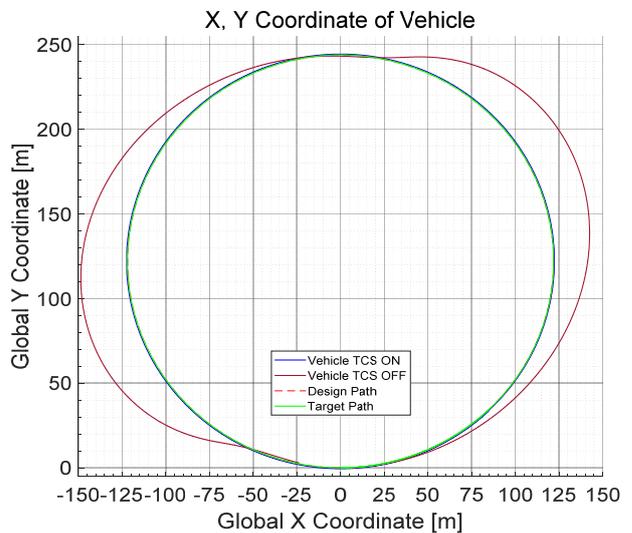


Fig. 17 X, Y Coordinate of Vehicle

평지 Low Mu 조건, 언덕길에서의 Split Mu 조건, Circle Road의 고속 조건에서 TCS의 유무에 따른 실제 차량의 거동과 TCS SILS에서의 TCS 동작 여부에 따른 구동력, 트레이스 제어 결과가 거의 유사하게 나타나는 것을 확인하였다.

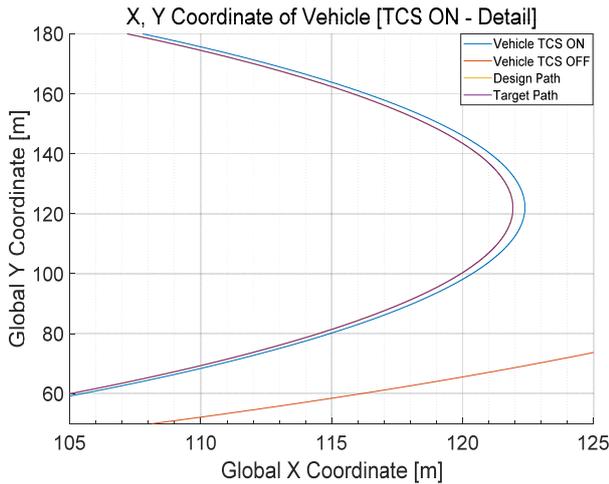


Fig. 18 X, Y Coordinate of Vehicle (Detail)

6. 결 론

본 논문에서는 MATLAB / Simulink 와 Carsim을 이용하여 차량 엔진, ABS 모듈 등의 H/W 및 TCS 제어 로직 S/W를 모델링하여 SILS 실시간 환경으로 개발하였다. 개발한 SILS를 이용하여 노면, 주행코스 등의 환경변수 변경해가며 시뮬레이션을 수행하여 설계한 알고리즘에 맞춰 작동하는 것을 확인하였다. 시뮬레이션의 결과가 실차의 TCS 실제 동작과 유사함⁹⁾으로, 본 연구에서 개발한 실시간 시뮬레이터를 TCS 제어 로직 개발에 사용하여 개발 기간의 단축과 비용 절감을 기대 할 수 있다.

향후, 시뮬레이션 결과와 실차 데이터를 비교 분석하고 H/W 모델을 보완하여 시뮬레이터를 보완 구축할 예정이며, 해당 모델을 전기차 모델로 수정하여 전기차의 ETCS 로직을 개발할 계획이다.

References

- 1) J. H. Ryu et al, "Development of Network - based Traction Control System and Study its on Performance Evaluation using Net-HILS", Transactions of the Korean Society of Automotive Engineers, Vol.14, No.5, pp.47-57, 2006.
- 2) Y. H. Yoon and K. W. Lee, "Development of SIL, HILS Environment for Electronic Control System", Journal of Drive and Control, Vol.5, No.1, pp.17-21, 2008.
- 3) G. H. Jung, "Development of Transmission Simulator for High-Speed Tracked Vehicles", Journal of Drive and Control, Vol.14, No.4, pp.29-36, 2017.
- 4) S. C. Han, "Development of an Electronic Simulator for Efficiency Improvement to Verify Electro-Hydraulic Servo Controllers in an Examination Set-up", Journal of Drive and Control, Vol.14, No.3, pp.50-57, 2017.
- 5) J. H. Song et al, "Development of a Control Method of Traction Control System Using Vehicle Model" Transactions of the Korean Society of Mechanical Engineers A, Vol.28, No.8, pp.1203-1211, 2004.
- 6) S. Y. Kim, "Traction Control with Brake Pressure Estimation", Journal of Drive and Control, Vol.10 No.3, pp.1-6, 2013.
- 7) S. Y. Cheon, S. W. Choi and S. Y. Yang, "Development of Real-time Simulator for Vehicle Electric Brake System" Journal of Drive and Control, Vol.16, No.1, pp.22-28, 2019.
- 8) J. Y. Yoon et al, "Evaluation of Unified Chassis Control System Using a Driving Simulator", Proceedings of KSAE 2009 Annual Spring Conference & Exhibition, pp.560-565, 2009.
- 9) B. C. Kim, J. B. Song, D. C. Shin, "Development of Engine Control Based TCS Slip Control Algorithm Based on Actual Vehicle Test", Proceedings of the KSME 1998 Spring Annual Meeting, pp.488-493, 1998.