Semnan University

*Journal of Modeling & Simulation in Electrical & Electronics Engineering (MSEEE)*

Journal homepage: **https://mseee.semnan.ac.ir/**

**ISSN: 2821-0786**

# Robust Trajectory Estimation for Maneuvering Targets Using an Adaptive Interacting Multiple Model Extended Kalman Filter

Alireza Jarrah[1], Hadi Asharioun[2]*and Mohammadhossein Hashemi[3]

***Abstract*-- Accurately tracking maneuvering targets remains a significant challenge in fields such as autonomous navigation and surveillance. This paper presents a robust solution using an Interacting Multiple Model Extended Kalman Filter (IMM-EKF). The proposed architecture adaptively combines three distinct kinematic models: a Near Constant Velocity (NCV) model for linear motion, a Coordinated Turn (CT) for constant turn rates, and a Coordinated Turn with Rate and Acceleration (CTRA) to handle aggressive maneuvers. The IMM framework dynamically weights each model's contribution based on the measurement likelihood, producing a fused state estimate that is more reliable than any single-model filter. The algorithm's performance was rigorously validated against ground truth data, demonstrating high precision with a position Root Mean Square Error (RMSE) of 0.3117 m and a yaw RMSE of 2.1614 degrees. Furthermore, the filter's statistical integrity was confirmed through consistency tests, with 94.16% of the Normalized Innovation Squared (NIS) values falling within the 95% confidence interval. These results underscore the effectiveness of the proposed multi-model approach for complex and dynamic trajectory estimation.**

***Index Terms*- State Estimation; Trajectory Estimation; Maneuvering Target Tracking; Extended Kalman Filter; Interacting Multiple Model; Motion Models**

## I. INTRODUCTION

THE advent of autonomous systems, particularly in vehicular navigation and advanced driver-assistance systems, has rendered the accurate and reliable tracking of dynamic objects a critical area of research. The ability to precisely estimate the state, including position, velocity, and orientation, of surrounding vehicles is fundamental for safe path planning, collision avoidance, and overall situational awareness. This concept of awareness is multifaceted, extending beyond just the external environment to include the internal state of the driver. For instance, recent research has focused on developing comprehensive Fuzzy Driver Monitoring Systems that integrate parameters such as vehicle speed and driver drowsiness to automatically identify dangerous behaviors [1]. A critical component of such systems is the accurate interpretation of non-verbal cues, where deep learning techniques such as Convolutional and Recurrent Neural Networks are employed to classify facial expressions and infer the driver's level of fatigue [2]. While these technologies focus on the ego-vehicle's driver, the work presented here addresses the complementary challenge of perceiving the state of external maneuvering targets. A primary

Received; 2025-11-05 Revised; 2026-01-21 Accepted; 2026-02-07

1. Faculty of Electrical Engineering, Shahid Beheshti University, Tehran, Iran.
2. Faculty of Electrical Engineering, Shahid Beheshti University, Tehran, Iran.
3. Faculty of Electrical Engineering, Shahid Beheshti University, Tehran, Iran.

*Corresponding author Email: asharioun@sbu.ac.ir

**Cite this article as:**
Jarrah, A., Asharioun, H. & Hashemi, M H. (2026). Robust Trajectory Estimation for Maneuvering Targets Using an Adaptive Interacting Multiple Model Extended Kalman Filter. *Journal of Modeling & Simulation in Electrical & Electronics Engineering (MSEEE)*. Semnan University Press . *6 (1)*, 53-65.
DOI: https://doi.org/10.22075/MSEEE.2026.39451.1233

challenge in this domain is the ability to maintain tracking precision when a target deviates from simple, predictable motion—that is, when it executes a maneuver. The dynamic nature of these targets, which can seamlessly transition between straight-line motion, coordinated turns, and sharp accelerations, poses a significant challenge to conventional estimation algorithms. Single-model filters, which operate under a fixed set of dynamic assumptions, often fail to adapt to these behavioral shifts, leading to a degradation in accuracy or, in worst-case scenarios, a complete loss of track.

To overcome these limitations, a substantial body of research has been devoted to developing more sophisticated state estimation techniques capable of handling target maneuvers. Study [3] introduces a novel IMM–MCEKF filter for GPS navigation, which combines the IMM and the maximum correntropy criterion. IMM is utilized to alter the covariance parameters of measurement noises in parallel filters to address measurement uncertainty. MCC replaces the minimal mean-square-error optimization criterion in the EKF to specifically enhance performance when dealing with non-Gaussian noise and outlier-type multipath interference. Gadsden et al. [4] introduced a novel SVSF-IMM method for nonlinear target tracking, comparing it with the conventional EKF-IMM approach in an Air Traffic Control scenario. The SVSF-IMM method achieved a 30% improvement in position accuracy and demonstrated robust, stable performance due to the SVSF switching gain. Study [5] compares the EKF and the IMM approach for nonlinear maneuvering target tracking. IMM utilized linear constant velocity and CT models. The study found that IMM is more robust and accurate for large ranges of nonlinearity and high noise, provided the model count is optimized. The IMM5CKF algorithm is proposed [6] to enhance the accuracy and quick response of maneuvering target tracking. Integrating a five-degree Cubature Kalman Filter (CKF) with the IMM framework, it simultaneously handles models via a Markov Chain. Gao et al. [7] propose four improved IMM-based algorithms for nonlinear maneuvering target tracking. They achieve better accuracy and efficiency than traditional methods, with Unscented Kalman Filter (UKF)-MIMM offering the best accuracy and EKF-SIMM the fastest performance.

Additionally, [8] proposes a mobile location estimation algorithm for harsh wireless environments using an interacting multiple model framework with a Markov chain to handle NLOS transitions. The method fuses TOA and RSS data and applies the CKF for nonlinear estimation, achieving accurate tracking of maneuvering mobile stations. An IMM-CKF algorithm is proposed [9] for tracking maneuvering targets using angular measurements. By integrating the CKF into the IMM framework, the method improves tracking accuracy and reduces computation time compared to the IMM-UKF approach. The Autoencoder IMM filter is proposed [10] for maneuvering target tracking. This hybrid framework embeds an IMM, which uses models like NCV and CT, within an autoencoder to facilitate nonlinear transformations. The goal is to help the IMM quickly identify mode changes, leading to improved state estimation compared to classical methods. Study [11] proposes a smart IMM filter using the EKF for bearing-only 2D maneuvering target tracking. To improve state estimation, the algorithm employs a second-order Markov model instead of the first-order IMM, allowing for a more accurate description of the target's maneuvering behavior. Simulation results confirmed that the second-order model provides more efficient tracking with reduced error. The IMM-MOT framework [12] introduces an IMM filter for 3D Multi-Object Tracking, overcoming the limitation of single motion models in fitting complex, varying motion patterns. The tracker dynamically combines models, including constant velocity, constant acceleration, constant turn rate and velocity, and CTRA, employing an EKF for nonlinear models. This approach yields more accurate predictions, achieving high performance on the NuScenes dataset.

The authors in [13] proposed a variational Bayesian adaptive framework that enhances traditional IMM methods by enabling faster model switching through real-time change-point updates. Their approach jointly optimizes state and model posteriors, achieving superior tracking performance, particularly in highly maneuverable target scenarios. Study [14] presents an intelligent tracking framework for highly maneuvering aerial targets by integrating a TCN–LSTM neural prediction model into a UKF. By leveraging recurrent architectures to approximate complex target dynamics and embedding them through the Unscented Transformation, the proposed TCN-LSTM-UKF method mitigates model-construction challenges and transition delays, achieving significantly improved tracking performance under high-maneuver conditions. An adaptive constant acceleration (ACA) model is integrated with a strong tracking square-root CKF (ST-SCKF) to improve maneuvering target tracking. By linking acceleration, velocity, and jerk through Taylor expansion and adaptively adjusting process noise, the ACA-ST-SCKF achieves higher accuracy, better adaptability, and lower computational complexity compared with traditional adaptive and IMM-based filters [15]. Leveraging variational inference, study [16] proposes an adaptive interacting multiple model algorithm for tracking multiple maneuvering extended targets. The method augments the state to capture time-varying orientation and shape and updates model probabilities in real-time, significantly improving tracking accuracy and robustness over conventional IMM approaches.

To address the limitations of single-model estimators in highly dynamic environments [12], this paper presents a comprehensive adaptive tracking framework based on the IMM-EKF. The distinct contribution of this work lies in the rigorous integration and statistical validation of a specialized kinematic model bank— NCV, CT, and CTRA. By explicitly incorporating the CTRA model, the framework accounts for complex dynamics involving simultaneous linear acceleration and variable turn rates, a capability often lacking in standard implementations. Furthermore, unlike studies that rely on position and velocity error metrics [4, 11], this research provides a holistic assessment of tracker reliability through extensive statistical consistency tests (NIS and NEES). This approach ensures the filter remains robust and statistically

sound across a challenging hybrid scenario encompassing both smooth S-curves and sharp 90-degree maneuvers.

The remainder of this paper is organized as follows: Section 2 details the proposed methodology, including the formulation of the motion models and the recursive steps of the IMM-EKF algorithm. Section 3 describes the experimental setup, the dataset generation, and the evaluation metrics. In Section 4, the simulation results are presented and comprehensively analyzed. Finally, Section 5 concludes the paper with a summary of the findings and suggestions for future work.

## II. Proposed Method

To address the challenge of tracking maneuvering targets, this paper proposes a hybrid estimation framework based on the IMM architecture [17]. The core philosophy of this approach is to not rely on a single, one-size-fits-all motion model, but to employ instead a bank of specialized estimators that operate in parallel.

The proposed methodology is structurally divided into two key components. First, a set of diverse kinematic models is established, where each model is optimized to describe a specific dynamic behavior, from simple linear motion to complex accelerating turns. Second, the IMM algorithm itself serves as the high-level probabilistic engine. This engine's role is to dynamically weigh the contribution of each specialized model based on real-time measurement data and intelligently fuse their estimates into a single, robust, and highly accurate output.

### A. Motion Models

Accurate tracking of a maneuvering target necessitates a model that can adapt to its changing dynamics. A single motion model is often insufficient, as it cannot adequately describe the full spectrum of behaviors, ranging from straight-line trajectories to sudden, aggressive turns. To address this limitation, this research employs a bank of three distinct motion models to cover a wide array of kinematic behaviors. This work integrates the NCV, the CT, and the CTRA models. The subsequent sections will provide a detailed formulation of each of these models.

- *Near Constant Velocity (NCV)*

The NCV model is the simplest representation of target dynamics, designed to describe motion along straight or near-straight paths. It operates under the assumption that the target maintains a constant velocity vector, making it highly effective for non-maneuvering phases of a trajectory. The state of the target is defined in a Cartesian coordinate system by the state vector (1):

$$x_{NCV} = [p_x, p_y, v_x, v_y]^T \tag{1}$$

where $(p_x, p_y)$ represents the target's position, and $(v_x, v_y)$ are the corresponding velocity components along the x and y axes.

The model's dynamics are governed by a linear state transition equation, which projects the state from a previous time step $k-1$ to the current time step $k$ (2):

$$x_k = F_{NCV} x_{k-1} + w_{k-1} \tag{2}$$

The state transition matrix (3), $F_{NCV}$, is derived from basic kinematic principles, updating the position based on the velocity over the time interval $\Delta t$:

$$F_{NCV} = \begin{bmatrix} 1 & 0 & \Delta t & 0 \\ 0 & 1 & 0 & \Delta t \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{3}$$

The term $w_{k-1}$ represents the process noise, which is modeled as a zero-mean white Gaussian noise with a covariance matrix $Q_{NCV}$. This noise accounts for slight, unmodeled accelerations or random deviations from a perfectly constant velocity, thereby preventing the filter from becoming overconfident. Despite its simplicity and efficiency, the primary limitation of the NCV model is its inherent inability to accurately track targets during turns or significant maneuvers, resulting in a rapid increase in estimation error.

- *Coordinated Turn (CT)*

For tracking targets during periods of sustained turning, the CT model is employed. This model is specifically designed to describe motion along an arc with a constant turn rate and speed. It provides a more accurate representation of maneuvering behavior compared to the NCV model by adopting a different kinematic state. The state vector for the CT model is defined as (4):

$$x_{CT} = [p_x, p_y, v, \psi, \dot{\psi}]^T \tag{4}$$

where $(p_x, p_y)$ are the Cartesian positions, $v$ is the magnitude of the velocity (speed), $\psi$ is the yaw angle (heading), and $\dot{\psi}$ (also denoted as ω) is the turn rate.

It is important to note that the CT and CTRA models are kinematic and operate under a zero sideslip assumption. Therefore, the vehicle's body orientation, or Yaw ($\psi$), is assumed to be instantaneously aligned with the velocity vector, or heading.

This zero-sideslip assumption is justified because the proposed method employs kinematic models rather than dynamic ones. In typical urban driving scenarios on paved roads with high friction coefficients, the vehicle's velocity vector aligns closely with its longitudinal axis, making this non-holonomic constraint a valid and computationally efficient approximation for trajectory estimation.

Unlike the NCV model, the state transition for the CT model is non-linear due to the trigonometric relationships governing circular motion. The predicted state is calculated based on the previous state components $(p_x, p_y, v, \psi, \omega)$. For a non-zero turn rate $(\omega \neq 0)$, the position is updated as in (5) and (6):

$$p_{x,k} = p_{x,k-1} + \frac{v}{\omega}\big(sin(\psi + \omega\Delta t) - sin(\psi)\big) \tag{5}$$

$$p_{y,k} = p_{y,k-1} + \frac{v}{\omega}(-cos(\psi + \omega\Delta t) + cos(\psi)) \tag{6}$$

The yaw angle is updated linearly: $\psi_k = \psi_{k-1} + \omega \Delta t$. To be used within the EKF framework, these non-linear equations must be linearized at each time step, which is accomplished by computing the state transition Jacobian matrix, $F_{CT}$.

The process noise for the CT model, with covariance $Q_{\mathrm{CT}}$, accounts for deviations from the ideal coordinated turn assumption. It models random fluctuations in the target's speed and variations in its turn rate, which are driven by the standard deviations $\sigma_v$ and $\sigma_{\dot{\omega}}$, respectively. While the CT model significantly improves tracking performance during coordinated turns, its primary limitation is the assumption of constant speed and turn rate. This assumption is violated during aggressive or non-coordinated maneuvers, when the target simultaneously accelerates and changes its rate of turn.

- *Coordinated Turn with Rate and Acceleration (CTRA)*

The CTRA model is the most sophisticated of the three, designed to handle complex scenarios where a target executes aggressive maneuvers. It extends the CT model by incorporating linear acceleration, thus allowing it to accurately describe motions involving simultaneous changes in both speed and turn rate. The state vector is augmented with an acceleration term, $a$, and is defined as (7):

$$x_{CTRA} = \left[p_x, p_y, v, \psi, \dot{\psi}, a\right]^T \tag{7}$$

The state transition equations for the CTRA model are highly non-linear, as they account for the influence of both the turn rate (ω) and linear acceleration (a) on the target's position. The velocity is predicted as $v_k = v_{k-1} + a\Delta t$, while the position update incorporates the effects of this changing velocity throughout the time interval $\Delta t$. For a non-zero turn rate, this results in a complex update mechanism for the position components. As with the CT model, these non-linear dynamics are linearized at each step via a state transition Jacobian matrix, $F_{CTRA}$, for use within the EKF algorithm.

The process noise in the CTRA model, defined by the covariance matrix $Q_{CTRA}$, is critical for its adaptability. It is applied to the turn rate and acceleration states, modeling the rate of change of the turn rate and the rate of change of acceleration (jerk). These are driven by the standard deviations $\sigma_{\dot{\omega}}$ and $\sigma_j$, respectively. By accounting for these higher-order dynamics, the CTRA model overcomes the limitations of the NCV and CT models, enabling the filter to maintain high tracking accuracy even during the most challenging and dynamic phases of a target's trajectory.

- *Motion Model State Transformations*

The NCV state vector is converted to the CTRA representation to align with a different motion model that allows for more flexible dynamics. This conversion is performed using (8) for the CTRA state $x_{CTRA}$ and (9) for the Jacobian matrix $J_{CTRA}$, which describes the linearized relationship between the original NCV model and the new CTRA model. This step ensures that the system's kinematic properties are accurately represented in the CTRA model, accounting for both position and velocity components.

$$x_{CTRA} = \begin{bmatrix} p_x \\ p_y \\ v = \sqrt{{v_x}^2 + {v_y}^2} \\ atan2(v_y, v_x) \\ 0 \\ 0 \end{bmatrix} \tag{8}$$

$$J_{CTRA} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & v_x/v & v_y/v \\ 0 & 0 & -v_y/v^2 & v_x/v^2 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} \tag{9}$$

The CT state vector is converted to the CTRA model by augmenting the state with zero acceleration and velocity components. As shown in (10) for $x_{CTRA}$ and (11) for $J_{CTRA}$, this transformation retains the position and velocity information from the CT model while adding extra states for acceleration (set to zero). This conversion provides a more flexible model that can later represent dynamic changes in velocity or acceleration, even though they are not explicitly present in the original CT model.

$$x_{CTRA} = \begin{bmatrix} x_{CT} \\ 0 \end{bmatrix} \tag{10}$$

$$J_{CTRA} = \begin{bmatrix} I_{5\times5} \\ 0_{1\times5} \end{bmatrix} \tag{11}$$

The CTRA state vector is converted to the NCV representation to project the system's state onto a simpler motion model that describes linear motion with constant velocity. This conversion is performed using (12) for the NCV state $x_{NCV}$ and (13) for the Jacobian matrix $J_{NCV}$, which captures the linearized relationship between the original CTRA model and the NCV representation. Through this transformation, the position and velocity components of the system are preserved, while higher-order dynamics such as acceleration and turn rate are omitted to align with the NCV model assumptions.

$$x_{NCV} = \begin{bmatrix} p_x \\ p_y \\ v\cos(\psi) \\ v\sin(\psi) \end{bmatrix} \tag{12}$$

$$J_{NCV} = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & \cos(\psi) & -v\sin(\psi) & 0 & 0 \\ 0 & 0 & \sin(\psi) & v\cos(\psi) & 0 & 0 \end{bmatrix} \tag{13}$$

To represent the system using a CT model, the CTRA state vector is projected onto a reduced state space that excludes acceleration. Equation (14) defines the resulting CT state $x_{CT}$ while (15) provides the Jacobian matrix $J_{CT}$, which linearizes the mapping from the original CTRA model to the CT representation. This conversion preserves the system's position, velocity, heading, and turn rate, but deliberately omits the acceleration component to comply with the assumptions of the

CT model.

$$x_{CT} = x_{CTRA}(1:5) \quad (14)$$

$$J_{CT} = [I_{5\times5} \quad 0_{5\times1}] \quad (15)$$

### B. IMM-EKF Algorithm

To effectively leverage the bank of motion models described previously, the IMM framework is employed. The core principle of the IMM is the parallel operation of multiple filters, each corresponding to a specific motion model. The final state estimate is then derived from a weighted fusion of the outputs from each filter, where the weights dynamically reflect how well each model describes the target's current behavior. Since the CT and CTRA models are non-linear, the EKF is utilized for their respective filtering steps, leading to the overall IMM-EKF architecture. The algorithm operates as a recursive cycle, executed at each time step, which consists of four fundamental stages: 1) Interaction/Mixing, 2) Parallel Filtering, 3) Model Probability Update, and 4) State Combination.

To facilitate a deeper understanding of the proposed method, the complete architectural flowchart of the IMM-EKF is depicted in Fig. 1. This diagram systematically visualizes the signal flow through the algorithm's four core modules within a single processing cycle. It emphasizes the parallel structure of the filtering bank, where the linear NCV model and the non-linear EKF-based CT and CTRA models operate simultaneously. Although presented as a linear flow, the diagram encapsulates the algorithm's recursive logic; the final posterior model states and probabilities ($x_k$, $P_k$, $\mu_k$) generated at the output stage inherently serve as the requisite inputs (k−1) for the interaction step of the subsequent iteration, thereby preserving the diversity of motion modes required for robust tracking.

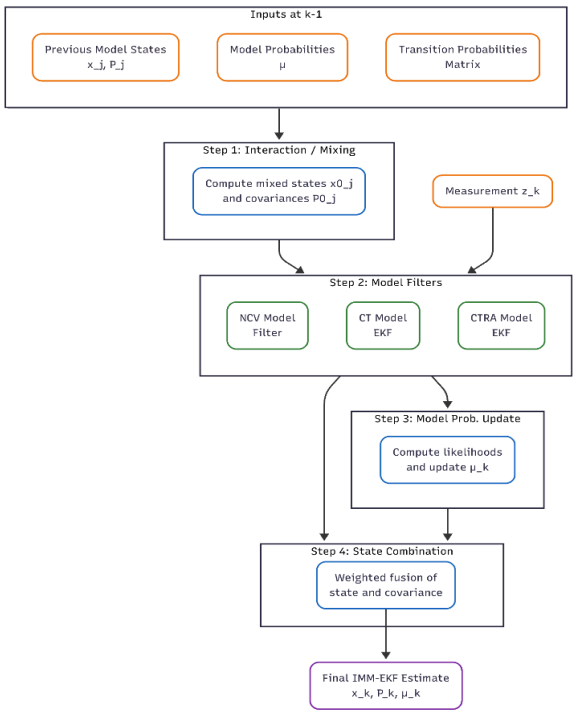


Fig. 1. Architecture of the proposed multi-model IMM-EKF tracking framework

- *Step 1: Interaction / Mixing*

The first stage in the IMM cycle is the interaction, or mixing, of the state estimates from the previous time step. The purpose of this step is to compute a mixed input for each filter based on the outputs of all filters from the prior step. This process ensures that each filter starts its prediction and update cycle with a more informed initial condition. The mixing probability, $\mu_{i|j,k-1}$, is calculated as (16):

$$\mu_{i|j,k-1} = \frac{1}{\bar{c}_j}\pi_{ij}\mu_{i,k-1} \quad (16)$$

where $\pi_{ij}$ is the transition probability from model $i$ to model $j$, $\mu_{i,k-1}$ is the probability of model $i$ at the previous step, and $\bar{c}_j$ is a normalization factor.

Using these probabilities, the mixed initial state (17), $\hat{x}_{0j,k-1}$, and covariance (18), $P_{0j,k-1}$, for each filter j, are computed:

$$\hat{x}_{0j,k-1} = \sum_{i=1}^{M} \mu_{i|j,k-1}\,\hat{x}_{i,k-1} \quad (17)$$

$$P_{0j,k-1} = \sum_{i=1}^{M} \mu_{i|j,k-1}[P_{i,k-1} + (\hat{x}_{i,k-1} - \hat{x}_{0j,k-1})(\hat{x}_{i,k-1} - \hat{x}_{0j,k-1})^T] \quad (18)$$

Here, M is the total number of models. It is crucial to note that before being mixed, the state vectors $\hat{x}_{i,k-1}$ and their corresponding covariances $P_{i,k-1}$ must be transformed into the state space of the target model $j$. This transformation involves a change of basis and requires the calculation of a Jacobian matrix to propagate the covariance correctly.

- *Step 2: Parallel Filtering*

Following the interaction stage, each of the M filters is executed independently. Each filter $j$ takes the mixed initial state, $\hat{x}_{0j,k-1}$, and covariance, $P_{0j,k-1}$, from the previous step as its input. It then performs a standard one-step Kalman filter (for the linear NCV model) or an EKF (for the non-linear CT and CTRA models) cycle using the current measurement, $z_k$. This cycle consists of a prediction step, where the state is propagated forward in time according to its specific motion model, and an update step, where this prediction is corrected based on the new measurement. For the linear NCV model, a standard Kalman Filter is used. For the non-linear CT and CTRA models, an EKF is employed.

While more advanced nonlinear filters, such as the UKF and CKF exist, which can offer superior accuracy in highly non-linear scenarios by avoiding the calculation of Jacobians [18], the EKF was deliberately chosen for this application. This decision is based on the optimal trade-off between estimation accuracy and computational efficiency.

The non-linearities present in the CT and CTRA motion models are generally smooth and mathematically tractable. Consequently, the first-order linearization used by the EKF

provides a sufficiently accurate approximation [19, 20] while maintaining a significantly lower computational footprint compared to the sigma-point propagation required by UKF or CKF. This reduced complexity is not only vital for ensuring real-time performance, but also translates directly to lower energy consumption, a critical consideration for the power-constrained embedded platforms found in autonomous vehicles [20]. Given that the EKF represents a practical, well-established benchmark with a clear algorithm structure, it was selected as the most appropriate filter for this study.

1. Prediction Step

The prediction step, also known as the time update, projects the state estimate and its uncertainty from the previous time step to the current one, based on the dynamics of the motion model. For each model $j$, the mixed state estimate $\hat{x}_{0\,j,k-1}$ is propagated through the non-linear state transition function $f_j$ to compute the a priori state estimate, $\hat{x}_{k|k-1}^{(j)}$ (19):

$$\hat{x}_{k|k-1}^{(j)} = f_j\left(\hat{x}_{0\,j,k-1}\right) \tag{19}$$

To propagate the covariance, the EKF linearizes this non-linear transformation using a first-order Taylor series expansion, represented by the Jacobian matrix of the state transition function, $F_{j,k}$. The a priori covariance (20), $P_{k|k-1}^{(j)}$, is then calculated as:

$$P_{k|k-1}^{(j)} = F_{j,k}\, P_{0\,j,k-1} F_{j,k}^{T} + Q_j \tag{20}$$

where $Q_j$ is the process noise covariance for model $j$, which accounts for the uncertainty inherent in the motion model itself.

2. Update Step

The update step, or measurement update, corrects the a priori estimate using the information from the new measurement $z_k$. The process begins by computing the measurement residual, or innovation, $y_k^{(j)}$, which quantifies the discrepancy between the actual measurement and the predicted measurement. The predicted measurement is obtained by mapping the a priori state into the measurement space via the non-linear measurement function $h_j$ (21) :

$$y_k^{(j)} = z_k - h_j(\hat{x}_{k|k-1}^{(j)}) \tag{21}$$

The Kalman Gain, $K_k^{(j)}$, is then computed. This gain acts as an optimal weighting factor that minimizes the a posteriori error covariance, effectively balancing the confidence in the predicted state against the confidence in the incoming measurement. It is calculated using the innovation covariance $S_k^{(j)}$ and the linearized measurement matrix (Jacobian) $H_{j,k}$ in (22) and (23):

$$S_k^{(j)} = H_{j,k} P_{k|k-1}^{(j)} H_{j,k}^{T} + R_k \tag{22}$$

$$K_k^{(j)} = P_{k|k-1}^{(j)} H_{j,k}^{T} (S_k^{(j)})^{-1} \tag{23}$$

where $R_k$ is the measurement noise covariance. Finally, the a priori state (24) and covariance (25) are corrected using the innovation and the Kalman Gain to yield the final a posteriori estimates for model $j$:

$$\hat{x}_{j,k} = \hat{x}_{k|k-1}^{(j)} + K_k^{(j)} y_k^{(j)} \tag{24}$$

$$P_{j,k} = (I - K_k^{(j)} H_{j,k}) P_{k|k-1}^{(j)} \tag{25}$$

This updated state $\hat{x}_{j,k}$ and covariance $P_{j,k}$ represent the final output of the EKF for model $j$ at time step $k$.

In addition to the updated state and covariance, this filtering stage yields the likelihood of the measurement, $\Lambda_{j,k}$, a value crucial for the model probability update that follows.

- *Step 3: Model Probability Update*

After each filter has processed the latest measurement, the probability of each motion model, $\mu_{j,k}$, is updated to reflect its performance. This update is based on how well each model's prediction matched the actual measurement, a quantity captured by the model-specific likelihood (26), $\Lambda_{j,k}$. The likelihood is calculated using the multivariate Gaussian probability density function of the measurement residual, $y_{j,k}$, with respect to its covariance, $s_{j,k}$ :

$$\Lambda_{j,k} = N\left(y_{j,k}; 0, s_{j,k}\right) \tag{26}$$

A higher likelihood value indicates a better fit between the model's prediction and the measurement.

The model probabilities are then updated using a Bayesian framework. The posterior probability of each model j is proportional to its prior probability (the predicted probability, $\bar{c}_j$, from the interaction step) multiplied by its calculated likelihood. The final updated probability (27), $\mu_{j,k}$, is found by normalizing these values:

$$\mu_{j,k} = \frac{\Lambda_{j,k}\bar{c}_j}{\sum_{i=1}^{M} \Lambda_{j,k}\bar{c}_i} \tag{27}$$

This mechanism ensures that models that more accurately describe the target's current dynamics are assigned a higher probability, thereby giving them more influence in the final state estimation.

- *Step 4: State Combination*

The final stage of the IMM cycle is the combination, or fusion, of the state estimates and covariances produced by the individual filters. This step generates a single, comprehensive state estimate that represents the overall output of the IMM filter at the current time step. The combined state (28), $\hat{x}_k$, is computed as a weighted average of the posterior state estimates from each filter, where the weights are the updated model probabilities, $\mu_{j,k}$:

$$\hat{x}_k = \sum_{j=1}^{M} \mu_{j,k}\, \hat{x}_{j,k} \tag{28}$$

Similarly, the overall covariance (29), $P_k$, is computed by combining the covariances from each filter. This calculation not only includes a weighted sum of the individual posterior covariances but also incorporates a term that accounts for the spread among the different state estimates, thereby providing a more accurate representation of the total uncertainty:

$$P_k = \sum_{j=1}^{M} \mu_{j,k}[P_{j,k} + (\hat{x}_{j,k} - \hat{x}_k)(\hat{x}_{j,k} - \hat{x}_k)^T] \tag{29}$$

Before this fusion, it is essential that all state estimates and covariances, which may exist in different state spaces (e.g., NCV vs. CTRA), are transformed into a common reference frame. The resulting combined state $\hat{x}_k$ and covariance $P_k$ are the final outputs for the time step k and serve as the inputs for the interaction stage of the next cycle.

## III. Experimental Setup

To evaluate the proposed algorithm, a simulation was conducted under conditions designed to emulate a real-world application. It is critical to note that the filter was not provided with the ground truth data directly. To simulate a realistic sensor measurement process, the measurement vector $z_k$ at each time step was generated by corrupting the true position and velocity data with zero-mean Gaussian noise. The standard deviations of this noise, $\sigma_p$ and $\sigma_{v,meas}$, are detailed in the Filter Parameters section

### A. Dataset

The dataset used for evaluating the filter performance was synthetically generated using the Driving Scenario Designer toolbox in MATLAB. This approach enabled the generation of a precise ground truth trajectory, which was then exported as a CSV file. The scenario was specifically designed to encompass a diverse range of dynamic behaviors representative of real-world driving conditions. The trajectory includes straight-line segments, gentle S-curves, and sharp turns. In addition to the primary state variables, the ground truth yaw angle ($\psi$), required for calculating the Yaw RMSE, was derived from the true velocity components ($v_x, v_y$) using the standard four-quadrant arctangent function.

Furthermore, the ego vehicle's speed profile is intentionally dynamic and context-aware. It realistically simulates higher speeds on straight sections, followed by the necessary decelerations for navigating these turns. This combination of linear and highly non-linear motion, coupled with variable acceleration, provides a challenging and comprehensive test case for validating the adaptive capabilities of the proposed IMM-EKF.

### B. Filter Parameters

The performance of the IMM-EKF tracker is critically dependent on the careful selection of its core parameters. These values define the filter's underlying assumptions about the target's motion dynamics (process noise), the sensor's precision (measurement noise), and the probability of transitioning between different behaviors (Transition Probability Matrix). The parameters used in this study were determined based on domain knowledge of typical vehicle dynamics and refined through empirical tuning to achieve optimal performance on the test dataset. The key parameters are summarized in Table I.

TABLE I
Configuration Parameters for the Proposed IMM-EKF Tracker

| Category | Parameter | Symbol | Value |
|---|---|---|---|
| IMM Framework | Transition Probability Matrix (TPM) | $\Pi$ | $\begin{bmatrix} 0.90 & 0.05 & 0.05 \\ 0.05 & 0.90 & 0.05 \\ 0.05 & 0.05 & 0.90 \end{bmatrix}$ |
| | Initial Model Probabilities | $\mu_0$ | $[0.90 \quad 0.08 \quad 0.02]^T$ |
| Process Noise | NCV Process Noise Std. Dev. | $\sigma_q$ | 0.6 $m/s^2$ |
| | CT Speed Noise Std. Dev. | $\sigma_{v,ct}$ | 2.4 $m/s^2$ |
| | CT Turn Rate Noise Std. Dev. | $\sigma_{\dot{\omega},ct}$ | 0.6 $rad/s^2$ |
| | CTRA Jerk Noise Std. Dev. | $\sigma_j$ | 20.0 $m/s^3$ |
| | CTRA Turn Rate Noise Std. Dev. | $\sigma_{\dot{\omega},ctra}$ | 3.0 $rad/s^2$ |
| Measurement Noise | Position Measurement Std. Dev. | $\sigma_p$ | 0.5 $m$ |
| | Velocity Measurement Std. Dev. | $\sigma_{v,meas}$ | 0.5 $m/s$ |

The TPM is chosen to be diagonally dominant, reflecting the high likelihood that the target will remain in its current motion state rather than abruptly switching at any given time step. The off-diagonal transition probabilities are set uniformly to 0.05. This uniformity reflects an uninformative prior assumption regarding specific maneuver transitions, since—without map information or knowledge of driver intent—there is no statistical basis to assume that a transition from NCV to CT is more or less likely than a transition to CTRA. Therefore, equal off-diagonal probabilities ensure that the filter remains unbiased and equally responsive to any potential change in dynamics, while still maintaining sensitivity to possible maneuvers.

Furthermore, the initial model probabilities are set with a strong bias toward the NCV model, based on the logical assumption that the vehicle begins its trajectory in a relatively stable, non-maneuvering state. The process and measurement noise parameters have been tuned to achieve a balance between tracking accuracy and filter robustness, enabling the more complex models to react effectively to maneuvers without introducing excessive noise into the final state estimates.

For reproducibility, the filter initialization details are as

follows: The state vectors are initialized using the first ground truth data point to isolate the tracking performance from initial transient errors. The initial error covariance matrices ($P_0$) are set diagonally. For the NCV model, $P_{0,NCV} = diag(0.5,0.5,1.0,1.0)$. For the CT model, $P_{0,CT} = diag(0.5,0.5,1.0,0.1,0.2)$. For the CTRA model, $P_{0,CTRA} = diag(0.5,0.5,1.0,0.1,0.2,1.0)$. Furthermore, the process noise covariance matrix $Q_{NCV}$ is implemented using the discrete white noise acceleration model, incorporating terms of $\Delta t^3/3$ and $\Delta t^2/2$."

### C. Evaluation Metrics

To rigorously assess the performance of the proposed IMM-EKF tracker, a comprehensive set of evaluation metrics is employed. These metrics are categorized into two groups: accuracy metrics, which quantify the closeness of the estimated trajectory to the ground truth, and consistency metrics, which evaluate the statistical validity of the filter's own uncertainty estimates.

- Root Mean Square Error (RMSE)

This is the primary metric for evaluating the overall tracking error. It is calculated for position, velocity, and yaw. For a total of N time steps, the position RMSE is defined as (30):

$$RMSE_{pos} = \sqrt{\frac{1}{N}\sum_{k=1}^{N}\left(p_{x,k} - \hat{p}_{x,k}\right)^2 + \left(p_{y,k} - \hat{p}_{y,k}\right)^2} \quad (30)$$

where $(p_x, p_y)$ is the true position and $(\hat{p}_x, \hat{p}_y)$ is the estimated position.

- Mean Absolute Error (MAE)

This metric (31) measures the average magnitude of position errors and is less sensitive to large, infrequent outliers compared to RMSE.

$$MAE_{pos} = \frac{1}{N}\sum_{k=1}^{N}\sqrt{\left(p_{x,k} - \hat{p}_{x,k}\right)^2 + \left(p_{y,k} - \hat{p}_{y,k}\right)^2} \quad (31)$$

- Final Displacement Error (FDE)

This metric measures the position error at the final time step of the trajectory, providing insight into the long-term drift of the tracker (32).

$$FDE = \sqrt{\left(p_{x,N} - \hat{p}_{x,N}\right)^2 + \left(p_{y,N} - \hat{p}_{y,N}\right)^2} \quad (32)$$

- Normalized Innovation Squared (NIS)

The NIS test evaluates whether the measurement residual (innovation) is consistent with its theoretical covariance. For a measurement of dimension m, the NIS value is calculated as (33):

$$\epsilon_{NIS,k} = y_k^T S_k^{-1} y_k \quad (33)$$

where $y_k$ is the innovation and $S_k$ is its covariance. Under the assumption of a correct model, the NIS values follow a Chi-squared ($\chi^2$) distribution with $m$ degrees of freedom.

- Normalized Estimation Error Squared (NEES)

The NEES test assesses the consistency of the state estimate itself by measuring the squared Mahalanobis distance between the estimated and true states. For a state vector of dimension n, the NEES is given by (34):

$$\epsilon_{NEES,k} = (x_k - \hat{x}_k)^T P_k^{-1} (x_k - \hat{x}_k) \quad (34)$$

where $(x_k - \hat{x}_k)$ is the true state error and $P_k$ is the estimated state covariance. Similar to NIS, the NEES values should follow a $\chi^2$ distribution with $n$ degrees of freedom for a consistent filter.

## IV. Results and Analysis

This section presents a comprehensive evaluation of the proposed IMM-EKF tracker by applying it to the synthetically generated trajectory described in the experimental setup. The analysis is structured to provide a multifaceted assessment of the filter's performance. It begins with a qualitative comparison of the estimated trajectory against the ground truth, followed by an in-depth analysis of the IMM framework's core adaptive mechanism—the dynamic evolution of model probabilities in response to target maneuvers. Subsequently, a detailed quantitative analysis of tracking accuracy is conducted by examining the position, velocity, and yaw errors over time. Finally, the statistical consistency of the filter is rigorously validated using the NIS and NEES tests to ensure the reliability of the filter's uncertainty estimates. Collectively, these results demonstrate the robustness and accuracy of the proposed multi-model approach for tracking maneuvering targets.

Fig. 2 presents a visual comparison between the ground-truth trajectory and the path estimated by the proposed IMM-EKF tracker. The true path is depicted by a solid blue line, while the filter's estimate is shown as a dashed red line. The scenario is designed to be challenging, encompassing straight sections, a gentle S-curve, and two sharp 90-degree turns.

A qualitative inspection of the results reveals an exceptionally close correspondence between the estimated and true paths throughout the entire trajectory. This visual evidence is strongly supported by quantitative metrics, with the overall Position RMSE calculated at only 0.3117 m. The tracker demonstrates remarkable robustness, accurately following the target through both the linear motion segments and the highly dynamic maneuvering phases. The minimal deviation, even during the sharp turns where motion models are most likely to fail, highlights the effectiveness of the adaptive IMM framework. In summary, Fig. 2 provides compelling evidence for the high accuracy and stability of the proposed algorithm.

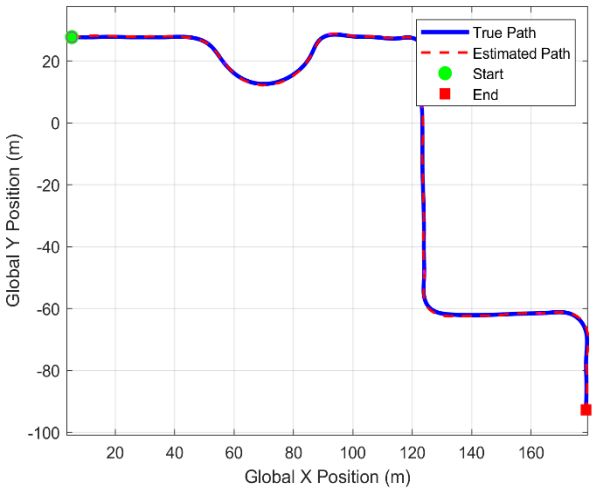


Fig. 2. Comparison of the estimated trajectory (dashed red line) from the IMM-EKF tracker against the ground truth path (solid blue line), demonstrating a high degree of tracking accuracy.

The IMM framework's dynamic adaptation is illustrated in Fig. 3, plotting each model's posterior probability over time. A model's probability increases when its predictions align with sensor measurements, indicating that it best describes the target's current behavior. Following an initial transient phase where the models fluctuate, the NCV model (blue) dominates during the stable straight segments.

Crucially, during complex maneuvers such as the S-curve and the transitional phases entering or exiting turns, transient spikes in the CTRA model's probability (red) demonstrate its essential role in capturing the complex dynamics involving simultaneous changes in speed and turn rate. Conversely, the CT model (green) becomes dominant during periods of sustained turning, such as the steady phases of 90-degree corners. This logical transition of probabilities between models is the fundamental mechanism enabling the IMM-EKF's superior tracking accuracy across the challenging trajectory.

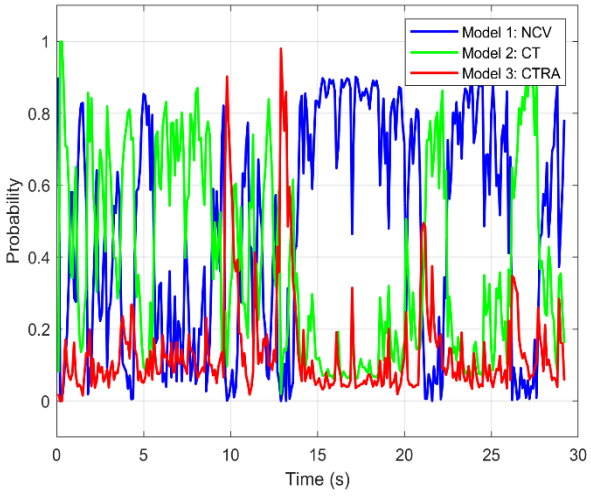


Fig. 3. Time evolution of the model probabilities for the NCV, CT, and CTRA models, illustrating the adaptive weighting mechanism of the IMM-EKF in response to target maneuvers.

A quantitative analysis of the filter's accuracy is provided in Fig. 4, which illustrates the magnitude of the position error over the duration of the trajectory. The error remains well-bounded throughout the entire run, with no evidence of divergence, which confirms the stability of the filter. The peaks observed in the error profile, such as the one around 22 seconds, are associated with the most challenging maneuvers in the trajectory (e.g., sharp turns). These temporary increases in error are expected as the filter adapts its model probabilities in response to abrupt changes in the target dynamics.

Despite these transient peaks, the tracker rapidly reconverges, keeping the error at a consistently low level. The visual data is supported by the overall performance metrics, which show a Maximum Position Error of 0.6722 m—aligning with the highest peak in the figure—and a Position RMSE of only 0.3117 m. The consistently low error magnitude across a trajectory with diverse dynamics underscores the tracker's high precision and its effective use of the multi-model framework to mitigate errors during maneuvers.

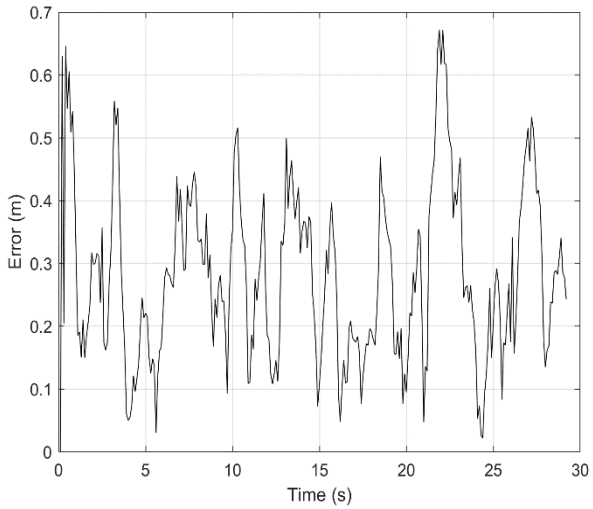


Fig. 4. Position error magnitude of the IMM-EKF tracker over time, demonstrating the filter's stability and high accuracy.

The analysis is extended to kinematic states beyond position in Fig. 5, which compares the true target speed with the IMM-EKF-estimated speed. The figure clearly shows that the estimated speed (dashed red line) accurately tracks the true speed profile (solid blue line) through all its dynamic variations.

The filter demonstrates a high degree of responsiveness, successfully capturing the initial sharp deceleration, the subsequent periods of acceleration and deceleration corresponding to maneuvers, and the final acceleration at the end of the trajectory. The minor oscillations observed in the estimated speed are an inherent artifact of the filtering process, resulting from the continual state update under measurement noise. The excellent tracking performance is quantitatively confirmed by a low Velocity RMSE of 0.5428 m/s. This result validates the effectiveness of including the CTRA model, which explicitly accounts for linear acceleration, enabling the filter to accurately estimate the target's speed profile even during aggressive maneuvers.

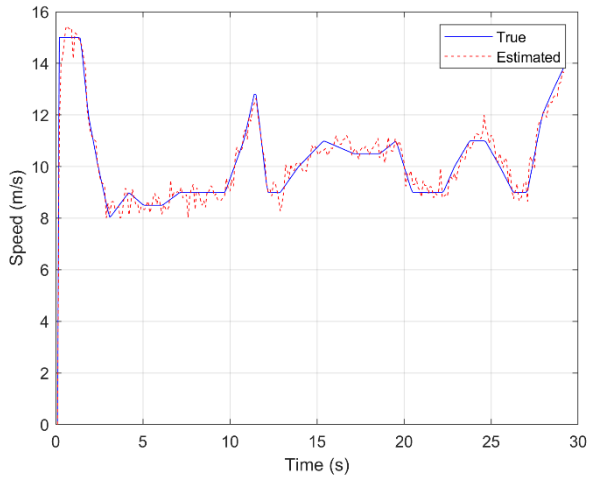


Fig. 5. Comparison of the true and estimated target speed over time, demonstrating the filter's ability to accurately track dynamic changes, including accelerations and decelerations.

Fig. 6 evaluates the filter's performance in estimating the target's orientation by comparing the true yaw (heading) angle with the estimated value. The plot demonstrates very high degree of accuracy, with the estimated heading (dashed red line) closely overlaying the true heading (solid blue line) throughout the entire scenario.

The filter successfully tracks both slow and rapid changes in the target's orientation. Most notably, it accurately captures the large, sweeping changes in yaw corresponding to the sharp 90-degree turns, where the heading changes by approximately 90 degrees in a short time frame. Accurate heading estimation is critical for maneuver recognition and short-term path prediction. The filter's ability to perform well in this regard highlights the effectiveness of the turn-based models (CT and CTRA). This visual assessment is quantitatively substantiated by a very low Yaw RMSE of 2.1614 degrees, confirming that the IMM-EKF provides a reliable and precise estimate of the target's heading, even during aggressive maneuvering.

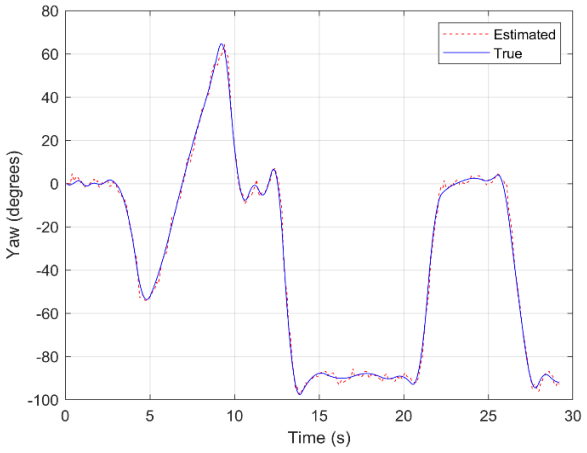


Fig. 6. Comparison of the true and estimated target yaw (heading) over time, highlighting the filter's high fidelity in tracking orientation during maneuvers.

To provide a more granular view of the tracking performance, Fig. 7 decomposes the total position error into its individual Cartesian components along the X and Y axes. Both subplots show that the estimation errors are zero-mean, indicating that the filter is unbiased and does not exhibit any systematic drift in either direction.

The magnitude of the error in both the X and Y components remains well-bounded within approximately $\pm 0.5$ meters for the entire duration. The fluctuations in the error signals are directly correlated with the target's dynamics; larger oscillations are observed during maneuvering periods, while the error tends to be smaller and more stable during straight-line motion. This detailed breakdown confirms that the low overall position error, as previously reported by the RMSE, is not masking poor performance in one dimension with a strong performance in another. Instead, the IMM-EKF demonstrates consistent and robust tracking accuracy in both axes independently, further validating the stability and precision of the estimation process.

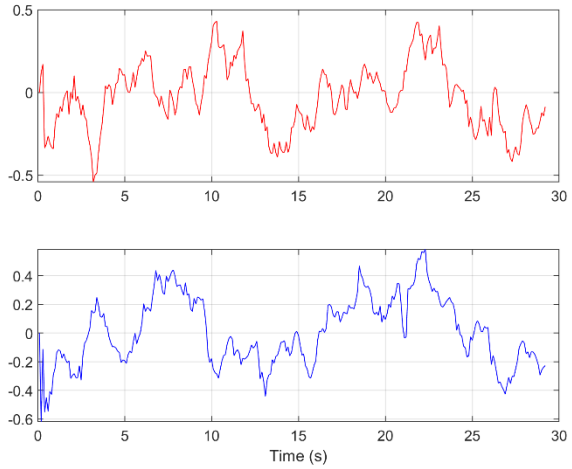


Fig. 7. Position error components along the X-axis (top) and Y-axis (bottom) over time, showing the unbiased and well-bounded nature of the state estimation.

Beyond tracking accuracy, it is crucial to validate the statistical consistency of the filter. Fig. 8 presents the results of the NIS test, which assesses whether the filter's model for measurement uncertainty is accurate. The plot displays the calculated NIS values at each time step against the 95% confidence bound (dashed red line). This bound is derived from a Chi-squared ($\chi^2$) distribution with 4 degrees of freedom, corresponding to the four-dimensional measurement vector (position and velocity).

For a well-tuned and statistically consistent filter, approximately 95% of the NIS values are expected to fall below this confidence threshold. The results show excellent adherence to this condition, with the vast majority of points lying within the valid region. This visual assessment is quantitatively confirmed by the fact that 94.16% of the NIS values are within the 95% confidence interval, a figure remarkably close to the theoretical target. Furthermore, the average NIS value across the trajectory is 4.3941, which is very close to the theoretical mean of 4 for a $\chi^2$ distribution with 4 degrees of freedom. These results collectively confirm that the filter is not over- or underconfident in its predictions and that its innovation covariance is a statistically sound representation of the true measurement uncertainty.

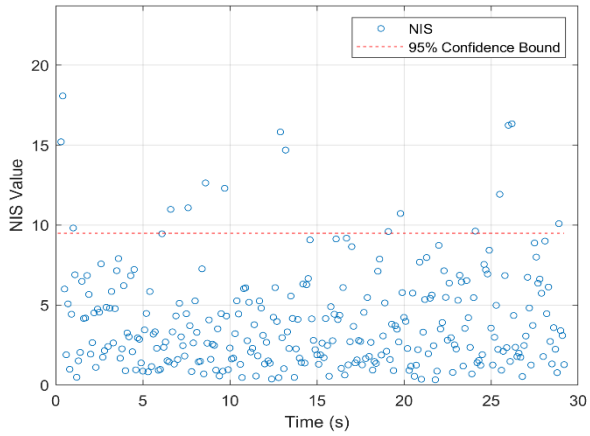


Fig. 8. NIS consistency test results, with NIS values (blue circles) plotted against the 95% confidence bound (dashed red line), confirming the statistical consistency of the filter's measurement uncertainty model.

The final validation step involves the NEES test, which directly assesses the consistency of the filter's state covariance matrix, $P_k$. Fig. 9 plots the NEES values for the four-dimensional Cartesian state (position and velocity) against the theoretical 95% confidence bound from a $\chi^2$ distribution with 4 degrees of freedom. This test is critical as it confirms whether the filter's own uncertainty estimate is a reliable measure of the true state error.

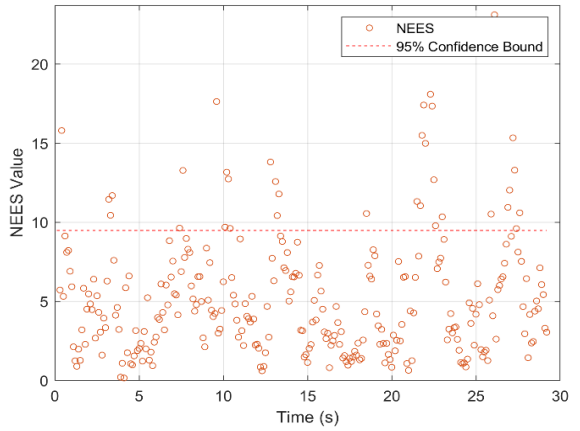


Fig. 9. NEES consistency test results for the 4-DOF Cartesian state. The NEES values (orange circles) are shown against the 95% confidence bound, validating the statistical consistency of the filter's state covariance matrix.

While the observed NEES consistency of 87.29% is below the theoretical 95% confidence level, this deviation arises from several intrinsic factors associated with the filtering process. During aggressive maneuvers or sudden changes in target dynamics, temporary mismatches between the assumed motion models and the actual system behavior can produce transient underestimation or overestimation of the state uncertainty. In addition, NEES is highly sensitive to finite sample effects and the stochastic properties of both process and measurement noise; limited trajectory length, abrupt state transitions, and random fluctuations in measurements naturally introduce variability in NEES values. Model simplifications, such as linearization approximations in the EKF and the finite set of motion models in the IMM framework, can also contribute to the underestimation of uncertainty during highly nonlinear or rapidly changing scenarios. To reduce such deviations, careful selection of model sets, inclusion of more representative motion models, and increasing the effective sample size through longer trajectories or repeated simulations can help stabilize NEES consistency and better align the estimated uncertainty with actual state errors. Despite the observed deviation, the average NEES of 5.5006, reasonably close to the theoretical mean of 4, along with consistent NIS results, confirms that the IMM-EKF provides reliable covariance estimates and maintains a statistically meaningful relationship between estimated uncertainty and actual state error.

Collectively, the analysis presented in Figs 2 through 9 confirms that the proposed IMM-EKF tracker provides a highly accurate, robust, and statistically consistent solution for maneuvering target tracking. The results have demonstrated the filter's capability to maintain low estimation errors while correctly adapting its internal models to the target's dynamics. To provide a concise, holistic view of the tracker's performance, the key quantitative metrics are consolidated and presented in Tables II and III. Table II summarizes the positional estimation accuracy metrics, including RMSE, MAE, and FDE. Meanwhile, Table III covers the wider estimation performance and consistency, detailing velocity error, yaw accuracy, and statistical tests (NEES and NIS). These tables numerically substantiate the qualitative findings from the graphical analysis.

TABLE II
Summary of Positional Estimation Accuracy Metrics

| Pos. RMSE (m) | Pos. MAE (m) | Max Pos. Err (m) | FDE (m) |
|---|---|---|---|
| 0.3117 | 0.2794 | 0.6722 | 0.2429 |

TABLE III
Estimation Performance and Consistency

| Vel. RMSE (m/s) | Yaw RMSE (deg) | NEES. Cons (%) | NIS. Cons (%) |
|---|---|---|---|
| 0.5428 | 2.1614 | 87.29 | 94.16 |

## V. Conclusion and Future Work

This paper presented a robust and adaptive solution for tracking maneuvering targets using an IMM-EKF. By synergistically combining three distinct kinematic models — NCV, CT, and CTRA — the proposed filter demonstrated a remarkable ability to accurately estimate a target's trajectory across a wide range of dynamic behaviors. The effectiveness of the algorithm was rigorously validated on a challenging, synthetically generated dataset that included straight-line motion, sustained turns, and aggressive maneuvers.

The experimental results confirmed the high performance of the tracker, achieving a very low Position RMSE of 0.3117 m and Yaw RMSE of 2.1614 degrees. The analysis of the model probabilities showcased the filter's core adaptive mechanism, which correctly inferred the target's motion mode in real-time. Furthermore, the filter's statistical consistency was successfully

verified through NIS and NEES tests, with an NIS consistency of 94.16%, confirming that the filter's uncertainty estimates are both reliable and statistically sound. In summary, this work validates the IMM-EKF as a highly effective and reliable framework for high-precision trajectory estimation.

While the proposed method has shown excellent results, several avenues for future investigation exist. The EKF can be susceptible to errors from linearization, especially in highly nonlinear systems. Future work could involve replacing the EKF with more advanced non-linear filters, such as the UKF [21] or a Particle Filter [22], which may offer improved accuracy during very aggressive maneuvers. The process and measurement noise covariances were hand-tuned. A significant enhancement would be to implement an adaptive noise estimation algorithm that can adjust these matrices online, making the filter more robust to varying sensor conditions and target behaviors. The current study was based on a synthetic dataset. The next logical step is to validate the algorithm's performance using real-world data collected from sensors such as GPS, IMU, and LiDAR to assess its effectiveness in practical, real-world scenarios.

## Funding Statement

This research did not receive any specific grant from funding agencies in the public, commercial, or not-for-profit sectors.

## Conflicts of Interest

The author declares that there is no conflict of interest regarding the publication of this article.

## STATEMENT ON THE USE OF GENERATIVE AI

Gemini was used for language editing and clarity enhancement.

## References

[1] S. Zargari, A. Jarrah, and F. Baghbani, "Fuzzy-based driver monitoring system to assess dangerous driving on roads," *Traffic Injury Prevention*, pp. 1-10, 2025, doi: 10.1080/15389588.2025.2539923.

[2] S. Zargari, A. Jarrah, and F. Baghbani, "Video-based Facial Expression Recognition Using DensNet121 and LSTM," in *The First Conference on Artificial Intelligence and Intelligent Processing*, 2022.

[3] D.-J. Jwo, J.-H. Lai, and Y. Chang, "Interacting multiple model filter with a maximum correntropy criterion for gps navigation processing," *Applied Sciences*, vol. 13, no. 3, p. 1782, 2023, doi: 10.3390/app13031782.

[4] S. A. Gadsden, S. R. Habibi, and T. Kirubarajan, "A novel interacting multiple model method for nonlinear target tracking," in *2010 13th International Conference on Information Fusion*, 2010: IEEE, pp. 1-8, doi: 10.1109/ICIF.2010.5712021.

[5] S. M. Aly, R. El Fouly, and H. Braka, "Extended Kalman filtering and interacting multiple model for tracking maneuvering targets in sensor networks," in *2009 Seventh Workshop on Intelligent solutions in Embedded Systems*, 2009: IEEE, pp. 149-156.

[6] W. Zhu, W. Wang, and G. Yuan, "An improved interacting multiple model filtering algorithm based on the cubature Kalman filter for maneuvering target tracking," *Sensors*, vol. 16, no. 6, p. 805, 2016, doi: 10.3390/s16060805.

[7] L. Gao, J. Xing, Z. Ma, J. Sha, and X. Meng, "Improved IMM algorithm for nonlinear maneuvering target tracking," *Procedia Engineering*, vol. 29, pp. 4117-4123, 2012, doi: 10.1016/j.proeng.2012.01.630.

[8] W. Li and Y. Jia, "Location of mobile station with maneuvers using an IMM-based cubature Kalman filter," *IEEE Transactions on Industrial Electronics*, vol. 59, no. 11, pp. 4338-4348, 2011, doi: 10.1109/TIE.2011.2180270.

[9] M. Wan, P. Li, and T. Li, "Tracking maneuvering target with angle-only measurements using IMM algorithm based on CKF," in *2010 International Conference on Communications and Mobile Computing*, 2010, vol. 3: IEEE, pp. 92-96, doi: 10.1109/CMC.2010.239.

[10] K. Vedula, M. L. Weiss, R. C. Paffenroth, J. R. Uzarski, and D. R. Brown, "Maneuvering target tracking using the autoencoder-interacting multiple model filter," in *2020 54th Asilomar Conference on Signals, Systems, and Computers*, 2020: IEEE, pp. 1512-1517, doi: 10.1109/IEEECONF51394.2020.9443396.

[11] M. Ebrahimi, M. Ardeshiri, and S. A. Khanghah, "Bearing-only 2D maneuvering target tracking using smart interacting multiple model filter," *Digital Signal Processing*, vol. 126, p. 103497, 2022, doi: 10.1016/j.dsp.2022.103497.

[12] X. Liu *et al.*, "IMM-MOT: A Novel 3D Multi-object Tracking Framework with Interacting Multiple Model Filter," *arXiv preprint arXiv:2502.09672*, 2025, doi: 10.48550/arXiv.2502.09672.

[13] J. Wang, X. Wang, Y. Chen, M. Yan, and H. Lan, "Model Adaptive Kalman Filter for Maneuvering Target Tracking Based on Variational Inference," *Electronics*, vol. 14, no. 10, p. 1908, 2025, doi: 10.3390/electronics14101908.

[14] Y. Dong, W. Li, D. Li, C. Liu, and W. Xue, "Intelligent Tracking Method for Aerial Maneuvering Target Based on Unscented Kalman Filter," *Remote Sensing*, vol. 16, no. 17, p. 3301, 2024, doi: 10.3390/rs16173301.

[15] J. Huang, J. Xie, H. Zhai, Z. Li, and W. Feng, "An Adaptive Constant Acceleration Model for Maneuvering Target Tracking," *Remote Sensing*, vol. 17, no. 5, p. 850, 2025, doi: 10.3390/rs17050850.

[16] S. Wang, R. Li, C. Men, Y. Cao, and T.-S. Yeo, "Adaptive IMM Algorithm Based on Variational Inference for Multiple Maneuvering Extended Targets Tracking," *Advances in Astronautics*, pp. 1-15, 2025, doi: 10.1007/s42423-025-00173-7.

[17] H. A. Blom and Y. Bar-Shalom, "The interacting multiple model algorithm for systems with Markovian switching coefficients," *IEEE Transactions on Automatic Control*, vol. 33, no. 8, pp. 780-783, 2002, doi: 10.1109/9.1299.

[18] E. A. Wan and R. Van Der Merwe, "The unscented Kalman filter for nonlinear estimation," in *Proceedings of the IEEE 2000 adaptive systems for signal processing, communications, and control symposium (Cat. No. 00EX373)*, 2000: IEEE, pp. 153-158, doi: 10.1109/ASSPCC.2000.882463.

[19] J. J. LaViola, "A comparison of unscented and extended Kalman filtering for estimating quaternion motion," in *Proceedings of the 2003 American Control Conference, 2003.*, 2003, vol. 3: IEEE, pp. 2435-2440, doi: 10.1109/ACC.2003.1243440.

[20] A. Valade, P. Acco, P. Grabolosa, and J.-Y. Fourniols, "A study about Kalman filters applied to embedded sensors," *Sensors*, vol. 17, no. 12, p. 2810, 2017, doi: 10.3390/s17122810.

[21] S. J. Julier and J. K. Uhlmann, "New extension of the Kalman filter to nonlinear systems," in *Signal processing, sensor fusion, and target recognition VI*, 1997, vol. 3068: Spie, pp. 182-193, doi: 10.1117/12.280797.

[22] N. J. Gordon, D. J. Salmond, and A. F. Smith, "Novel approach to nonlinear/non-Gaussian Bayesian state estimation," in *IEE proceedings F (radar and signal processing)*, 1993, vol. 140, no. 2: IET, pp. 107-113, doi: 10.1049/ip-f-2.1993.0015.

## Biographies

**Alireza Jarrah** received his B.Sc. degree (with Honors) in Electrical Engineering (Control) from Semnan University, Semnan, Iran, in 2023. He is currently pursuing his M.Sc. in Electrical Engineering (Control) at Shahid Beheshti University (SBU), Tehran, Iran. His research interests include autonomous vehicles, fuzzy systems, and artificial intelligence.

**Hadi Asharioun** is a faculty member at Shahid Beheshti University (SBU) in Tehran, Iran. He earned his B.Sc. from Sharif University of Technology and his M.Sc. from Iran University of Science and Technology. In 2014, he received his Ph.D. from Universiti Sains Malaysia, Pulau Pinang, Malaysia. His research interests include computer and industrial networks, the Internet of Things, artificial intelligence, machine learning, and wireless sensor networks.

**Mohammadhossein Hashemi** is currently a Ph.D. student in Electrical Engineering (Control) at Shahid Beheshti University (SBU), Tehran, Iran. He received his M.Sc. in Electrical Engineering (Control) and his B.Sc. in Computer Engineering, both from Shahid Beheshti University. His research interests include observer design.