Zie Eya Ekolle ¹ and Ryuji Kohno¹

¹Affiliation not available

December 22, 2023
CoPreMo: A Collaborative Predictive Model in Time Series and its Application to Radar Target Tracking for ADAS/AD Vehicles

Zie Eya Ekolle, Member, IEEE, Ryuji Kohno, Life Fellow, IEEE

Abstract—The use of radar sensors in the detection and ranging of targets is an important technology that plays a leading role in the operation of many modern technologies such as the automotive driving assistant systems (ADAS) and the automated driving (AD) technologies. ADAS/AD are technologies that enable unmanned vehicle control along a trajectory. Some of the challenges of using these technologies in vehicles include the risk of misdirection and collision of the vehicle with an obstacle along its trajectory. To avoid these, many technologies such as radar are being used to detect and track targets and trajectories of ADAS/AD vehicles. In this study, we focus on radar tracking technologies and propose a collaborative predictive model in time series, called CoPreMo, for this purpose. We carried out experiments with the model on a simulated radar system to track the range of a target in an ADAS/AD scenario and achieved a range tracking performance that surpasses those of the presented baseline models.

Index Terms—target tracking, radar sensors, ADAS, automated driving, collaborative models, time-series prediction.

1 INTRODUCTION

The testing and use of ADAS/AD vehicles are no longer a future but a present we are experiencing. On the 30th of May 2022, the United Nations Economic and Social Council (ECOSOC) adopted a resolution to extend automated driving in certain traffic environments from the current limit of 60 km/h to up to 130 km/h [1]. This is an important improvement for the use of ADAS/AD vehicles but at the same time presents a new milestone for ADAS/AD vehicle technology. Coupled with the fact that most cars will be fully autonomous by 2030 as predicted in [2], the pressure to meet standards and demands will be huge for ADAS/AD vehicle stakeholders.

One advantage of promoting such technology is the fact that it will eliminate many traffic accidents caused by human factors, such as tiredness, drunkenness, fading memory, poor sight, and so on. However, such innovations cannot be reliable if their performance is not well optimized, because this may instead lead to more accidents. Furthermore, while this trend in ADAS/AD vehicles may not stop because of the significance of these technologies to automation as explained in [3], one of the main risks posed by these technologies in vehicular automation is the risk of collision in a traffic environment.

Actually, vehicle collision is one of the major risk concerns for both manned and unmanned vehicles. Reports from the World Health Organization (WHO) [4] and the International Transport Forum (ITF) [5] show that about 1.3 million people die every year due to manned vehicle collision. While the advert and use of unmanned vehicles equipped with ADAS/AD technologies promised some solution to the problem, there is still risk related to their technology. The failure of such technology can lead to more disastrous collisions and loss of more human lives. This implies the reliability of such technology needs to be optimal to increase human trust and safety. Amongst the technologies used in ADAS/AD vehicles as explained in [6] is the radar target tracking technology.

Technically, target tracking technologies are used to estimate the future value of a target’s property in time series given the present value, while memorizing the past values. This property can vary depending on the tracking application and sensor. In the case of radar applications, the property can be the range, speed, and/or angle of a target. In this study, we focus on range tracking. In a radar system, the target tracking technology is an additional function to the target detection and ranging functions, and it is carried out after the target detection and ranging functions. The general architecture of a radar system is presented in Figure [1] together with common algorithms as explained in [7].

As shown in Figure [1], detected signals related to the target are captured by the receiving antenna and processed to remove noise. The denoised and pre-processed signal is then used to estimate the range of the target using different techniques such as the Fast Fourier Transform (FFT). The estimated range is then passed to the tracker where predictive operations about the range of the target are performed. The tracker collects the series of range values in time series and uses them to predict the next range value of the target. The tracker also memorized the past predicted values to keep track of the target over time. In some cases, the output of the tracker is sent to the controller which enables the vehicle to take action concerning the tracked target. This action may include stopping the vehicle, slowing its speed, or changing its direction of motion, to avoid a collision. In this study, we focus only on the tracking function.

Generally, target-tracking models can be classified into different categories; linear or non-linear, single reflection point or multiple reflection points, single-target or multi-target, single-property or multi-property, single-modal or multi-modal, and state-estimator-based or target-motion-based models. There are many types of target-tracking models used in research and industry, but irrespective of their types and functionality, they all fall into one or more of these categories.

Linear target-tracking (LTT) models capture the linear rela-
tionship between the present and future property value of the target. Common examples of this model include the $\alpha - \beta$ filter, the Gaussian filter (GF), and the linear Kalman filter model (KF). However, such a relationship is not mostly linear, leading to the use of non-linear target tracking (NLTT) models, which capture the non-linear relationship between the past and future property value of the target. Common examples of NLTT include the Bayes filter (BF), the extended Kalman filter (EKF), the unscented Kalman filter (UKF), and the cubature Kalman filter (CKF). The Kalman-based nonlinear filters mostly use the white Gaussian noise to characterize the noise statistics of the tracking process.

Single-point source (reflection point) target tracking (SP-STT) models assume a single detection from each target, while multiple-point source target tracking (MP-STT) models allow multiple detections to originate from each target. In reality, a single target usually generates multiple reflection points for the radar to process and track. Rather than using all these points, some tracking models use properties of the same modality, while others use properties with different modalities. The former is considered a single-modal tracking (SMT) model, while the latter is a multi-modal tracking (MPT) model. Single-property tracking (SPT) models use a single property to track a target, while multiple-property tracking (MPT) models use multiple properties to track a target. In general, most target tracking models can be upgraded from single-property to multiple properties but with varying resource performance requirements.

A special type of property-based tracking categorization is the modality of the property. The modality of the tracking property defines the data type of the property value captured by the tracking sensor. These data types can be image data, audio data, or kinematic data of the target, captured using a camera, sonar, and radar sensors, respectively. Some tracking models use properties of the same modality, while others use properties with different modalities. The former is considered a single-modal tracking (SMT) model, while the latter is a multi-modal tracking (MMDT) model.

Lastly, some models are built as state estimators without taking into account the motion model of the target, while others are built using a specific target motion model. The former is considered as state-estimator-based tracking models and includes models such as the recursive Bayesian estimator (GNNT), and interactive multiple model tracker (IMMT).

In some target motions, many properties may be required to effectively track the target, while in other cases, just a single property is enough to track the target. Generally, the higher the number of properties used to track a target the more statistically performant (accuracy, precision, and error) the tracking process will be, however, more computational and memory resources will be needed to process the many properties. The multiple properties are usually of different spatial dimensions such as distance and speed, or different directions such as front and rear distance with the sensor (e.g., radar sensor). Single-property tracking (SPT) models use a single property to track a target, while multiple-property tracking (MPT) models use multiple properties to track a target. In general, most target tracking models can be upgraded from single-property to multiple properties but with varying resource performance requirements.
(i.e., Bayes filter), Kalman filter estimator, and Particle filter, while the latter is considered as target-motion-based models and includes models such as the constant velocity (CV) and constant acceleration (CA) models. State estimators are prolific in tracking models and can further be categorized into parametric models such as Kalman filters and non-parametric models such as Bayes filters and Particle filters.

In this study, we focus on a non-linear, single-point source, single-target, single-property, single-modal, state-estimator-based target tracking model. The main contributions of this study consist of a collaborative tracking technique in time series based on partial tracking actions, an algorithm of the tracking process, and an experiment to prove its significance.

The rest of the article is summarized as follows, section 2 discusses related literature about radar target tracking. Section 3 focuses on the modeling techniques used to implement the proposed tracking model together with those of conventional models. Section 4 focuses on the experimental results and discussions. Section 5 is the conclusion of the study.

2 Related Works

There have been many related studies on radar target tracking in the research literature and industry.

Cao et al. [8], proposed an automotive radar-based extended object tracker that jointly estimates the kinematic state and the extension of a vehicle. Their technique involves using a rectangular shape to describe the target and partitioning the rectangle into multiple regions while assuming that at each region the scattering centers have a simple distribution.

Qiao et al. [9], proposed an optimized Sage–Husa adaptive Kalman filter (SHAKF) for eliminating white noise triggered by an unmanned surface vehicle (USV) and target oscillations. They achieved this through a novel square root decomposition method in the SHARKF algorithm for decomposing the covariance matrix of SHAKF to assure its non-negative definiteness.

Kim et al. [10], proposes an extended Kalman filter (EKF) that fuses the distance characteristics of lidar and radar sensors to track the position of a target. The tracking accuracy of position was improved by identifying the sensor errors according to distance.

Honer et al. [11], proposed a closed-form Bayesian recursion for tracking an extended target by the use of random set cluster process and the spatial distribution of measurements generated by a target vehicle is learned via a variational Gaussian mixture (VGM) model.

Zhai et al. [12], proposed a target tracking model based on adaptive Sage–Husa Kalman filter algorithm to track radar signals. Their proposed model can estimate the real-time state of the system together with the ability to modify the parameters of the system and its statistical noise, so that the model is closer to the current real state of the system, thus improving the accuracy of the target tracking.

Thormann et al. [13], proposed an extended radar target tracking model based on an extended Kalman filter and using a Gaussian Process (GP) to model the shape of the target. Their tracking model uses the cartesian point measurements from the target’s contour as well as the Doppler range rate provided by the radar to track a target vehicle’s position, orientation, and translational and rotational velocities.

Gunnarsson et al. [14], proposed a multiple-point source target tracking model where similar hypotheses are joined into groups to solve the high complexity problem in data association algorithms with a large number of association hypotheses. Based on this model, basic data association algorithms can be implemented with less complexity.

3 Modeling

Many approaches have been proposed to track the range of targets using radars. In this section, we present the conventional approach and our proposed approach.

3.1 Conventional approach

There are many conventional approaches to target tracking using radars. We shall focus on non-linear approaches, precisely the Bayes filter, which is the foundational model from which many filters, including the Kalman filter and Particle filter, can be derived.

The Bayes filtering process is a recursive estimation process that consists of two steps: the prediction step and the update step. The prediction step uses the state transition probability \( P(x_{i} | x_{i-1}) \) to predict the next posterior state of the property. The state transition probability \( P(x_{i} | x_{i-1}) \) is the probability of a target’s property in the state \( x_{i-1} \) at time \( i - 1 \) to transition to the state \( x_{i} \) at time \( i \). The update step consists of the measurement model, which is based on the sensor measurements, to update (i.e., revise) the previous prediction.

\[
P(x_{i} | z_{1:k}) = \int P(x_{i} | z_{1:k}) \, dP(x_{i} | z_{1:k})
\]

Fig. 2. A Bayes filter model for target tracking.

Generally, the Bayes filter as shown in Figure 2 is based on the first-order Markov assumption, which is an assumption that the future and past state of the system are independent given the present. In other words, the future state depends only on the present state and not the past state(s). This is mathematically defined as follows

\[
P(x_{i} | x_{i-1}, x_{i-2}, \ldots, x_{0}) = P(x_{i} | x_{i-1}) \tag{3.1}
\]

where \( P(\cdot) \) is a probability function, \( x \) is the true state of the property, and \( i \) is the time step.

Using this assumption that the true states of the tracking property are related based on a Markov property, and considering the measurement state \( z_{i} \) of the property is a reflection of the true state \( x_{i} \), the measurement state \( z_{i} \) of the property

There have been many related studies on radar target tracking in the research literature and industry.
3.2 Proposed approach

Our proposed approach is based on a collaborative technique for model prediction. In this approach, the partial causal actions of a model(s) on a target are mutually aggregated or integrated to form a unified (collaborative) causal action on the target. The action in this study is to predict the future value of the range of a target in time series. The tracked property is a range of the target.

Our model is based on the following axiom and proposition

Axiom 3.1. (Conditional independence of measurement values of target property)

\[
P(z_k|x_{k-1} = x_k) = P(z_k|x_{k+1})
\]

Proposition 3.1.

\[
x_{k+1|k} = P(x_{k+1}|z_1:k) = \frac{1}{P(x_{k+1})} \prod_{i=1}^{k} P(x_{k+1}|z_i) \left( \frac{P(z_i|\sum_{\mu=1}^{i-1} z_{\mu})}{P(z_i)} \right)^{-1}
\]

where \( x_{k+1} \) is the future state value at time step \( k+1 \) of the tracked property, \( z = (z_1, z_2, \ldots, z_k) \) is the vector of the present state value \( z_k \) and past state value \( z_{1:k-1} \) of the tracked property, \( \hat{x}_{k+1} \) is the predicted future state value of the tracked property at time step \( k+1 \) given \( z \) at time steps \( 1:k \), \( P(x_{k+1}) \) is the prior probability of \( x \) at time step \( k-1 \), \( P(x_{k+1}|z_i) \) is the partial posterior probability of \( x \) given \( z_i \), \( P(z_i) \) is the prior probability of \( z_i \) based on \( \bigcap_{\mu=1}^{i-1} P(z_{\mu}) \), \( P(z_i|\bigcap_{\mu=1}^{i-1} z_{\mu}) \) is the posterior probability of \( z_i \) given \( \bigcap_{\mu=1}^{i-1} z_{\mu} \), \( i \) is the time steps (or sequential states) of the tracking process, and \( k \) is the cumulative time steps (or sequential states).

As earlier explained, the action to predict the future is defined by the collaboration of causal and mutual actions in different sequential states. The causal and mutual actions can be distinguished in Proposition 3.1 as follows

\[ C_i \triangleq P(x_{k+1}|z_i) \quad \text{(causal action)} \]

\[ M_i \triangleq P(z_i|\bigcap_{\mu=1}^{i-1} z_{\mu}) \quad P(z_i) \quad \text{(mutual action)} \]

Assuming a 1st order Markov property over \( M_i \),

\[ M_i = \frac{P(z_i|z_{i-1})}{P(z_i)} \]

Considering that \( \forall x, z \in \mathbb{R}^k \), a continuous distribution or any formal approximation is required to estimate \( C_i \) and \( M_i \). A common distribution that can be used is the Gaussian distribution, where \( (x, z) \sim \mathcal{N}(\mu, \sigma^2) \) and \( P(.) = \mathcal{N}(\mu, \sigma^2) \).

Applying \( C_i \) and \( M_i \) to Proposition 3.1 will result to

\[ C_{k+1} = \frac{1}{W_k} \prod_{i=1}^{k} C_i M_i, \quad \text{where} \quad W_k = P(x_{k+1})^{k-1} = c_k^{(k-1)} \]

Given that \( n \) is the number of available event space at any time step and \( n > 1 \), the tracking process can be divided into two main iterative steps, local estimation, and global (collaborative) update, as described below.

- Initialize the process at state \( k = 1 \), where \( x_1 \) is initialized to a random value with a probability \( P(x_1) = C_1 \), \( C_1 = P(x_2|z_1) \), \( M_1 = f(z_1) = 1 \), and \( W_1 = C_1^{(0)} = 1 \).
• Updates the state $k = 1$ with the global causal action $C_2$ using the initialized local values $C_1, M_1, W_1,$ and $z_1$ of state $k = 1; C_2 = C_1M_1/W_1 = C_1M_1.$

• Hops to the next state $k = 2$ and estimate the local actions $C_1, C_2, M_1, M_2,$ and $W_2$ using all previously hops measurements $z_{1:k-1}$ including $z_k$ of the current state $k = 2; C_1 = P(x_3|z_1), C_2 = P(x_3|z_2), M_1 = f(z_1) = 1, M_2 = f(z_2),$ and $W_2 = C_2^{(1)}.$

• Update the state $k = 2$ with the global causal action $C_3$ using those of the local actions $C_1, C_2, M_1, M_2, W_2,$ and $z_2$ of state $k = 2; C_3 = C_1C_2M_1M_2/W_2.$

• Continue updating the next global causal actions $C_{k-1}$ with the local estimates $C_k, M_k,$ and $W_k$ in the same way after hopping to the next state $k + 1,$ until $k = n.$

The implementation of the tracking process is presented in Figure 3 and described in Algorithm 1.

![Figure 3: A 1-step 1-D action hopping of the CoPreMo model on a target tracking process.](image)

As shown in Figure 3, once the property values (in this case range values) of the target are measured by the sensor (in this case radar), they enter the CoPreMo tracking system represented in Figure 3. In this system, the measured range values of the target are tracked sequentially. The tracking process consists of predicting the next range value of the target in a time series. To achieve this, a sequence of mutually chained instances of causal tracking actions is used.

The optimization of the tracking process can be done using either a model-free or model-based approach. In a model-free approach, the optimization of the estimation can be done by expanding the sequential memory allocation of the measurement values. In the model-based approach, a referenced true value of the property is used to train the model, such that less sequential memory allocation can be used. The former can be considered as an unsupervised approach to sequential estimation while the latter can be considered as a supervised approach. In this study, we focus on the unsupervised approach.

Furthermore, this variant of CoPreMo, as shown in Figure 3 and described in Algorithm 1, where the action hops sequentially over a fixed block of sequential event states is considered in this study as an "action hopping" technique. Also, the action uses a one-step hoping process, where the immediate past state is mutually related to the current state to predict the future state. This result is because of the application of the 1st-order Markov property to the mutual value. However, more than one-step hoping can also be used by extending this property. The importance of the action-hopping technique is that the action does not only predict the future input state but also captures the mutuality between the past and present input states. Lastly, the study focuses on 1-D property, where only one property is tracked.

Algorithm 1 Action Hopping Algorithm for target tracking

Require: $z$ (measurement), $n$ (number of event space)
Ensure: $C_{k+1} = P(x_{k+1}|z_{1:k})$

1. $C \sim N(\mu, \sigma^2), M_1 \leftarrow 1, C_1 \leftarrow 1,$
2. for $k = 1$ to $n$
   a. $C \leftarrow C/C$
   b. $M \leftarrow M/M$
   c. $C_k \leftarrow C, M_k \leftarrow M$
   d. $W_k \leftarrow C_{k-1}^{(1)}$
   e. $C_{k+1} \leftarrow \frac{1}{W_k}C_kM_k$

Once predicted, the value is sent to the deterministic phase to classify the range into ultra-short, short, medium, or long ranges using the discrete numeric values 1,2,3,4, respectively. Considering [15], [16], the range limits for each category are given as an ultra-short range (<30m), short-range (0-30m), mid-range (0-100m), and long-range (0-150m).

Algorithm 2 Deterministic operation for range classification

Require: $\hat{x}$ (predicted measurement)
Ensure: $c = [1, 2, 3, 4]$

if $\hat{x}_{k+1} < 30$ then
  $c \leftarrow 1$
else if $\hat{x}_{k+1} \leq 30$ then
  $c \leftarrow 2$
else if $\hat{x}_{k+1} \leq 100$ then
  $c \leftarrow 3$
else if $\hat{x}_{k+1} \leq 150$ then
  $c \leftarrow 4$
end if

One difference between the CoPreMo models and the Bayes filter models such as Kalman and particle filter is that the Bayes filter models have a recursive action process whereas CoPreMo models have a collaborative (or distributed) action process. This will influence their computational complexities as we shall show in the next section.

3.3 Performance measures and Analytical Comparisons

Different performance measures can be used to evaluate a tracking process as explained in [17], [18]. This includes statistical measures such as accuracy and error measures used mostly to evaluate the statistical performance of models. However, apart from such statistical performance measures, it is also important to evaluate the system performance of the model, that is the algorithmic or implementation performance of the model, and common measures under this category include the time and space complexity of the algorithm.
The statistical performance measures used in this study are the accuracy and root mean square error (RMSE) of the tracker. The system performance measures used are the time and space complexity based on the Big O notation. These performance measures are presented below, including the analytical comparison of the time and space complexity of the CoPreMo model and different Bayes filter models.

i) Statistical performance measures

The accuracy and RMSE are used in this study as the statistical performance measures. The accuracy measure captures the performance of the tracker at the discrete (classification) output, while the RMS captures the performance of the tracker at the continuous output. The classification is multi-class, with four range classes; ultra-short, short, medium, and long ranges. These are defined as

\[ \text{Accuracy} = \frac{P_c}{P_a} \]

where \( P_c \) is the correct predictions and \( P_a \) is all predictions.

\[ \text{RMSE} = \sqrt{\frac{\sum_{i=1}^{k} (x_i - \hat{x}_i)^2}{k}} \]

where \( x_i \) is the true value and \( \hat{x}_i \) is the predicted value.

ii) System performance measures

The time and space complexity of the Bayes filter and its variants differs based on the design technique of the algorithm used to implement the models. We consider a general worst-case scenario in defining their computational complexities. Furthermore, the time and space complexity of the CoPreMo model used in this study is based on Algorithm 1. The comparison of the time and space complexities for different models is presented in Table I as defined in [19]–[21].

<table>
<thead>
<tr>
<th>Models</th>
<th>Time Complexity</th>
<th>Space Complexity</th>
</tr>
</thead>
<tbody>
<tr>
<td>BF</td>
<td>( O(k^2 + d^2) )</td>
<td>( O(d) )</td>
</tr>
<tr>
<td>KF</td>
<td>( O(k^2 + d^2 + n) )</td>
<td>( O(d) )</td>
</tr>
<tr>
<td>EKF</td>
<td>( O(k^2 + d^2) )</td>
<td>( O(d) )</td>
</tr>
<tr>
<td>GSF</td>
<td>( O(mv^3k^3 + md^2) )</td>
<td>( O(v + d) \approx O(v) )</td>
</tr>
<tr>
<td>CoPreMo</td>
<td>( O(kv + d^2) )</td>
<td>( O(nd) )</td>
</tr>
</tbody>
</table>

\( k \) is the number of time step, \( v \) is the number of data points per time step, \( d \) is the dimension of the property, \( n \) is the allocated event space per time step, and \( m \) is the number of Gaussian components (kernels).

As shown in Table I, the worst-case time complexities of BF and KF for a 1-D property, i.e., \( d = 1 \), is \( O(k^2) \), less than that of CoPreMo 1 and 2, which are based on Algorithm 1. The time complexity for EKF and GSF are \( O(k^3) \) and \( O(mv^3k^3 + m) \), respectively, higher than that for BF, KF, and CoPreMo.

Furthermore, the worst-case space complexity of the models shows that the CoPreMo model has the highest space complexity of \( O(nd) \). However, since the number of point sources needed to effectively capture the target at any given time step in a GSM model is high, one can consider that for a 1-D property, \( v > n \Rightarrow O(v) > O(n) \). The least space complexity models are the BF, KF, and EKF models with \( O(d) \), since they use a single-point source per time step. This will be \( O(1) \), i.e., constant space, for a 1-D property.

Therefore, comparing the models with respect to their performances, the CoPreMo model has a better time complexity as compared to the other models but with a worse space complexity. Selecting the models based on computational performance will require the consideration of their space and time complexities trade-off.

4 EXPERIMENTAL RESULTS AND DISCUSSIONS

The experiment aims to demonstrate the performance of the proposed CoPreMo model and compare the results with those of conventional models. To validate the performance of our model, we carried out simulation experiments using a road scenario dataset generated from the Matlab Radar data generator and compared the results with other models.

4.1 Experimental Setup

The experiment consists of an egocar equipped with a radar sensor that moves along a highway and another car that enters the highway from a road inlet until it collides with the egocar as shown in Figure 4(a),(b),(d). The radar sensor is placed at the left corner of the egocar as shown in Figure 4(c). The simulation was recorded when the egocar started observing the target car and the range of the target car as it approached the egocar was recorded.

The dataset for the range estimations was generated using the Matlab Radar data generator. The total number of range measurement points of the target estimated by the radar when the target moves under the radar’s field of view is 541 points. These points represent the measurement values of the tracker. The goal of the tracker is to predict the next measurement value given the current and past measurements of the dataset.

Also, these predicted continuous output measurements are classified into one of four discrete-output classes at the deterministic logic added to the tracker. The tracking results for the different trackers are presented in the next section.

4.2 Results and Discussions

After running the simulation, the performance results for both the continuous-output and discrete-output phases of the proposed and baseline tracking models are presented in Table II. The error during the tracking process is presented in Figure 6. To capture the computational cost of each model, their execution time and memory usage were recorded.

As shown in Figure 6, the RMSE value of the CoPreMo model is lowest over the entire time steps as compared to the other models. This is followed by the EKF model. The BF using direct application of Gaussian distribution has the highest RMSE value over time.

The summarized numerical results are presented in Table II. From these summarized results, the BF model has an accuracy far less than the KF, EKF, GSF, and CoPreMo. However, CoPreMo has the lowest run-time but a high memory usage as compared to BF, KF, and EKF.

One reason for such high predictive performance of the CoPreMo model is the use of multiple causal values to determine the global output and the mutual value consideration to capture the correlation between the measurement values.
Radar position on egocar: left side and 30cm from the ground.

Target car approaching the highway of the egocar.

With no detection and tracking of the target car by the egocar, a collision may occur.

Top view of collision at road merging.

Fig. 4. Simulation of an ADAS/AD vehicle collision scenario with corner radar detection and tracking.

Table II

<table>
<thead>
<tr>
<th>Models</th>
<th>Accuracy (%)</th>
<th>RMSE</th>
<th>Run-time</th>
<th>Memory-used</th>
</tr>
</thead>
<tbody>
<tr>
<td>BF</td>
<td>91.37</td>
<td>0.51</td>
<td>72ms</td>
<td>16 bytes</td>
</tr>
<tr>
<td>KF</td>
<td>93.42</td>
<td>0.48</td>
<td>74ms</td>
<td>18 bytes</td>
</tr>
<tr>
<td>EKF</td>
<td>97.95</td>
<td>0.26</td>
<td>79ms</td>
<td>19 bytes</td>
</tr>
<tr>
<td>GSF</td>
<td>98.31</td>
<td>0.31</td>
<td>100ms</td>
<td>37 bytes</td>
</tr>
<tr>
<td>CoPreMo</td>
<td>98.81</td>
<td>0.21</td>
<td>67ms</td>
<td>27 bytes</td>
</tr>
</tbody>
</table>

5 Conclusion

In this study, we presented a collaborative predictive model in time series for target tracking. The model was defined for a 1-D tracking property and the range property was considered. The mathematical framework on which the model operates is proposed and a simulation experiment was carried out to track the range property of a target car in a Matlab driving scenario.

The performance results were compared with those of BF, KF, EKF, and GSF in terms of their accuracy, RMSE, runtime, memory usage, time complexity, and space complexity. From the results, the accuracy and RMSE of the CoPreMo model are better than those of the other models. Furthermore, the CoPreMo model has a lower runtime but a higher memory usage as compared to the BF, KF, and EKF models. The high memory usage of the CoPreMo model is a drawback that we look forward to optimizing.

Apart from being used in this study for tracking, CoPreMo can also be used in different time series applications such as in signal processing, filtering, coding theory, control systems, machine learning, and natural language processing. We look forward to implementing it on these different applications.
APPENDIX

PROOF OF PROPOSITION

Proof of Proposition (3.1)

Consider the joint probability distribution \( P(z_1, z_2, z_3, x_4) \), \( \forall x, z \in \mathbb{R} \), \( x \equiv z \), then we can express it as

\[
P(z_1, z_2, z_3, x_4) = P(z_1)P(z_2|z_1)P(z_3|z_1, z_2)P(x_4|z_2, z_1, z_3)
\]  
(A.1)

Equating (A.1) and (A.2),

\[
P(x_4|z_1, x_4)P(z_2|z_1, x_4)P(z_3|z_2, z_1, x_4)
\]

(A.2)

\[
P(z_1)P(z_2|z_1)P(z_3|z_1, z_2)
\]

(A.3)

Applying Axiom (3.1),

\[
P(x_4|z_1, z_2, z_3) = P(x_4|z_1)P(x_4|z_2)P(x_4|z_3)
\]

(A.4)

Applying Bayes rule to (A.1) and (A.2),

\[
P(x_4|z_1, z_2, z_3) = P(x_4|z_1)P(x_4|z_2)P(x_4|z_3)
\]

\[
\frac{P(z_2|z_1)P(z_3|z_1, z_2)}{P(x_4|z_2)}
\]

(A.5)

Therefore, for \( P(x_{k+1}|z_1, z_2, z_3, \ldots, z_k) \)

\[
P(x_{k+1}|z_1, z_2, z_3, \ldots, z_k) = \frac{1}{P(x_{k+1}|z_1, z_2, z_3, \ldots, z_k)}
\]

\[
\prod_{i=1}^{k} \left( \frac{P(z_i)}{P(z_{i+1})} \right)^{-1}
\]

(A.6)

REFERENCES


[11] Zie Eya Ekolle (Member, IEEE) received the Master in Engineering from the Department of Electrical and Computer Engineering, Yokohama National University, Japan, in 2019. He received his Ph.D. in Engineering from the Department of Electrical and Computer Engineering, Yokohama National University, Japan, in 2023. His current fields of research are computational theory, information systems, information geometry, dependable systems, cognitive and rational agents, explainable and ethical AI, and collaborative machine learning.

[12] K. Zephyr, “Technical characteristics for radiolocation equipment for ground-based vehicular applications within the frequency range 77ghz to 81ghz,” ETSI TR 103 593 VI.1.1.5 2020.


[16] Referees