

이동 객체 추정 성능 향상을 위한 LSTM 모델 기반 칼만필터 오차 보상 기법

LSTM-based Kalman Filter Error Compensation for Enhanced Moving Object Tracking

김 소 현¹, 최 재 훈¹, 정 다 연¹, 이 석 원^{1*}

(Sohyun Kim¹, Jaehoon Choi¹, Dayeon Jeong¹, and Seokwon Lee^{1,*})

¹Department of Mechanical Engineering, Chung-Ang University

Abstract: The accurate and easy estimation of system models and parameters of a moving object remains an issue that requires further improvement. Although the interacting multiple model filter (IMMF) has been employed in previous studies to achieve good results, the importance of model and parameter selection cannot be undermined. Research on estimating the desired states using deep learning models has been actively pursued recently. In this study, the state of external objects has been estimated using a constant velocity model-based Kalman filter (CVKF). The estimation accuracy was improved by using long short-term memory (LSTM) to predict and compensate for estimation errors caused by acceleration or rotational motion. The proposed model exhibited a reduction of 27.09% and 4.70% in the root mean square error (RMSE) compared to the extended Kalman filter (EKF) and IMMF, respectively, which were selected as comparison targets. The state estimation model offers a wide range of output distribution for position estimation, resulting in substantially decreased estimation performance in the tail part of the distribution. However, the error compensation model has the advantage of being robust to the estimated position range as it outputs the errors.

Keywords: Constant Velocity model-based Kalman Filter, long short-term memory, error compensation, state estimation

I. 서론

제어 시스템에서 상태 추정의 중요성은 매우 크다. 모든 상태 변수를 직접 측정할 수 없거나 노이즈와 외란이 섞여 측정값의 신뢰도가 낮은 경우, 상태 추정 기법을 통해 현재 상태를 정확하게 파악하여 효과적인 제어를 수행할 수 있다. 칼만 필터[1]는 이전 단계의 상태 추정값과 공분산을 활용하여 새로운 측정값을 얻음으로써, 보다 적은 연산 비용으로 상태 추정이 가능하다는 장점을 가지고 있다. 하지만 프로세스 노이즈와 측정 노이즈가 가우시안 분포를 따라야 하며, 대부분의 실제 시스템은 비선형적인 특성을 보이기 때문에 칼만 필터의 사용은 제한적이다. 이러한 한계를 극복하기 위해 EKF (Extended Kalman Filter), UKF (Unscented Kalman Filter) [2,3]가 제안되었다. EKF는 비선형 모델을 선형화하여 칼만 필터를 적용한 확장형이며, UKF는 비선형 함수를 시그마 포인트를 사용하여 직접 전파하여 선형화 과정의 근사화 오류를 줄인 상위 필터이다. 이러한 필터들은 항공기, 선박 그리고 로봇이나 자율 주행 차량의 여러 센서 데이터를 통합하여 정확한 위치 추정에 널리 사용된다[4].

칼만 필터를 활용한 객체의 상태 추정은 시스템 모델의 불확실성으로 인해 성능이 크게 달라질 수 있다. 사용된 시스템 모델의 정확도에 따라 칼만 필터의 성능이 크게 좌우되며,

필터에 사용된 모델과 다른 객체를 추정하는 경우 큰 오차가 발생할 수 있다. 적절하지 않은 프로세스 노이즈와 측정 노이즈의 공분산 행렬 선정으로 인한 오차 또한 중요한 문제이다. IMMF (Interacting Multiple Model Filter) [5]는 등속 운동 모델, 등가속도 운동 모델, 회전 운동 모델 등 다양한 동적 모델을 동시에 고려하여 객체의 상태를 추정한다. 따라서 모델의 불확실성이 높고, 객체의 운동 패턴이 빈번하게 변하는 문제에 유용하다. 하지만 잘못된 가중치, 공분산 행렬 등 매개변수 선정으로 인한 성능 저하 문제는 여전히 존재한다.

필터링 기법의 한계를 극복하기 위해 AI와 칼만 필터를 결합하는 연구가 진행되고 있다. 참고문헌 [6]은 RNN (Recurrent Neural Network)을 사용하여 CVKF (Constant Velocity model-based Kalman Filter)의 프로세스 노이즈 공분산을 조정하여 성능을 향상시켰고, 참고문헌 [7]는 LSTM (Long Short-Term Memory)을 사용하여 EKF의 프로세스 노이즈 공분산과 측정값 공분산을 조정하여 추정 성능을 향상시켰다. 매개변수 조정 방법 이외에 필터의 추정 오차를 예측하여 보상하는 방법이 있다. 오차를 예측하여 직접 보상하는 방법은 간단한 프레임워크로 다양한 응용 분야에 적용 가능하다는 장점이 있다[8]. 참고문헌 [9]는 UKF를 사용한 기동 목표물 추정에서 발생하는 모델의 부정확함으로 생기는 오

* Corresponding Author

Manuscript received August 16, 2024; revised September 23, 2024; accepted November 12, 2024

김소현: 중앙대학교 기계공학부 학부생(r1athgus0417@cau.ac.kr, ORCID^{id} 0009-0002-3031-8338)

최재훈: 중앙대학교 기계공학부 학부생(cj9681197@cau.ac.kr, ORCID^{id} 0009-0002-8897-7096)

정다연: 중앙대학교 기계공학부 학부생(wjdekudus224@cau.ac.kr, ORCID^{id} 0009-0006-3858-6352)

이석원: 중앙대학교 기계공학부 조교수(seokwonlee@cau.ac.kr, ORCID^{id} 0000-0001-9560-9262)

※ 본 연구는 정부(과학기술정보통신부)의 재원으로 한국연구재단의 지원을 받아 수행된 연구임 (No. RS-2023-00251551).

차를 AI 통해 예측하고 보상하여 추정 성능을 개선하였고, 참고문헌 [10]은 미세전자기계시스템(micro electro mechanical systems) 자이로스코프에서 발생하는 무작위 드리프트 오차를 AI를 사용하여 예측하고 보상함으로써 성능을 개선한다. 필터링 기법 외에 시계열 예측에 사용되는 기존의 대표적인 ARIMA (Auto Regressive Integrated Moving Average)[11]은 고차원 데이터와 비선형성을 처리하는데 한계가 있다. 이를 극복하기 위해 딥러닝 모델을 활용하는 연구가 진행되고 있다[12]. 참고문헌 [13]은 교통 흐름 예측에서 LSTM과 GRU (Gated Recurrent Unit) 모델을 사용하여 ARIMA 방법보다 더 낮은 오차를 통해 성능을 입증하였다.

칼만 필터를 사용하여 상태를 추정하기 위해서는 시스템 모델을 구성해야 하나, 외부 객체의 경우 시스템 모델과 프로세스 노이즈를 정확히 알기 어렵다는 문제가 있다. 이에 본 논문은 외부 객체의 시스템 모델을 등속도 모델과 가속도 노이즈로 구성된 프로세스 노이즈로 구성하고, 시스템 모델과 다른 움직임으로 인한 오차를 LSTM 예측으로 보상하여 추정 성능을 향상한다. 또한 측정치를 바탕으로 상태변수를 추정하는 모델과 비교하여 오차 보상 방식의 이점을 분석한다. 몬테카를로 시뮬레이션을 통해 IMM과 EKF와의 성능 비교를 진행함으로써, 제안된 모델이 기존의 주요 기법들을 효과적으로 대체할 수 있음을 보인다.

II. AI를 사용한 칼만 필터 오차 보상

1. 칼만 필터

칼만 필터의 두 가지 주요 과정으로는 예측 단계와 측정 업데이트 단계가 있다. 첫 번째 예측 단계는 이전 추정 상태 (\hat{x}_{k-1})와 오차 공분산(P_{k-1})을 사용하여 다음 시점에서의 상태 예측치(\hat{x}_k)와 오차 공분산 예측치(P_k)를 계산한다. 두 번째 단계인 측정 업데이트 단계에서는 상태 예측치(\hat{x}_k)와 측정치(z_k) 간의 차이를 이용하여 칼만 이득(K_k)을 계산한다. 칼만 이득은 예측의 불확실성과 측정의 불확실성 사이의 상대적 신뢰도를 나타내며, 이를 사용하여 상태 추정치와 오차 공분산을 업데이트한다.

예측 단계:

$$\begin{aligned}\hat{x}_k &= A\hat{x}_{k-1} \\ P_k &= AP_{k-1}A^T + Q_k\end{aligned}\quad (1)$$

측정 업데이트 단계:

$$\begin{aligned}K_k &= P_k H^T (H P_k H^T + R_k)^{-1} \\ \hat{x}_k &= \hat{x}_k + K_k (z_k - H \hat{x}_k) \\ P_k &= P_k - K_k H P_k\end{aligned}\quad (2)$$

2. 등속도 모델 기반 칼만 필터

상태 벡터(x, v_x, y, v_y)는 x 위치, x 속도, y 위치, y 속도이며, 객체가 등속도 운동을 한다고 가정한 CVKF의 시스템 모델(A)은 식 (5)와 같다. 측정치(x_{ob}, y_{ob})는 측정된 객체의 x, y 위치며 상태 벡터와의 관계(H)는 식 (7)과 같다. 프로세스 노이즈(w_k)와 측정 노이즈(v_k)는 각각 공분산 행렬 Q_k, R_k 로

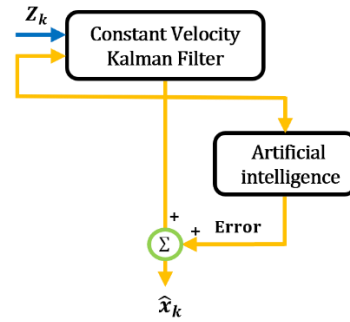


그림 1. 등속도 모델 기반 칼만 필터의 AI를 사용한 오차 보상 흐름도.

Fig. 1. Flow diagram of AI error compensation in constant velocity model-based Kalman filter.

이루어진 평균이 0인 가우시안 백색잡음이다. Q_k 는 시스템 모델의 불확실성, R_k 는 측정치의 불확실성을 나타내며 필터의 성능을 결정하는 중요한 매개변수이다.

$$\begin{aligned}x_{k+1} &= Ax_k + w_k \\ z_k &= Hx_k + v_k\end{aligned}\quad (3)$$

$$\text{state vector: } x_k = [x, v_x, y, v_y]^T \quad (4)$$

$$\text{state transition matrix: } A = \begin{bmatrix} 1 & dt & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & dt \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (5)$$

$$\text{measurement vector: } z_k = [x_{ob}, y_{ob}]^T \quad (6)$$

$$\text{measurement matrix: } H = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \quad (7)$$

$$\text{Process noise: } w_k \sim N(0, Q_k) \quad (8)$$

$$\text{Measurement noise: } v_k \sim N(0, R_k) \quad (9)$$

3. AI 기법을 사용한 CVKF 오차 보상

외부 객체를 추정하는 상황에서는 시스템 모델과 매개변수를 정확히 파악하기 어려운 문제가 있다. 이에 외부 객체의 시스템 모델을 등속도 모델과 가속도 노이즈로 구성된 프로세스 노이즈로 설정하여 상태를 추정하고 예측하기 어려운 가속 또는 회전 운동, 프로세스 노이즈 선정으로 인한 CVKF의 추정 오차는 AI를 사용하여 보상한다. 제안하는 모델의 흐름도는 그림 1에 나타내었다. CVKF에서 재귀적으로 사용되는 내부 상태 벡터는 오차 보상이 이루어지지 않으며, 출력 상태 벡터에 대해서만 AI가 예측한 오차 보상을 적용하여 최종 위치를 추정한다. CVKF의 추정 상태와 측정치를 바탕으로 CVKF의 오차가 증가하는 상황을 인지하도록 하기 위해 AI의 입력으로 CVKF의 추정 상태 벡터와 측정치를 사용한다.

III. 학습 데이터 생성 및 AI 설계

1. 기준 경로 데이터 생성

초기 위치 $x_0 = 0, y_0 = 0$ 에서 경로가 생성된다. 초기 속도는

$v_0 = 0 \sim 30m/s$ 사이 무작위 값을 가진다. 초기 각도 또한 $\theta_0 = 0 \sim 360^\circ$ 사이 무작위 값을 가진다. 가속도 (a_k) 는 $\pm(1 \sim 5)m/s^2$, 각가속도(ω_k) 는 $\pm(0.035 \sim 0.175)rad/s^2$ 범위의 무작위 값을 갖는다. 기준 경로(x_{GT} , Ground-truth path) 데이터 생성 로직은 식 (10)과 같다. 무작위로 결정된 가속도와 각가속도를 바탕으로 생성한 경로의 예시를 그림 2에 나타내었다. 그림 3은 곡선 주로에 진입하기에서 앞서 10% 확률로 $-20m/s^2$ 가속도로 급감속하고, 이후 $\pm(2 \sim 3.5)m/s^2$ 범위의 무작위 가속도로 가속하며 곡선주로를 진행하는 급감속 상황이 추가된 경로를 나타낸다.

$$x_{GT}: \begin{cases} x_{k+1} = x_k + v_k * dt * \cos(\theta_k) \\ y_{k+1} = y_k + v_k * dt * \sin(\theta_k) \\ \theta_{k+1} = \theta_k + \omega_k * dt \\ v_{k+1} = v_k + a_k * dt \\ \omega_{k+1} = \omega_k + \dot{\omega}_k * dt \end{cases} \quad (10)$$

2. 측정치 데이터 생성

기준 경로의 위치 데이터에 평균(μ) 0, 표준편차(σ) 3의 가우시안 백색잡음 ($\omega(z)$)를 더해 측정치를 생성한다.

$$z_k: \begin{cases} x_{Ob} = x_k + \omega(z) \\ y_{Ob} = y_k + \omega(z) \end{cases} \quad (11)$$

$$\omega(z) = \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{(z-\mu)^2}{2\sigma^2}} \quad (12)$$

3. CVKF 추정 상태 데이터 생성

Q_k, R_k 를 $\sigma_Q^2 = 10, \sigma_R^2 = 9, dt = 0.1$, 식 (13), (14)와 같이 설정하고 CVKF 추정 상태 벡터를 생성한다. 그림 4는 기준 경로, 측정치 그리고 CVKF 추정 경로를 나타낸다. 직선 경로 구간에 비해 곡선 구간에서 CVKF 추정 위치에 오차가 증가하는 것을 확인할 수 있다.

$$Q_k = \begin{bmatrix} dt^4/4 & dt^3/2 & 0 & 0 \\ dt^3/2 & dt^2 & 0 & 0 \\ 0 & 0 & dt^4/4 & dt^3/2 \\ 0 & 0 & dt^3/2 & dt^2 \end{bmatrix} \sigma_Q^2 \quad (13)$$

$$R_k = \begin{bmatrix} \sigma_R^2 & 0 \\ 0 & \sigma_R^2 \end{bmatrix} \quad (14)$$

4. 학습 데이터 생성 및 LSTM 모델 설계

오차 보상 AI 모델과 상태 추정 AI 모델의 입력 데이터는 시퀀스 길이에 해당하는 CVKF 추정 상태 벡터와 측정치로 동일하다. 오차 보상 모델의 경우 시퀀스 마지막 시점의 기준 경로의 위치와 CVKF 추정 위치의 오차를 출력하고, 상태 추정 모델의 경우 시퀀스 마지막 시점의 추정 위치를 출력한다. 시퀀스 길이는 LSTM 네트워크가 과거 정보를 얼마나 활용할 수 있는지를 결정하는 중요한 요소이다. 상태 추정 모델의 경우 시퀀스 길이에 측정치와 CVKF 추정치 데이터의 특성이 충분히 담겨야 하고, 오차 보상 모델의 경우 시퀀스

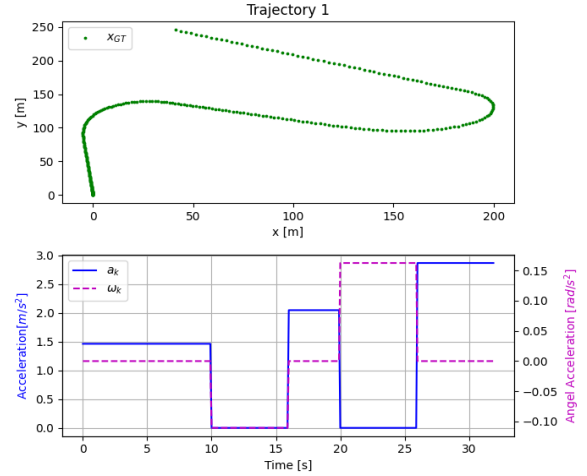


그림 2. 랜덤으로 생성된 경로 중 급감속 상황이 없는 경로.
Fig. 2. Randomly generated paths without sudden deceleration scenario.

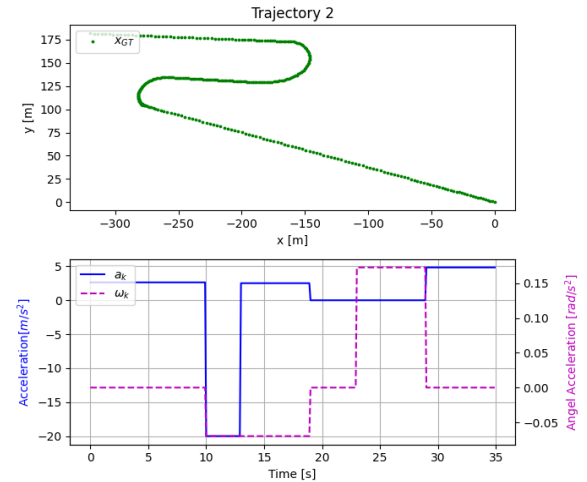


그림 3. 랜덤으로 생성된 경로 중 급감속 상황이 있는 경로.
Fig. 3. Randomly generated paths with sudden deceleration scenario.

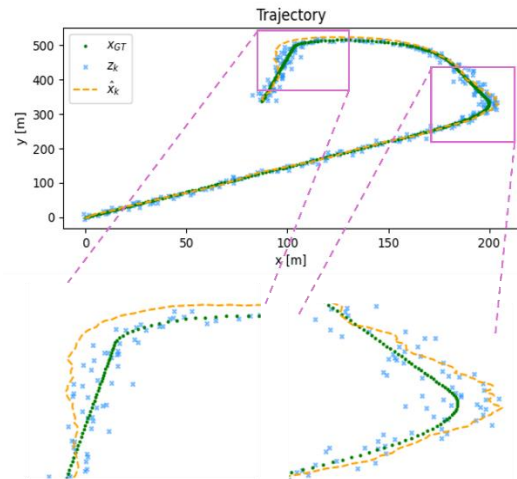


그림 4. 기준 경로, 측정치 그리고 CVKF 추정 경로.
Fig. 4. Ground-truth, observation and CVKF estimation path.

길이는 CVKF의 성능이 저하되는 곡선 혹은 감가속 구간에 있다는 정보 또한 담겨야 한다. 샘플링 주기 0.1초로 무작위로 선택된 가속도로 객체의 움직임을 기록하였을 때 14개의 시퀀스에 길이에 객체의 기동 변화의 정보가 충분히 담겨 시퀀스 길이를 14로 설정하였다.

생성된 하나의 경로를 시퀀스 길이만큼 자르고, 3개의 시점을 건너뛰어 다시 시퀀스 길이만큼 데이터를 자르는 슬라이딩 윈도우 방식을 사용한다. 이를 모든 경로에 반복 적용하여 학습 데이터를 생성한다. 이렇게 생성된 데이터들 사이에는 겹치는 시퀀스가 존재하기 때문에, 전체 10,000개의 경로를 먼저 7:2:1의 비율로 훈련, 검증, 테스트 세트로 분할한 후 각 세트에 슬라이딩 윈도우 방식을 적용하여 데이터 누수를 방지하였다. 해당 과정은 그림 5에 나타내었다. 테스트 데이터는 이후 몬테카를로 시뮬레이션을 통해 모델의 성능 평가에 사용된다. 표 1은 모델별 훈련 및 검증 출력 데이터의 평균과 표준편차를 정리한 표이다. 필터링된 데이터는 최소한 측정치 노이즈로 인한 오차보다 작아야 하므로, 모델의 출력 분포는 측정치 노이즈의 표준편차인 3이내의 분포를 추정할 정도의 정밀도가 필요하다. 표준편차가 2.84, 2.86인 분포를 예측하는 오차 보상 모델은 표준편차가 210.66, 213.75인 분포를 예측하는 상태 추정 모델보다 더욱 정밀한 예측이 가능하다.

LSTM은 순환 신경망의 한 종류로, 장기 의존성을 효과적으로 학습할 수 있는 모델이다. 과거의 정보를 장기간에 걸쳐 유지하거나 필요에 따라 업데이트할 수 있는 셀 상태라는 정보 흐름이 추가되었으며 입력, 삭제, 출력 게이트 구조를 도입하여 설계되었다. LSTM은 게이트에 사용되는 시그모이드 함수와 tanh 함수를 사용하여 비선형성을 도입하므로, 복잡한 비선형성을 처리할 수 있다. 본 연구에서는 비선형성을 효과적으로 처리할 수 있고, 시계열 데이터의 과거 정보를 바탕으로 현재 시점의 CVKF 오차를 예측하거나 상태를 추정할 수 있는 LSTM을 사용하는 것이 적합하다고 판단하였다. 그림 6은 본 논문에서 사용되는 오차 보상 LSTM 네트워크 구조도이다. Tensorflow 라이브러리를 사용하여 LSTM을 구현하였다. 첫 번째 LSTM 레이어에는 128개의 유닛을 사용하고 return sequences는 True로 설정하여 다음 LSTM 레이어로 시퀀스 데이터를 전달하였다. 이를 통해 다층 LSTM 구조를 형성하여 모델이 더 복잡한 패턴과 장기 의존성을 학습할 수 있게 한다. 두 번째 LSTM 레이어에는 32개의 유닛을 할당하여 모델의 복잡성을 조절하고, 과적합을 방지하였다. 두 번째 레이어는 return sequences를 False로 설정하여 시퀀스 마지막 시점의 출력만을 다음 레이어로 전달하였다. 출력 레이어인 Dense 레이어는 2개의 출력을 가지며 활성화 함수는 적용하지 않았다. 상태 추정 모델은 x, y 추정 위치(\hat{x}_e, \hat{y}_e), 오차 보상 모델의 경우 CVKF 추정 x, y 위치의 오차(\hat{x}_e, \hat{y}_e)를 출력한다. 두 개의 LSTM 레이어에는 L2 정규화를 적용하였으며, 정규화 강도는 0.001로 설정하여 모델의 일반화 성능을 향상하고 과적합을 방지하였다. 또한, 첫 번째 LSTM 레이어 뒤에 드롭아웃 레이어를 추가하여 무작위로 선택된 10%의 유닛을 비활성화하여 과적합을 방지하였다. 최적화는 Adam 옵티마이저를 사용하였고, 학습률은 0.001로 설정하였다. 평균

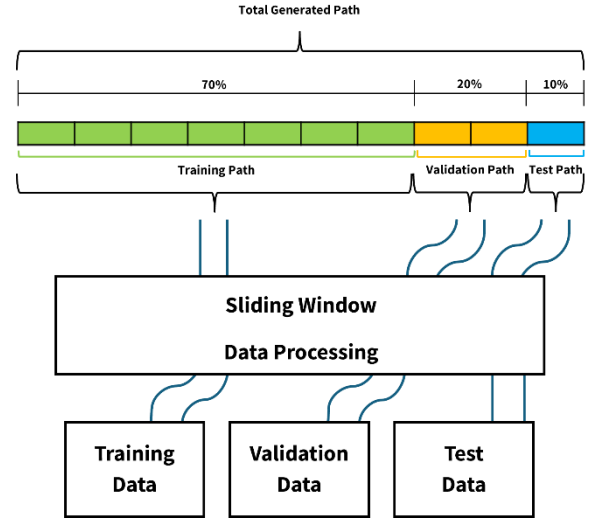


그림 5. 데이터 누수 방지를 위한 학습 데이터 생성 과정.
Fig. 5. The process of generating learning data to prevent data leakage.

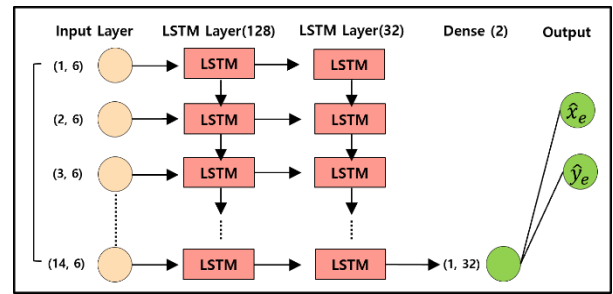


그림 6. 오차 보상 LSTM 네트워크 구조도.
Fig. 6. Schematic of error compensation LSTM network structure.

표 1. 훈련 출력 데이터의 모델별 평균과 표준편차.

Table 1. Mean and standard deviation of training output data by model.

Training Data	Model	Output X mean	Output Y mean
	오차 보상	-0.00126	-0.00468
	상태 추정	1.01	0.0437
	Model	Output X std	Output Y std
오차 보상	2.84	2.86	
상태 추정	210.66	213.75	

제곱오차(MSE, Mean Squared Error)를 손실 함수로 사용하고 훈련 데이터에 이상치가 많은 경우 학습이 불안정해지는 문제가 있다. 본 논문에서는 시뮬레이션 기반 경로 데이터를 사용하여 훈련을 진행하기 때문에 이상치가 존재하지 않는다. 따라서 회귀 문제에 널리 사용되는 평균제곱오차를 손실 함수로 학습을 진행하였다.

NVIDIA GeForce RTX 3050 ti 그래픽 카드로 배치 사이즈(batch size) 3072로 학습을 진행하였다. 오차 보상 모델과 상태 추정 모델은 동일한 입력 데이터, 다른 출력 데이터로 학습된다. 그림 7은 오차 보상, 상태 추정 모델의 학습 곡선이다. 검증

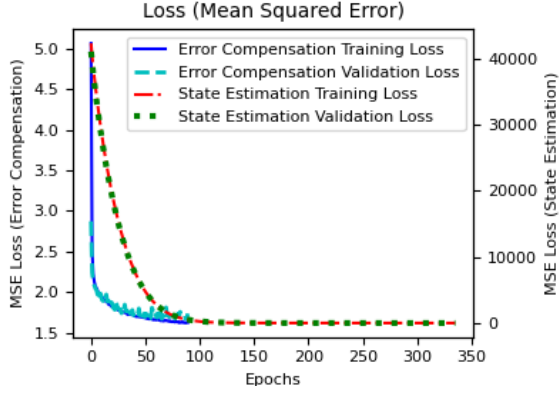


그림 7. 오차 보상, 상태 추정 모델 학습 곡선.

Fig. 7. Error compensation, state estimation model learning curve.

손실을 기준으로 15회 이상 개선되지 않을 경우 학습을 조기 종료하도록 설정하여 과적합을 방지하였다. 오차 보상 모델의 경우 89 에포크(epoch)에서 훈련 손실 1.62, 검증 손실 1.74로 학습이 종료되었으며, 상태 추정 모델의 경우 335 에포크에서 훈련 손실 5.77, 검증 손실 4.93으로 학습이 종료되었다. 오차 보상 모델은 2.84, 2.86의 표준편차를 갖는 출력 분포를 학습하나, 상태 추정 모델은 210.66, 213.75의 표준편차를 갖는 출력 분포를 학습한다. 이에 초기 훈련 손실이 5.07, 42310으로 크게 차이 난다.

IV. 시뮬레이션

1. EKF, IMM 선택

본 논문에서 제안하는 오차 보상 모델의 성능을 평가하기 위해 EKF, IMM을 비교 대상으로 선정하였다. EKF는 비선형 시스템의 상태 추정 문제에서 널리 사용되며, 선형화 과정을 통해 비선형 시스템을 근사하여 추정한다. 이는 물체의 동적 모델이 미리 알려진 경우 최적의 추정치를 제공하는 것으로 알려져 있다. 본 논문에서는 경로 생성에 사용된 운동 방정식을 기반으로 상태 천이 행렬을 식 (16)과 같이 선정하였다.

동적 시스템이 다양한 상태 변화 모델을 가질 수 있기 때문에 단일 시스템 모델 필터로 이를 효과적으로 처리하는데 한계가 있다. 반면 IMM은 다양한 동적 시스템 모델을 동시에 고려하고, 각 모델의 가중치를 조절하여 최종 상태를 추정한다. 이는 모델의 불확실성이 높고, 객체의 운동 패턴이 빈번하게 변하는 상황에 유용하다. 따라서 참고문헌 [14, 15]를 바탕으로 등속도 모델(CV, Constant Velocity model), 등가속도 모델(CA, Constant Acceleration model), 등속선회율 모델(CT, Constant Turn rate model), 세 가지 동적 시스템 모델을 바탕으로 IMM을 설계해 비교군으로 선정하였다. 각 필터에 사용된 상태변수와 시스템 모델 그리고 시스템 노이즈의 공분산을 식 (18) ~ (26)에 나타내었다. σ_Q^2 는 10, dt 는 0.1로 모든 필터가 동일하고, 측정 노이즈의 공분산 또한 모든 필터가 식 (14)와 동일하다.

EKF:

$$\text{State vector: } x_k = [x, y, \theta, v, \omega]^T \quad (15)$$

$$\begin{aligned} x_{k+1} &= x_k + v_k * dt * \cos(\theta_k) \\ y_{k+1} &= y_k + v_k * dt * \sin(\theta_k) \\ \theta_{k+1} &= \theta_k + \omega_k * dt \\ v_{k+1} &= v_k \\ \omega_{k+1} &= \omega_k \end{aligned} \quad (16)$$

$$Q_k: \begin{bmatrix} dt^4/4 & 0 & 0 & dt^3/2 & 0 \\ 0 & dt^4/4 & 0 & dt^3/2 & 0 \\ 0 & 0 & dt^4/4 & 0 & dt^3/2 \\ dt^3/2 & dt^3/2 & 0 & dt^2 & 0 \\ 0 & 0 & dt^3/2 & 0 & dt^2 \end{bmatrix} \sigma_Q^2 \quad (17)$$

CV Filter:

$$\text{State vector: } x_{CV} = [x, v_x, y, v_y]^T \quad (18)$$

$$A_{CV}: \begin{bmatrix} 1 & dt & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & dt \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (19)$$

$$Q_k: \begin{bmatrix} dt^4/4 & dt^3/2 & 0 & 0 \\ dt^3/2 & dt^2 & 0 & 0 \\ 0 & 0 & dt^4/4 & dt^3/2 \\ 0 & 0 & dt^3/2 & dt^2 \end{bmatrix} \sigma_Q^2 \quad (20)$$

CA Filter:

$$\text{State vector: } x_{CA} = [x, v_x, a_x, y, v_y, a_y]^T \quad (21)$$

$$A_{CA}: \begin{bmatrix} 1 & dt & dt^2/2 & 0 & 0 & 0 \\ 0 & 1 & dt & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & dt & dt^2/2 \\ 0 & 0 & 0 & 0 & 1 & dt \\ 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix} \quad (22)$$

$$Q_k: \begin{bmatrix} dt^5/20 & dt^4/8 & dt^3/6 & 0 & 0 & 0 \\ dt^4/8 & dt^3/3 & dt^2/2 & 0 & 0 & 0 \\ dt^3/6 & dt^2/2 & dt & 0 & 0 & 0 \\ 0 & 0 & 0 & dt^5/20 & dt^4/8 & dt^3/6 \\ 0 & 0 & 0 & dt^4/8 & dt^3/3 & dt^2/2 \\ 0 & 0 & 0 & dt^3/6 & dt^2/2 & dt \end{bmatrix} \sigma_Q^2 \quad (23)$$

CT Filter:

$$\text{State vector: } x_{CT} = [x, v_x, y, v_y, \omega]^T \quad (24)$$

$$A_{CT}: \begin{bmatrix} 1 & \sin(\omega * dt) & 0 & -\frac{1-\cos(\omega * dt)}{\omega} & 0 \\ 0 & \cos(\omega * dt) & 0 & \frac{\sin(\omega * dt)}{\omega} & 0 \\ 0 & \frac{1-\cos(\omega * dt)}{\omega} & 1 & \frac{\sin(\omega * dt)}{\omega} & 0 \\ 0 & \sin(\omega * dt) & 0 & \cos(\omega * dt) & 0 \\ 0 & 0 & 0 & 0 & 1 \end{bmatrix} \quad (25)$$

$$Q_k: \begin{bmatrix} dt^3/3 & dt^2/2 & 0 & 0 & 0 \\ dt^2/2 & dt & 0 & 0 & 0 \\ 0 & 0 & dt^3/3 & dt^2/2 & 0 \\ 0 & 0 & dt^2/2 & dt & 0 \\ 0 & 0 & 0 & 0 & dt \end{bmatrix} \sigma_Q^2 \quad (26)$$

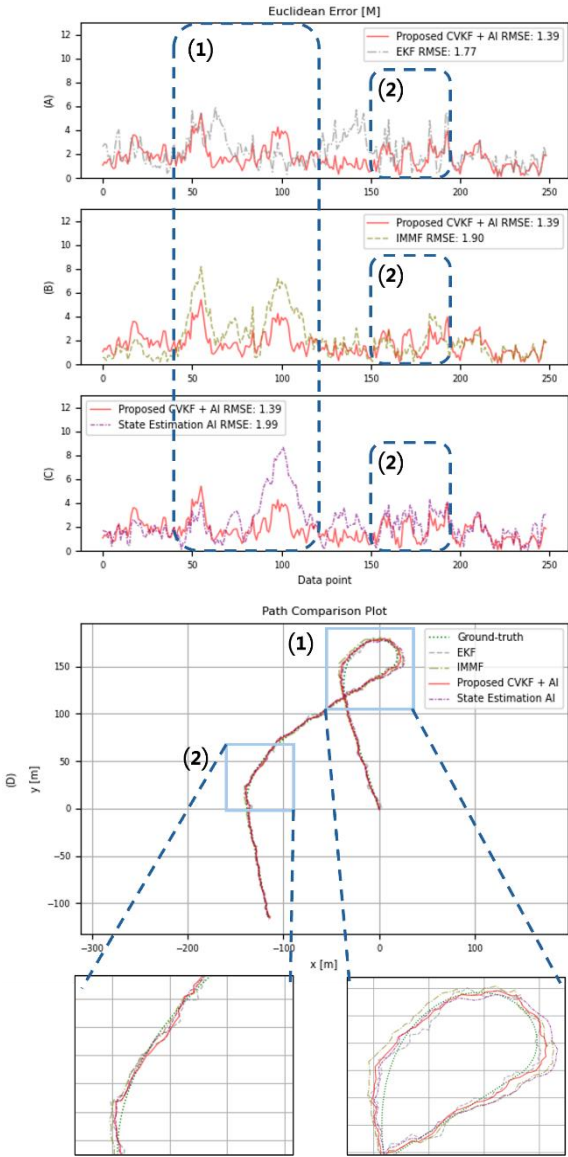


그림 8. 무작위로 생성된 한 경로의 추정 결과.
 Fig. 8. Estimation results for a randomly generated path.

2. 시뮬레이션 결과 분석

89 에포크 동안 학습된 오차 보상 모델(CVKF + AI)과 335 에포크 동안 학습된 상태 추정 모델(State Estimation AI)의 성능 평가를 위해 테스트 경로 중 하나를 선택하여 시뮬레이션을 진행하였다. 그림 8은 무작위로 생성된 경로 중 급감속 상황이 없는 경우를, 그림 9는 급감속 상황이 포함된 경로를 나타낸다. 가시성을 높이기 위해 측정치는 제외하고 기준 경로를 기준으로 EKF, IMMf, 상태 추정 모델, 오차 보상 모델의 추정 경로를 D에 나타내었다. 기준 경로와 필터들이 추정한 경로 사이 유클리드 거리 오차를 (A), (B), (C)에 나타내었으며, 표 2에는 위치 RMSE (Root Mean Square Error)를 정리하였다. 그림 8의 (1), (2) 구간은 직선 경로 이후 -0.148 rad/s^2 , 0.08 rad/s^2 의 각가속도로 곡선 주로를 진행하는 구간이다. 구간 (1)에서 IMMf의 오차는 8까지 상승하지만, EKF와 오차 보상 모델은 오차를 6 이하로 유지한다. 상태 추정 모델은 곡선 주로 초반에는 추정 위치 오차를 5 이하로 유지하지만,

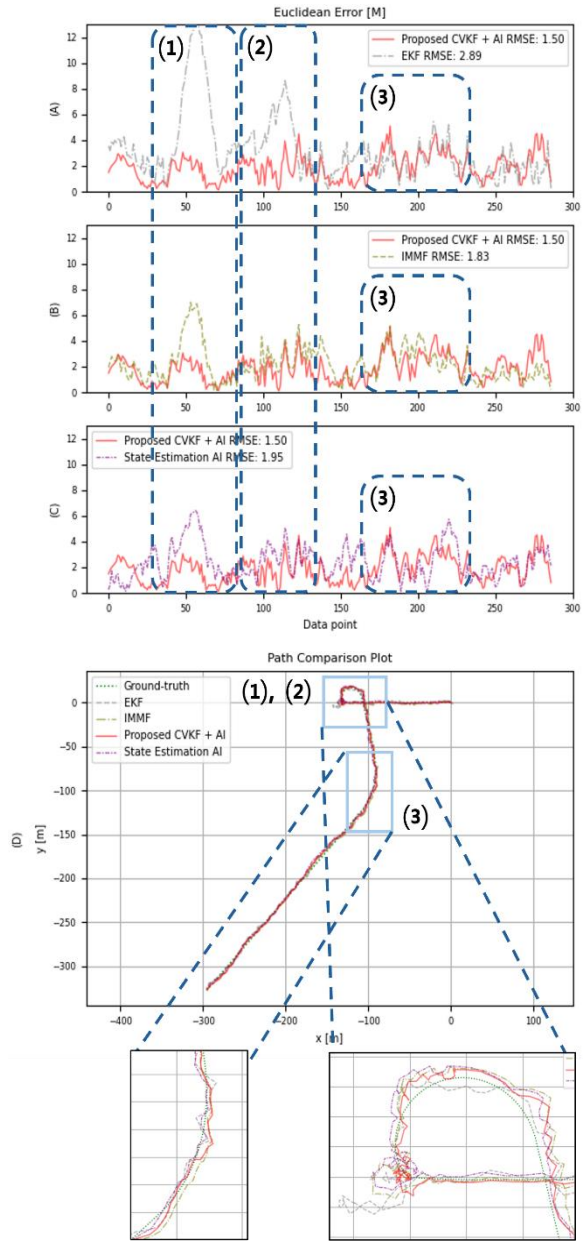


그림 9. 무작위로 생성된 경로 중 급감속 경로 추정 결과.
 Fig. 9. Estimation of random sudden deceleration path.

표 2. 모델별 추정 위치 RMSE 결과.

Table 2. Estimation position RMSE results for each model.

Model	EKF	IMMF	State Estimation AI	CVKF + AI
Fig. 8	1.77	1.90	1.99	1.39
Fig. 9	2.89	1.83	1.95	1.50

각속도가 큰 곡선 주로 후반에서 오차가 8 이상으로 증가한다. 구간 (2)는 낮은 각가속도와 짧은 조향 시간으로 인해 완만한 곡선 주로를 진행하여 모든 모델의 오차가 6 이하로 유지된다. 오차 보상 모델의 위치 RMSE는 1.39로 EKF: 1.77, IMMf: 1.90, 상태 추정 모델: 1.99와 비교했을 때 가장 낮은 값을 보인다. 그림 9의 구간 (1)은 곡선 주로에 진입하기 전 급감속하는 구간이다. 급감속으로 인해 구간 (1)에서 EKF 오

차는 12 이상으로 크게 증가하고, IMM의 오차는 6.9까지 상승한다. 상태 추정 모델의 오차는 6.4로 IMM과 비슷한 성능을 보인다. 오차 보상 모델의 오차는 3.05로 급감속 상황에서 가장 안정적으로 위치를 추정함을 확인할 수 있다. 그림 9의 구간 (2)는 급감속 이후 다시 가속하여 곡선 구간을 주행하는 구간이다. EKF를 제외한 다른 모델의 경우 오차가 5.4 이하로 유지되나, EKF의 경우 오차가 8.65까지 상승한다. 구간 (2)의 특징인 급감속 이후 다시 가속하며 곡선 경로를 추정하는 데 EKF가 취약한 것을 확인할 수 있다. 그림 9의 구간 (3)은 -0.105 rad/s^2 의 각가속도로 곡선 경로를 진행하고 직선 경로로 이어지는 구간이다. 그림 8의 구간 (2)와 그림 9의 구간 (3)을 보면 완전한 곡선 경로를 추정하는 상황에서 오차 보상 모델의 성능이 기존 필터와 비슷한 것을 확인할 수 있다. EKF와 IMM의 추정 경로가 오차 보상 모델보다 우수한 구간도 일부 존재하지만, 경로 전체에 대한 위치 RMSE를 비교할 때 EKF - (1.77, 2.89), IMM - (1.90, 1.83), 상태 추정 모델 - (1.99, 1.95) 그리고 오차 보상 모델 - (1.39, 1.50)로 나타나 논문에서 제안하는 오차 보상 모델이 가장 우수한 성능을 보였다.

위치를 추정하는 경우 외부 객체의 움직임이 한 방향으로 진행되어, 추정해야 할 위치의 범위가 넓어질 수 있다. 그림 10은 한 방향으로 진행된 경로의 각 모델의 추정한 경로이다. 구간 (1)을 보면 객체는 x 위치가 -400을 넘어 진행된다. -400은 출력 데이터 분포의 분위수 하위 2.31%에 위치하는 값으로 출력 분포의 꼬리 부분에 해당한다. 객체의 위치를 추정하는 상황에서 학습한 출력 분포의 꼬리 혹은 넘어가는 위치를 추정하는 상황이 생기게 되고 이때 상태 추정 모델의 추정 위치의 오차가 크게 증가한다. 반면에 오차 보상 모델의 경우 CVKF의 오차를 학습하고 예측하여 보상하기 때문에, 추정 위치의 범위가 넓은 상황에서도 안정적으로 동작함을 그림 10의 구간 (1)을 통해 확인할 수 있다.

오차 보상 모델의 경우 89 에포크, 상태 추정 모델의 경우 335 에포크에서 학습이 종료되었다. 그림 11은 두 모델의 학습이 약 33% 진행된 30, 110 에포크에서 학습된 모델을 사용하여 그림 9의 경로를 추정한 결과이다. 그림 11의 (A)는 오차 보상 모델이 89 에포크와 30 에포크 동안 학습된 모델의 유클리디안 거리 오차를 나타내며, (B)는 상태 추정 모델이 335 에포크와 110 에포크 동안 학습된 모델의 유클리디안 거리 오차를 보여준다. (C)는 네 가지 모델이 추정한 경로를 기준 경로와 비교한 그림이다. 오차 보상 모델의 89 에포크 모델 RMSE는 1.46, 30 에포크 모델 RMSE는 1.64로 89 에포크 모델이 약 12% 더 성능이 우수하다. 반면, 상태 추정 모델의 335 에포크 모델 RMSE는 2.09, 110 에포크 모델 RMSE는 12.17로 5배 이상의 차이를 보인다. 그림 10의 상태 추정 335 에포크 모델은 추정 위치가 -400을 넘어가는 상황에서 오차가 크게 증가했지만, 그림 11의 구간 (1)에서 상태 추정 110 에포크 모델은 -230을 넘어가는 위치에서 오차가 크게 상승하였다. 상태 추정 모델은 학습이 진행될수록 출력 분포의 꼬리 부분을 학습하며, 출력 분포가 큰 상태 추정 모델은 출력 분포가 작은 오차 보상 모델에 비해 더 많은 에포크 수가 필요하게 된다.

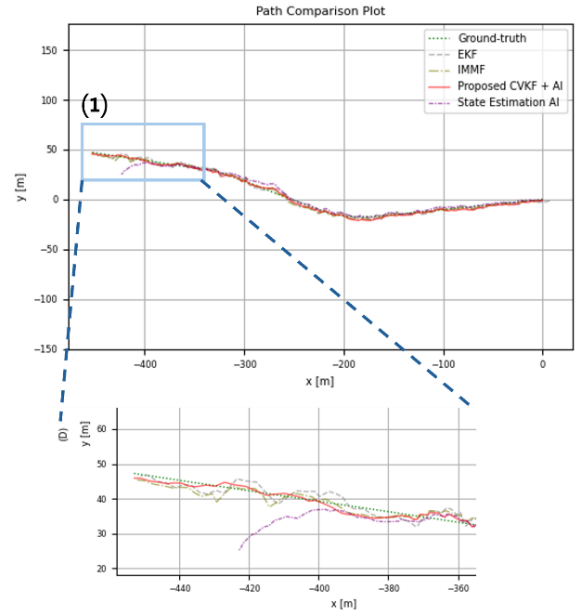


그림 10. 한 방향으로 진행된 경로의 각 모델의 추정 결과.
Fig. 10. Estimation results of each model along the trajectory in a single direction.

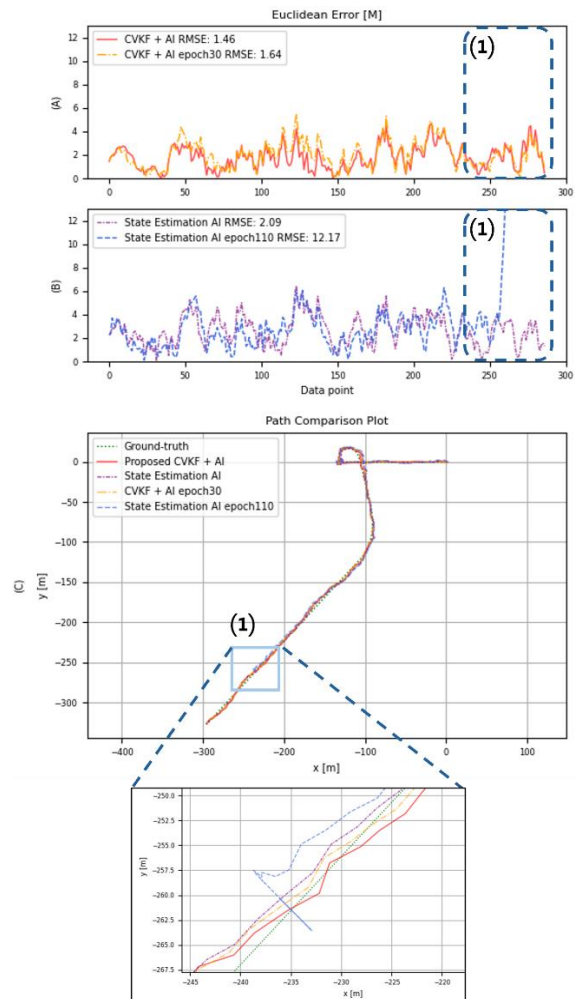


그림 11. 모델 에포크 별 급감속 경로 추정 결과.
Fig. 11. Sudden deceleration path estimation results by model epoch.

표 3. 테스트 경로 RMSE 평균, 표준편차 결과.

Table 3. Test path RMSE mean and standard deviation results.

Model	EKF	IMMF	State Estimation AI	CVKF + AI
Average RMSE	1.93	1.48	2.06	1.41
std	0.35	0.17	0.5	0.14

표 4. 기준 경로 범위 제한 경로 RMSE 평균, 표준편차 결과.

Table 4. Ground-truth path range-constrained path RMSE mean and standard deviation results.

Model	State Estimation AI	CVKF + AI
Average RMSE	1.97	1.35
std	0.34	0.12

3. 몬테카를로 시뮬레이션

몬테카를로 시뮬레이션을 통해 본 논문에서 제안하는 오차 보상 모델의 우수성을 검증하였다. 1000개의 테스트 경로를 바탕으로 RMSE 평균과 표준편차(std, standard deviation)를 표 3에 정리하였다. 오차 보상 모델은 EKF 대비 27.09%, IMMF 대비 4.70%의 성능 향상을 보였다. 또한 표준편차는 0.14로, EKF: 0.35, IMMF: 0.17, 상태 추정 모델: 0.50에 비해 낮은 값을 보여 다양한 조건에서도 안정적이고 일관된 성능을 유지함을 의미한다. 반면, 상태 추정 모델은 EKF 대비 6.50%, 그리고 IMMF 대비 39.12% 성능 저하를 보였다. 표준편차는 0.50로 나타났는데, 이는 그림 10의 학습된 출력 분포 범위를 넘어서는 위치를 추정하는 상황에서 오차가 크게 증가한 것으로 분석된다.

상태 추정 모델의 출력 분포를 넘어서는 출력으로 인해 추정 성능이 감소하는 경우를 제외하고 모델의 성능을 비교하기 위해, 테스트 경로 중 기준 경로의 범위가 학습 출력 분포의 표준편차인 -210~210 사이에 존재하는 100개를 바탕으로 평균 RMSE와 표준편차를 표 4에 나타내었다. 평균 RMSE는 1.97로 오차 보상 모델의 1.34 보다 크게 나타났으며, 표준편차 또한 0.34로 0.12에 비해 크게 나타나 오차 보상 모델의 경로 추정 성능 상태 추정 모델에 비해 뛰어난 것을 보여준다. 이는 출력 분포의 차이로 인해 정밀한 예측이 가능해져 추정 정확도의 차이가 발생한 것으로 분석된다.

V. 결론

본 논문은 외부 객체의 상태 추정 정확도 향상을 위해 연구되었다. CVKF를 사용하여 외부 객체의 상태를 추정하고, 모델과 파라미터의 부정확함으로 인한 오차는 LSTM을 사용하여 보상하였다. 평균 위치 RMSE가 EKF 대비 27.09%, IMMF 대비 4.70% 감소하였고, RMSE의 표준편차는 0.14로 EKF와 IMMF에 비해 안정적이고 일관된 성능을 보였다. 또한, 넓은 범위의 위치를 추정할 때 상태 추정 모델과 달리 오차 보상 모델은 추정 위치 범위에 제한받지 않고 안정적으로 동작함을 확인하였다. 기준 경로의 x, y 위치 범위가 -

210~210 사이에 존재하는 100개의 경로를 선정해 학습된 출력 범위를 넘는 추정을 하는 경우를 제외하여 몬테카를로 시뮬레이션을 진행하였다. 이를 통해 작은 분포를 출력하는 오차 보상 모델이 상태 추정 모델에 비해 정밀한 출력으로 추정 정확도를 향상시킬 수 있음을 확인하였다. 다만 실제 실험을 통한 검증이 이루어져야 한다는 점과 오차 보상 모델의 출력 분포가 커지는 경우 상태 추정 모델의 출력 분포 범위에 따른 문제가 발생할 수 있다는 점은 극복해야 할 점으로 남는다.

REFERENCES

- [1] R. E. Kalman, "A new approach to linear filtering and prediction problems," *Journal of Basic Engineering*, vol. 82, no. 1, pp. 35-45, 1960.
doi: <https://doi.org/10.1115/1.3662552>
- [2] S. J. Julier and J. K. Uhlmann, "New extension of the Kalman filter to nonlinear systems," *Signal Processing, Sensor Fusion, and Target Recognition VI*, vol. 3068, edited by I. Kadar, International Society for Optics and Photonics, SPIE, pp. 182-193, 1997.
doi: <https://doi.org/10.1117/12.280797>
- [3] E. A. Wan and R. Van Der Merwe, "The unscented Kalman filter for nonlinear estimation," *Proceedings of the IEEE 2000 Adaptive Systems for Signal Processing, Communications, and Control Symposium*, Lake Louise, AB, Canada, pp. 153-158, 2000.
doi: <https://doi.org/10.1109/asspcc.2000.882463>
- [4] D. B. Jeong, N. Y. Ko, and H.-T. Choi, "Vessel location tracking by using unscented Kalman filter implemented in nonlinear attitude space," *Journal of Institute of Control, Robotics and Systems (in Korean)*, vol. 29, no. 4, pp. 347-354, 2023.
doi: <http://dx.doi.org/10.5302/J.ICROS.2023.22.8007>
- [5] H. A. P. Blom, "An efficient filter for abruptly changing systems," *The 23rd IEEE Conference on Decision and Control*, Las Vegas, NV, USA, pp. 656-658, 1984.
doi: <https://doi.org/10.1109/cdc.1984.272089>
- [6] S. Jouaber, S. Bonnabel, S. Velasco-Forero, and M. Pilté, "NNAKF: A neural network adapted Kalman filter for target tracking," *2021 IEEE International Conference on Acoustics, Speech and Signal Processing*, Toronto, ON, Canada, pp. 4075-4079, 2021.
doi: <https://doi.org/10.1109/icassp39728.2021.9414681>
- [7] H.-J. Yang and S. M. Seong, "Extended Kalman filter covariance tuning using LSTM for navigation system," *Journal of Institute of Control, Robotics and Systems (in Korean)*, vol. 28, no. 2, pp. 130-137, 2022.
doi: <http://dx.doi.org/10.5302/J.ICROS.2022.21.0192>
- [8] S. Kim, I. Petrunin, and H. -S. Shin, "A review of Kalman filter with artificial intelligence techniques," *2022 Integrated Communication, Navigation and Surveillance Conference*, Dulles, VA, USA, pp. 1-12, 2022.
doi: <https://doi.org/10.1109/icns4818.2022.9771520>
- [9] Z. -J. Yu, S. -L. Dong, J. -M. Wei, T. Xing, and H. -T. Liu, "Neural network aided unscented Kalman Filter for maneuvering target tracking in distributed acoustic sensor networks," *2007 International Conference on Computing: Theory and Applications*,

Kolkata, India, pp. 245-249, 2007.

doi: <https://doi.org/10.1109/iccta.2007.88>

- [10] D. Li, J. Zhou, and Y. Liu, "Recurrent-neural-network-based unscented Kalman filter for estimating and compensating the random drift of MEMS gyroscopes in real time," *Mechanical Systems and Signal Processing*, vol. 147, p. 107057, 2021.
doi: <https://doi.org/10.1016/j.ymssp.2020.107057>
- [11] C. Liu, S. C. Hoi, P. Zhao, and J. Sun, "Online ARIMA algorithms for time series prediction," *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 30, no. 1, Feb. 2016.
doi: <https://doi.org/10.1609/aaai.v30i1.10257>
- [12] L. N. N. Do, N. Taherifar, and H. L. Vu, "Survey of neural network-based models for short-term traffic state prediction," *WIREs Data Mining and Knowledge Discovery*, vol. 9, no. 1, p. e1285, 2019.
doi: <https://doi.org/10.1002/widm.1285>
- [13] R. Fu, Z. Zhang, and L. Li, "Using LSTM and GRU neural network methods for traffic flow prediction," *2016 31st Youth Academic Annual Conference of Chinese Association of Automation*, Wuhan, China, pp. 324-328, 2016.
doi: <https://doi.org/10.1109/yac.2016.7804912>
- [14] A. Khan and D. A. Iqbal Bhatti, "Comparison between Kalman Filter and interacting multiple model using 2-D trajectories," *2019 International Conference on Electrical, Communication, and Computer Engineering*, Swat, Pakistan, pp. 1-6, 2019.
doi: <https://doi.org/10.1109/ICECCE47252.2019.8940755>
- [15] M. Efe and D. P. Atherton, "Maneuvering target tracking using adaptive turn rate models in the interacting multiple model algorithm," *Proceedings of 35th IEEE Conference on Decision and Control*, Kobe, Japan, vol. 3, pp. 3151-3156, 1996.
doi: <https://doi.org/10.1109/cdc.1996.573613>



김 소 현

2019년~현재 중앙대학교 기계공학부 재학 중. 관심분야는 자율 및 지능형 시스템, 항법 및 가이드نس 필터, 객체 추적, 인공지능.



최 재 훈

2018년~현재 중앙대학교 기계공학부 재학 중. 관심분야는 로보틱스, 자율주행, 인공지능.



정 다 연

2021년~현재 중앙대학교 기계공학부 재학 중. 관심분야는 로보틱스, 인공지능, 자율주행.



이 석 원

2023년~현재 중앙대학교 기계공학부 조교수로 재직 중. 관심 분야는, 무인 항공 시스템 유도제어, 비선형 및 적응 제어 응용, 의도 추론 및 상태추정, 학습 기반 제어.