Al-Furat Journal of Innovations in Electronics and Computer Engineering (FJIECE)
ISSN -2708-3985

Neural-Aided State Estimation for Nonlinear Vehicle Tracking using KalmanNet and GRU

Sakina Hassan Saeed [1], Mohanad Al-Ibadi [1]

[1] Technical Engineering College, Al-Furat Al-Awsat Technical University, Najaf 540001, Iraq
Email: sakenahass2000@gmail.com
Email:mohanad.alibadi@atu.edu.iq

Article's Information

Received: 19.01.2026
Accepted: 13.02.2026
Published: 31.03.2026

Keywords:
KalmanNet
Gated Recurrent Unit (GRU)
Autonomous Vehicle Tracking
Nonlinear State Estimation
Deep Learning

Abstract

Accurate tracking of vehicles is a key problem in autonomous driving, especially when the maneuvers are complex and/or under high noise sensing measurements. Traditional model-based estimators, such as the EKF, can easily be affected by linearization errors and are dependent on precise mathematical modelling and carefully calibrated noise characteristics. In this paper, we propose an enhanced learning-aided state estimator called KalmanNet GRU that maintains the recursive property of Kalman filtering and introduces a neural dynamics learner based on a GRU (gated recurrent unit) cell and a learnable Kalman gain in order to achieve robustness against nonlinearities and measurement noises. The performance of the proposed approach is demonstrated via the HDVT problem with nonlinear range–bearing radar measurements at an inverse sensing noise level of 20 dB. The experimental results reveal that KalmanNet significantly outperforms the EKF as it can attain an average test MSE of −7.91 dB in contrast to 9.85 dB resulted by the EKF, which means achieving nearly 17.76 dB improvement in performance. The model proposed also has a stable estimation performance and yet holds an acceptable processing time for real-time autonomous navigation. These results verify the utility of recursive filtering by data-driven dynamics learning for nonlinear state estimation in ITS.

https://doi.org/10.46649/fjiece.v5.1.6a.31.3.2026
*Corresponding author: sakenahass2000@gmail.com

1. INTRODUCTION

State estimation is a fundamental problem in many engineering fields, such as autonomous driving, radar tracking, navigation systems and wireless networks. In such applications, system and observation dynamics are often non-linear systems affected by process and measurement noise, which makes it difficult to obtain reliable state estimates. The classical model-based estimation methods, like the Kalman Filter (KF) and its nonlinear counterpart, have enjoyed wide applications due to their recursive nature and relatively low computational load.

The EKF is one of the most popular and commonly used techniques for dealing with nonlinear systems. The EKF is founded on the first-order Taylor series approximation of the non-linear motion and observation models around a current state estimate. This approach would be successful for systems that are weakly nonlinear, however, the performance is very sensitive to the accuracy of the system model and tuned noise covariance matrices. For the strongly nonlinear case, or if there is model uncertainty in models exist potential linearization error collecting errors later, accuracy of estimation will be reduced and sometimes filters will diverge [1], [2].

In recent years, data-driven techniques based on deep learning which offer powerful alternatives to classical filtering methods have attracted extensive interest RNN such as LSTM also demonstrate strong capability of model- ing temporal dynamics and nonlinearities given data. However, purely data-driven approaches might lack interpretability and require large amounts of training data in order to achieve good generalization performance, particularly in safety-critical applications [3,4,27]

To address these deficiencies, there has been a recent push towards developing hybrid state estimation methods that could leverage the robustness of physics-based theory with the high flexibility offered by modern deep learning [5]. One of the most popular architectures is KalmanNet, which structures the conventional Kalman Filter preserved but replaced its sensitive parts by customized neural networks [6]. Unlike the traditional approaches, which are underpinned by deterministic transition models with fixed mathematical functions, the Dynamics Learner that forms our Predictive Dynamics is defined using a GRU (Gated Recurrent Unit). The choice arises because complex maneuvers and/or high steering turn angles cannot be defined by static equation [7]. When historical trajectory is given the GRU learns layers of compound dynamics and a local 'Neural Predictor' can estimate the next state Qi more adaptively better than 1st Order Taylor expansion [8,28]

Another key point in nonlinear filtering is the computation of the Kalman gain (K This is achieved by tuning the amount of model estimation and sensor observation noise. This gain is solved in the classical EKF by solving very difficult Riccati equations (which require a priori knowledge of the statistics of noise Which are often time-variant or simultaneously undetermined R adar tracking [9]). We side-step this issue by modeling the Kalman Gain as a learnable, data-dependent parameter. The network learns to optimize this gain to act as an ``Intelligent Balance'' through backpropagation and minimizing the Mean squared-error (MSE). It adaptively attaches weight to the innovation, i.e., the value between radar measurement and GRU prediction, where despite of sharp turning or sensor blackout happening often, the final state updating is almost optimal [10].

By plugging these neural blocks into a recursive filter loop, we propose a consistent method to track targets when the measurements are range-bearing ones models of nonlinearities. It is not a "black-box" estimator, and thus preserves the interpretable structure of the Kalman but uses deep learning to learn obvious uncertainties [11]. This hybrid approach is particularly promising once the dynamics of the system are not ideal and shows how (real-time) classical estimation theory can be complemented by modern artificial intelligence in a scalable way. [12]

In this paper , we make the robustness of keep it either \non-linears or \nonlinear in enterprises. state estimation by introducing a new learning paradigm. Specifically, Transforming the old-fashioned linear state-transition models with GRU-based neural dynamics learner allows the system to handle complex motion trajectories and maneuvers that does not have a hope in being adequately described by simple mathematical formulae. For further improving the estimated result, we use a trainable Adaptive Kalman Gain built with supervised learning in order to avoid an analytical noise covariance estimation and to achieve a novel balance between data driven neural weather forecasts and radar measurements. For this

Al-Furat Journal of Innovations in Electronics and Computer Engineering (FJIECE)
ISSN -2708-3985

purpose, we design a hybrid recursive pipeline that simultaneously processes non-linear observation models—mapping state estimates from planar to range and bearing coordinates—with deep learning structures so that physical constraints and tracking performance are assured. At last, extensive empirical comparisons demonstrate that the proposed model achieves better performance on tracking accuracy, computational robustness and capability of adapting to complex maneuvering patterns than conventional EKF. The remaining part of this paper is organized as follows.Section II introductionsthe system model and the discrete-time state-space representationof the target dynamics. Section III reviews the EKF 3 as baseline for nonlinear state estimation Section IV describes the proposed KalmanNet framework emphasizing on its efficient combination of the GRU-based dynamics learner with adaptive kalman gain. In Section V, we present our results, including discussion of training convergence and empirical performance. Section VI concludes the paper and lists the future work.

2. SYSTEM MODEL

2.1. STATE-SPACE REPRESENTATION

The nonlinear dynamic of the target is assumed to be a discrete-time state-space model. The system state vector at the t^{th}instant of time is configured as:

$$x_t = [x_t\ y_t\ v_{x,t}\ v_{y,t}]^T \qquad (1)$$

where $x_t$and $y_t$denote the Cartesian position of the target, while $v_{x,t}$ and $v_{y,t}$ represent the corresponding velocity components along the x- and y-axes.
The state evolution is governed by a nonlinear motion model:

$$x_{\ t} = f(x_{t-1}) + w_t \qquad (2)$$

$$F(x_{t-1}) = \begin{bmatrix} p_{x,t-1} + V_{x,t-1}\Delta t \\ p_{y,t-1} + V_{y,t-1}\Delta t \\ (V_{t-1} + a\Delta t)\cos(\psi_{t-1} + \dot{\psi}_{t-1}\Delta t) \\ (V_{t-1} + a\Delta t)\, sin(\psi_{t-1} + \dot{\psi}_{t-1}\Delta t) \end{bmatrix} \qquad (3)$$

where $f(\cdot)$ denotes the nonlinear state transition function describing the vehicle dynamics, and $w_t \sim N(0, Q)$ represents zero-mean Gaussian process noise with covariance matrix Q.
Such nonlinear motion models are commonly used to represent realistic vehicle kinematics, including acceleration and turning behaviour [13,14].

2.2. Nonlinear Observation Model

The measurement process is modeled using a nonlinear radar-like observation model that provides range and bearing measurements. The observation vector is defined as:

$$y_t = \begin{bmatrix} \rho_t \\ \phi_t \end{bmatrix} = h(x_t) + v_t \qquad (4)$$

$$h(x_t) = \begin{bmatrix} \sqrt{p_x^2 + p_y^2} \\ \text{atan2}(p_y, p_x) \end{bmatrix} \quad (5)$$

where the nonlinear observation function $h(\cdot)$ is given by:

$$\rho_t = \sqrt{x_t^2 + y_t^2} \quad (6)$$

$$\phi_t = \text{atan}\left(\frac{y_t}{x_t}\right) \quad (7)$$

and the measurement noise vector v_t is modeled as a zero-mean Gaussian random variable with covariance matrix R, $(0, R) \sim v_t$

which accounts for sensor imperfections and environmental disturbances affecting the measurement process [15,16]

3. EXTENDED KALMAN FILTER (EKF)

EKF is a widely accepted algorithm for state estimation of nonlinear systems. It is time-efficient to linearize the nonlinear motion model and the measurement model with first-order Taylor series approximation around the current estimate, so that recursive estimation can be done with relatively low computational burden [17,18]. This paper also takes the EKF as a benchmark to demonstrate the effectiveness of learning-aided scheme in nonlinear vehicle moving and radar observation.

3.1. The Prediction Phase

In this stage, the filter projects the current state and uncertainty forward in time.

1. State Extrapolation : The predicted state $\hat{x}_{t|t-1}$ is calculated by passing the previous estimate through the nonlinear state-transition function $f(\cdot)$

$$\hat{x}_{t|t-1} = f(\hat{x}_{t-1|t-1}) \quad (8)$$

2. Linearization of Dynamics : Since we cannot propagate covariance directly through a nonlinear function, we calculate the Jacobian Matrix F_t ,which represents the partial derivatives of f with respect to x [17,18] :

$$G = \frac{\partial f}{\partial x}\bigg|_{x_{t-1|t-1}}$$

3. Covariance Projection : The uncertainty (Error Covariance P) is projected forward, accounting for the process noise

$$P_{t|t-1} = f_t . P_{t-1|t-1} * f_t^{\,T} + Q \quad (9)$$

Al-Furat Journal of Innovations in Electronics and Computer Engineering (FJIECE)
ISSN -2708-3985

3.2 The Update Phase

In this stage, the filter corrects the prediction using the actual radar measurement.

1. Observation Prediction: The filter predicts what the radar should see based on the predicted state:

$$\hat{y}_t = h(\hat{x}_{t|t-1}) \tag{10}$$

2. Linearization of Observation: A second Jacobian Matrix (H_t) is computed to linearize the radar's measurement function:

$$H_t = \frac{\partial h}{\partial x}\Big|_{\hat{x}_{t|t-1}} \tag{11}$$

3. Innovation (Measurement Residual): Calculating the difference between the real measurement and the prediction:

$$r_t = y_t - \hat{y}_t \tag{12}$$

4. Innovation Covariance : Estimating the uncertainty in the innovation by factoring in the sensor noise R:

$$S_t = H_t . P_{t|t-1} * H_t{}^T + R \tag{13}$$

5. Analytical Kalman Gain Calculation :The "weight" given to the new measurement is computed. This is the most complex step in EKF:

$$K_t = P_{t|t-1} \,.\, H_t{}^T \,. S_t^{-1} \tag{14}$$

This step involves matrix inversion, which is computationally demanding and may lead to numerical sensitivity in higher-dimensional systems [18], [19]**.**

6. State Correction: Updating the state estimate by adding the weighted residual:

$$\hat{x}_{t|t} = \hat{x}_{t|t-1} + K_t . \boldsymbol{r_t} \tag{15}$$

7. Covariance Update: The final step is to refine the error covariance for the next recursive cycle:

$$P_{t|t} = (I - k_t H_t) P_{t|t-1} \tag{16}$$

3.3 Technical Limitations

The EKF implementation faces several critical challenges in real-time nonlinear tracking. First, it depends on Jacobian-based linearization, which may become inaccurate under sharp nonlinear maneuvers or large sampling intervals, potentially causing estimation degradation or divergence [20]**.** Second, EKF assumes that Q and R are known and constant, while in practice they can be uncertain or time-varying, especially in radar tracking applications [18,21]**.** Finally, the matrix inversion in s_t^{-1} may become a computational bottleneck and can be numerically unstable in high-dimensional settings [18, 19]**.**

4. KALMANNET

In actual vehicle tracking tasks, it is hard to model the dynamic of the tracked system in a precise way due to complexity and non-linearity of motion pattern and/or perturbations of environment including modeling errors. Thus, it makes difficult to model an explicit and precise state transition function that may cause performance degradation of the model-based filters (e.g., Extended Kalman Filter(EKF)) [22,23].

To address this issue, the current work proposes a KalmanNet-inspired method with neural network as a surrogate of the state transition function. More specifically, a teacher network is introduced to learn from data directly the physics of the vehicle dynamics and avoid making assumptions on global features of allowed motion.

Furthermore, as opposed to the previous method that had predefined noise covariance matrices used in the calculation of the Kalman gain matrix, we treat this as a learnable quantity so it can be optimized for each dataset during running. The other operations of the Kalman filter, including updating of innovation and state equations, are not affected. This integrated model combines the generalization faculty of data-driven learning property and the recursive structure of standard Kalman filter.

We present the details of our proposed KalmanNet framework including networks and training in the following subsections. [24]

4.1. KalmanNet Network Architecture

The resulting KalmanNet architecture is intended to keep the recursive structure of the standard Kalman filter, but replacing the explicit state transition sequence with a data-driven scheme based on neural networks. In this setting, a recurrent neural network is used to learn the system dynamics and a correction step that follows the classical Kalman filter form with train-able Kalman gain.

A. Neural State Transition Model

In this approach, the state transition function $f(\cdot)$ is estimated by a GRU model [26]. At every time step, we input the posterior state estimate at the previous iteration to a GRU, allowing for the learning of temporal dependencies and nonlinear motion pattern modeling present in vehicle dynamics. The GRU output is then fed through fully connected layers to generate the estimated prior state prediction.

$$\hat{x}_{t|t-1} = f_{-NN}\left(\hat{x}_{t-1|t-1}\right) \qquad (17)$$

This is particularly appealing if we want to learn the modeling of the system directly from data, without having a known dynamics model.

B. Observation Prediction

Given the predicted prior state, the corresponding observation is obtained by applying the known nonlinear measurement function, consistent with the state-space model introduced in Section 2.

$$\hat{y}_t = h\left(\hat{x}_{t|t-1}\right) \qquad (18)$$

C. Learnable Kalman Gain

Rather than using pre-defined noise covariance matrices to analytically compute the Kalman gain, the proposed method has an end-to-end learning of a Kalman gain matrix. This matrix is optimized during the training step and actually shared over time steps, thus making it possible for the filter to adapt its correction step directly based on data $k_{-NN} \epsilon R^{m\times n}$

D. State Update Equation

The posterior estimate of state is corrected toward the measurements according to an innovation calculated to be the difference between measured and predicted observations.

$$\hat{x}_{t|t} = \hat{x}_{t|t-1} + k_{-NN}(y_t - \hat{y}_t) \qquad (19)$$

This update preserves the recursive estimation structure of the Kalman filter while incorporating learned components to enhance robustness against modeling uncertainties.

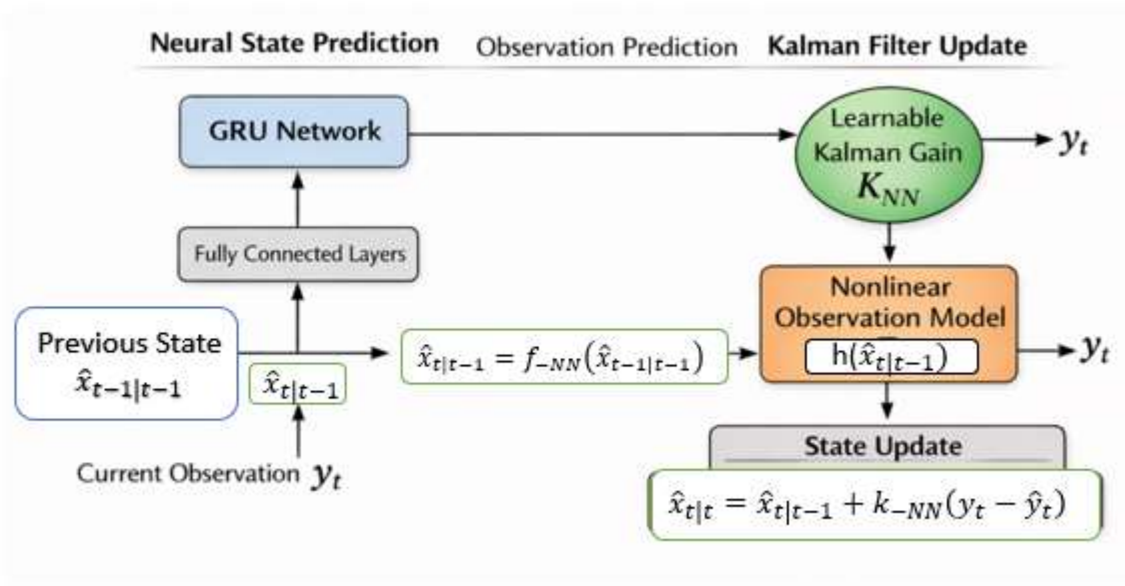

Fig. 1 Overall KalmanNet network architecture for nonlinear vehicle tracking

Al-Furat Journal of Innovations in Electronics and Computer Engineering (FJIECE)
ISSN -2708-3985

To further clarify the internal structure of the proposed KalmanNet implementation, Fig. 2 presents the detailed network architecture, including the GRU-based dynamics learner, the fully connected layers, and the GainNet used to generate the learnable Kalman gain matrix.

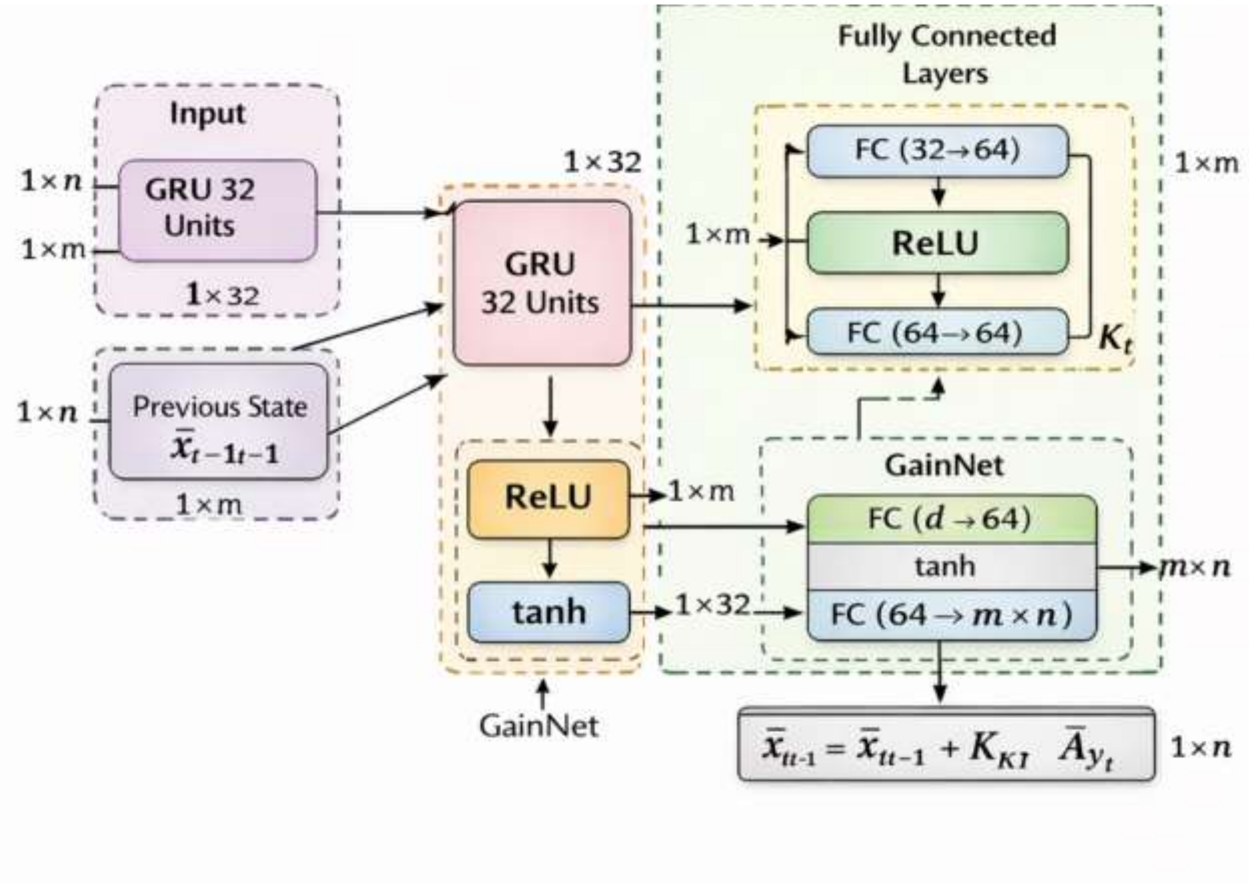

Fig.2 Detailed architecture of the proposed KalmanNet, showing the GRU-based state prediction module and the GainNet used to generate the learnable Kalman gain k_{NN}.

4.3 Training Procedure

The action of the KalmanNet model is learnt in a supervised way to minimize the difference between the predicted posterior states and ground-truth states. This training jointly optimizes the GRU-based state transition model f_{-NN} and the learnable Kalman gain k_{-NN} so that the filter can adapt to nonlinear vehicle dynamics and measurement noise without manually modeling the system.

Training and test data are sampled according to the state-space system model introduced in Section II. For each trajectory we simulate the ground-truth states x_t as well as the measurements y_t by applying the non-linear observation function and then adding Gaussian noise to it. Batches of the data are used to enable gradient based training of both the GRU network and the learnable Kalman gain over several trajectories at once.

We train the network to reduce the mean squared error (MSE) between the posterior state estimates and their respective ground-truth states. The loss for a trajectory of trajectory length T is given by

$$\mathcal{L} = \frac{1}{T}\sum_{t=1}^{T}\left\|\hat{x}_{t|t} - x_t\right\|^2 \qquad (20)$$

where $\hat{x}_{t|t}$ denotes the posterior state estimate and x_t the true state. During training, the posterior state estimate from the previous time step $\hat{x}_{t-1|t-1}$ is fed into the GRU network to produce the prior estimate $\hat{x}_{t|t-1}$, which is then mapped to the predicted observation in question (17) using the known nonlinear measurement function. The innovation, defined as the difference between the actual measurement and the predicted observation, is computed and used together with the learnable Kalman gain k_{-NN} to update the posterior state estimate according to the standard Kalman filter correction equation (18)

The training of GRU over the time is achieved through backpropagation through time (BPTT), and both the GRU parameters and learnable Kalman gain are updated by optimizer (e.g., Adam) to reduce the overall MSE. The initial posterior state $\hat{x}_{0|0}$ can be initialized to the first true state or some prior known guess. Through this process, the network is effectively training itself to model the system dynamics and make corrections to its state estimates such that it learns to track well in a robust manner even in nonlinear and noisy situations

5. RESULTS AND DISCUSSION

5.1. Simulation Environment and Parameters

To assess the performance of the proposed KalmanNet framework in nonlinear vehicle tracking, we performed a series of simulations on tracking a maneuvering vehicle within a nonlinear state-space model. The target state vector is represented by in (1) and the nonlinear motion model describing how the state evolves is given by (2). The simulated vehicle trajectory is curved and nonlinear by constant acceleration commands and constant turn-rate of the simulated vehicle and presents a practical tracking problem.

The radar sensor gives the nonlinear noisy range–bearing measurements. Model (3) represents the measurement model and the associated measurement equations for range, bearing (bearing), are given by models (4, 5). The noise level is set to an inverse-noise value around 20 dB, leading to a challenging condition where raw observations scatter significantly. This renders the direct estimation based on measurements unreliable and emphasises the importance of nonlinear filtering.

The KalmanNet model was implemented with PyTorch and trained, using the hyper-parameters listed in Table I, by an Adam optimizer together with a composition loss to integrate state estimation precision and measurement consistency regularization. Early stopping on validation performance was also used to avoid overfitting and select the best model checkpoint.

Al-Furat Journal of Innovations in Electronics and Computer Engineering (FJIECE)
ISSN -2708-3985

Table I: Simulation Setup and Training Hyperparameters

Parameter	Value
N_E (Training trajectories)	1000
N_CV (Validation trajectories)	100
N_T (Test trajectories)	200
T	50
T_{test}	50
State dimension m	4
Measurement dimension n	2
r^2(measurement variance)	Noise
$10\ \log_{10}\left(\frac{1}{r^2}\right)$	Noise
$Q = Q\ q^2_{stracture}$	Noise
$R = Rr^2_{stracture}$	Noise
Optimizer	Training
Learning rate (lr)	Training
Batch size	Training
Max steps	Training
Weight decay (wd)	Training
Composition Loss	Loss
α	Loss
Early stopping	Training

5.2. Motion and Measurement Visualization (Tracking Challenge)

The first step of the evaluation is to visualize the tracking performance under non-linear vehicle maneuver and noisy radar observation. Figure 2 The annotated vehicle motion path by the ground truth vehicle trajectory follows a curved maneuvering path, which is generated as described by the egocentric model (2). The noisy radar measurements are depicted as point clouds, which for each row due to the high measurement noise is influenced by our observation model (3)(5). This proves that pure observations-based compensation for image tracking is challenging, since the measured points are far off from the true trajectory.

This visualization inspires the design of KalmanNet, which provides a recursive estimation following the structure of the Kalman filter; however, it substitutes for a closed-form solution to compute an analytical Kalman gain with one that is learned and thus more robust to nonlinear dynamics and observation.

Al-Furat Journal of Innovations in Electronics and Computer Engineering (FJIECE)
ISSN -2708-3985

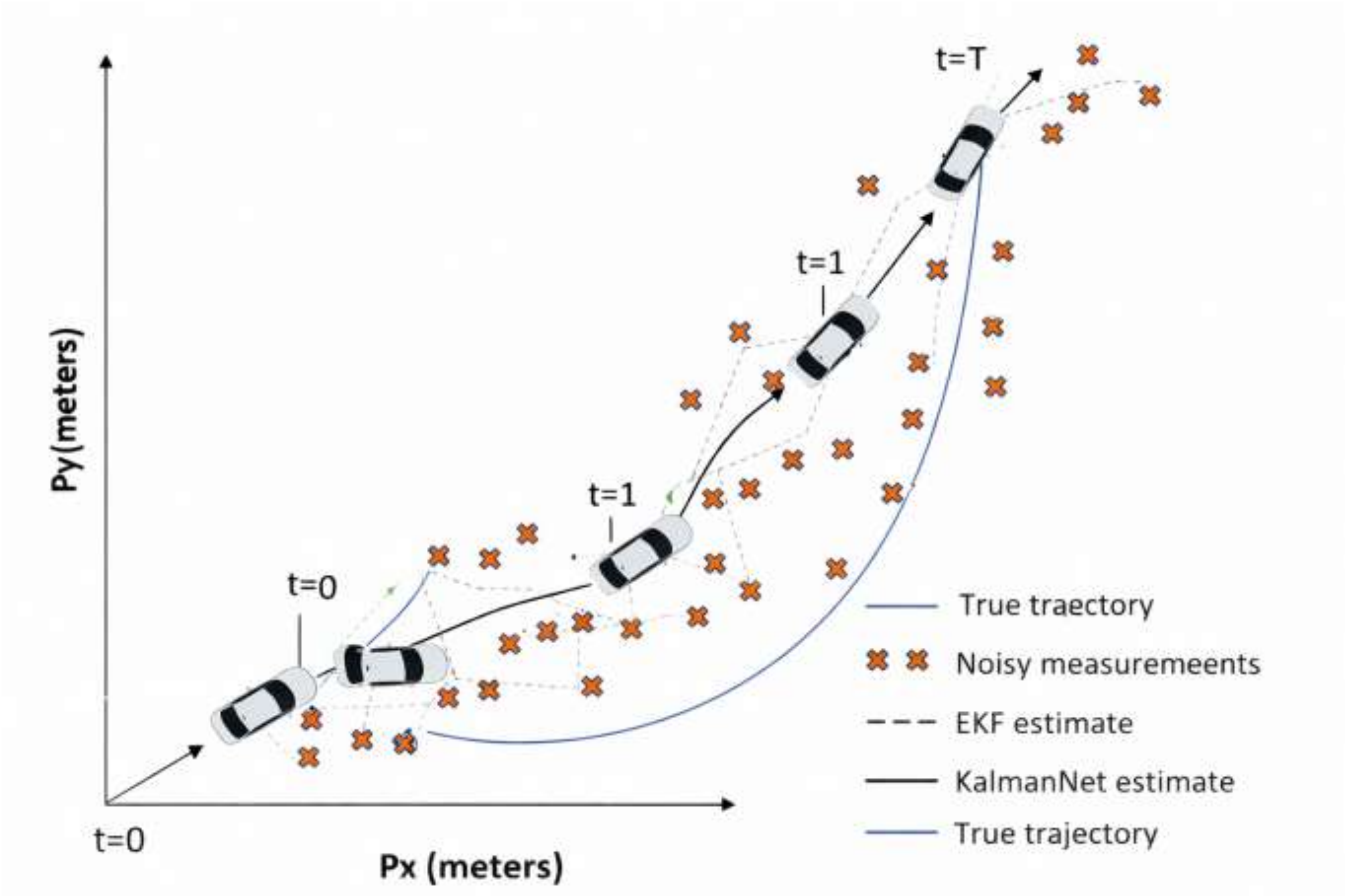

Fig.3 Vehicle motion visualization showing the true trajectory and noisy radar range–bearing measurements.

The nonlinear trajectory shown in Figure 3 is generated according to the kinematic model defined in Eq. (3) and Eq. (5), where a constant acceleration of 2.0 $m\ /\ s^2$ and a turn rate of 0.5 rad/s are applied to simulate a maneuvering vehicle.

5.3. Training Convergence and Generalization

The training and validation MSE curves are plotted through the optimization iterations to inspect the learning behaviour. This can be seen from Figure 3 as both training and validation errors consistently decrease, which is indicative of well-behaved convergence without divergence during the learning phase. The repeated close trend of the training and validation curves also indicates that the KalmanNet model generalizes well and is not subject to severe overfitting.

There are two horizontal baselines in Figure 3 representing the average test errors of EKF and KalmanNet, respectively. The EKF baseline resulted in an average test error of around 9.85 dB, whereas KalmanNet achieved a significantly lower average testing error of ~ −7.91 dB. This large performance difference supports the idea that learning an adaptive Kalman gain allows to better compensate non-linearities compared to Jacobian-based EKF linearization.

Al-Furat Journal of Innovations in Electronics and Computer Engineering (FJIECE)
ISSN -2708-3985

From a mathematical point of view, the EKF fails to capture linearization errors when motion and observation functions are very nonlinear (like in the case of (2) or (3)(5)). In contrast, KalmanNet

learns nonlinear correction patterns directly from data, resulting in more accurate and stable state estimates.

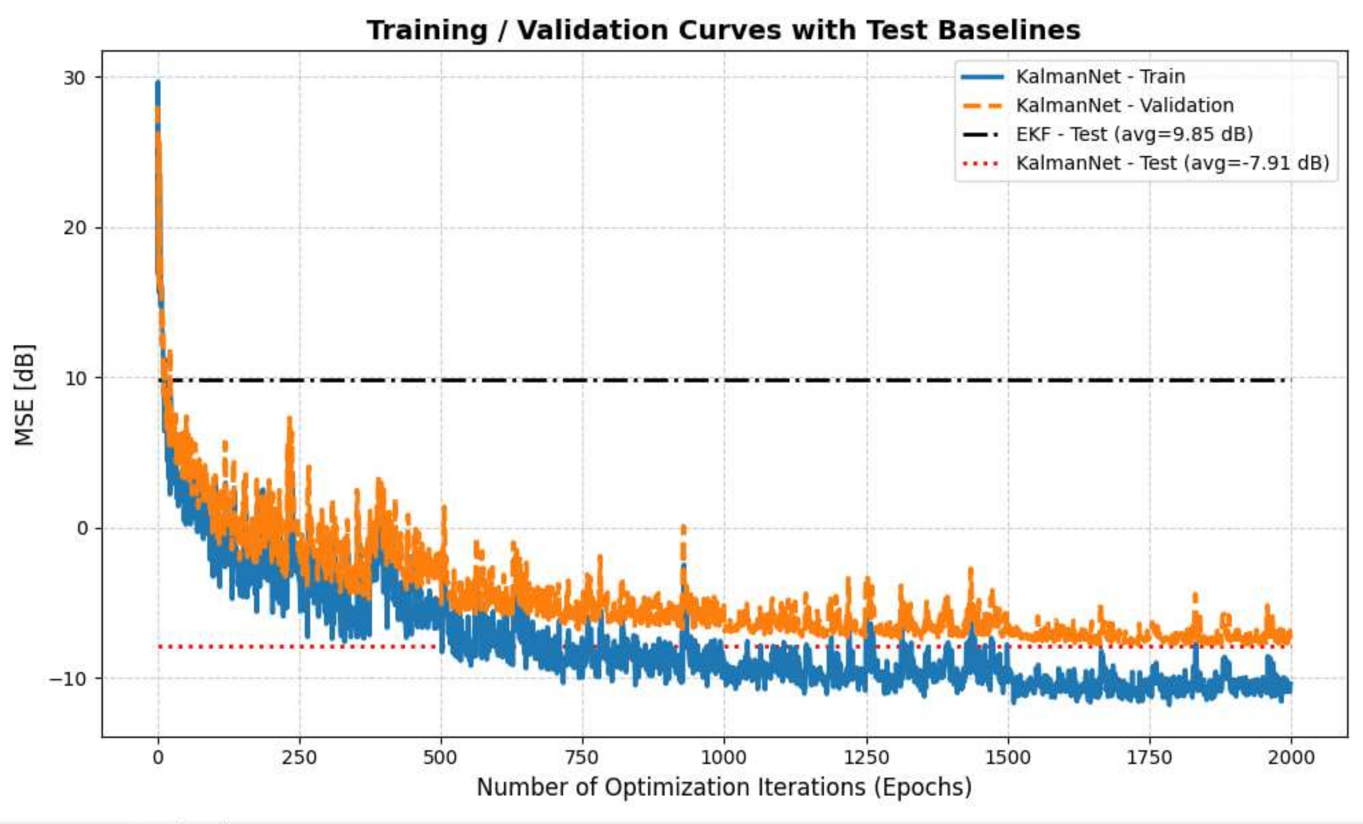

Fig.4 Training and validation MSE (dB) curves with EKF and KalmanNet test baselines.

5.4. Tracking Accuracy Over Time (NMSE Analysis)

Furthermore, to more robustly investigate accuracy of tracking across rolling sequence length, we examined the error evolution of normalized mean squared error (NMSE) in dB. NMSE brings a fair comparison through the estimation error normalization by true state power. The NMSE performance is reported in Figure 4 through a couple of complementary plots:

(A) NMSE per step and (B) Cumulative average NMSE.

As can be seen from Figure 4(A), the per-step NMSE of KalmanNet is consistently lower than that of EKF at all time steps. It suggests that KalmanNet can have strong instantaneous tracking ability even in the severely corrupted observations. The EKF error still remains quite large because the filter update is based

Al-Furat Journal of Innovations in Electronics and Computer Engineering (FJIECE)
ISSN -2708-3985

on first order linearization about the current estimate, which may be grossly inappropriate during the nonlinear maneuver.

In addition, cumulative average NMSE pattern have been given in Figure 4(B) depicting long term performance comparison. KalmanNet also achieves much lower total NMSE throughout the trajectory, indicating good quality of estimation and enhanced robustness in a long horizon. This shows that besides error suppression at individual steps, KalmanNet also maintains high accurate throughout the whole tracking duration.

Finally, the NMSE results demonstrate that KalmanNet is able to estimate accurately the state of target under nonlinear motion (2) and nonlinear radar observation (3)(5), showing significant improvements over EKF across the entire sequence.

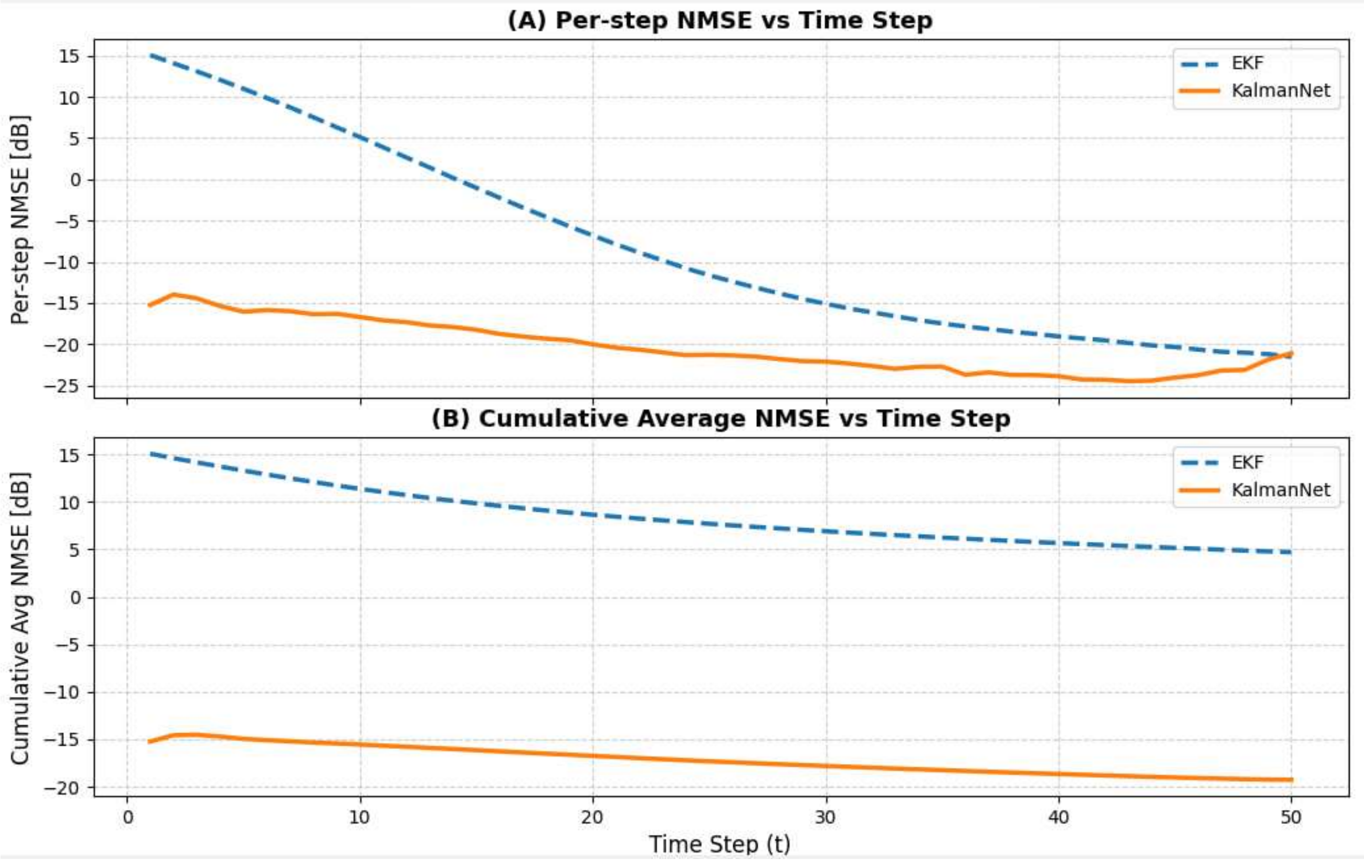

Fig.5 (A) Per-step NMSE (dB) vs time step and (B) cumulative average NMSE (dB) vs time step: EKF vs KalmanNet.

5.5. Computational Efficiency and Robustness Discussion

As for computational efficiency, the state space dimension in this work is reasonably low (from Eq.(1)) so that EKF matrix calculations are still possible. Nevertheless, EKF performance becomes restricted in higher nonlinear regime owing to Jacobian calculation and approximation errors, and its numerical stability may be worsted with growing nonlinearity.

KalmanNet addresses these limitations by performing an end-to-end learning of the Kalman gain, which makes it more robust and less susceptible to errors introduced due to approximation. Inference time measurements reveal that KalmanNet is still in the scale of practical for on-line processing such as near real-time tracking. Furthermore, the more regular NMSE profile implies less estimation jerking behavior and enhanced stability with respect to the EKF, which is essential in order to guarantee navigation safety.

In conclusion, the simulation results show that this application of embedding a learned Kalman gain within a recursive Kalman filtering framework is robust and effective for nonlinear vehicle tracking with noisy radar measurements.

6. CONCLUSION AND FUTURE WORK

In this work, we have demonstrated for the first time a robust KalmanNet-based vehicle tracking framework that can handle nonlinear systems using maneuvering dynamics with noisy nonlinear radar measurements. The proposed method obeys the recursive Kalman filtering framework and leads to learning an adaptive Kalman gain which could be able to manage better nonlinear state transitions (2) and nonlinear measurements (3),(5), comparing with classical EKF linearization.

Quantitative results indicated that the proposed method outperformed EKF with a significant margin. In particular, EKF achieves an average test error of around 9.85 dB and KalmanNet demonstrates a lower test error of around −7.91 dB, clearly indicating the significant gain in estimation accuracy and robustness.

Moreover, the convergence of training/validation and the analysis of NMSE show that KalmanNet is stable and reliable in providing accurate tracking trajectories with the lower sensitivity to measurement noise and nonlinear maneuvering.

This work could be extended to more complex cases like multi-sensor fusion (Radar+LiDAR+Camera) in the future. Furthermore, studying the KalmanNet robustness against sensor attacks, missing measurements and adverse weather conditions is a promising direction to further improve its performance for safety-critical autonomous navigation.

REFERENCES

[1] D. J. Jwo, “Critical remarks on the linearised and extended Kalman filters with geodetic navigation examples,” *Measurement*, vol. 43, no. 9, pp. 1077–1092, 2010.

[2] S. Yazdkhasti and M. Pakmehr, “Adaptive H-infinity extended Kalman filtering for a maneuvering target tracking,” *Transactions of the Institute of Measurement and Control*, 2023.

[3] S. Hochreiter and J. Schmidhuber, “Long short-term memory,” *Neural Computation*, vol. 9, no. 8, pp. 1735–1780, 1997.

[4] J. P. Llerena Caña, F. Herrero, and M. A. Gómez-Moreno, “Forecasting Nonlinear Systems with LSTM: Analysis and Comparison with Filtering Approaches,” *Sensors*, vol. 21, no. 5, 2021.

[5] X.-B. Jin, R. Jeremiah, T. Su, Y. Bai, and J.-L. Kong, "The New Trend of State Estimation: From Model-Driven to Data-Driven and Hybrid-Driven Methods," *Sensors*, vol. 21, no. 6, p. 2085, 2021.

[6] G. Revach, N. Shlezinger, X. Ni, A. L. Escoriza, R. J. G. van Sloun, and Y. C. Eldar, "KalmanNet: Neural Network Aided Kalman Filtering for Partially Known Dynamics," *IEEE Transactions on Signal Processing*, vol. 70, pp. 1532–1547, 2022.

[7] K. Cho, B. van Merriënboer, C. Gulcehre, D. Bahdanau, F. Bougares, H. Schwenk, and Y. Bengio, "Learning Phrase Representations using RNN Encoder–Decoder for Statistical Machine Translation," in *Proc. 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, Doha, Qatar, 2014.

[8] Y. Lin et al., "A hybrid data-driven model for state of health estimation," *Engineering Applications of Artificial Intelligence*, 2025.

[9] M. S. Grewal and A. P. Andrews, *Kalman Filtering: Theory and Practice with MATLAB*, 4th ed. Hoboken, NJ, USA: Wiley, 2015.

[10] D. Simon, *Optimal State Estimation: Kalman, H∞, and Nonlinear Approaches*. Hoboken, NJ, USA: Wiley, 2006.

[11] G. Revach, N. Shlezinger, X. Ni, A. L. Escoriza, R. J. G. van Sloun, and Y. C. Eldar, "KalmanNet: Neural Network Aided Kalman Filtering for Partially Known Dynamics," arXiv:2107.10043, 2021.

[12] X.-B. Jin, R. Jeremiah, T. Su, Y. Bai, and J.-L. Kong, "The New Trend of State Estimation: From Model-Driven to Data-Driven and Hybrid-Driven Methods," *Sensors*, 2021

[13] J. L. Crassidis and J. L. Junkins, *Optimal Estimation of Dynamic Systems*, 2nd ed. Boca Raton, FL, USA: CRC Press, 2011.

[14] P. S. Maybeck, *Stochastic Models, Estimation, and Control*, vol. 1. New York, NY, USA: Academic Press, 1979.

[15] E. A. Wan and R. Van Der Merwe, "The Unscented Kalman Filter for Nonlinear Estimation," in *Proc. IEEE Symposium 2000 (AS-SPCC)*, Lake Louise, Alberta, Canada, 2000.

[16] M. S. Grewal and A. P. Andrews, *Kalman Filtering: Theory and Practice with MATLAB*, 4th ed. Hoboken, NJ, USA: Wiley, 2015.

[17] D. Simon, *Optimal State Estimation: Kalman, H∞, and Nonlinear Approaches*. Hoboken, NJ, USA: Wiley, 2006.

[18] S. Haykin, *Kalman Filtering and Neural Networks*. New York, NY, USA: Wiley, 2001.

[19] R. E. Kalman, "A New Approach to Linear Filtering and Prediction Problems," *Journal of Basic Engineering*, vol. 82, no. 1, pp. 35–45, 1960.

Al-Furat Journal of Innovations in Electronics and Computer Engineering (FJIECE)
ISSN -2708-3985

[20] D. J. Jwo, “Critical remarks on the linearised and extended Kalman filters with geodetic navigation examples,” *Measurement*, vol. 43, no. 9, pp. 1077–1092, 2010.

[21] A. H. Mohamed and K. P. Schwarz, “Adaptive Kalman Filtering for INS/GPS,” *Journal of Geodesy*, vol. 73, pp. 193–203, 1999.

[22] D. J. Jwo, “Critical remarks on the linearised and extended Kalman filters with geodetic navigation examples,” *Measurement*, vol. 43, no. 9, pp. 1077–1092, 2010.

[23] J. L. Crassidis and J. L. Junkins, *Optimal Estimation of Dynamic Systems*, 2nd ed. Boca Raton, FL, USA: CRC Press, 2011.

[24] G. Revach, N. Shlezinger, X. Ni, A. L. Escoriza, R. J. G. van Sloun, and Y. C. Eldar, “KalmanNet: Neural Network Aided Kalman Filtering for Partially Known Dynamics,” *IEEE Transactions on Signal Processing*, vol. 70, pp. 1532–1547, 2022.

[25] S. Haykin, *Kalman Filtering and Neural Networks*. New York, NY, USA: Wiley, 2001.

[26] K. Cho, B. van Merriënboer, C. Gulcehre, D. Bahdanau, F. Bougares, H. Schwenk, and Y. Bengio, “Learning Phrase Representations using RNN Encoder–Decoder for Statistical Machine Translation,” in *Proc. EMNLP*, Doha, Qatar, 2014.

[27] M. S. Ibrahim, "Motion Estimation for Multi-Object Tracking using KalmanNet with Semantic-Independent Encoding," *IEEE Access*, vol. 13, pp. 1142-1155, 2025.

[28] Y. Shen, J. Zhang, and L. Wang, "KalmanFormer: Using transformer to model the Kalman Gain in Kalman Filters," *Frontiers in Neurorobotics*, vol. 18, pp. 88-102, 2024.