|  |
| --- |
|  |
| **석사학위논문** |
|  |
| **LSTM 기반 차량 모델 불확실성 변수 예측을 통한 차량 Torque Vectoring에서의 Sliding Mode Control 적용 가능성 연구** |
|  |
| **A Study on the Application of Sliding Mode Control in Vehicle Torque Vectoring by prediction of LSTM-based Vehicle Model Uncertainty Variables** |
|  |
| **김 진 민** |
|  |
| **한양대학교 대학원** |
|  |
| **2024년 8월** |
|  |

|  |
| --- |
|  |
| **석사학위논문** |
|  |
| **LSTM 기반 차량 모델 불확실성 변수 예측을 통한 차량 Torque Vectoring에서의 Sliding Mode Control 적용 가능성 연구** |
|  |
| **A Study on the Application of Sliding Mode Control in Vehicle Torque Vectoring by prediction of LSTM-based Vehicle Model Uncertainty Variables** |
|  |
| **지도교수 조 한 신** |
|  |
| **이 논문을 공학 석사학위논문으로 제출합니다.** |
|  |
| **2024년 8월** |
|  |
| **한양대학교 대학원** |
|  |
| **미래자동차공학과 (미래자동차-SW 융합전공)** |
|  |
| **김 진 민** |
|  |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | | | | |
| **이 논문을 김 진 민의 석사학위 논문으로 인준함** | | | | |
|  | | | | |
| **2024년 8월** | | | | |
|  | | | | |
|  | **심사위원장:**  **심사위원:**  **심사위원:** | **윤 영 두**  **윤 상 원**  **조 한 신** | **(인)**  **(인)**  **(인)** |  |
|  | | | | |
| **한양대학교 대학원** | | | | |
|  | | | | |

차 례

[차 례 i](#_Toc167702142)

[List of Tables iii](#_Toc167702143)

[List of Figures iv](#_Toc167702144)

[국문 요지 v](#_Toc167702145)

[제1장 서 론 1](#_Toc167702146)

[1.1 연구 배경 및 필요성 1](#_Toc167702147)

[1.2 연구 목표 3](#_Toc167702148)

[제2장 이론적 배경 4](#_Toc167702149)

[2.1 Vehicle Yaw Moment Equation 4](#_Toc167702150)

[2.2 Tire Model 6](#_Toc167702151)

[2.3 Steering Model 8](#_Toc167702152)

[2.4 Sliding Model Control 11](#_Toc167702153)

[2.5 Long Short-Term Memory 14](#_Toc167702154)

[2.6 Boruta Algorithm 16](#_Toc167702155)

[제3장 데이터 수집 및 모델 학습 18](#_Toc167702156)

[3.1 차량 설정 18](#_Toc167702157)

[3.2 데이터 수집 및 선정 19](#_Toc167702158)

[3.3 모델 학습 21](#_Toc167702159)

[3.4 LSTM 예측 결과 23](#_Toc167702160)

[제4장 차량 시뮬레이션 및 결과 25](#_Toc167702161)

[4.1 SMC 제어기 설계 25](#_Toc167702162)

[4.2 시뮬레이션 결과 28](#_Toc167702163)

[4.2.1 Race Track 주행 테스트 28](#_Toc167702164)

[4.2.2 Sinus 테스트 31](#_Toc167702165)

[제5장 결론 33](#_Toc167702166)

[Reference 34](#_Toc167702167)

[ABSTRACT 36](#_Toc167702168)

List of Tables

[Table. 3.1 Vehicle’s spec]

[Table. 3.2 Boruta Result]

[Table. 3.3 LSTM Computational Cost]

List of Figures

[Fig. 2.1] Planar Model Full Car Model

[Fig. 2.2] Kamm’s Circle

[Fig. 2.3] Slip Angle of Tire

[Fig. 2.4] Bicycle Model

[Fig. 2.5] Ackerman Angel

[Fig. 2.6] Under Steer, Neutral Steer, Over Steer

[Fig. 2.7] LSTM

[Fig. 3.1] Custom Vehicle

[Fig. 3.2] Plots of The LSTM Result and Ground Truth

[Fig. 4.1] Sliding Mode Control Part

[Fig. 4.2] Distribution Part

[Fig. 4.3] LSTM Prediction Part

[Fig. 4.4] Racing Track

[Fig. 4.5] Racing Track LSTM Result

[Fig. 4.6] Racing Track Yaw Rate

[Fig. 4.7] Section 1 and Section 2

[Fig. 4.8] Sinus Test Yaw Rate

[Fig. 4.9] Sinus Test Section 1

국문 요지

Torque Vectoring은 각 바퀴의 출력을 다르게 하여 추가적인 Yaw Moment를 생성함으로써 차량의 조향 성능을 향상시키는 기술이다. 전기자동차는 각 바퀴별로 모터를 장착하여 원하는 토크를 출력하는 것이 용이하며, 이로 인해 Torque Vectoring에 큰 이점을 가진다. 이러한 장점으로 인해 전기 자동차의 발전에 따라 Torque Vectoring 기술을 적용한 제품이 지속적으로 등장하고 있다. 이러한 Torque Vectoring 기술을 사용하기 위해서는 필요한 Yaw Moment 출력을 위한 제어기 설계가 필요하고 제어기 성능을 높이기 위해서는 정확한 차량 모델이 필요하다. 하지만 차량의 타이어는 비선형성 및 불확실성으로 인해 측정에 어려워 이를 극복하기 위한 예측 기법이 필요하다.

본 논문에서는 타이어의 인해 불확실성 변수와 외부 변수로 인한 외란에 강건한 제어를 위해 Sliding Mode Control(SMC)을 사용하였다. 하지만 불확실성 변수 및 외란의 크기가 클수록 SMC의 단점 중 하나인 Chattering 또한 심해진다. 이를 위해 Long Shrot-Term Memory(LSTM)을 통해 차량 타이어로 인한 불확실성 변수를 예측하여 Chattering 현상을 저감하는 것을 목표로 한다.

본 연구에서는 가상환경 시뮬레이션인 CarMaker를 사용하여 데이터 수집 및 실험을 진행하였다. 딥러닝 모델 학습을 위해 차량 타이어에 횡방향 마찰력과 센서 데이터를 수집하고, 민감도 분석 기법인 Boruta Algorithm을 사용하여 예측값과 관련성이 높은 입력 변수를 선택하여 LSTM 학습에 이용하였다.

먼저 CarMaker를 통해 수집한 데이터를 통해 LSTM을 학습한 결과 높은 정확성으로 예측하는 것을 확인하였다. 이를 통해 LSTM을 적용한 제어기와 적용하지 않은 제어기 두 가지를 설계하여 주행 테스트를 진행하였다. 시뮬레이션 결과, LSTM을 사용한 모델과 사용하지 않은 모델 모두 수렴하는 것을 확인하였으며, 이때 LSTM을 사용한 모델에서 Chattering 현상이 저감된 것을 확인할 수 있었다.

따라서 본 연구에서 제안된 LSTM 모델을 통해 차량의 불확실성 변수의 값을 이용하여 SMC 제어기에 적용이 가능하다는 것을 확인할 수 있었다.

제1장 서 론

## 1.1 연구 배경 및 필요성

배터리의 발전으로 인해 충분한 항속거리를 보유한 전기 자동차가 출시함에 따라 현재 많은 자동차 회사에서 전기 자동차를 출시하고 있다. 전통적인 내연기관 자동차의 엔진은 낮은 회전속도에서 낮은 토크와 효율성 문제로 인해 출력을 높이고 높은 효율 구간에서 작동할 수 있게 트랜스미션이 필요하다. 그러나 전기자동차에서 사용하는 전기모터는 작동영역 대부분에서 안정적인 토크를 제공하며, 엔진에 비해 크기가 작아 설치가 용이하고 트랜스미션이 필요 없어 차량 설계에 많은 자유도를 준다는 장점이 있다. 이로 인해 현재 전기 자동차는 여러 모터를 사용하여 바퀴를 각기 제어하는 등 다양한 구동 모델들이 개발되고 있다.

Torque Vectoring은 각 바퀴의 출력을 다르게 하여 추가적인 Yaw Moment를 생성하여 차량의 조향 성능을 향상하는 것을 말한다. 기존의 내연기관 자동차는 원하는 토크 조절의 어려움과 차량 바퀴에 동력을 전달할 때 사용되는 Differential Gear로 인해 출력 분배가 어려워 Limited Slip Differential(LSD)와 같은 부품으로 제한적으로 적용이 되었다. 반면, 전기자동차는 각 바퀴 별로 모터를 장착하여 원하는 Torque를 출력하는 것이 기존 내연기관에 비해 쉽다는 점에서 Torque Vectoring에서 큰 이점을 가질 수 있다. 또 이러한 Torque Vectoring을 통해서 더 안정된 차량 제어를 제공할 수 있고 더 상황에 따라 회전 반경을 조절하여 차량 주행에 재미를 추가할 수 있다. [1]

Torque Vectoring을 적용하기 위해서는 차량을 원하는 상태로 만들어내기 위한 Yaw Moment 입력값 계산이 중요하다. 이러한 Yaw Moment는 에너지 효율 향상, Understeer 조향, 그리고 Oversteer 조향등 목표로 하고 싶은 Yaw Rate를 고려하여 여러가지 제어 기법들을 적용하여 생성할 수 있으며, 제어 기법들은 사용할 상황에서의 단점과 장점들을 고려하여 선택할 필요가 있다. 또한 효율적인 제어를 하기 위해서는 정확한 모델링이 필요하다. 하지만 차량의 경우 타이어의 비선형성 및 센서로 측정이 불가능한 변수들로 인해 정확한 모델을 만드는데 어려움이 있어 이러한 값들을 추정하는 것이 중요하다. [2]

Torque Vectoring은 일반 차량뿐만 아니라 Formula Student 차량과 같은 개인이 제작한 차량에서도 충분히 적용 가능하다. 특히 높은 속도에서도 작은 회전반경을 만들어야 하는 레이싱 상황에서 탁월한 성능을 발휘할 수 있고, 일반적으로 공개되지 않는 차량의 내부 정보들도 이용할 수 있다는 점에서 데이터를 수집하고 사용하는데 큰 이점이 있다.

## 1.2 연구 목표

본 논문은 각 바퀴마다 모터를 장착하고 있는 4륜 구동을 적용한 Formula Student 차량에 Torque-Vectoring 적용하기 위한 제어기 설계 및 딥러닝 모델을 통한 제어기 성능 향상을 연구한다.

제어기로는 불확실성 변수와 외부에서 발생하는 외란에 대해서 강건하게 대응할 수 있는 Sliding Mode Control(SMC)을 사용하였다. 또한 시퀀스 데이터에 효과적인 Long Short-Term Memory(LSTM)을 통해 차량 불확실성 변수를 예측함으로써 Sliding Mode Control의 단점이 Chattering 현상을 저감하는 것을 목표로 하였다.

본 연구에서는 가상 환경 시뮬레이션인 CarMaker를 사용하여 데이터 수집 및 실험을 진행하였다. 딥러닝 모델 학습을 위해 CarMaker를 통해 차량 타이어에 횡방향 마찰력과 차량의 센서 데이터 및 컨트롤 데이터들을 수집하였고 민감도 분석 기법 중 하나은 Boruta Algorithm을 사용하여 예측값과 관련성이 높은 입력 변수들만을 선택하여 LSTM 학습에 이용하였다. 차량 제어기는 Mathworks 사의 Simulink를 CarMaker와 연동하여 SMC 제어기를 제작하였고 차량의 Yaw Moment를 출력하면서 목표 값과 수렴하는 정도를 확인하였다.

제2장 이론적 배경

2장에서는 본 연구 논문을 이해하기 위해 필요한 기본적인 내용을 기술한다. 본 연구 주제인 Four Wheel Drive Vehicles에 횡방향 제어 향상을 위해 Vehicle Dynamics Model, Sliding Mode Control(SMC), 마지막으로 LSTM을 사용한 시스템 모델 불확실성 예측 기법에 대해 설명한다.

## 2.1 Planar Full Car Model

차량은 타이어에 의한 마찰력으로 가감속 및 조향 제어를 한다. 그러므로 차량의 운동방정식은 차량 타이어에 관련된 변수들로 이루어져 있으며 Four Wheel Drive Vehicles의 횡방향 제어를 위한 평형식은 식 2.1과 같다.

여기서 차량의 z축 관성 모멘트, 차량의 Yaw 각가속도, 4 바퀴와 지면에서 발생하는 x축 마찰력, 4 바퀴와 지면에서 발생하는 y축 마찰력, 는 차량의 조향각, a 무게 중심과 바퀴 앞의 축 간의 거리, b 무게 중심과 바퀴 뒤의 축 간의 거리, M은 바퀴에서 발생하는 Torque Alignment, 은 각각 전륜과 후륜 바퀴 축 사이 간격을 뜻한다.

식 2.1의 우변의 경우 두개 파트로 분리할 수 있다. 식 2.2의 값은 의 값으로 이루어져 구동계의 출력을 조절해서 조절 및 예측이 가능한 항이지만, 식 2.3의 경우 바퀴의 횡방향 마찰력인 를 포함하는데 이는 타이어의 비선형성 및 직접적인 측정이 불가능하기 때문에 예측에 한계가 있다. [3][4]

[Fig. 2.1 Planar Full Car Model]

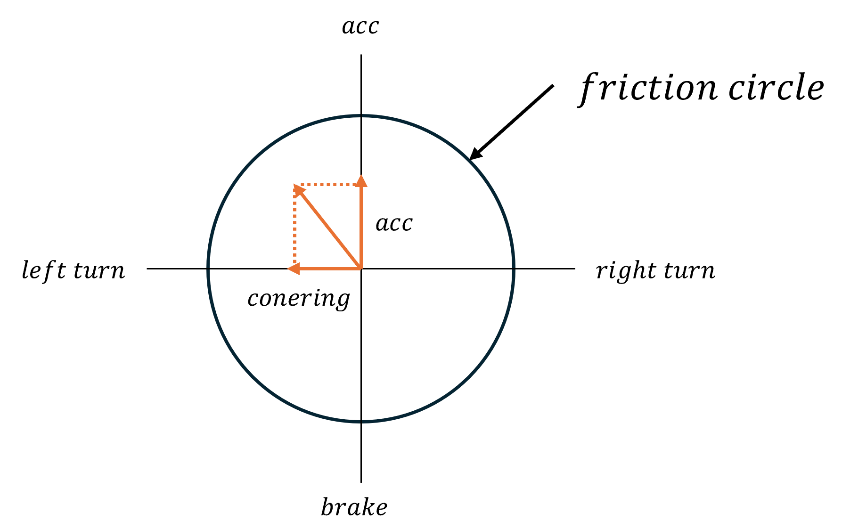


## 2.2 Tire Model

차량은 지면과 타이어의 상호작용을 통해 추진력을 얻기에 차량 동역학 모델을 제작함에 Tire Model은 필수적으로 고려해야 한다.

마찰력은 기본적으로 두 물체의 작용되는 힘과 마찰계수의 곱이상의 힘을 낼 수 없다. 그렇기에 차량의 작용하는 마찰력 힘의 합인 은 차량의 수직 항력에 마찰계수 곱 이상을 낼 수 없다. 그림 2.2는 이를 표현한 Kamm’s circle로 차량의 횡방향과 종방향 힘의 혼합은 원 안에서 발생해야 하고 이를 벗어날 경우 접지력을 잃으면서 미끄러지기 시작할 것이다. [5]

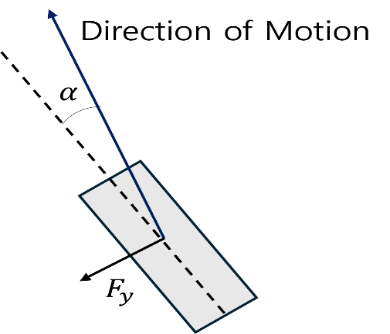
[Fig. 2.2 Kamm’s Circle]



타이어는 표면 특성과 변형 등과 같은 여러 요인들로 인해 미끄러지는 현상이 발생한다. 차량 속도와 바퀴 속도가 맞지 않는 것을 Slip ratio () 차량이 향하고 있는 방향과 실제 방향이 틀어지는 각도를 Slip angle이라 부른다.

차량의 횡방향 힘은 이 Slip Angle에 따라 달라지게 된다. 일반적으로 선형 모델을 사용할 경우 Cornering Stiffness를 적용하여 Slip Angle에 따른 횡방향 힘을 계산한다. 그러나 선형 모델이 아닌 비선형 모델을 사용할 경우 Magic Formula를 사용한다. 여기서 Magic Formula는 타이어의 복잡한 비선형 특성을 표현하기 위해 사용되는 경험적 공식이다. 식 2.4는 Magic Formula 공식으로 B,C,D는 실험을 통해 측정되는 식이다. 또한 차량의 횡방향 힘의 경우 차량 타이어의 변형으로 인해 즉각적인 반응이 일어나지 않고 입력과 출력의 지연이 발생한다. [6][7]

[Fig. 2.3 Slip Angle of Tire]



## 2.3 Steering Model

차량 횡방향 운동 방정식은 기본적으로 4 바퀴를 전부 고려해야 한다. 하지만, 이 방법은 계산에 복잡성을 증가시키고, 일반적인 주행조건에서는 조향각 의 값이 상대적으로 작기 때문에 Bicycle Model로 고려하여 계산하여도 큰 차이가 없다. 따라서 본 논문에서 목표 조향값 계산에서는 제어기의 간소화를 위하여 Bicycle Model로 고려하여 진행하였다. [8]

바퀴의 미끄러짐을 고려하지 않을 경우, 전륜 조향에 따른 차량의 회전 반경은 식 2.5와 같고 이를 Ackerman Angle으로 정의한다. 여기서 는 차량의 조향각, R 차량의 회전 반경, a는 차량의 중심과 전방 바퀴 축 간의 거리, b는 차량의 중심과 후방 바퀴 축 간의 거리를 뜻한다. 타이어의 Slip angle를 고려한 차량 회전반경은 식 2.6와 같다. 여기서 L은 차량의 길이, R은 차량의 회전 반경, K는 Understeer gradient이다. 여기서 Understeer gradient인 K의 값에 따라 차량의 조향이 Ackerman Angle와 같은 중립 조향(Neutral Steer, K=0), 보다 작은 부족 조향(Under Steer, K> 0), 보다 과해지는 과대 조향(Over Steer, K< 0)이 발생한다. [9]

일반적인 차량은 주행 목적에 맞게 위의 3가지 상태가 일어나도록 설계된다. 본 논문에서는 차량 설계가 아닌 Torque Vectoring을 활용하여 이러한 상태를 조절하며, 본 논문은 중립 조향(Neutral Steer) 상태로 제어하는 것을 목표로 한다. 이를 위해, 차량 조향각에 따른 차량의 Yaw 속도는 식 2.7와 같다. 여기서 는 차량의 속도를 뜻한다. 본 논문의 목표는 중립 조향이므로 식 2.7에 K=0을 대입하면 목표 Yaw 속도()가 도출된다. [9]

[Fig. 2.4 Bicycle Model]



[Fig. 2.6 Under Steer, Neutral Steer, Over Steer]



[Fig. 2.5 Ackerman Angle]



## 2.4 Sliding Model Control

Sliding Mode Control(SMC)은 비선형 제어 기법 중 하나로, 외부의 변동이나 불확실성에 대한 높은 저항성(Robust)을 보여주는 제어 방법이다. SMC의 주요 특징은 제어 입력을 조절하여 시스템의 상태를 특정한 슬라이딩 표면(Sliding Surface)이라는 또 하나의 상태 공간으로 이동시키는 것이다. 시스템이 이 슬라이딩 표면 위에 도달하면 슬라이딩 표면의 경계(Boundary)를 따라 미끄러지면서(Slide) 수렴하게 되는데, 이러한 과정 때문에 Sliding Surface Control라고 불러진다. 하지만 SMC는 제어 신호가 빠르게 전환되면서 시스템에 불필요한 성능 유발하는 Chattering이라는 단점이 있다. 많은 연구에서 SMC에서의 Chattering을 저감하는 연구를 진행하고 있다.[10]

SMC의 Chattering을 저감하기 위해 연속적인 제어 법칙을 적용하는 Second-order SMC와 Boundary Layer 추가 등이 제안되었다. Second-order SMC는 연속적인 제어 입력을 통해 Chattering을 저감할 수 있지만, 이산 신호 처리 지연과 통신 지연과 같은 요소로 인해 Discrete 환경에서 성능이 감소할 수 있다. Boundary Layer를 추가하는 방법은 Chattering을 줄이는 대신, 정확도의 저하와 강건성에 부정적인 영향을 줄 수 있다. [10] Chattering의 진폭 크기는 불확실성과 외란을 보상하기 위한 Switching Term의 크기로 결정된다. 그렇기에 정확한 모델을 통해 Switching Term의 크기를 줄임으로서 Chattering 현상을 줄일 수도 있다. 본 논문에서는 LSTM을 통해 정확한 불확실성 변수를 예측하여 Switching Term의 값을 감소시키는 것으로 Chattering 현상의 저감을 목표로 하였다.

SMC의 설계과정은 다음과 같다. 먼저 제어하려는 시스템을 모델링한다. 이는 2.1장을 통해 유도되었다. 다음으로 원하는 상태로 수렴하기 위한 슬라이딩 표면을 선택한다. 슬라이딩 표면은 일반적인 n차 시스템에서 2.8같이 표현된다. 하지만 본 논문에서 사용한 시스템은 1차 시스템으로 이는 식 2.9와 같다. 계속해서 설정한 슬라이딩 표면 위에서 유한 시간 내에 원점으로 향하도록 제어 입력 값을 설정해야 한다. 이를 위해서는 슬라이딩 표면이 0으로 수렴하고 안정적인 제어를 위해서 슬라이딩 표면의 미분 값들이 0으로 수렴하도록 설정해야 한다. 식 2.10은 슬라이딩 표면의 미분값을 식 2.1을 이용하여 정리한 식이다. 이를 바탕으로 제어 입력 값을 계산하면 식 2.11이 된다. 여기서 d(t)는 시스템의 불안정성을 위해 추가한 항이고 는 입력 값, 는 예측이 필요한 항, 는 예측에 발생할 수 있는 오차를 표현한 항이다. 식 2.11은 제어 입력 값 의 식으로 식 2.10에서 모델 외부에서 발생하는 외란 와 모델 불확실성 를 제외한 후 나머지 변수들을 우변에 정리하 후 앞서 제외한 두 값을 보상해줄 Switching Term 을 우변에 추가한 것이다. 마지막으로 설정한 제어 입력 값이 시간에 따라 수렴을 하는지 확인하기 위하여 Lyapunov stability를 실시한다. Lyapunov stability에서 locally positive definite function 는 을 만족하고, 을 만족할 때 안정성을 보장한다. 으로 정의한 후, 을 만족하는 식을 찾으면 식 2.12와 같다. 을 만족시키기 위하여 이를 만족시키는 K를 설정한다면 위의 모델은 항상 수렴할 수 있다.

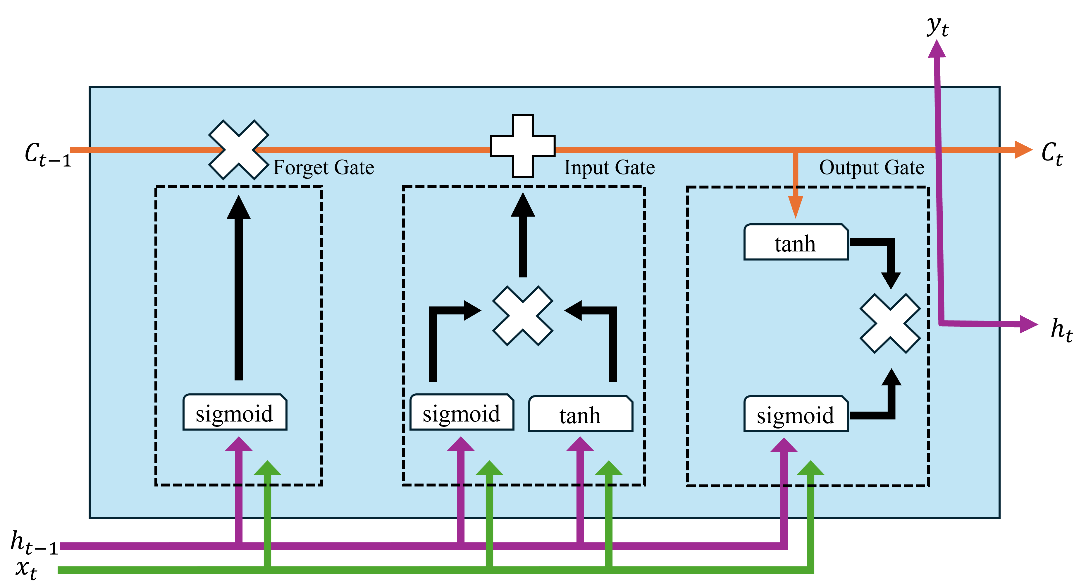
## 2.5 Long Short-Term Memory

앞서 2.1.2장에서 차량의 횡방향 힘은 여러 요인에 따라 달라지게 되고 이는 센서로 측정하기 힘든 지표이다. 이에 따라 [12], [13]에서는 다양한 Kalman Filter를 사용하여 횡방향 힘을 예측하였고, [14]에서는 DNN을 사용하여 횡방향 힘을 추정하였다. 본 논문에서는 딥러닝 모델을 사용하여 횡방향 힘으로 구성된 값을 차량 내부 센서 데이터 및 입력 데이터를 기반으로 예측한다. 딥러닝 모델은 타이어의 입력 지연을 고려하여 시간 데이터에 좋은 성능을 보이는 LSTM을 사용하였고, 예측된 값을 사용하여 2.2절에 언급된 SMC의 제어 성능 향상을 목표로 하였다.

순환 신경망(Recurrent Neural Networks, RNN)은 시간에 따라 변화하는 데이터를 처리하는데 있어 사용되어 왔었다. 하지만 연속된 데이터가 길어질 수록 앞서 입력된 정보를 장기간 유지하지 못하던 장기 의존성 문제라는 한계가 있었다. LSTM은 이러한 한계를 해결하기 위해 고안되었으며, 이후 많은 연소된 데이터의 처리에 활용되어졌다. [11]

LSTM의 핵심 구조는 Cell State 한 개와 Input Gate, Forget Gate, Output gate 3개를 추가하여 Cell State 정보를 계속 해서 조정해서 다음 단계로 전달하는 것이다. 여기서 Input Gate는 새로운 정보를 Cell State에 추가할 것인지를 결정하고, Forget Gate는 어떤 정보를 제거할 것인지, 마지막으로 Output Gate는 어떠한 정보를 출력할지 결정한다. Fig 2.6은 LSTM 모델을 도식화한 모습으로 C는 Cell State, h는 hidden State, x는 입력 값, y는 출력 값에 해당한다.

[Fig. 2.7 LSTM]



## 2.6 Boruta Algorithm

딥러닝은 Scale이 커질수록 정확도가 높아지는 경향성을 보이지만, 이러한 Scale 증가는 더 많은 Computing Power를 요구하며, 이는 연산 시간의 증가로 직결된다. 이는 Real-Time System에서 치명적인 문제를 발생시킨다. 따라서, 딥러닝 모델의 경량화를 통해 정확도와 연산 속도를 최적화하는 것이 중요하다. [15]

모델 경량화를 위한 전략 중 하나는 불필요한 변수를 제거하여 모델의 복잡성을 줄이고 추론속도를 가속화하는 것이다. [16] 이 과정에서 Sensitivity Analysis가 중요한 역할을 한다. Sensitivity Analysis는 다양한 입력 변수들이 모델 출력에 미치는 영향을 평가하여, 결과에 가장 큰 영향을 주는 변수를 식별하는 방법으로, 이를 통하여 필요한 변수만을 선택함으로 모델의 효율성을 증대할 수 있다.

본 논문에서는 Sensitivity Analysis 방법들 중 Boruta Algorithm을 사용하였다. Boruta Algorithm은 Random Forest를 기반으로 하며, 이를 통해 변수들의 중요도를 정량적으로 평가한다. Boruta Algorithm은 자동화된 선택 과정을 통해 많은 변수들을 사용자 개입을 최소화하고 분석할 수 있다는 장점이 있다. 본 방법론의 절차는 다음과 같다. 먼저 데이터셋에 존재하는 모든 변수들과 그들을 무작위로 섞은 복사본을 함께 사용하여 Random Forest를 학습시킨다. 학습을 통해 각 변수의 중요도가 Random Forest의 특성 중요도(Feature Importance)에 의해 평가되게 된다. 훈련된 모델로부터 원래 변수와 복사본의 중요도를 전부 추출하여 복사본 보다 낮은 중요성을 보여주는 변수들을 제거하는 식으로 진행한다. Random Forest는 학습 때마다 다른 결과를 보여줄 수 있으므로 위의 과정을 반복하여 실제로 중요한 변수들을 선별하는 방법이다. [17]

제3장 데이터 수집 및 모델 학습

본 논문에서 LSTM을 위한 데이터 셋을 제작하기 위하여 IPG사에서 개발한 차량용 가상환경 시뮬레이터인 CarMaker 11.2ver을 사용하였다.

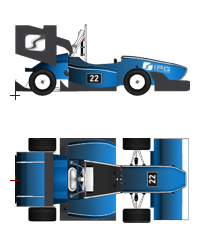
## 3.1 차량 설정

차량은 학생용 Fomula차량 모델을 사용하여 CarMaker 상에서 Custom Vehicle을 제작하였다. 차량의 상세 스펙은 Table 3.1과 같다. 타이어는 CarMaker에서 제공하는 “FS\_205\_50R13”모델을 사용하였다. 차량 운전자의 무게는 70kg으로 설정하여 차량 주행을 진행하였다.

|  |  |  |  |
| --- | --- | --- | --- |
| Name | Value | Name | Value |
| Mass [kg] | 226.26 | Stiffness of Front Spring [N/m] | 35000 |
| Length of Body [m] | 1.6 | Stiffness of Rear Spring [N/m] | 35000 |
| Distance CoG to Front[m] | 0.813 | Damping of Front [m/s] | 2500 |
| Distance CoG to Real[m] | 0.787 | Damping of Rear [m/s] | 2500 |
| Moment of Inertia x [kgm2] | 53.624 | Moment of Inertia Motor [kgm2] | 0.01 |
| Moment of inertia y [kgm2] | 134.629 | Max Motor Torque [Nm] | 30 |
| Moment of inertia z [kgm2] | 146.827 | Gear Ratio | 12;3 |

[Table. 3.1 Vehicle’s Spec]

[Fig. 3.1 Custom Vehicle]



## 3.2 데이터 수집 및 선정

주행 데이터를 수집하기 위해 맑은 날씨의 평지 아스팔트 도로 위(일정한 마찰계수 1을 유지)에서 차량의 속도(0-100km/h)와 바퀴의 조향각(-34 – 34 deg)을 다양하게 조정하며 10ms 주기마다 데이터를 저장하였다. 논문의 목표 대상인 Formula Student 차량은 고성능 칩셋을 사용하기에는 제한사항이 있어 모델 축소화가 굉장히 중요하다. 이를 위해 2.4장의 Boruta Algorithm을 사용하여 CarMaker에서 취득가능한 변수들 중 일부 데이터를 선발하였다. Table 3.2는 Boruta Algorithm을 통해 우선 선발한 데이터들에 한번 Boruta Algorithm을 사용하여 중요도가 복사본을 넘긴 횟수(HInts)와 중요도의 합산(Importances)를 확인한 값이다. 여기에는 차량에서 취득가능한 센서(IMU) 데이터 및 제어 신호등이 포함되었다. Tables 3.2에서 Hints가 한번 실패한 Az값과 Motor Speed의 값과 Velocity의 시간에 따른 값 변환 추이가 크게 차이가 없음을 확인하여 Motor Speed를 제외한 [Steering Angle, Motor Torque FL, Motor Torque FR, Motor Torque RL, Motor Torque RR, ax, ay, Yaw velocity, Vehicle Speed] 총 9개의 변수를 선택하였다. 또한 이러한 데이터는 단위와 데이터의 분포가 균일하지 못하면 학습 효과가 떨어지므로 입력 변수들의 최댓값과 최솟값을 찾아 Min-Max Normalization을 진행해주었다.

|  |  |  |
| --- | --- | --- |
| Name | Hints | Importances |
| Steering [rad] | 100 | 1.124178721 |
| Motor Speed FL [rad/s] | 100 | 0.160317027 |
| Motor Speed FR [rad/s] | 100 | 4.491055273 |
| Motor Speed RL [rad/s] | 100 | 1.223174742 |
| Motor Speed RR [rad/s] | 100 | 1.469073461 |
| Motor Torque FL [N/m] | 100 | 3.474548487 |
| Motor Torque FR [N/m] | 100 | 11.01922642 |
| Motor Torque RL [N/m] | 100 | 10.56697544 |
| Motor Torque RR [N/m] | 100 | 3.09991089 |
| Ax [m/s^2] | 100 | 0.197302616 |
| Ay [m/s^2] | 100 | 41.39588124 |
| Az [m/s^2] | 99 | 0.05426448 |
| Roll velocity [m/s] | 100 | 2.202003413 |
| Yaw velocity [m/s] | 100 | 1.743530038 |
| Pitch velocity [m/s] | 100 | 0.014232528 |
| Velocity [m/s] | 100 | 17.76381043 |

[Table. 3.2 Boruta Result]

## 3.3 모델 학습

LSTM 모델을 학습하기 위해 Python의 Pytorch를 사용하였다. 모델 크기에 따른 성능 변화 및 계산량을 알아보기 위해 hidden size를 10~125까지 5개 단위로 변화하며 학습을 진행하였다. 계산량은 FLOPs(Floating Point Operations)를 사용하여 비교하였다. FLOPs란 딥 러닝 모델에서 수행되는 계산량을 측정하는 한 방법으로, 사칙연산을 포함하여 얼마나 많은 계산이 진행되었는지를 뜻한다. 사용한 LSTM 모델의 FLOPs 계산 식은 식 3.1와 같다. 각 항의 의미는 다음과 같다. 는 각 Gate들을 통해 이루어지는 계산을 나타낸다. 는 Sigmoid와 Tanh 함수에 의해 발생하는 계산량을 의미한다. 는 이전 Cell State와 현재 gate에서 출력된 값으로 Cell State를 업데이트 하는 계산을 나타낸다. 마지막으로 e는 Fully Connected Layer에서 발생하는 계산을 의미한다.

Table 3.2는 FLOPs와 400 Epoch를 완료했을 때의 Train Error와 Validation Error를 정리한 표이다. hidden size가 커질수록 Error가 더 낮은 값에서 수렴하는 것을 확인할 수 있다. 모델의 크기가 커질수록 Error가 낮아지므로 예측 성능이 높다고 할 수 있다. 하지만 FLOPs또한 크게 증가하기에 계산량이 증가한 만큼 실질적인 효과를 얻었는지 확인할 필요가 있다.

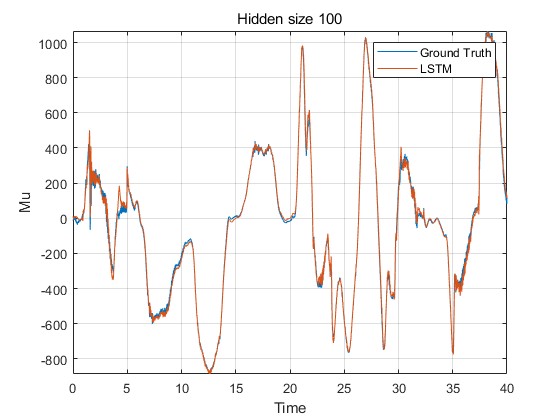
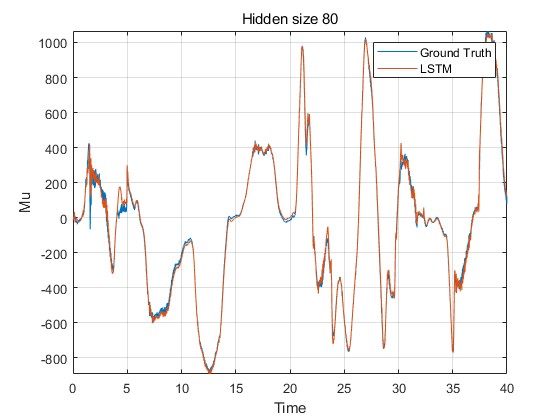
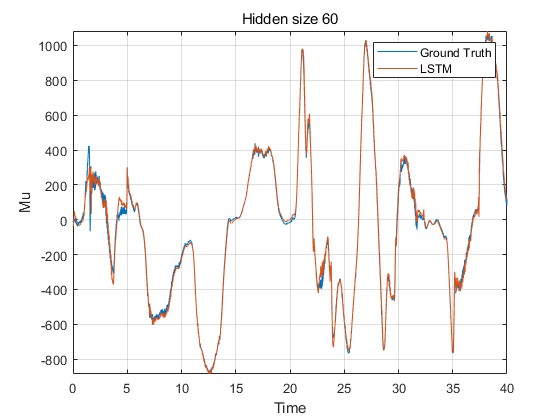
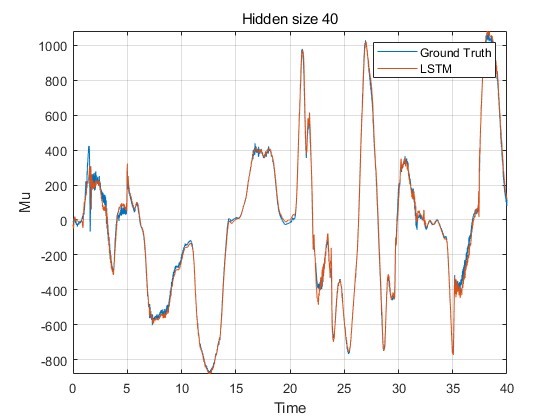
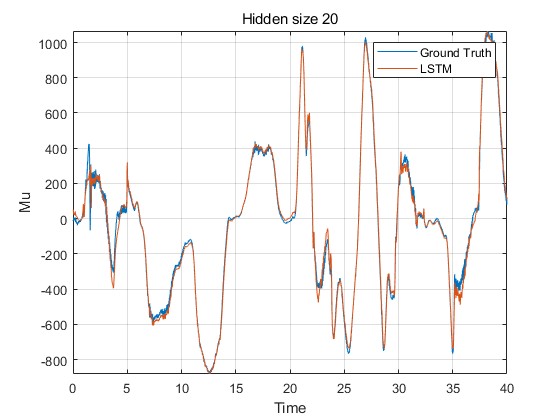
|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Hidden Size | FLOPs | Train Error | Validation Error | Hidden Size | FLOPs | Train Error | Validation Error |
| 10 | 350 | 6.18e-5 | 6.20e-5 | 70 | 23100 | 8.45e-6 | 8.87e-6 |
| 15 | 900 | 5.28e-5 | 5.63e-5 | 75 | 26250 | 8.86e-6 | 1.02e-5 |
| 20 | 1650 | 4.01e-5 | 4.24e-5 | 80 | 296000 | 7.79e-6 | 9.38e-6 |
| 25 | 3750 | 2.67e-5 | 2.91e-5 | 85 | 33150 | 6.5e-6 | 7.04e-6 |
| 30 | 5100 | 2.47e-5 | 2.5e-5 | 90 | 16900 | 5.82e-6 | 6.46e-6 |
| 35 | 6650 | 2.37e-5 | 3.18e-5 | 95 | 40850 | 6.38e-6 | 6.7e-6 |
| 40 | 8400 | 1.74e-5 | 1.88e-5 | 100 | 45000 | 5.88e-6 | 6.98e-6 |
| 45 | 10350 | 1.55e-5 | 1.63e-5 | 105 | 49350 | 5.90e-6 | 6.41e-6 |
| 50 | 12500 | 1.75e-5 | 2.34e-5 | 110 | 53900 | 4.79e-6 | 5.38e-6 |
| 55 | 14850 | 1.11e-5 | 1.25e-5 | 115 | 58650 | 4.57e-6 | 5.29e-6 |
| 60 | 17400 | 1.05e-5 | 1.16e-5 | 120 | 63600 | 4.61e-6 | 5.02e-6 |
| 65 | 20150 | 8.53e-6 | 8.56e-6 | 125 | 68750 | 4.58e-6 | 6.40e-6 |

[Table. 3.2 LSTM Model Computational Cost]

## 3.4 LSTM 예측 결과

LSTM으로 예측한 결과를 확인하기 위해 학습에 이용하지 않은 데이터를 사용하여 전체 학습 모델 중 hidden size가 20, 40, 60, 80, 100인 모델들의 예측 결과를 확인하였다. Fig 3.2는 LSTM 예측 결과와 실제 값(Ground Truth)을 나타낸 그림이다. Fig.3.2의 그래프들을 통해 hidden size가 증가함에 따라 정확도가 크게 증가하지 않는 것을 확인할 수 있다. LSTM을 통해 예측하는 값은 최대 4000[Nm]에서 최소 -4000[Nm] 범위를 가진다. 하지만 계산량을 높임으로써 얻는 정확도 증가는 이에 비해 작으므로 모델의 크기를 계속해서 키울 필요가 없다. 그러므로 FLOPs가 10000이 넘지 않는 hidden size가 40인 모델을 사용하여 CarMaker 시뮬레이션 실험을 진행하였다.

[Fig. 3.2 Plots of The LSTM Result and Ground Truth]



제4장 차량 시뮬레이션 및 결과

본 장에서 3장에서 제작한 LSTM을 적용하여 SMC 제어기 설계하고 LSTM을 사용하지 않은 모델과 사용한 모델 두개의 결과를 비교 논의한다.

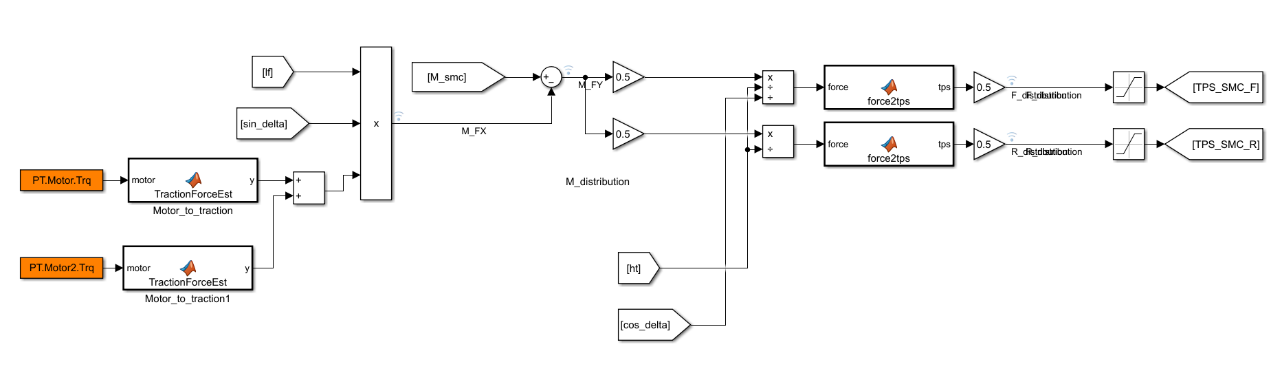
## 4.1 SMC 제어기 설계

2.2장에서 시스템에 대한 입력 값으로 식 2.11로 설정하였다. 이를 Simulink 환경에서 구현하면 그림 4.1과 같다. 여기서 는 식 2.7을 사용하여 계산하였고 이의 미분 값이 필요하므로 Simulink의 Derivative 블록과 LPF를 추가하여 높은 주파수 영역을 차단하여 급격한 제어를 방지하였다. 또한 LSTM의 예측 값이 조향각이 없을 때는 동력 분배가 필요 없으므로 직선 주행시 제어기 출력을 제거하도록 설정하였다. 제어기의 Switching Term의 Gain의 값은 모델 예측에 오차와 차량 외란 두가지를 고려하여 설정해야 한다. 가상환경은 이상적인 환경을 제공해주므로 외란의 값을 크게 고려하지 않았다. LSTM을 사용한 모델의 경우 3.3장에서 오차가 거의 발생하지 않았지만 고려하지 못한 오차가 발생할 수 있기에 500의 Gain 여유값을 두었고, LSTM이 없을 시에는 불확실성 변수인 의 최대 범위(-4000~ 4000[Nm])를 넘도록 설정해야 하므로 4500의 Gain 값을 설정하였다.

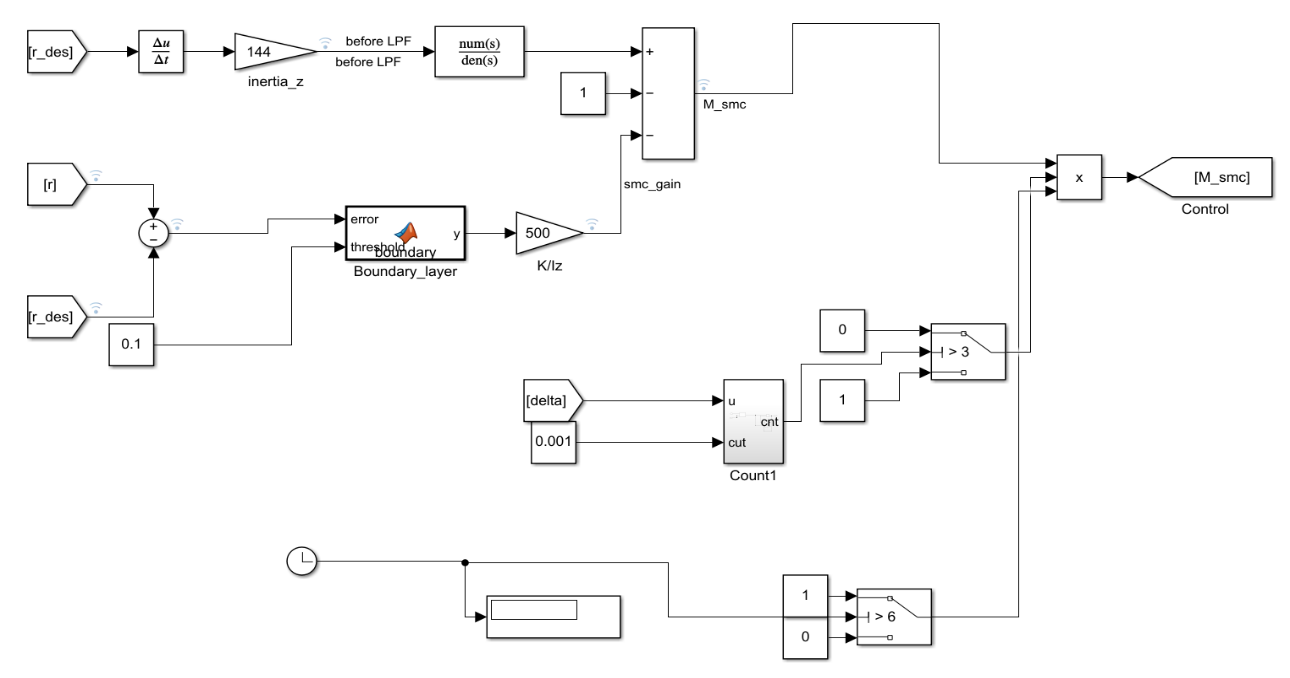
다음으로 SMC 제어기를 통해 출력된 입력값을 시스템의 제어 입력인 식 2.2에 대입하여 실제로 필요한 분배량을 계산하였다. 식 2.2는 두가지 요소, 바퀴 조향각에 의해 발생하는 Moment(식 4.1)와 토크 분배로 인해 발생하는 Moment(식 4.2)을 포함한다. 제어기에서 출력된 Moment에서 바퀴 조향에 따라 발생하는 부분(식 4.1)을 제거한 후 남은 값을 각 차량 바퀴에 토크를 분배해서 생성하였다. 본 논문에서는 전후의 동력 배분을 5:5로 설정하였고, 좌우는 필요한 Moment를 생성하기 위한 힘의 차이를 균등하게 배분하는 식으로 진행하였다.

Simulink와 CarMaker는 1ms 고정 시간 간격으로 시스템이 구동된다. 하지만 LSTM 모델 제작시에 10ms 주기로 데이터를 수집하여 학습을 진행하였다. 이를 고려하여 10ms 주기로 데이터를 수집하여 저장하는 기능을 추가하여 200개의 데이터를 저장하였고 이를 입력 데이터로 사용하여 예측 값을 계산하였다.

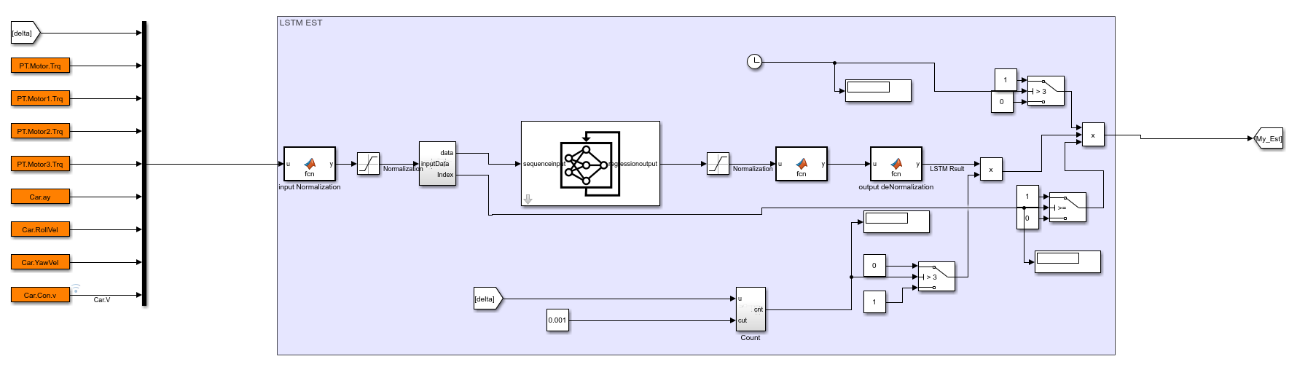
[Fig. 4.2 Distribution Part]



[Fig. 4.1 Sliding Mode Control Part]



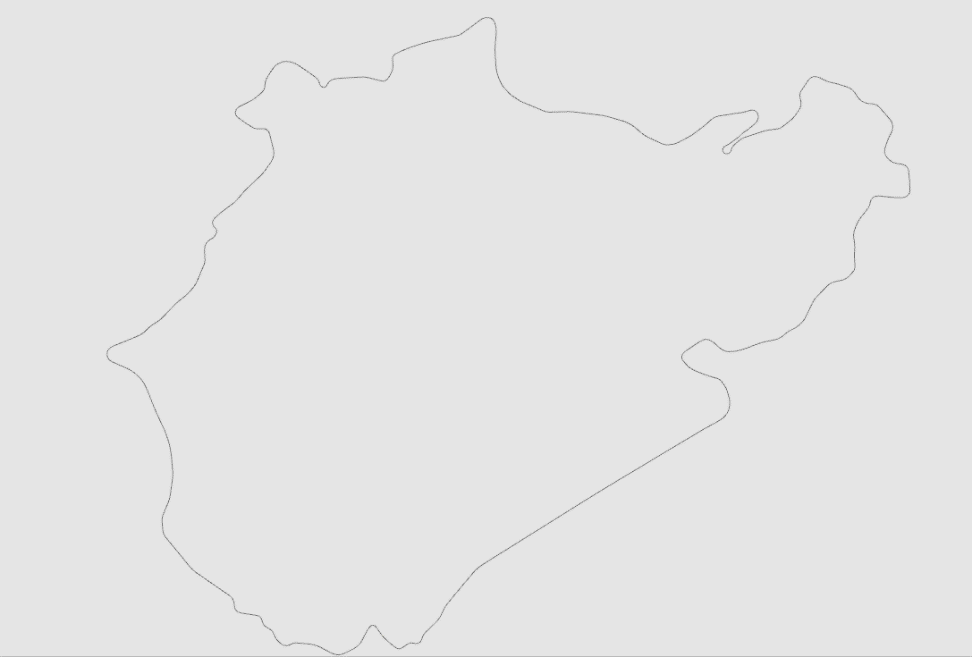
[Fig. 4.3 LSTM Prediction Part]



## 4.2 시뮬레이션 결과

시뮬레이션은 CarMaker에서 제공하는 기본 주행기능을 사용하여 레이싱 Track과 주기적으로 handle을 조작한 Sinus 테스트 3가지를 LSTM 사용한 모델과 사용하지 않은 모델 두 가지로 진행하였고, Sliding Mode Control의 성능을 확인하기 위하여 목표 변수인 Yaw rate의 값을 확인하였다.

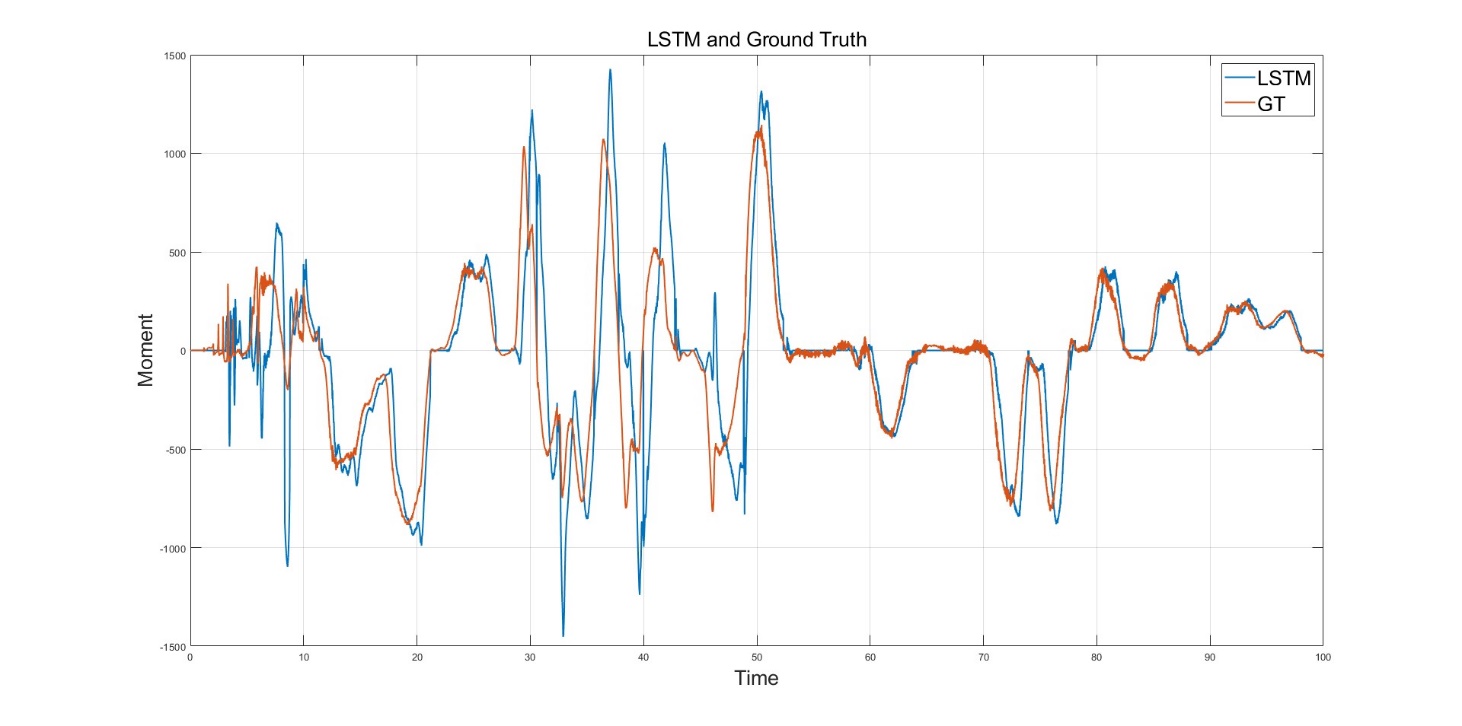
[Fig. 4.4 Racing Track ]



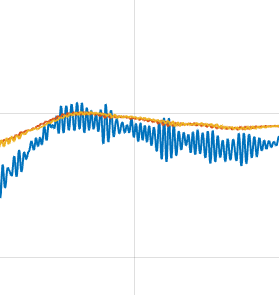
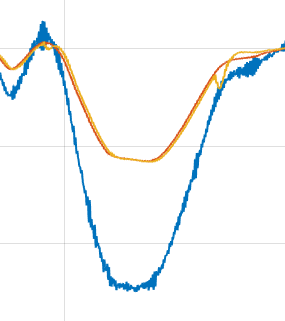
### 4.2.1 Race Track 주행 테스트

Fig 4.5은 Racing Track을 주행하면서 LSTM 결과(파란색)와 실제 값을 보여주는 그래프이다. 초기 LSTM을 위한 데이터가 쌓이기 전에 데이터에서는 낮은 예측을 보이지만 초기를 지나고서는 높은 정확도를 보였다. 초기 정확도가 낮은 부분의 결과를 반영하지 않게 하므로 이 부분을 부분적으로 해결하였다. Fig 4.6은 Track 1에서 차량의 조향각에 따른 Desired yaw rate(파란색), LSTM에서의 yaw rate(빨간색), 마지막으로 LSTM을 사용하지 않은 모델의 yaw rate(노란색)를 나타낸 그래프이다. Fig.4.6의 Section 1에 파란색 그래프를 보면 Desired yaw rate가 크게 흔들리는 것을 확인할 수 있다. 이에 따라 결과의 흔들림이 SMC에 인한 것인지 아니면 Desired yaw rate의 흔들림으로 인한 것인지 확인하기 어렵다. 이러한 현상이 발생한 이유로 Desired yaw rate를 차량의 조향각에 따라 생성하였고, CarMaker의 주행기능은 설정한 주행경로를 따라가게 된다. 이에 따라 주행경로를 유지하기 위해 조향이 진동하게 되고 이 진동에 따라 Desired yaw rate가 흔들리는 것으로 예상된다. 또한 Fig.4.6의 Section 2를 확인할 시 파란색 선에 수렴하지 않는 것을 확인할 수 있다. 이는 차량의 Torque Vectoring 과 조향에 따라 발생하는 Yaw Moment 최대값의 한계로 인해 수렴을 하기 전에 Desired yaw rate의 상태가 변화했기 때문에 발생하는 것으로 여겨진다.

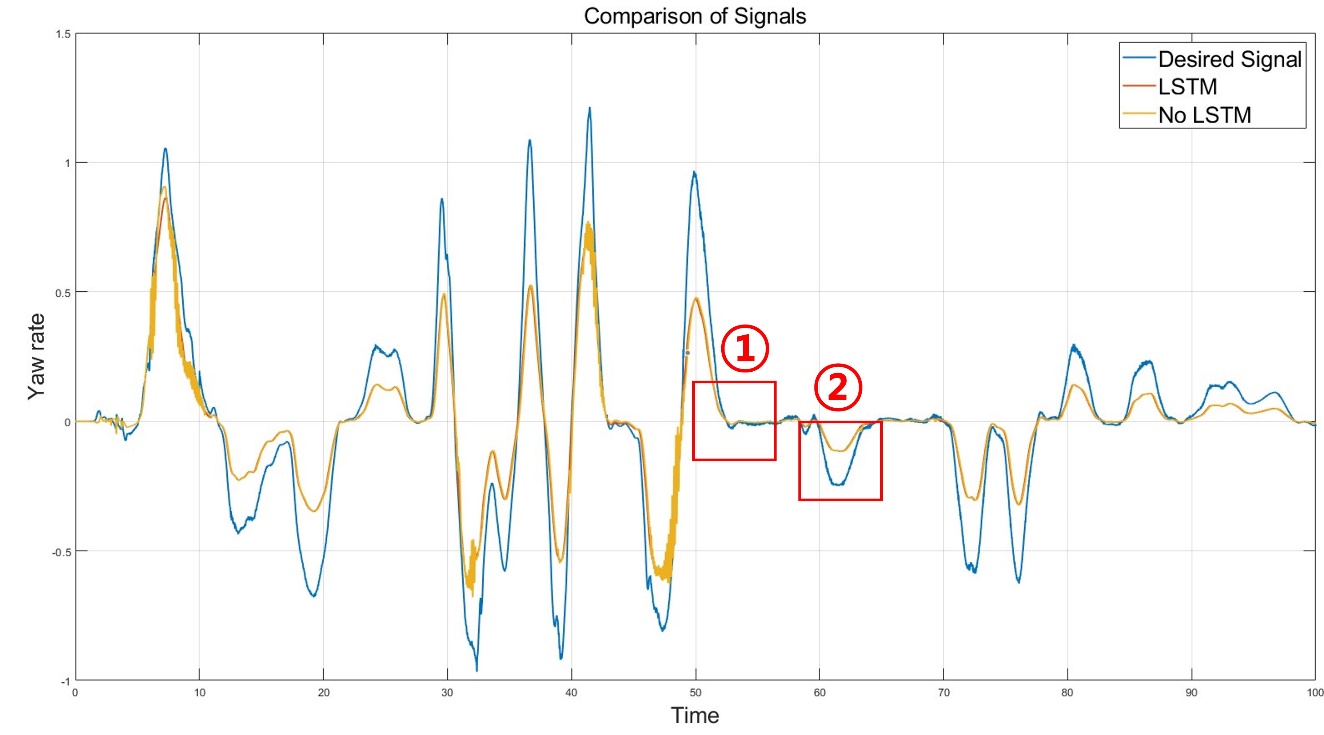
[Fig. 4.5 Racing Track LSTM Result]



[Fig. 4.7 Section 1 and Section 2]



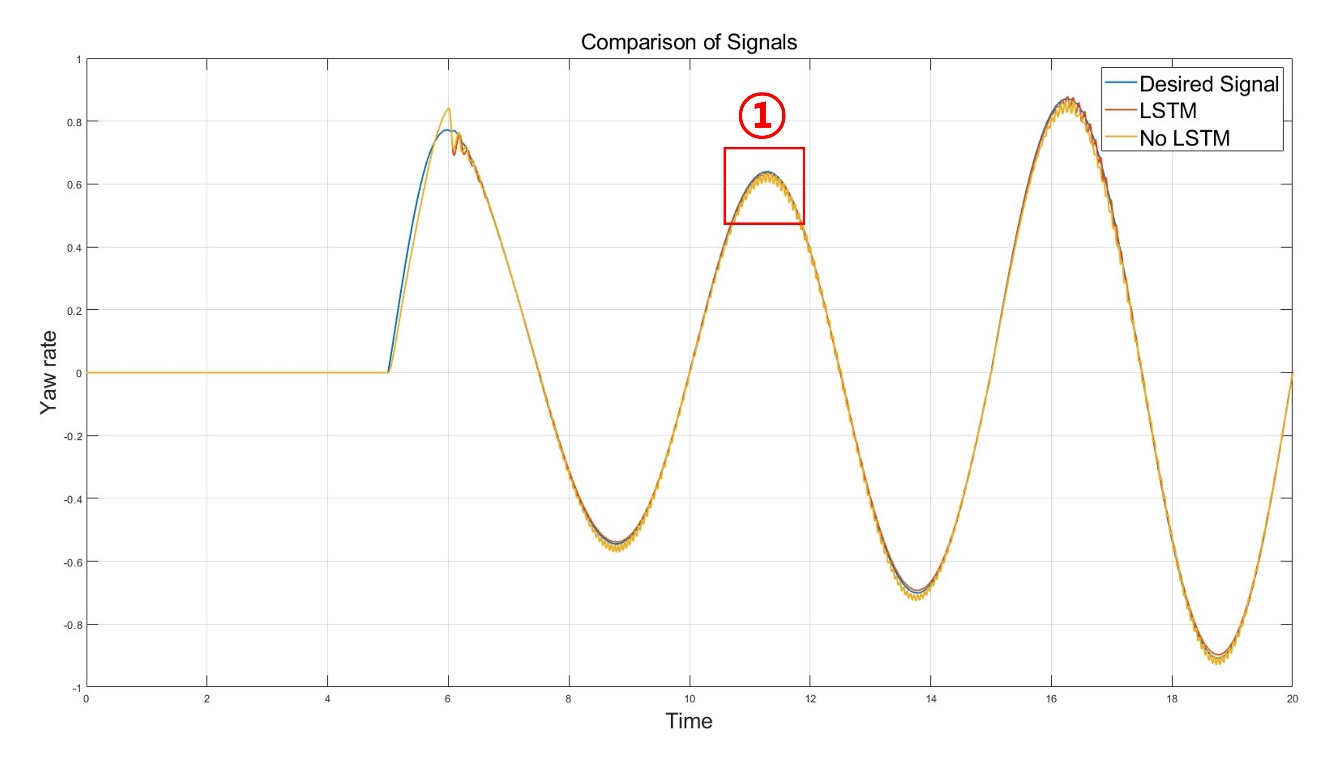
[Fig. 4.6 Racing Track Yaw rate]



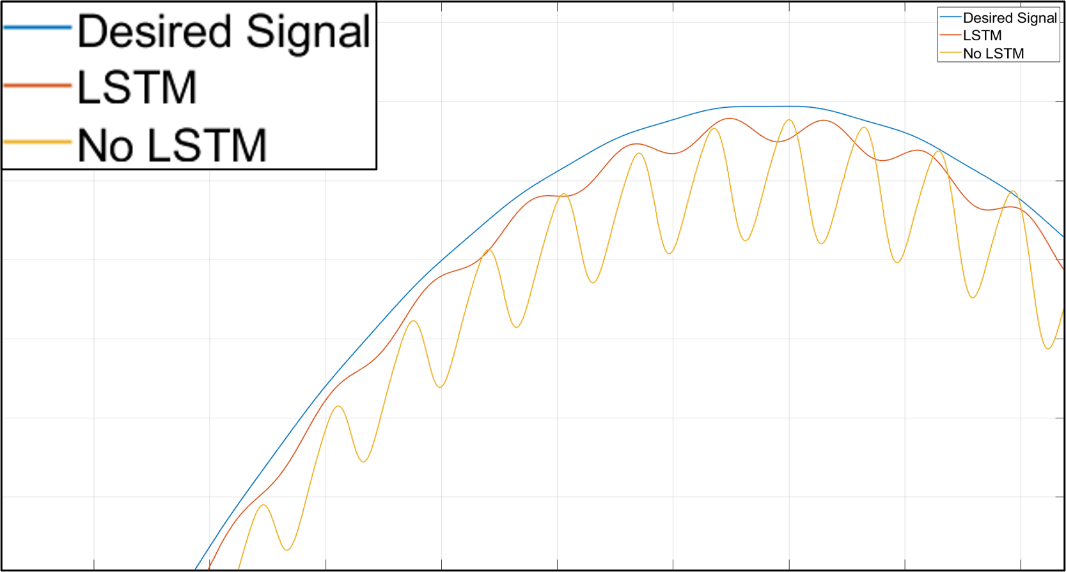
### 4.2.2 Sinus 테스트

Racing Track Test에서는 CarMaker의 주행기능과 급격한 Desired yaw rate의 변화를 따라가지 못하여 LSTM을 통한 Chattering 저감을 확인할 수 없었다. 또한 CarMaker의 자동 주행기능에서 발생한 입력의 진동을 없애기 위하여 CarMaker의 주행 기능을 사용하지 않고 일정한 속도에서 주기성을 가진 조향 입력을 줄 수 있는 Sinus 테스트를 진행하였다. Fig. 4.8는 시속 40km/h에서 5초의 주기로 조향 테스트를 진행한 결과이다. Fig 4.8에서 LSTM을 사용한 모델과 사용하지 않은 모델 모두 Desired Signal을 잘 추정하는 것을 확인할 수 있다. 이를 통해 더 낮은 Gain 값을 설정하여도 수렴하는 것을 확인할 수 있고, Fig 4.9에서 LSTM을 사용한 모델에 진폭이 더 낮을 것을 통하여 Chattering 현상의 저감을 확인할 수 있다.

[Fig. 4.8 Sinus Test Yaw rate]



[Fig. 4.9 Sinus Test Section 1]



제5장 결론

본 연구에서는 4개의 구동 모터를 사용하는 Formula Student 차량의 Torque Vectoring 적용을 위한 Sliding Mode Control 제어기와 LSTM을 활용한 Sliding Mode Control의 Chattering 저감 가능성에 대해 연구하였다. IPG 사의 CarMaker를 사용하여 데이터를 수집하여 LSTM 모델을 학습하고 주행 테스트를 진행하면서 목표 값인 Yaw rate의 값을 확인하였다. LSTM 사용하여 Switching Term의 Gain 값을 낮춘 상태에서도 목표 yaw rate가 수렴하는 것을 확인하였고, Switching Term의 Gain의 값이 낮아 Chattering 현상도 감소하는 것을 확인하였다.

하지만 Track에서 주행하는 테스트 결과 차량의 종방향 힘과 Torque Vectoring의 최댓값 합보다 큰 Yaw Moment는 생성하지 못하여 급격한 Yaw rate 변화가 발생하는 상황에서는 수렴하지 못하는 한계가 있다. 또한 가상환경은 이상적인 상황을 가정하므로 현실 상황에서도 사용가능한지에 대한 추가 검증이 필요하다. 또한, 다양한 외란 조건에서의 성능 평가가 추가로 요구됩니다.

Reference

[1] De Novellis, L., Sorniotti, A., & Gruber, P. (2013). Wheel torque distribution criteria for electric vehicles with torque-vectoring differentials. IEEE transactions on vehicular technology, 63(4), 1593-1602.

[2] De Novellis, L., Sorniotti, A., Gruber, P., & Pennycott, A. (2014). Comparison of feedback control techniques for torque-vectoring control of fully electric vehicles. IEEE Transactions on Vehicular Technology, 63(8), 3612-3623.

[3] Goggia, T., Sorniotti, A., De Novellis, L., Ferrara, A., Gruber, P., Theunissen, J., ... & Zehetner, J. (2014). Integral sliding mode for the torque-vectoring control of fully electric vehicles: Theoretical design and experimental assessment. IEEE Transactions on Vehicular Technology, 64(5), 1701-1715.

[4] Mikuláš, E., Gulan, M., & Takács, G. (2018, June). Model predictive torque vectoring control for a formula student electric racing car. In 2018 European Control Conference (ECC) (pp. 581-588). IEEE.

[5] Kumar, A., & Wagatsuma, H. (2022). A Kamm’s Circle-Based Potential Risk Estimation Scheme in the Local Dynamic Map Computation Enhanced by Binary Decision Diagrams. *Sensors*, *22*(19), 7253.

[6] Bhoraskar, A., & Sakthivel, P. (2017, January). A review and a comparison of Dugoff and modified Dugoff formula with Magic formula. In 2017 international conference on nascent technologies in engineering (ICNTE) (pp. 1-4). IEEE.

[7] Lee, E., Jung, H., & Choi, S. (2018). Tire lateral force estimation using Kalman filter. International Journal of Automotive Technology, 19, 669-676.

[8] Wang, R., Zhang, H., & Wang, J. (2013). Linear parameter-varying controller design for four-wheel independently actuated electric ground vehicles with active steering systems. IEEE Transactions on Control Systems Technology, 22(4), 1281-1296.

[9] Handbook, A. (2011). Bosch Automotive Handbook.

[10] Utkin, V., Guldner, J., & Shi, J. (2017). Sliding mode control in electro-mechanical systems. CRC press.

[11] Hochreiter, S., & Schmidhuber, J. (1997). Long short-term memory. Neural Computation, 9(8), 1735-1780.

[12] 이동훈, 김인근, & 허건수. (2012). 비선형 Kalman Filter 를 사용한 타이어 횡력 추정 시스템. 한국자동차공학회논문집, 20(6), 126-131.

[13] Xu, N., Askari, H., Huang, Y., Zhou, J., & Khajepour, A. (2020). Tire force estimation in intelligent tires using machine learning. IEEE Transactions on Intelligent Transportation Systems, 23(4), 3565-3574.

[14] Cho, W., Yoon, J., Yim, S., Koo, B., & Yi, K. (2009). Estimation of tire forces for application to vehicle stability control. IEEE Transactions on Vehicular Technology, 59(2), 638-649.

[15] Wu, Y., Ruan, X., Zhang, Y., Zhou, H., Du, S., & Wu, G. (2019). Lightweight architecture for real-time hand pose estimation with deep supervision. Symmetry, 11(4), 585.

[16] Sui, L., Du, B., Zhang, M., & Sun, K. (2021, May). A new variable selection algorithm for LSTM neural network. In 2021 IEEE 10th Data Driven Control and Learning Systems Conference (DDCLS) (pp. 571-576). IEEE.

[17] Kursa, M. B., Jankowski, A., & Rudnicki, W. R. (2010). Boruta–a system for feature selection. Fundamenta Informaticae, 101(4), 271-285.

ABSTRACT

**A Study on The Application of Sliding Mode Control in Vehicle Torque Vectoring by Prediction of LSTM-based Vehicle Model Uncertainty Variables**

Jinmin Kim

Dept. of Automotive Engineering

(Automotive-Computer Convergence)

Graduate School of Hanyang University

Torque Vectoring is a technology that enhances vehicle handling performance by generating additional yaw moments through distributing the output to each wheel. The electrical vehicle has a big advantage to output the desired torque to output the desired torque Vector by each wheel. Due to this advantage, the vehicle that applied to Torque Vectoring technology is continuously appearing. To use these Torque Vectoring technology, it is necessary to design a controller for yaw moment output and accurate vehicle model. However, due to the nonlinearity and uncertainties of tires, predicting these variables are very important.

In this paper, Sliding Mode Control (SMC) is employed to ensure robustness against uncertainties and disturbances caused by tire characteristics and external factors. However, increased uncertainties and disturbances aggravate one of SMC’s disadvantages: chattering. To prevent this, Long Shrot-Term Memory(LSTM) networks are used to predict the uncertainties variables, thereby reducing chattering

Data collection and experiments were conducted using the virtual environment simulator CarMaker. For training the deep learning model, lateral tire friction and sensor data were collected. The Boruta Algorithm, a sensitivity analyses technique, was used to select input variables with highly related to the prediction targets for LSTM training.

The results from training the LSTM with data collected from CarMaker demonstrated high prediction accuracy. Two controllers, one using LSTM predictions and one without, were designed and tested in driving simulations. Both models converged, but the LSTM-based model showed less chattering.

Thus, this study confirms that the proposed LSTM mdeol can effectively be applied to SMC controllers by utilizing predicted values of vehicle uncertainties.