

전기자동차 설계 프로젝트 2차 보고서



Group 8

이반석 2012312854 권준윤 2013311415
백승한 2013313342 안병준 2013311142
전민수 2013312697 조성준 2014314247

Class 41

2017-11-20

Sungkyunkwan University, School of Mechanical Engineering

목차

전기자동차 설계 프로젝트 2차 보고서

1) 전기자동차 설계 사양

----- 3p

2) 시뮬레이터 개발

----- 4p

3) 시뮬레이터 검증

----- 15p

4) 회생제동

-- 28p

5) UDDS 사이클 주행시 SOC 변화

----- 30p

6) UDDS 사이클 주행시 전비

----- 33p

1. 전기자동차 설계 사양

대상 전기자동차 선정

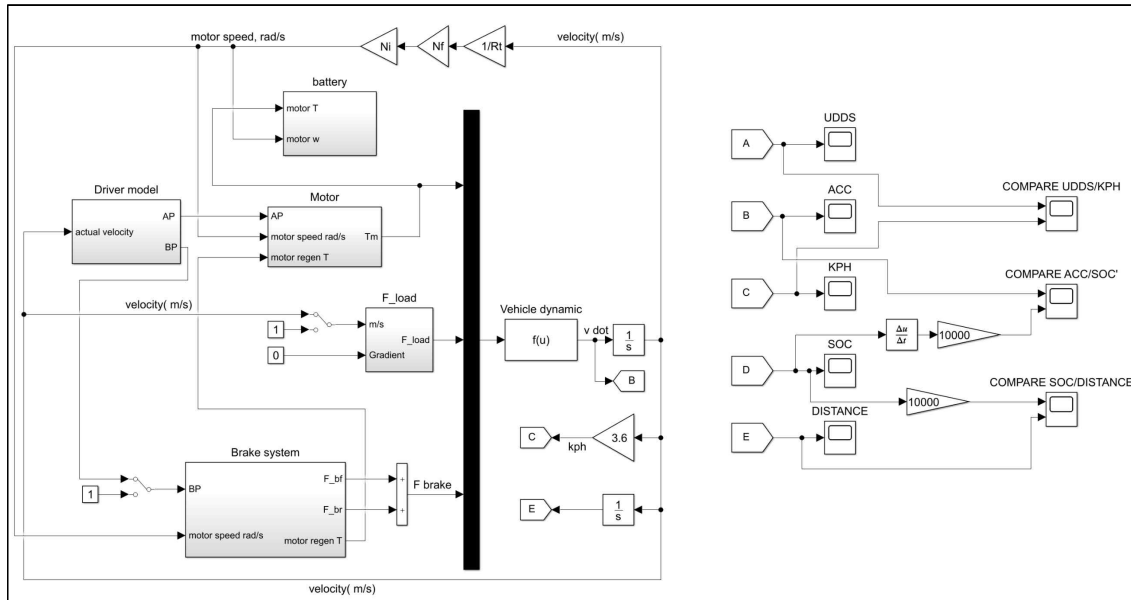
저희 조는 회생제동이 추가된 이번 과제에서 회생제동을 수행하기 앞서 1차 과제에서 설계한 전기자동차 모델의 수정하여 **최대전력 101.7kW, 최대속도 200kph, 제로백 10초**로 설계한 임의의 전기자동차를 설정하고 1차 과제의 항목들을 수행하기 위해 다시 시뮬레이터를 돌려보았습니다.

* Simulated Car Properties

		Simulated Car
Engine/Motor	Power (kW)	101.7
	MAX Torque (Nm)	259.07
	Base RPM (rpm)	3750.6
Electric	Energy (kWh)	28
	Voltage (V)	360
Dimensions/Weight	Weight Empty (kg)	N/A
	Weight Full (kg)	1800
	Passenger	5
Performance	0-100 (s)	10.0
	Top speed (kph)	200
Constants	Transmission gear ratio	2.3
	Final gear ratio	4
	Inertia of motor (kg*m^2)	0.05
	Inertia of gear (kg*m^2)	0.00226
	Inertia of a wheel (kg*m^2)	0.905
	Radius of tire (m)	0.316
	Gravity (m/s^2)	9.81
	Coefficient of Rolling resistance	0.01
	Air density (kg/m^3)	1.166
	Corfficient of air resistance	0.35
	Frontal area of vehicle (m^2)	2

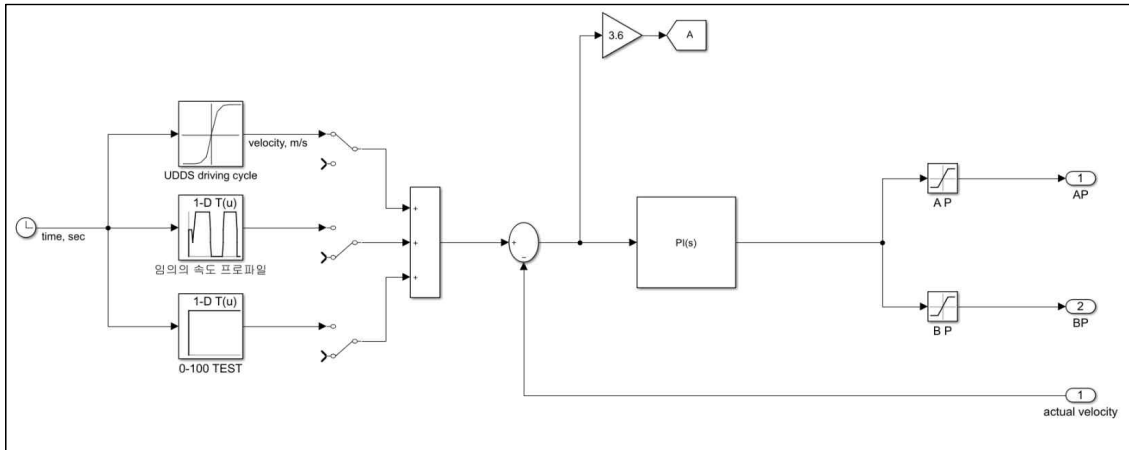
2. 시뮬레이터 개발

전체 시뮬레이터 모델



저희 조에서 구성한 시뮬레이터의 전체적인 모습입니다. 자동차 모델이 구동되는 부분과 결과를 확인하는 부분을 구분 지어서 한눈에 보이도록 구성하였습니다. 왼쪽의 시뮬레이터 부분을 보면 크게 운전자 모델, 브레이크 시스템, 배터리, 모터, 하중조건, 차량동역학 부분으로 나눌 수 있습니다. 이번에 설계한 전기자동차 시뮬레이터는 실제로는 복잡한 시스템을 단순화하는 방법을 이용하여 만들어진 시뮬레이터입니다. 아래에서 각각 요소들을 분석하고 살펴보겠습니다.

운전자 모델



1) 속도 프로파일

운전자 모델에서 요구되는 속도 프로파일을 Input 값으로 넣어 줍니다. 이때 스위치를 통해 UDDS, 임의의 속도 프로파일, 0-100 TEST의 세가지를 전환하여 넣을 수 있게 만들었습니다.

2) PI Control

여러 P gain, K gain값을 테스트 해보기 위해 시뮬링크에 내장되어 있는 PI제어기를 이용하였습니다. 더블클릭 하여서 간단하게 gain값을 조정할 수 있습니다.

3) AP&BP

Error (Demend Velocity - Actual Velocity) 값에 따라서 가속이 필요하면 0~1의 AP값을 주고 감속이 필요하면 -1~0의 BP값을 주게 됩니다. 이때 AP값은 모터 모델로 들어가게 되고 BP값은 브레이크 모델로 들어가게 됩니다.

동시에 값이 출력되는 일이 없도록 Saturation block을 이용하였습니다.

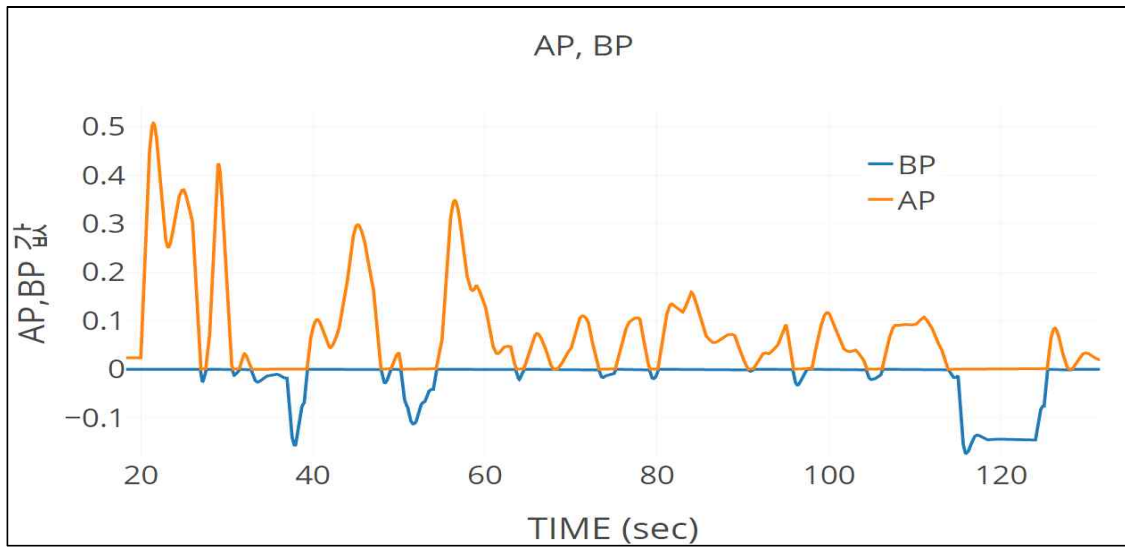
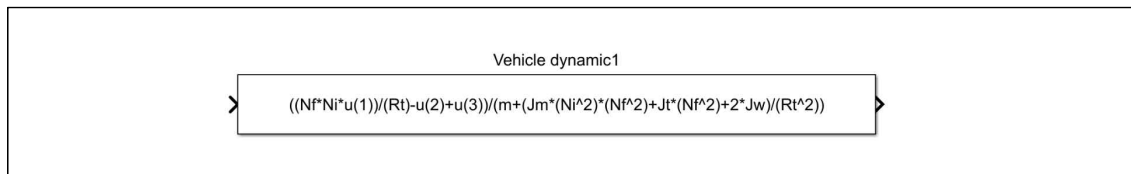
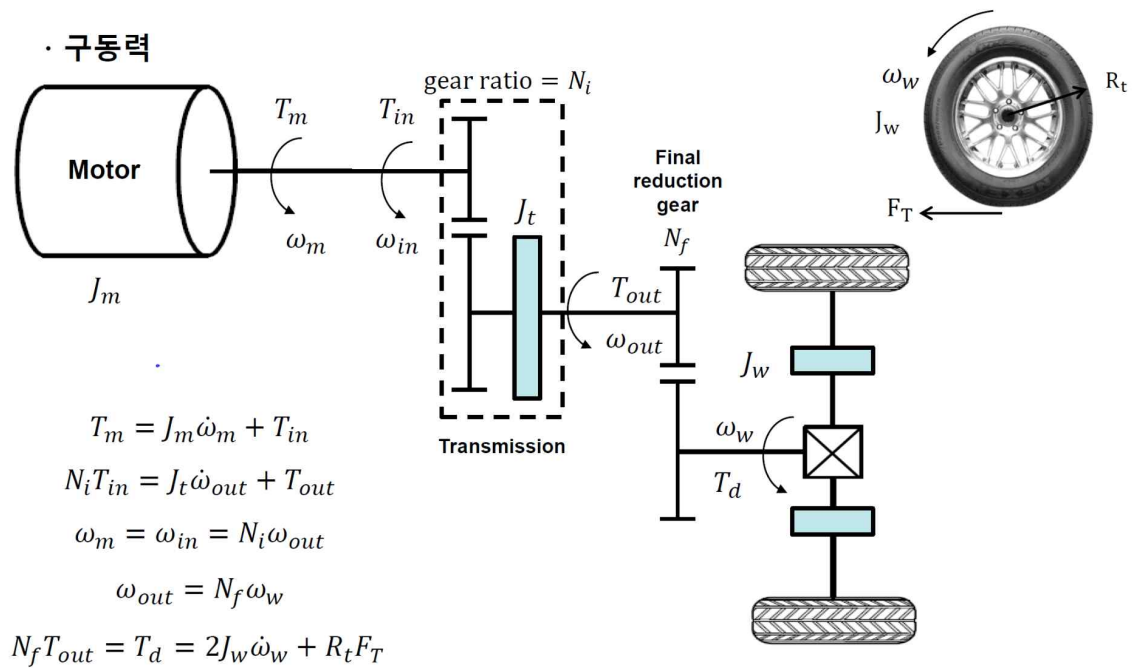


Figure 1. AP,BP-시간 그래프

차량동역학 모델



1) 차량 동역학 모델



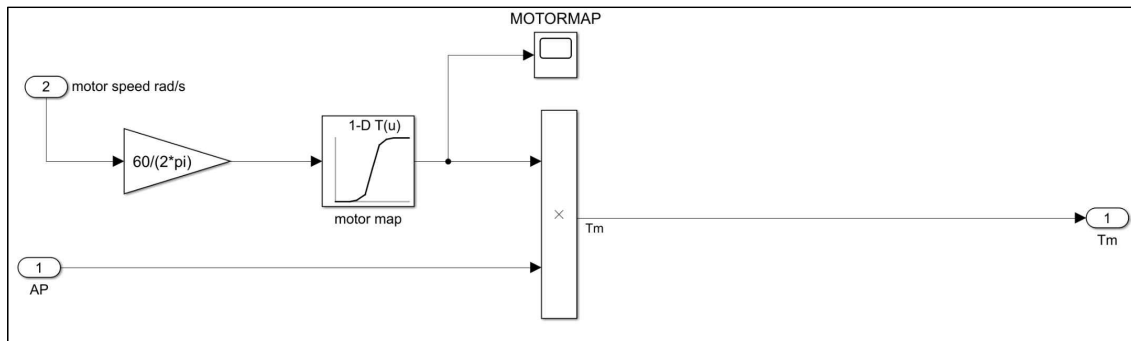
차량 동역학 모델은 위 그림의 단순화된 차량 모델에서 유도가 됩니다. 위 모델은 1개의 State Variable (\dot{V})이 있기 때문에 하나의 Equation 으로 정리가 됩니다.

2) 차량 동역학 함수

$$\dot{V} = \frac{\frac{1}{R_t}(N_f N_i T_m) - F_{Load} - F_{Break}}{m + \frac{1}{R_t^2}(J_m N_i^2 N_f^2 + J_t N_f^2 + 2J_w)}$$

$$F_{Load} = \frac{1}{2} \rho C_d A V^2 + f m g \cos \theta + m g \sin \theta$$

모터 모델



1) 모터 특성곡선 함수

차량 모터 용량 선정에서 구한 모터의 최대 RPM, Base RPM 그리고 최대 토크를 이용해서 아래의 함수를 만들어 보았습니다.

$$F_n = @(x) 259.1 \cdot (x < 3750.6) + ((101700 \cdot 60) / (2 \cdot \pi \cdot x)) \cdot (15446 > x) \cdot (x \geq 3750.6) + 0 \cdot (20000 > x) \cdot (x \geq 15446)$$

여기서 최대 RPM, Base RPM, 최대토크, 최대전력 다음과 같은 과정을 통해 구합니다.

i) 최대 RPM(w_{\max})

$$\text{최대속도 } V_{\max} = 200 \text{ km/h} = 55.56 \text{ m/s}$$

$$V_{\max} = R_t w_{\text{wheel}}$$

$$55.56 \text{ m/s} = 0.316 \text{ m} \times w_{\text{wheel}} \quad \therefore w_{\text{wheel}} = 175.82 \text{ rad/s}$$

$$w_{\max} = w_{\text{wheel}} \times \text{감속비} = 175.82 \times 2.3 \times 4 = 1617.544 \text{ rad/s}$$

$$\therefore w_{\max} = 15446 \text{ rpm}$$

ii) 최대토크(T_{\max})와 최대전력(P_{\max})

제로백 시간을 10초로 만족시키는 전력, 토크와 33%의 경사를 0.1g로 주행시킬 수 있는 전력과 토크 중에 큰 값을 선택합니다.

ii-1) 제로백 10초

등가속도이므로 가속도 $a = 2.78 \text{ m/s}^2$, $\theta = 0$,

$$V_{\text{avg}} = \frac{2}{3} \times 100 \text{ km/h} = 18.52 \text{ m/s}$$

$$\begin{aligned} F_t &= m^* \dot{V} + F_{\text{Load}} = m^* \dot{V} + F_{\text{roll}} + F_{\text{drag}} + F_{\text{grade}} \\ &= m^* \dot{V} + \frac{1}{2} \rho C_d A V^2 + f m g \cos \theta + m g \sin \theta \\ &= 1800 \times 2.78 + \frac{1}{2} \times 1.166 \times 0.35 \times 2 \times 27.78^2 \\ &\quad + 0.01 \times 1800 \times 9.81 \times \cos 0 + 1800 \times 9.81 \times \sin 0 \\ &= 5490.8 \text{ N} \end{aligned}$$

$$T = F_t \times R_t \div \text{기어비} = 188.6 \text{ Nm}$$

$$P = F_t \times V_{\text{avg}} = 101.7 \text{ kW}$$

ii-2) 33% 등판성능

가속도 $a = 0.1g$, $\theta = \tan^{-1} 0.33$, $V_{\text{avg}} = 45 \text{ km/h} = 12.5 \text{ m/s}$

$$\begin{aligned} F_t &= m^* \dot{V} + F_{\text{Load}} = m^* \dot{V} + F_{\text{roll}} + F_{\text{drag}} + F_{\text{grade}} \\ &= m^* \dot{V} + \frac{1}{2} \rho C_d A V^2 + f m g \cos \theta + m g \sin \theta \\ &= 1800 \times 0.981 + \frac{1}{2} \times 1.166 \times 0.35 \times 2 \times 13.6^2 \\ &\quad + 0.01 \times 1800 \times 9.81 \times \cos \theta + 1800 \times 9.81 \times \sin \theta \\ &= 7542.6 \text{ N} \end{aligned}$$

$$T = F_t \times R_t \div \text{기어비} = 259.07 \text{ Nm}$$

$$P = F_t \times V_{avg} = 94.3 \text{ kW}$$

$$\therefore T_{\max} = 259.07 \text{ Nm}, P_{\max} = 101.7 \text{ kW}$$

iii) Base RPM(w_{\square})

$$w_{\square} = P_{\max} \div T_{\max} = 3750.6 \text{ rpm}$$

2) 모터 특성곡선 Look up table

위에서 구한 함수를 시뮬링크에서 이용하기 위해서 0~15446RPM 까지 15447개의 x좌표값을 대입합니다. 이때 넣어서 나오는 15447개의 좌표점을 Look up table에 넣어주게 되면 아래 그림과 같이 plot이 만들어 집니다.

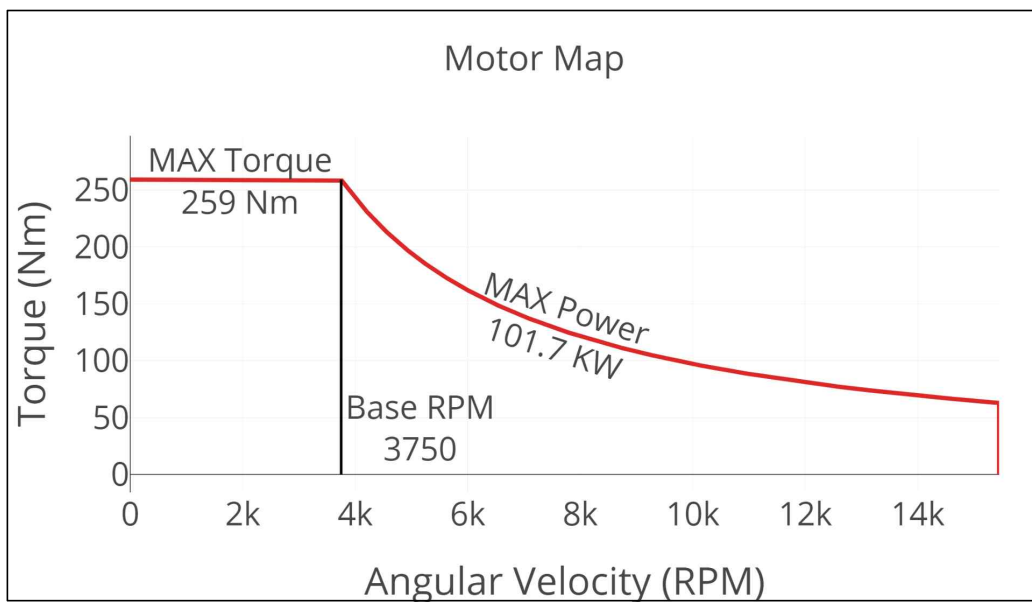
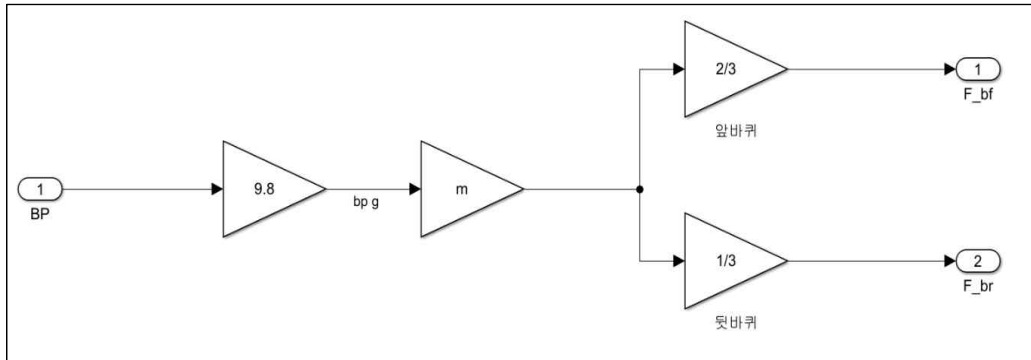


Figure 2. 모터 특성 곡선

브레이크 모델

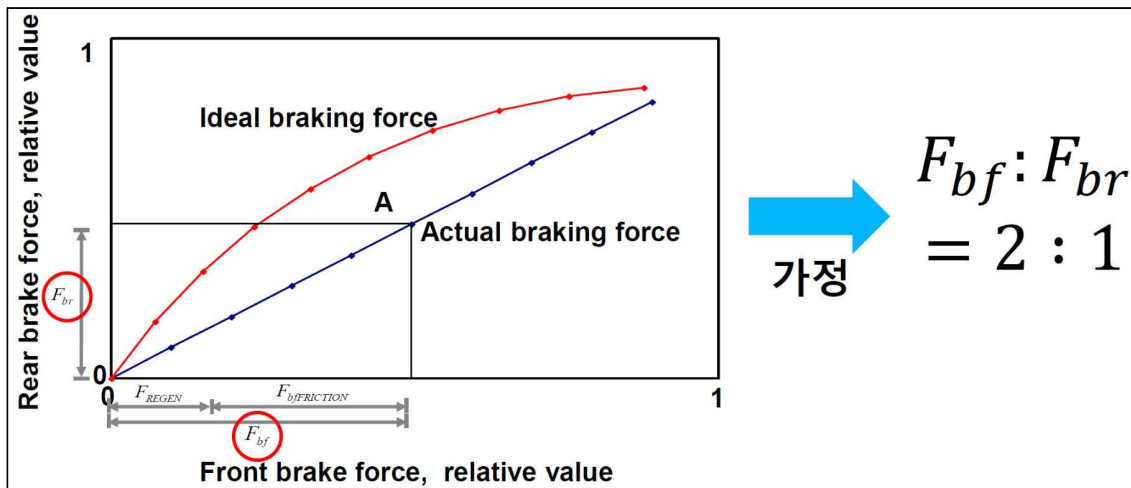


1) 제동력

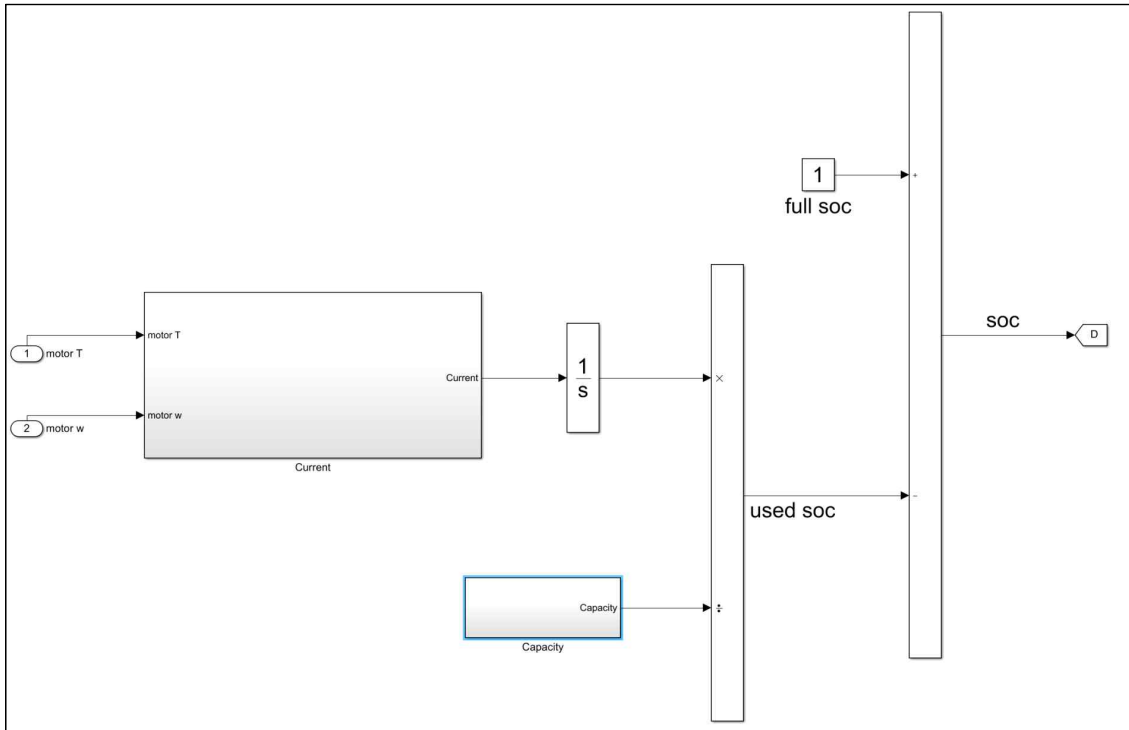
운전자 모델에서 들어온 BP값에 차량의 무게를 곱해준 값을 전체 제동력 값으로 설정 하였습니다. 이는 실제 차량이 설계될 때 최대 제동력이 1G 값을 기준으로 설계되는 것을 반영한 것입니다.

2) 제동력 분배

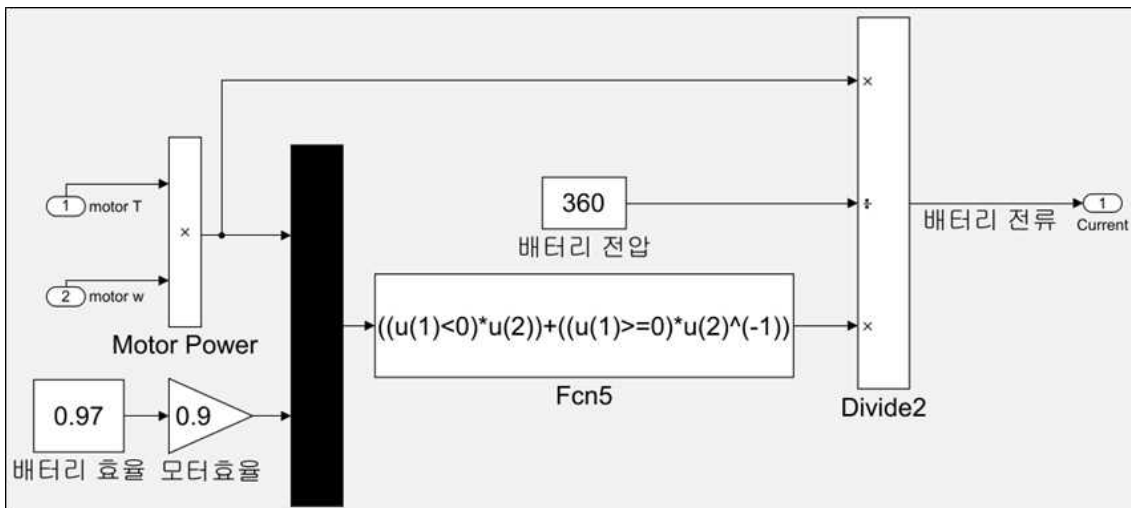
전체 제동력이 앞바퀴에 2/3, 뒷바퀴에 1/3 작용한다고 가정을 하고 분배 해 주었습니다. 결국 차량 동역학 식으로 들어갈 때 합쳐지게 되지만 실제와 유사한 모델을 만들기 위해서 값이 나뉘는 것으로 표현하였습니다.



배터리 모델



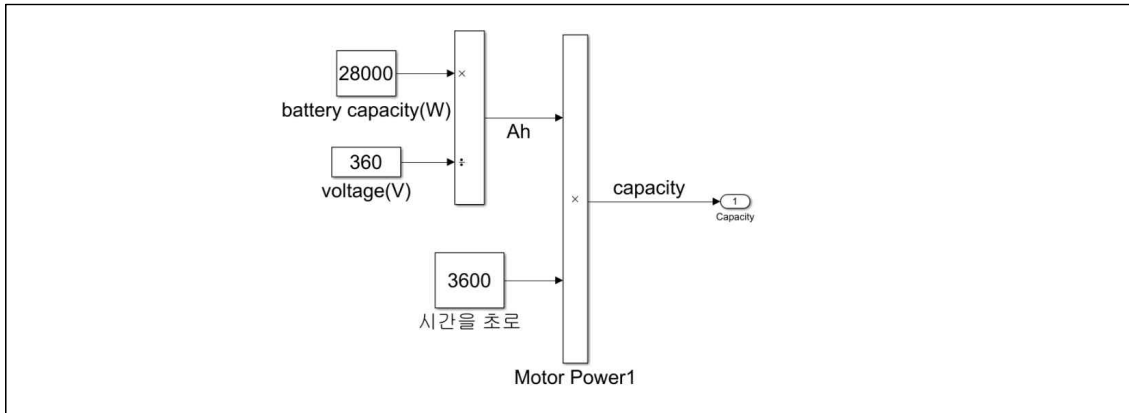
1) Current



모터의 각속도 값과 토크값을 곱해주면 모터의 Power값이 구해지게 됩니다. 이번 과제에서는 회생제동을 하지 않기 때문에 단순히 배터리 효율과 전압을 Power에서 나누는 것만으로 전류를 구할 수 있습니다.

$$i_{battery} = \frac{T_{motor} \times \omega_{motor}}{V_{battery} \times \eta_{battery}} \text{ (discharging)}$$

2) Capacity



28kWh의 용량을 가진 배터리를 각 요소들로 분해하여 나타내어 보았습니다.

$$Q_{battery} = 3600Ah = 3600 \frac{Wh}{V}$$

3) SOC

처음 SOC값을 1이라고 했을 때 아래 식을 이용해서 남은 SOC값을 구할 수 있습니다.

$$SOC = SOC_{initial} - \frac{1}{Q_{battery}} \int i_{battery} dt$$

결과값 확인

From/Go to블록을 이용해서 결과를 보여주는 부분을 따로 분리하였습니다. 이때 각각 Scope는 다음과 같습니다.

1) Single Input Source Scope

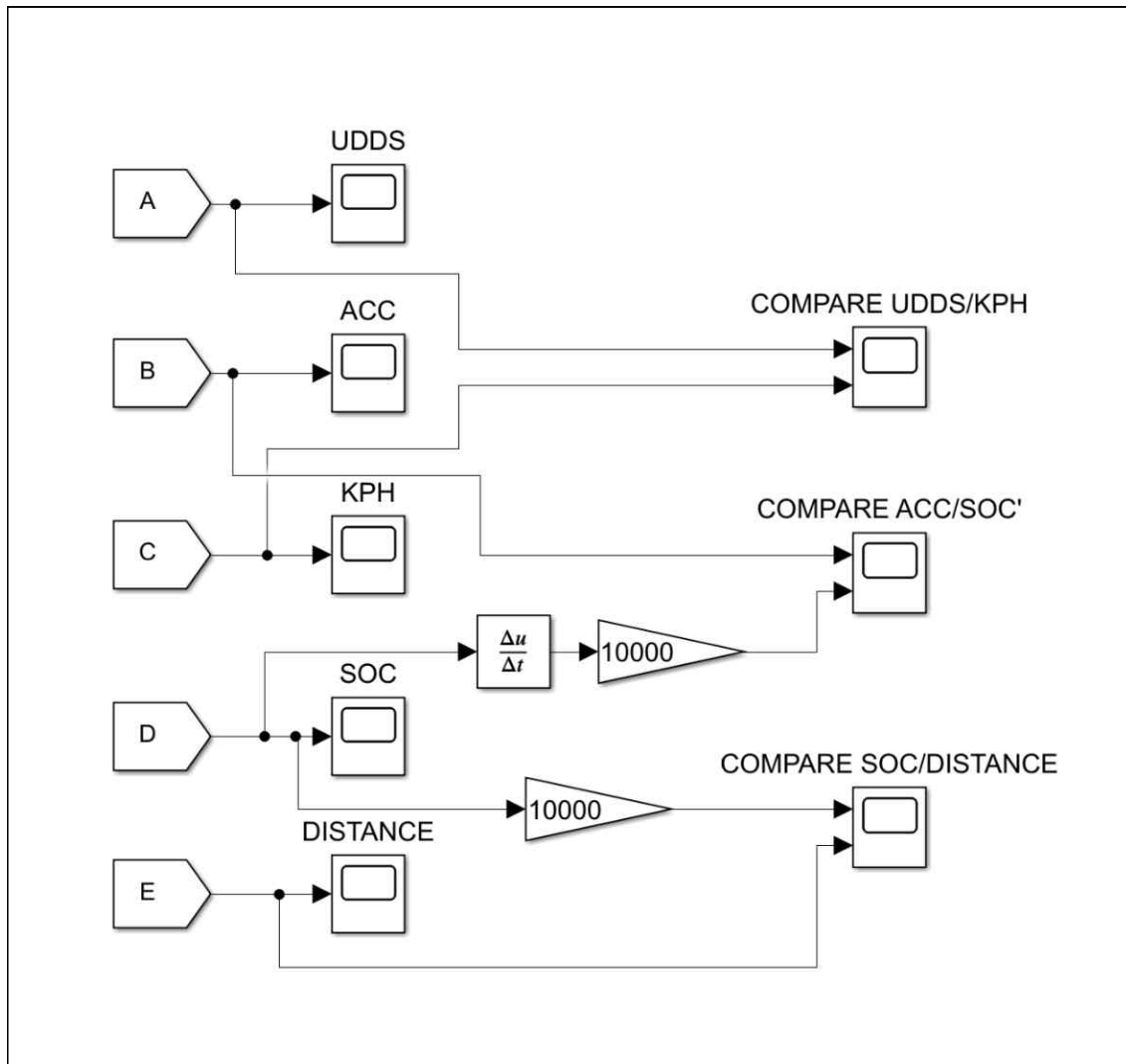
UDDS : 원하는 속도 프로파일

ACC : 차량의 가속도

KPH : 차량의 실제 속도

SOC : 잔여 SOC양

DISTANCE : 차량의 주행거리



2) Double Input Scope

Compare UDDS with 실제속도 : 원하는 속도 프로파일과 실제 속도 프로파일을 한눈에 비교 가능합니다

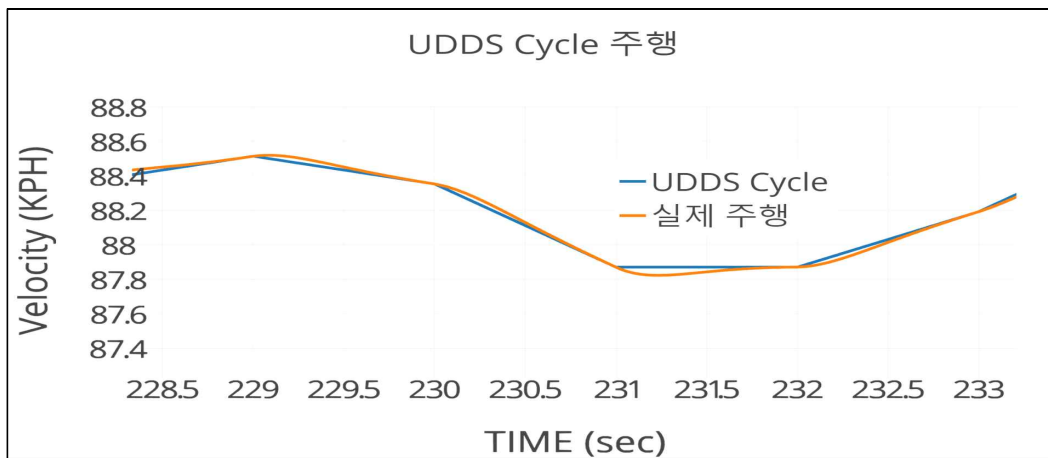


Figure 3. UDDS Cycle 주행시 속도-시간 그래프

Compare ACC with SOC' : 차량의 가속도와 배터리 소모율을 한눈에 비교해서 보여줍니다.

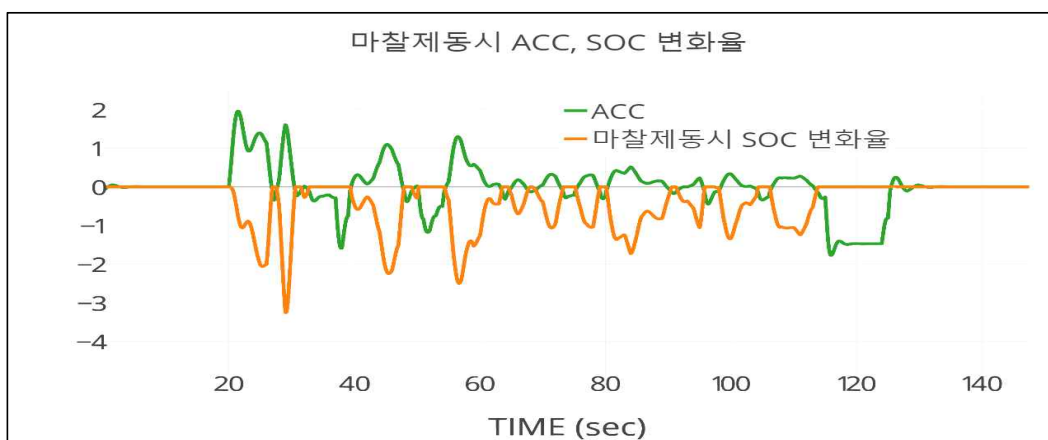


Figure 4. ACC, SOC-시간 그래프

3. 시뮬레이터 검증

가속성능 비교

1) Top speed & 0 to 100 by simulink

저희가 목표로한 최고속도는 200kph, 제로백은 10초로 설계한 후 이것을 Simulink를 통한 시뮬레이터로 가속성능을 시험해 보았습니다. AP=1, BP=0으로 설정하여 최대가속 상황을 설정하고 시뮬레이터를 실행시켰을 때 결과는 다음 그래프와 같습니다.

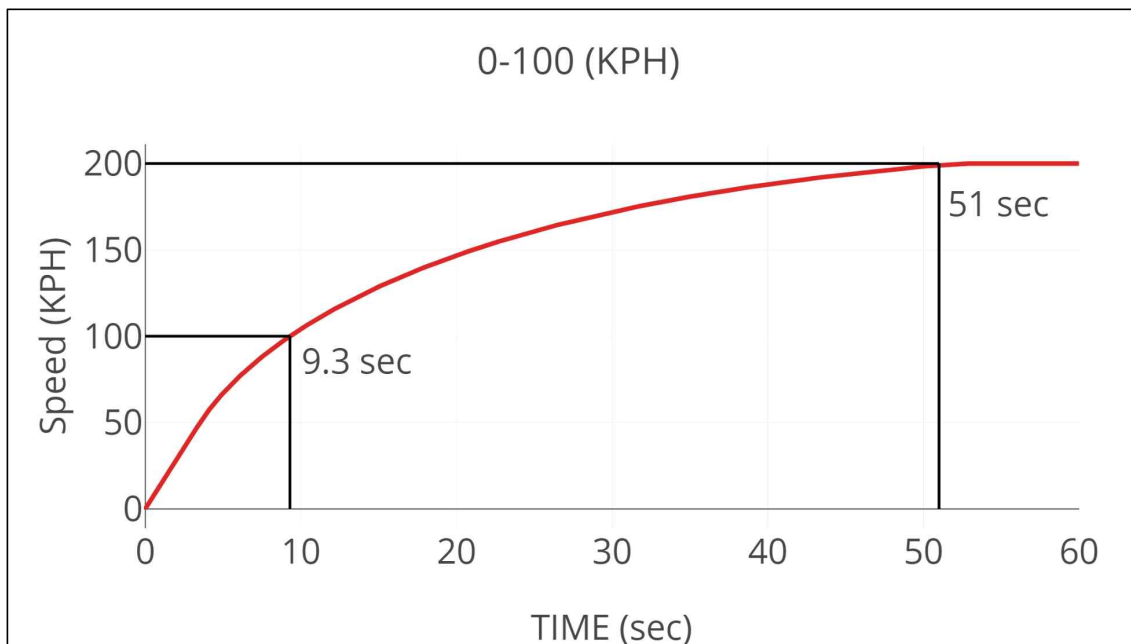


Figure 5. 제로백 속도-시간 곡선

시뮬레이터 결과, 최고속력은 51초에서 200kph, 제로백은 9.3s로 나타났습니다.

	설계	시뮬레이터결과
중량	1800 kg	1800 kg
제로백	10.0 s	9.3 s
최고속도	200 kph	200 kph

2) Discussion

제로백 성능과 최고속도를 비교해 본 결과 최고속도는 목표로 한 값과 차이가 없었지만 제로백 시간은 0.7s 만큼 빠른 결과를 얻을 수 있었습니다. 이러한 오차가 발생한 이유를 고찰해 본 결과는 다음과 같았습니다.

- 실제와 달리 시뮬레이터가 모터의 성능을 90%의 매우 높은 수치로 가정
- 동력이 전달될 때 기어에서 발생하는 손실 그 밖의 동력이 전달되는 과정에서 발생하는 마찰과 열로 인한 손실을 무시
- 0 to 100 test에서 가속 시 바퀴가 헛돌지 않는다고 가정하여 마찰계수를 일정하게 유지
- 모터특선곡선에서 제로백을 10초로 설정한 후 필요로 하는 power를 구할 때 이용한 식 중 $Ft \cdot V$ 에서 적분식을 이용하지 않고 V 를 평균속도인 $100\text{kph} \cdot 2/3$ 로 가정하였으므로 오차가 생겼을 것입니다.
- 모터특선곡선에서 제로백 속도를 만족시키기 위한 최대 파워를 설정하는 구간에서 관성을 고려하지 않았지만 실제에서는 그러한 값들이 영향을 끼쳤을 것입니다.

등판성능 고찰

저희가 앞서 설계한 모터제원이 등판성능을 만족하는지 테스트를 해보았습니다. 33%인 경사를 50km/h로 주행하는 것이 저희가 설정한 등판성능이었고 결과는 다음과 같았습니다.

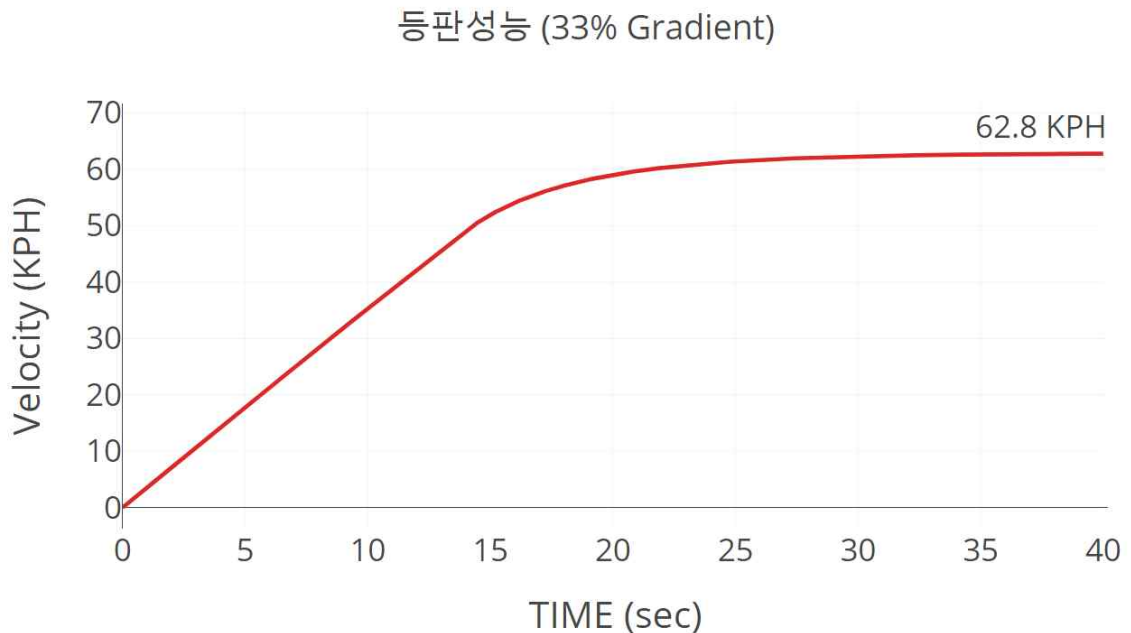


Figure 6. 등판성능 속도-시간 그래프

시뮬레이터 결과 33%의 경사를 최대 62.8km/h로 주행할 수 있었습니다. 저희가 목표로 한 성능보다 높게나온 이유는 등판성능을 만족시키는 파워보다 제로백 성능을 만족시키는 파워가 크기 때문입니다.

임의의 속도 프로파일을 추종하는 운전자 모델

1369초동안 시내주행과 고속도로에서의 주행속도 프로파일을 추종하는 운전자 모델을 선정하였습니다. 속도 프로파일 안에는 도로주행과 고속도로상의 속도를 반영하기 위하여 넓은 속도구간을 가지게 설정하였으며, 실제 주행상황과 유사한 분포를 보이도록 속도의 변화량이 연속적이도록 설정하였습니다.

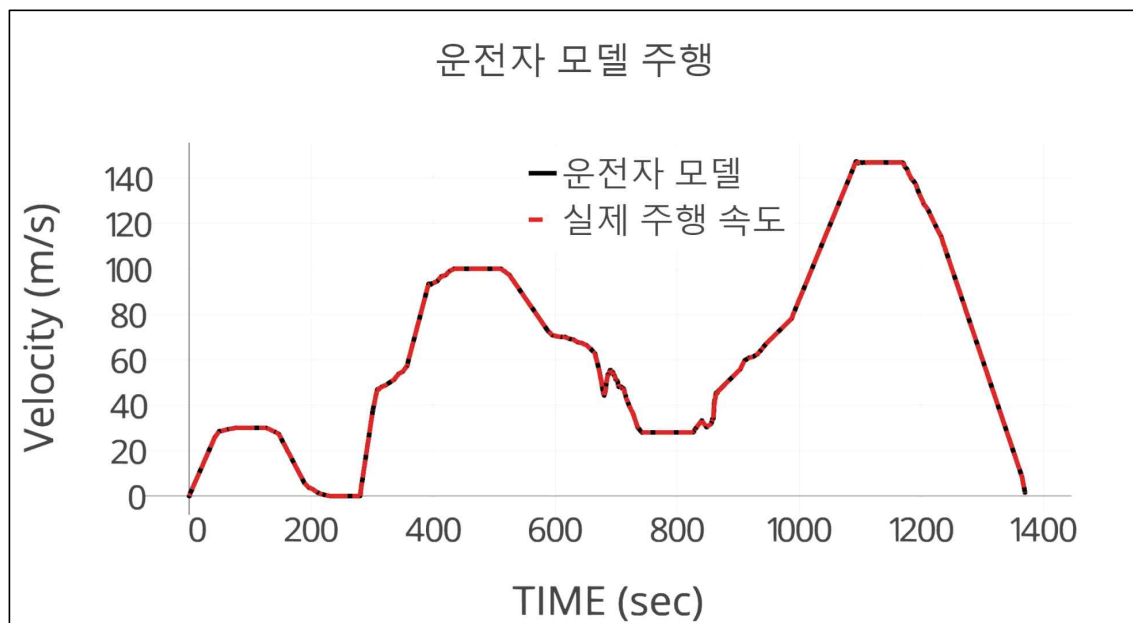


Figure 7. 운전자 모델 주행과 실제 주행 속도-시간 그래프 비교

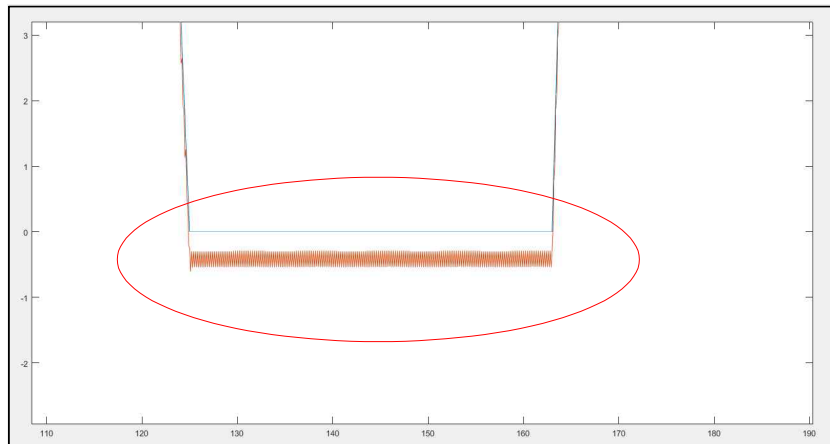
시뮬레이션 결과 급격한 변화가 있는 구간에서 아주 미세한 차이(0.1% 이하)를 제외하고는 거의 완벽한 일치율의 결과 값을 보였는데, 미세한 차이가 발생하는 이유는 급격한 변화가 있는 구간에서는 변화량이 연속적이지 않은 분포를 보이므로 관성에 의해 완벽히 따라갈 수 없는 한계를 가지기 때문에 PI제어기를 통해서 조정하였다 하더라도 약간의 진동이 발생할 수 밖에 없었습니다.

UDDS cycle 주행 시 차속 고찰

1) PI제어기의 상수 값 결정

먼저 실제차속과 요구차속이 일치하도록 PI제어기의 상수를 정했습니다.

K_p 가 5보다 클 때 속도가 0인 부근에서 값이 진동하기 때문에 K_p 는 5 이하여야 합니다.

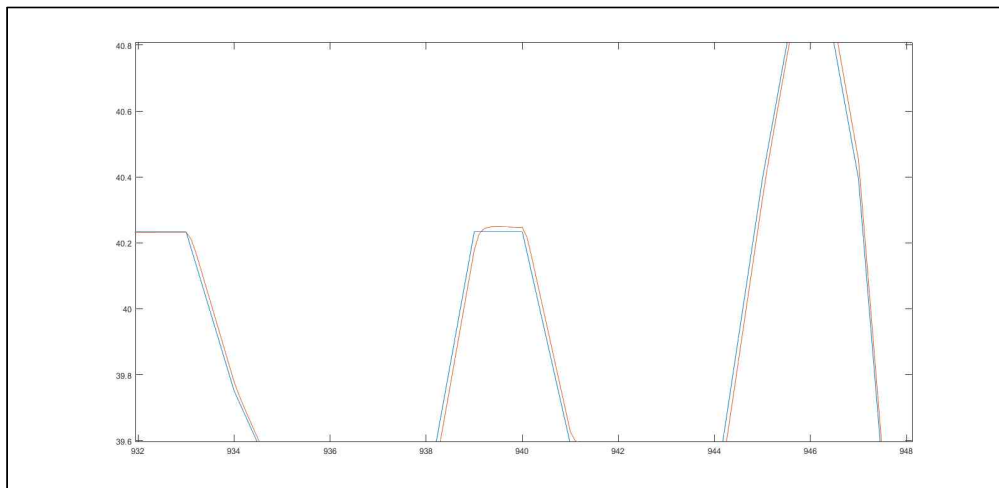


[K_p 가 5일 때 (주황색 : 실제차속, 파란색 : 요구차속)]

따라서 5이하인 K_p 중에서 K_p 값들을 대입하면, $K_p=3$ 일 때 전반적인 요구차속과 실제차속이 가장 일치하게 되는 것을 발견하였습니다.

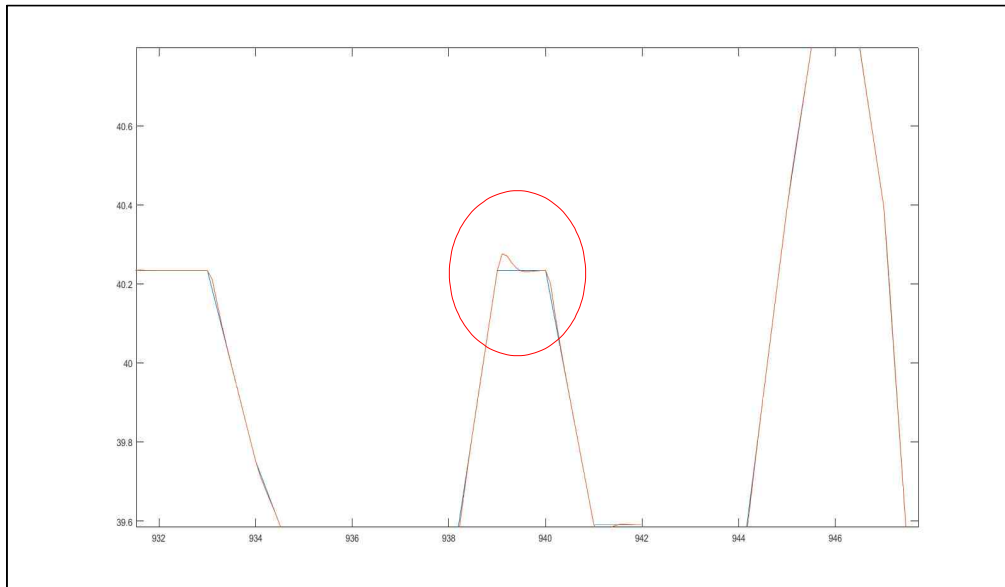
다음으로 K_i 에 대해서 여러 값을 대입하면서 응답을 관찰하였습니다.

K_i 가 1 이하일 때 다음 그림과 같이 응답 지연이 발생하였습니다.



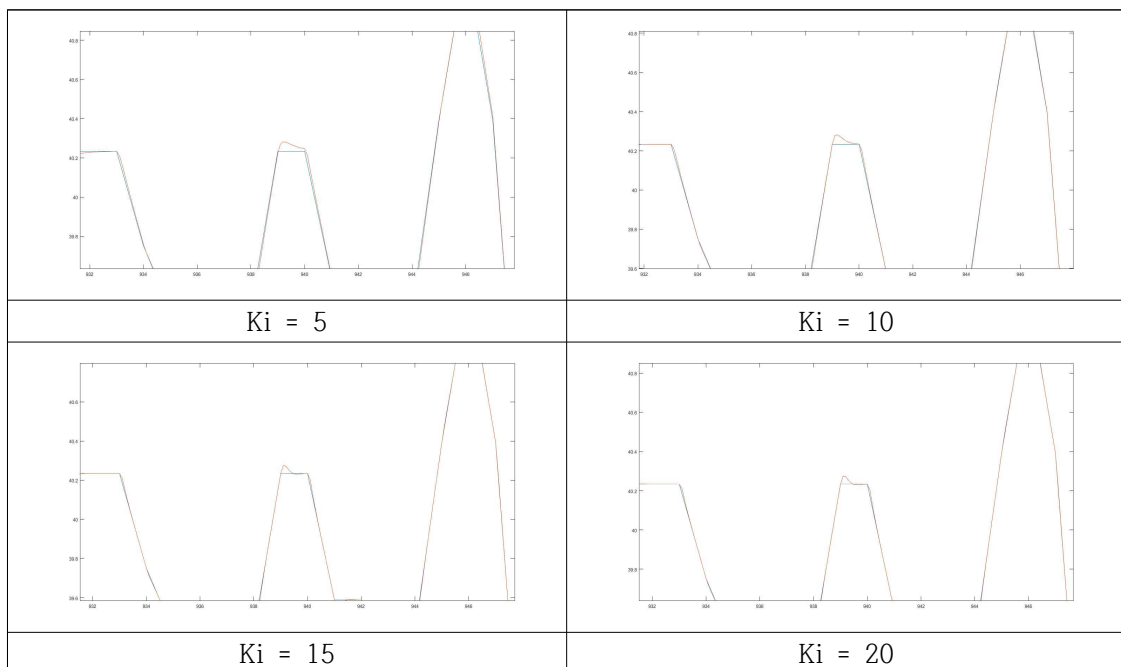
[K_p 가 3, K_i 가 1일 때 (주황색 : 실제차속, 파란색 : 요구차속)]

Ki를 1 이상으로 설정해야 응답지연이 발생하지 않았습니다. 또한, Ki가 너무 커지면 (15이상) 응답의 settling time이 커져서 Output이 요동치는 부분 (939s~940s)이 발생하였기 때문입니다. 따라서 Ki는 그 사이의 값으로 설정해 주었습니다.



[Kp가 3, Ki가 15일 때 (주황색 : 실제차속, 파란색 : 요구차속)]

또한 Ki가 커질수록 실제차속이 요구차속 그래프에 더욱 가까워졌습니다. Ki=10일 때 가장 실제차속과 요구차속이 일치하는 것을 발견하여 PI제어기의 상수 값은 Kp=3, Ki=10으로 결정하였습니다.



2) UDDS 사이클 주행 시 실제차속과 요구차속 비교

앞에서 설정한 상수 값을 대입한 PI제어기를 이용한 차량 모델의 실제차속과 요구차속의 그래프를 비교하면 다음과 같습니다. 실제 주행 모델과 UDDS모델의 그래프가 대부분 일치하는 것을 알 수 있습니다.

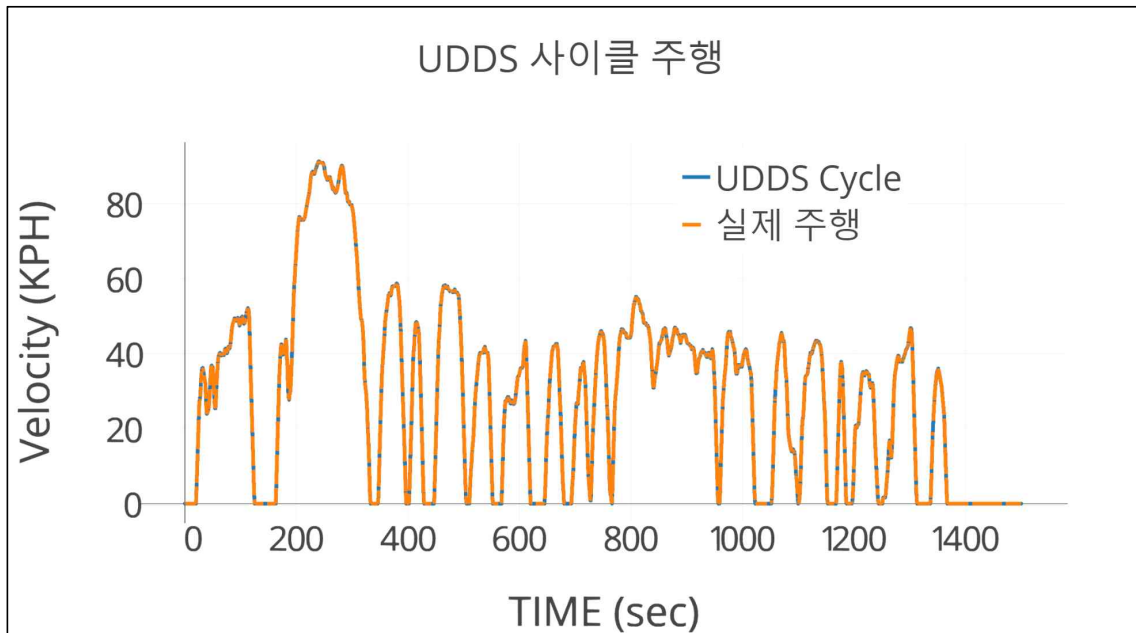


Figure 8. UDDS Cycle 주행 시 속도-시간 그래프

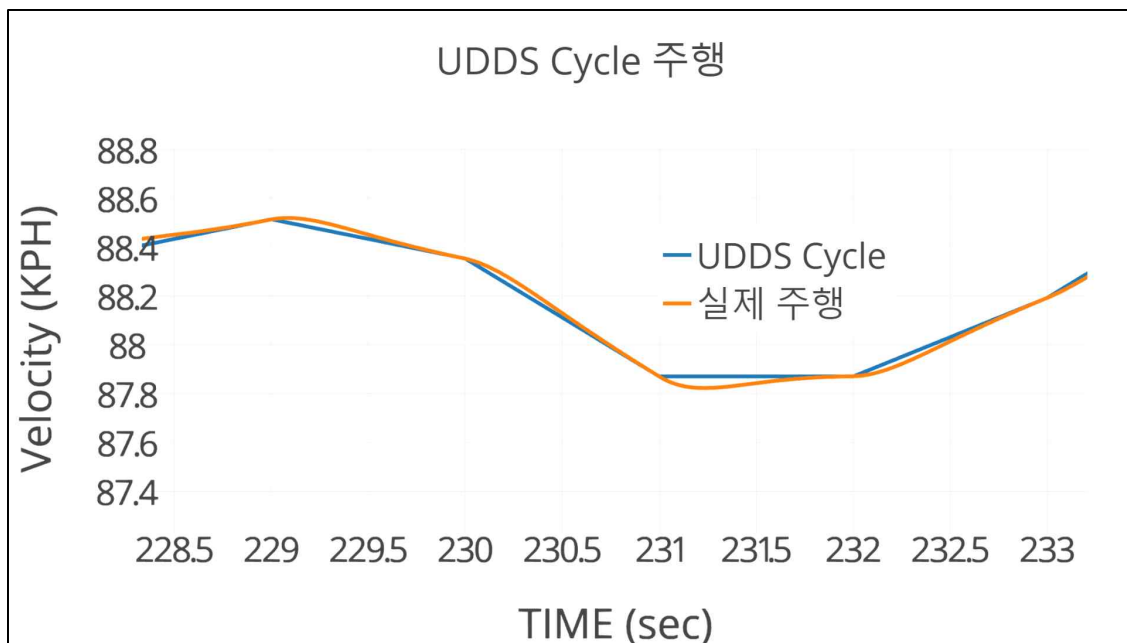


Figure 9. UDDS Cycle과 실제 주행 속도 비교

3) Discussion

두 그래프를 비교하면 거의 대부분의 데이터가 일치하는 것을 확인할 수 있습니다. 작게나마 발생하는 오차의 원인에 대해 분석한다면 PI 제어기의 한계를 이유로 들 수 있습니다.

PI제어기를 사용해서 요구차속과 실제차속을 최대한 일치하게 만들더라도 입력 값이 갑자기 변하는 부분에 대해서는 대처하기는 어렵습니다. 예를 들어, 위 그래프를 보면 속도가 0인 지점으로 급격히 감소할 때 (-) 방향으로 약간의 오차(0.1m/s)가 발생하는 것을 볼 수 있습니다.

UDDS Cycle 주행 시 SOC변화

다음 그래프는 설계한 자동차를 Simulink를 이용하여 UDDS 주행을 완료했을 때 배터리 SOC의 변화를 나타낸 것입니다.

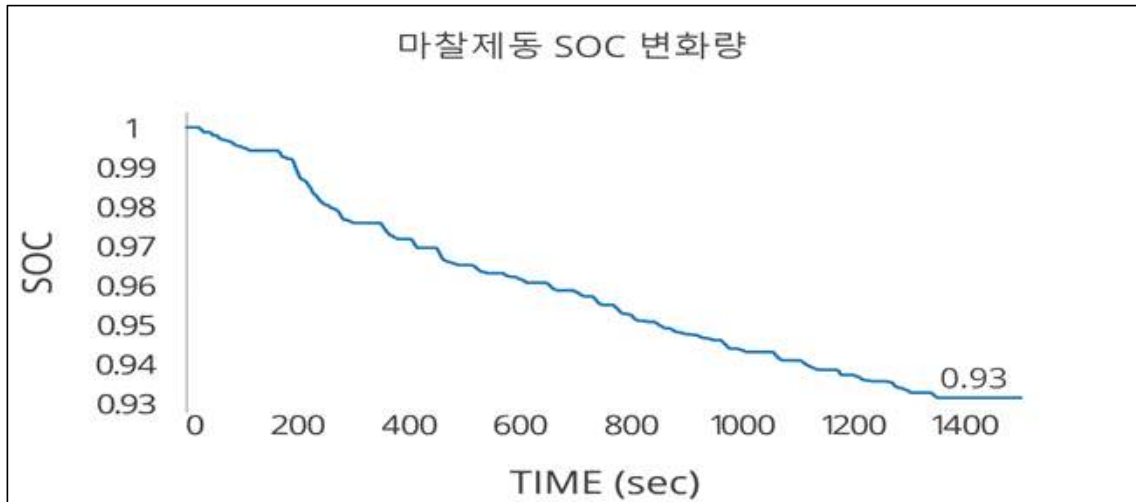


Figure 10. 마찰제동 시 SOC 변화량-시간 그래프(1)

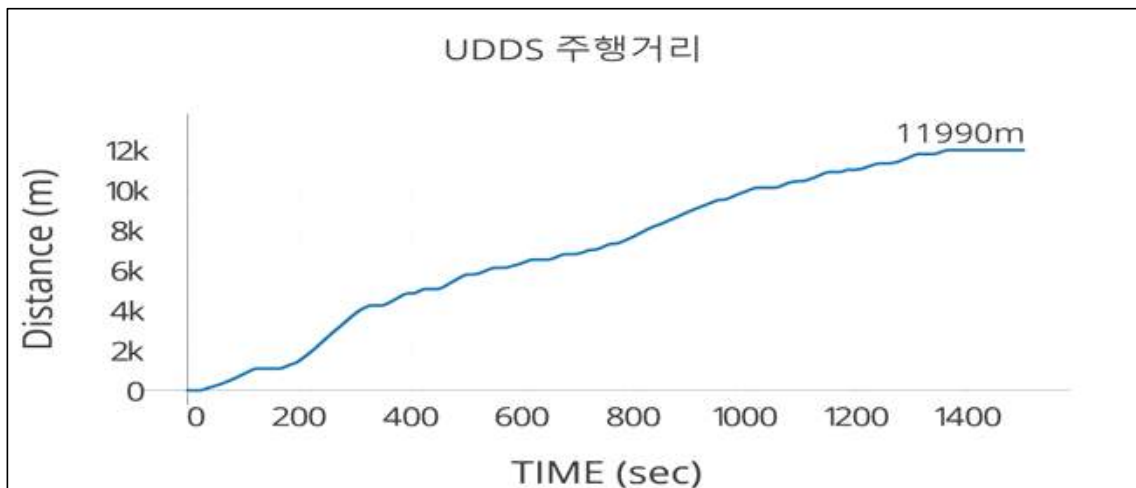


Figure 11. UDDS 주행거리-시간 그래프

회생제동을 실시하지 않고 마찰제동만 수행했을 때, 배터리 SOC는 당연하게도 우하향으로 감소하는 그래프의 모양을 가지게 되며 Figure 10에선 주행시간이 1369초 일 때, y축 SOC의 값은 93%로 변하는 것을 알 수 있습니다.

그런데 그래프를 자세히 분석해 보면 시작부터 약 20초까지 SOC의 값은 1로 일정(이는 배터리가 사용되지 않았음을 뜻함)하며 그 이후에도 간간히

그래프의 기울기가 0인 지점이 나타나며, 그래프에서 매초마다 그래프의 순간 감소되는 기울기가 다른 것을 알 수 있습니다.

SOC-시간 그래프에서 평균변화율은 일정한 경향을 가지고 있으나 파란색 부분(기울기가 0인 지점이 다소 긴 지점)을 지난 부분인 약 170초부터 약 280초까지의 평균변화율이 다른 지점보다 가파릅니다.

이러한 결과는 아래의 거리-시간 그래프의 경향과 일치하였으며 이는 Simulink를 돌리기 전에 움직이는 거리와 순간 속도의 크기에 따라 배터리 SOC의 값이 감소하는 정도가 달라질 것이라 예상한 우리의 가정과 일치하는 결과입니다.

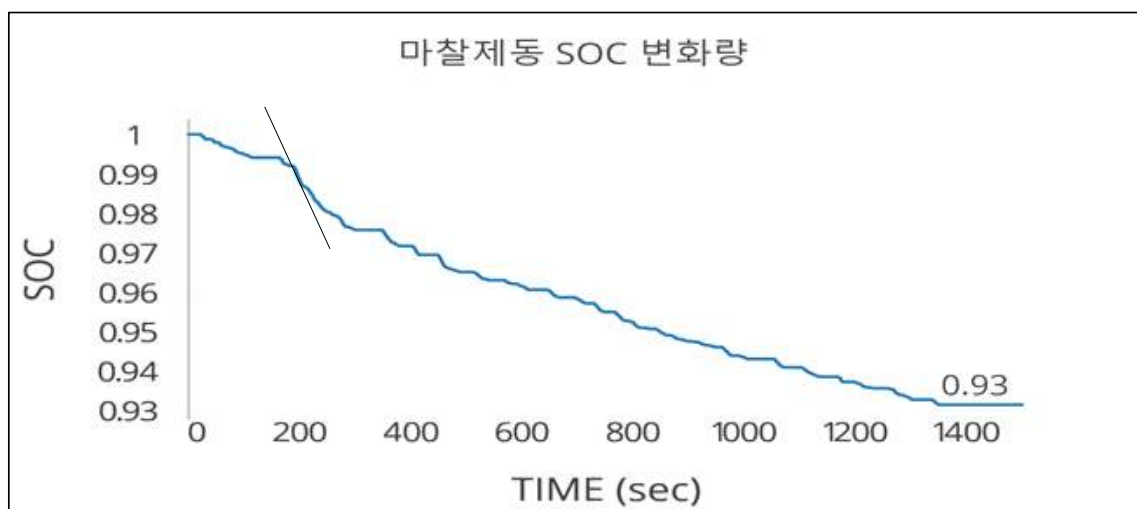


Figure 12. 마찰제동 시 SOC변화량-시간 그래프(2)

또한 Figure 12에서 순간변화율이 가장 클 때는 약 190초~200초 부근인데 이를 ACC/SOC 그래프와 관련지어 보았을 때 SOC 변화 값이 최소 값을 갖는 지점과 일치하는 것을 알 수 있었습니다.

UDDS Cycle 주행 시 전비 전비 고찰

1) 일반적인 전기자동차 전비 측정 방법

전기자동차의 에너지 소비효율을 측정하는 방법은 크게 두 가지로 나뉘집니다. 1회 충전시 최대가능 주행거리를 측정한 후 완전 방전된 자동차를 충전기로 완충될 때 충전된 전력량을 측정하여 비교하는 AC 에너지소비효율(식 1), 전기자동차 주행 중 이동거리에 따른 배터리로부터 공급되는 전력을 측정하는 DC 에너지 소비효율(식2)로 나누어집니다.

$$AC \text{ 에너지효율} \left(\frac{km}{ACWh} \right) = \frac{\text{주행거리}(km)}{AC \text{ 에너지충전량}(ACWh)} \quad \dots (\text{식1})$$

$$DC \text{ 에너지효율} \left(\frac{km}{DCWh} \right) = \frac{\text{주행거리}(km)}{\text{주행시 에너지충전량}(ACWh)} \quad \dots (\text{식2})$$

미국을 중심으로 한 자동차기술학회(Society of Automotive Engineers)에서는 전기자동차의 전기에너지 소비 효율에 대해서 규정하고 있으며, 주로 AC 에너지효율을 전기자동차의 전비측정기준으로 다섯가지 실주행여건(5-Cycle, 주행거리 3000km)으로 시가지주행모드(FTP-75, UDDS), 고속도로주행모드(HWFET), 급가속모드(US06), 고온환경에서 에어컨 동작 시 주행모드(SO03), 그리고 저온환경(Cold UDDS)에 따라 각각 측정하여 사용하고 우리나라 또한 이를 따르고 있습니다.

이중에서 전기자동차 제품의 전비를 나타낼 때 사용하는 주행거리는 배터리 방전에 따라 각 주행모드의 속도 허용오차를 4초 이상 벗어날 때까지의 주행거리를 ‘1회충전주행거리’라 하고, 도심주행시험(UDDS)과 고속도로주행시험(HWFET)을 반복측정 후 복합해서 나타내는데, 이를 ‘복합1회 충전주행거리’라고 하며 아래와 같이(식3)로 나타냅니다. 전기자동차의 경우 온도, 배터리 상태, 운전자의 조건 등에 따라 주행거리가 크게 변하므로 보정계수 0.7을 도입합니다.

1회충전주행거리를 통해 AC에너지 효율을 사용하여 연료소비율을 측정하고 이를 복합하면 우리가 사용하는 전기자동차의 전비가 나타나게 됩니다.

복합 1회 충전 주행거리(km)

$$= 0.55 \times \text{도심 1회충전주행거리} + 0.45 \times \text{고속도로 1회충전주행거리} \quad \dots (\text{식3})$$

도심 1회 충전 주행거리(km)

$= 0.7 \times FTP75$ 모드에서 UDDS에 따라 반복주행하며 측정된 1회충전주행거리 ... (식4)

고속도로 1회 충전 주행거리(km)

$= 0.7 \times HWFET$ 모드에서 반복주행하며 측정된 1회충전주행거리 ... (식5)

2) UDDS Cycle 주행방법

현재 우리나라는 1975년 미국 도심지역 차량 흐름을 반영한 FTP-75 모드를 연비측정 방법으로 사용하여 왔습니다.

주행거리 : 17.77 km (11.04 mile)

소요시간 : 1874 초 (31.2 분)

최고속도 : 91.2 km/h (56.7 mph)

평균속도 : 34.2 km/h (21.2 mph)

전기자동차를 위와 같은 상황에서 운전한다고 가정하여 반복 측정된 도심 1회 측정주행거리를 이용하여 연료효율을 구하는 것이 UDDS 반복주행에 따른 연료소비율입니다.

3) MATLAB Simulink 배터리 전비 측정

우리가 제원으로 사용한 차량의 경우 배터리 용량 28kWh입니다. 배터리 효율 97%라고 가정하고, 이를 Simulink를 사용하여 UDDS 주행을 완료 했을 때 시뮬레이션 결과는 다음과 같습니다.

소모된 배터리SOC(%) : 7%

주행거리 : 11990m

주행시간 : 1369sec

배터리 에너지 용량을 사용하여 전비계산을 실시하면 다음과 같습니다.

$$\text{소모된 배터리 } SOC = 28kWh \times 0.07 = 1.96kWh$$

$$\text{Simulink에서의 에너지효율} = \frac{\text{주행거리}(km)}{\text{소모된 배터리}SOC(kWh)} = 6.12km/kWh$$

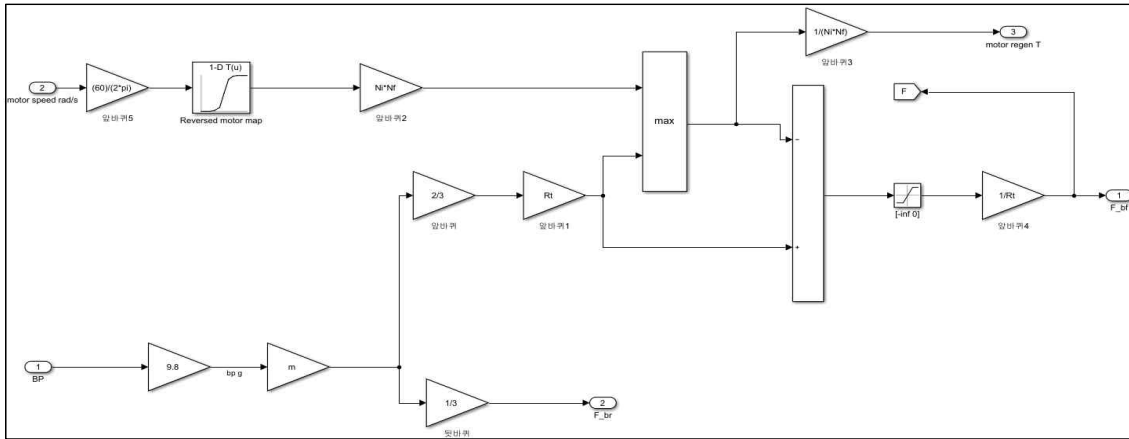
4) 배터리 SOC의 전비 오차의 원인

- 측정 방법 차이

Simulink를 활용한 에너지효율은 UDDS 주행 시 주어진 UDDS과정 전체에 대해서 주행거리, 주행 시 소모된 SOC를 활용하여 구한 값입니다. 실제 UDDS 주행과정을 모델로 하여 주행에 사용된 배터리 SOC와 주행거리 사이의 비율을 나타낸 것이 Simulink 에너지 효율이라면, 설정한 값으로 구한 에너지효율은 전체 배터리 용량과 전체 주행가능거리 사이의 단순 비율이므로 에너지 효율에 차이가 생기게 됩니다.

4. 회생제동

회생제동 모델



전륜에서만 회생제동이 일어난다고 가정하였을 때, 회생제동 모델을 위해 저희가 설계한 simulink는 위와 같습니다. max항을 이용하여 차량 요구 제동력을 회생제동이 만족시키지 못할 때 회생제동 협조제어가 가능하도록 설계하였습니다.

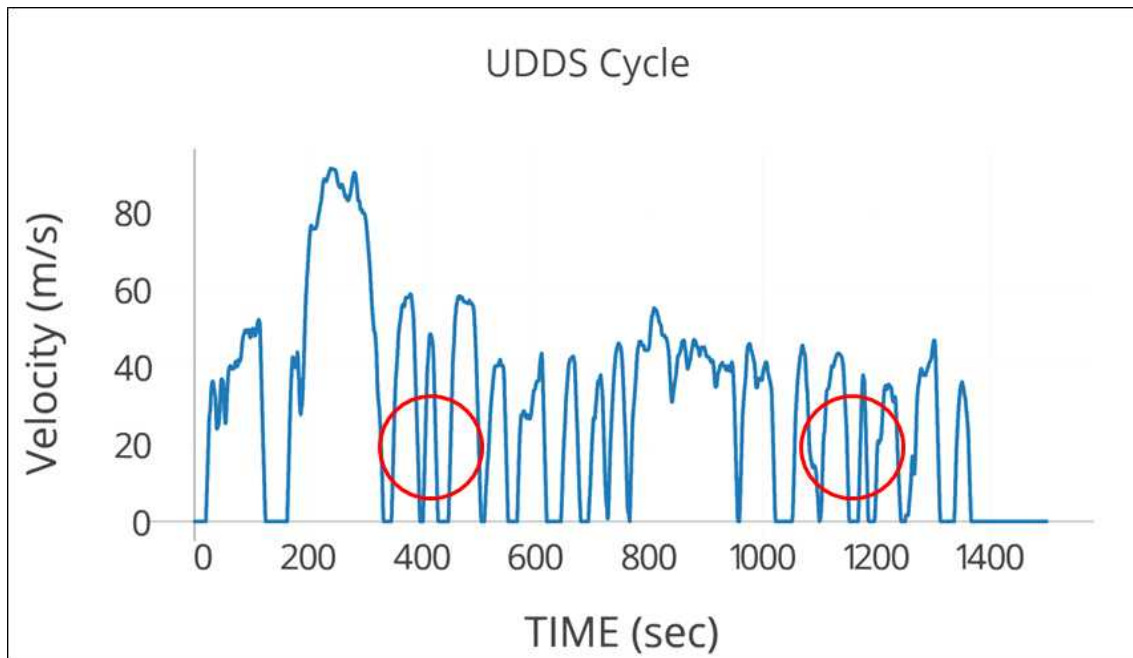


Figure 13. UDDS Cycle 주행 시 속도-시간 그래프

실제로 UDDS cycle을 이용해 시뮬레이터를 사용해본 결과 원으로 표시한 부분인 급 감속구간에서만 협조제어를 통하여 제동하였는데 이는 그

외의 구간에서는 회생제동만으로도 차량의 요구 제동력을 만족시킬 수 있었다는 것을 의미합니다.

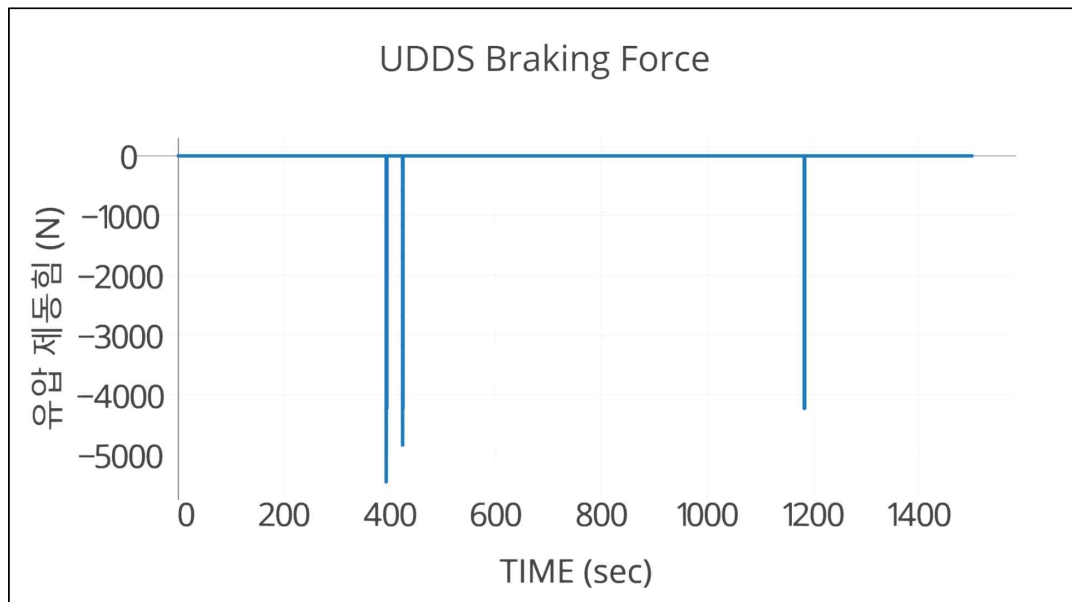


Figure 14. UDDS 제동력-시간 그래프

우수한 결과값이 나온 것으로 볼 수 있는데, 이러한 결과값이 나온 이유로는 모터의 성능 자체가 뛰어나 회생제동으로 얻을 수 있는 제동력이 큰 반면 급감속이 적고 대체로 저속으로 운전하는 UDDS cycle에서는 차량 요구 제동력이 낮기 때문으로 생각할 수 있습니다. 아래의 그래프는 UDDS cycle에 gain을 통해 그 속도 폭을 1.5배 늘려주었을 때 협조제어시 필요한 힘을 나타낸 것인데, 이 경우 회생제동 협조제어가 더 빈번히 나타남을 알 수 있습니다.

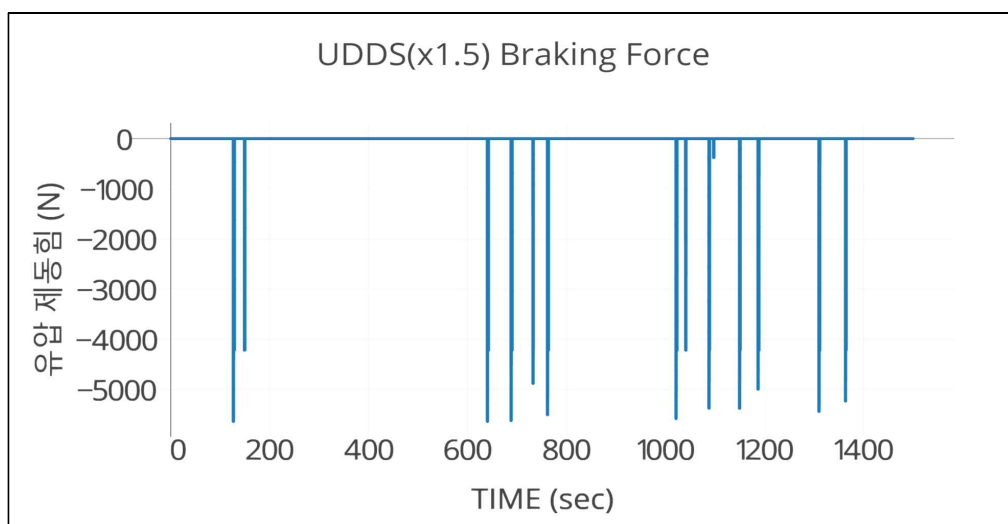


Figure 15. UDDS 제동력(x1.5배)-시간 그래프

5. UDDS사이클 주행시 SOC변화

회생제동, 마찰제동의 배터리 SOC 변화

Figure 16의 회생제동과 마찰제동의 배터리 SOC변화를 살펴보면 마찰제동에서는 배터리SOC가 유지되거나 지속적으로 감소하지만 회생제동에서는 이에 더해서 배터리 SOC가 증가하는 구간이 존재함을 알 수 있습니다.

또한 전체 UDDS사이클 주행 시 회생제동과 마찰제동 두 경우 Figure 17과 같이 모두 동일한 주행거리를 주행하지만 배터리 소모량이 마찰제동은 7% 회생제동은 5.3%로 1.7%의 차이가 남을 알 수 있습니다.

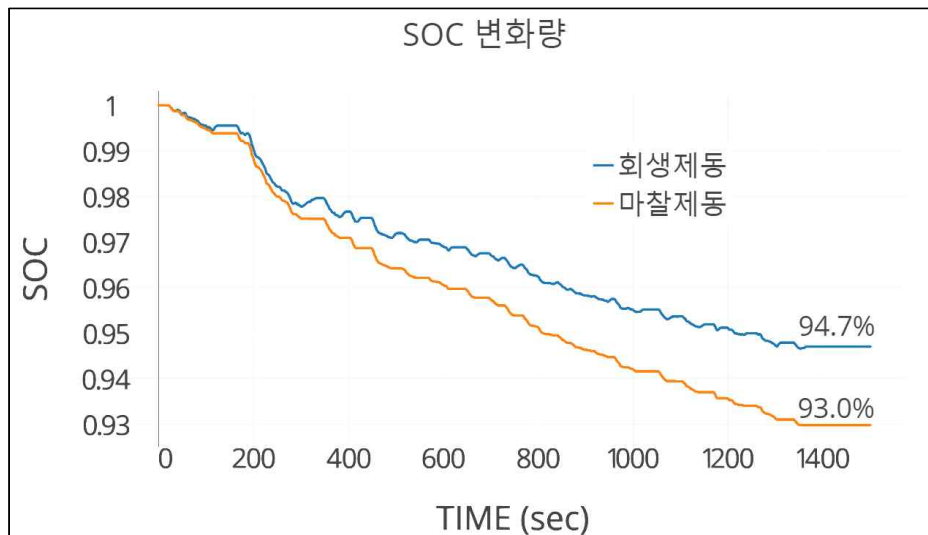


Figure 16. SOC 변화량

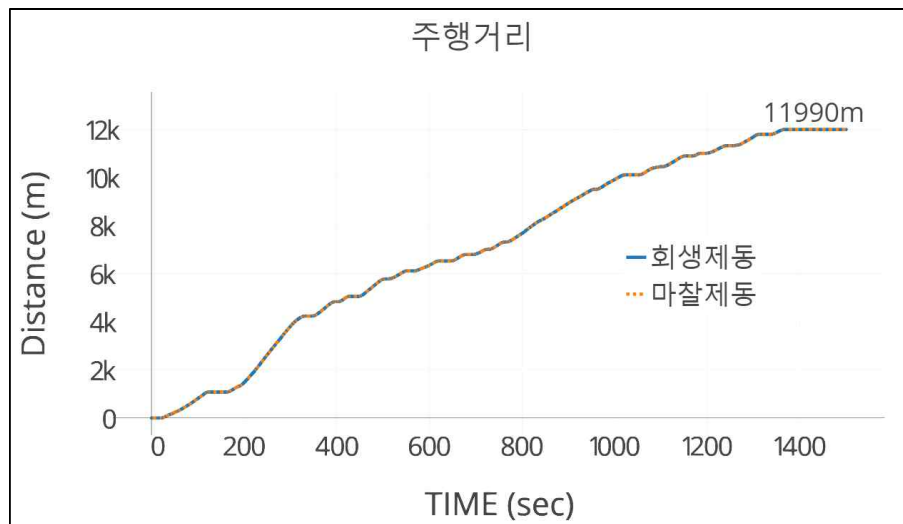


Figure 17. UDDS 사이클 주행거리

Figure 18은 SOC변화량 그래프를 특정 구간만 확대한 그래프입니다. 이처럼 UDDS 사이클의 전체 구간을 세 가지 가속, 감속, 정지 구간으로 나눌 수 있습니다. 가속 구간에서는 회생제동과 마찰제동이 같은 기울기로 SOC가 감소하게 됩니다. 감속 구간에서는 마찰제동은 SOC기울기가 0이고 회생제동은 양의 기울기 값을 가집니다. 정지구간은 마찰제동 회생제동 모두 0의 기울기 값을 가집니다. 이러한 세 구간이 반복되면서 누적되는 차이만큼 회생제동과 마찰제동의 SOC소모량이 달라지는 결과가 도출되었습니다.

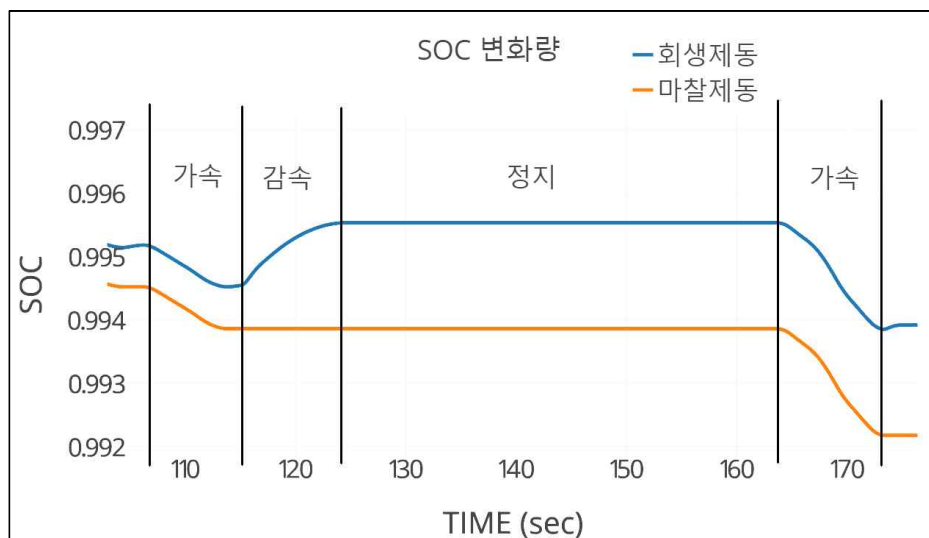


Figure 18. SOC 변화량 확대

회생제동, 마찰제동의 가속도와 배터리 SOC 변화율

Figure 19와 Figure 20은 차량의 가속도와 SOC변화율을 10000배한 값을 비교한 그래프입니다. 앞에서 보았던 가속, 감속, 정지 구간을 그림4와 그림5에서도 확인할 수 있습니다. 가속구간에서 마찰제동과 회생제동의 두 경우 모두 앞서 그림3에서처럼 SOC변화율이 음수 값을 가지게 되는데 그 크기가 대략적으로 가속도의 크기에 비례함을 확인할 수 있습니다. 감속 구간 에서 마찰제동은 변화율이 0의 값을 가지고 회생제동은 변화율이 대략적으로 감속력 즉 제동력에 비례하여 커짐을 확인할 수 있습니다. 정지 구간 에서는 두 경우 모두 변화율이 0임을 알 수 있습니다.

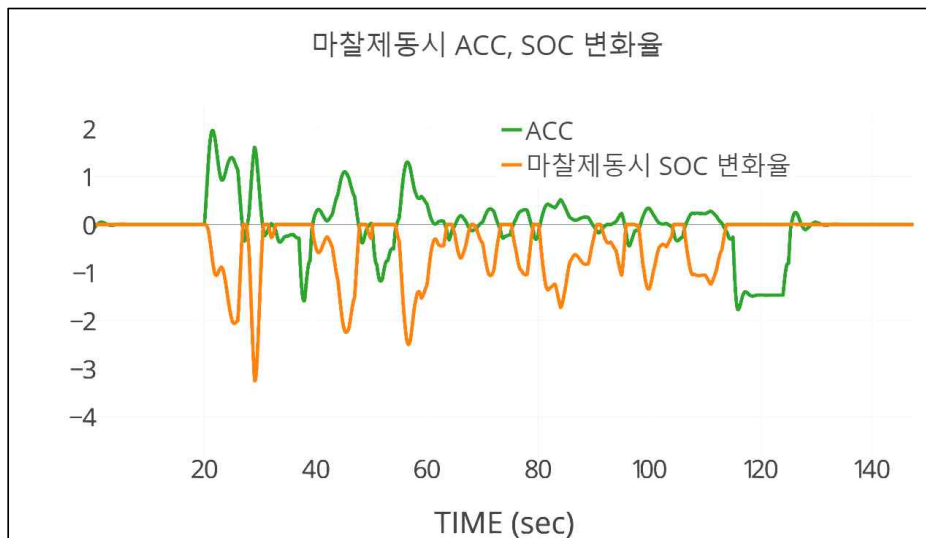


Figure 19. 마찰제동 ACC, SOC변화율

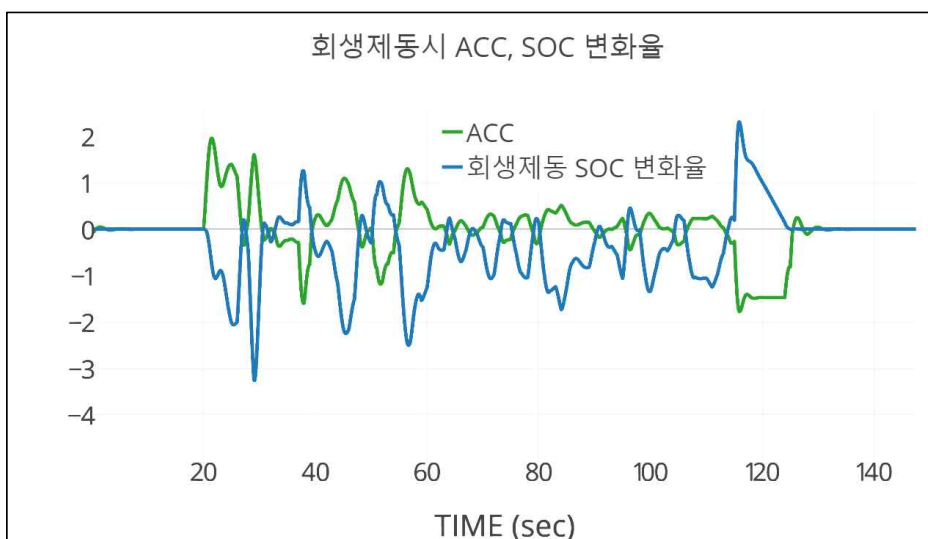


Figure 20. 회생제동 ACC, SOC변화율

6. 회생제동 주행 시 전비변화

UDDS Cycle에서 SOC 변화

주행거리 : 11.991km

주행시간 : 1369sec

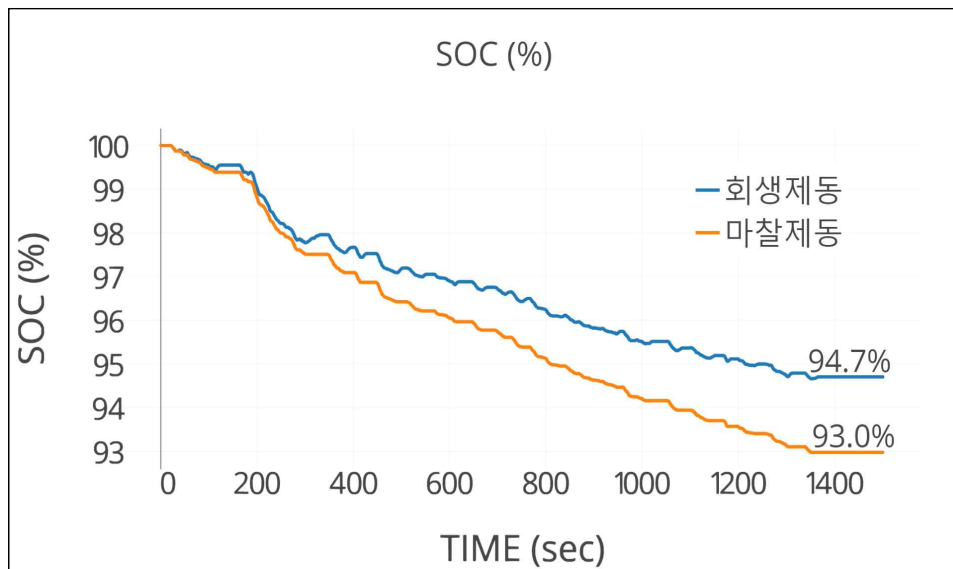


Figure 21. 회생제동과 마찰제동 시 SOC 변화량 비교(%)

마찰 제동 사용시 소모되는 배터리 SOC(%) : 7%

회생 제동 사용시 소모되는 배터리 SOC(%) : 5.3%

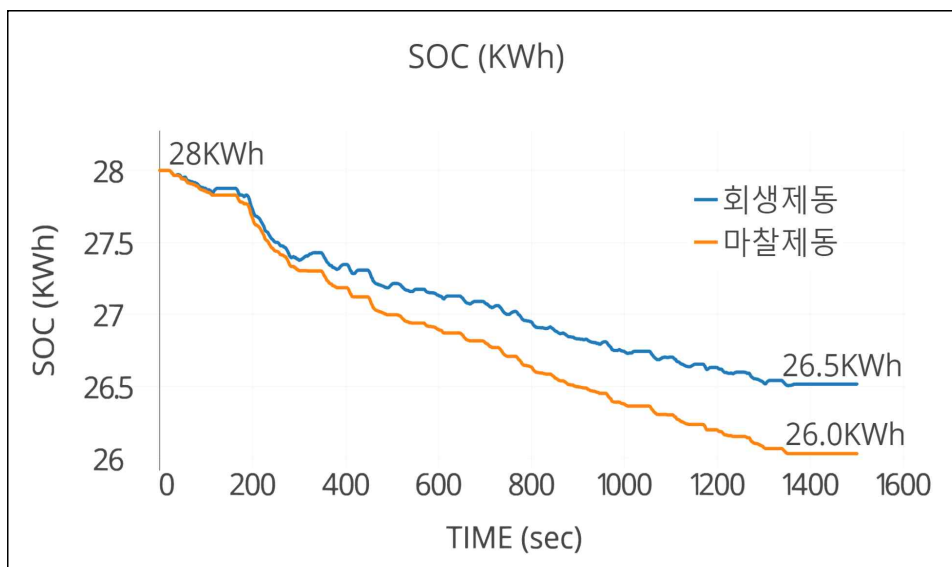


Figure 22. 회생제동과 마찰제동 시 SOC 변화량 비교(kWh)

절약된 배터리 SOC : 0.5kWh

UDDS Cycle 전비계산

배터리 용량 : 28kWh

배터리 효율 : 97%

UDDS Cycle의 전체 거리와 소모된 총 SOC로 전기자동차의 에너지 효율을 구하는 방식을 단순화하였습니다. Matlab Simulink에서 나타난 그래프는 배터리 효율이 모두 고려된 SOC 소모량이므로 다음 식과 같이 단순한 계산만으로 에너지효율을 계산하였습니다.

마찰제동

$$\text{소모된 배터리 SOC} = 28kWh \times 0.07 = 1.96kWh$$

$$\text{Simulink에서의 에너지효율} = \frac{\text{주행거리}(km)}{\text{소모된 배터리SOC}(kWh)} = 6.12km/kWh$$

회생제동

$$\text{소모된 배터리 SOC} = 28kWh \times 0.053 = 1.5kWh$$

$$\text{Simulink에서의 에너지효율} = \frac{\text{주행거리}(km)}{\text{소모된 배터리SOC}(kWh)} = 8.0km/kWh$$

UDDS Cycle 주행동안 회생제동이 작동하지 않고 마찰제동만 작동할 때 총 소모된 SOC는 7%이고, 회생제동이 작동하면 5.3%가 소모되면서 회생제동이 작동하여 1.7%의 배터리 절약효과를 얻을 수 있었습니다.

이는 UDDS Cycle동안 회생제동을 사용하면서 마찰제동에 비해 32% 효율성 향상을 의미합니다.

감속구간에서 회생제동이 배터리를 회복하므로 감속이 많은 도심주행에서 회생제동의 효과가 크게 나타납니다. 반면 고속도로주행과 같이 감속구간이 적은 주행에서는 회생제동과 마찰제동의 배터리 소모율과 전비의 차이가 크지 않을 것입니다.

