Journal of Systems Engineering and Electronics
Vol. 33, No. 5, October 2022, pp.1195 – 1209

Maneuvering target state estimation based on separate modeling of target trajectory shape and dynamic characteristics

ZHANG Zhuanhua and ZHOU Gongjian*

School of Electronics and Information Engineering, Harbin Institute of Technology, Harbin 150001, China

Abstract: The state estimation of a maneuvering target, of which the trajectory shape is independent on dynamic characteristics, is studied. The conventional motion models in Cartesian coordinates imply that the trajectory of a target is completely determined by its dynamic characteristics. However, this is not true in the applications of road-target, sea-route-target or flight route-target tracking, where target trajectory shape is uncoupled with target velocity properties. In this paper, a new estimation algorithm based on separate modeling of target trajectory shape and dynamic characteristics is proposed. The trajectory of a target over a sliding window is described by a linear function of the arc length. To determine the unknown target trajectory, an augmented system is derived by denoting the unknown coefficients of the function as states in mileage coordinates. At every estimation cycle except the first one, the interaction (mixing) stage of the proposed algorithm starts from the latest estimated base state and a recalculated parameter vector, which is determined by the least squares (LS). Numerical experiments are conducted to assess the performance of the proposed algorithm. Simulation results show that the proposed algorithm can achieve better performance than the conventional coupled model-based algorithms in the presence of target maneuvers.

Keywords: maneuvering target tracking, separate modeling, natural parametric function, interacting multiple model (IMM) filter, data fitting, state augmentation.

DOI: 10.23919/JSEE.2022.000115

1. Introduction

State estimation has been widely studied in many different applications in ground surveillance [1], trajectory determination [2], attitude control [3,4], and navigation [5]. State estimation for dynamic systems is concerned with not only estimation algorithms but also the mathematical modeling of systems. A motion model, which describes the evolution of the target state with respect to time, can be put in a state estimation framework in order to improve tracking performance. Knowledge of target dynamics is available and a good model-based estimation algorithm will greatly perform better than any model-free estimation algorithm if the underlying model is a good one [6]. Various motion models have been developed to describe target dynamics for target tracking and/or estimation [1,4,6−11]. Motion models can be classified into two classes: one is a non-maneuvering model that describes a uniform motion and the other is a maneuvering model. A nearly constant velocity (NCV) model is used to describe non-maneuvering motion, which is the straight. To describe maneuvering behaviors, a nearly constant acceleration (NCA) and a nearly coordinated turn (NCT) models are developed. The NCT model achieves the turning of an aircraft with a constant turn rate, whose position evolves along circular arcs. When an estimator based on a single model was not adequate, an interacting multiple model (IMM) approach was presented to provide the state-of-the-art solutions to many problems such as air traffic control tracking [12,13], in which a two-model IMM with the NCV and NCT models is used to perform maneuvers. The Singer model is a popular model for characterizing target maneuvers [14]. In [15], this model was modified to be the mean-adaptive acceleration model. The jerk model [16] was discussed for highly maneuvering targets.

Typically, above motion models are constructed in Cartesian coordinates, where the trajectory shape of a target is completely determined by its dynamic characteristics. In this case, the target trajectory shape is coupled with the target velocity, and thus the evolution of target trajectory shape is developed by target velocity properties described by the motion models. The motion models in Cartesian coordinates are generally effective to describe the target dynamics in free space. However, when a target is subject to various constraints imposed by external environments such as roads, trees, buildings, sea-

Manuscript received April 22, 2021.
*Corresponding author.
This work was supported by the National Natural Science Foundation of China (61671181).

routes, and terrains, the target trajectory shape is independent of target velocity properties. This has a lot of applications in practice. For example, to improve navigability, a target may make a speed maneuver [17−19], but its trajectory shape is a straight line when the target moves along a straight road. Another example is to assume that the shape of a road on which a target is moving is a circular arc. In general, the faster the target travels, the larger the radius of the circular arc is; the slower the target travels, the smaller the radius of the circular arc is. In this case, the target trajectory shape is a circular arc and is irrelevant to the speed. Although the above NCT model generates a circular trajectory, the target speed described by it is constant. Actually, the speed of a target moving on a circular arc road may be changed. The NCA model in Cartesian coordinates characterizes the target's maneuvering behavior, but it may not yield a circular arc trajectory since it is a motion model underlying free space, and thus the knowledge of target trajectory shape is unavailable from it. Therefore, the separate modeling (SM) of target trajectory and target dynamic characteristics is proposed.

The one-dimensional (1-D) target motion representation defines the traveled distance (mileage/arc length) and its derivatives as states in mileage coordinates. This 1-D modeling has been proven to be a state-of-the-art framework to model the on-road target motion [20]. The 1-D representation satisfies the constraint imposed by a road and is capable of improving tracking performance by integrating the road constraint. The 1-D representation of target trajectory, regardless of the speed, is modeled by a function of the arc length. The 1-D representation of a trajectory network has been explored in the literature. Similar to [20], Chen et al. [21] discussed the approximation of a curve trajectory by straight line segments and circular arc segments. In [17,22−27], the polygonal curve, which is composed of a sequence of linear segments, was used to approximate acurve trajectory. Another way to describe a curve trajectory is the cubic splines [28,29]. When the prior information about a road such as the map is available, the known arc length function describing the target trajectory can be directly integrated into a state estimation framework to improve tracking performance. However, to the best of our knowledge, the practical situation where the map or other prior information to determine the target trajectory may not always be available has not been considered in on-road target tracking. Since the independence of on-road target trajectory shape and target dynamic characteristics requires the SM of target trajectory and velocity properties, the description of target trajectory, fed into a filter to improve tracking performance, is available with some unknown parameters to be estimated.

While the target trajectory over a sliding window is described by continuous time functions, which are first-degree polynomials determined by the least squares (LS) in [30], this approach implies that the target trajectory is coupled with target dynamic characteristics. However, as mentioned above, the determination of on-road target trajectory shape is irrelevant to target dynamic characteristics.

This paper aims to investigate target tracking constrained by external environments such as roads based on SM of target trajectory and target dynamic characteristics, as well as integrating the target trajectory derived from measurements into a modified IMM framework without knowing prior information about the target trajectory.

The target dynamics are described by motion models in mileage coordinates, in which a state vector consists of mileage and its derivatives. A description of the target's maneuvering behavior is given by multiple models. A sliding window of the latest k_{l} measurements is utilized to solve the problem where no prior information about the target trajectory is available. The target trajectory in each window is represented by natural parametric functions, which are defined by three attributes, starting point, direction vector, and starting arc length. To determine the functions, an augmented system is derived by denoting the unknown function coefficients as states propagated with an identity transition matrix and artificial process noise with zero-mean white Gaussian distribution. An SM-based algorithm, which estimates simultaneously the target trajectory and velocity properties, is proposed. At every estimation cycle except the first one, the interaction (mixing) stage of the proposed estimation algorithm starts from the latest base state estimate and a reset parameter vector (i.e., stacked coefficients of the function), which is determined by the LS method using the functions to fit measurements in a window except the new measurement. Numerical experiments in four situations are conducted to verify the effectiveness of the proposed estimation algorithm compared with the Kalman filter (KF), IMM extended KF (IMM-EKF), and online fitting [30].

The remainder of the paper is organized as follows: Section 2 introduces the problems about maneuvering target tracking. In Section 3, the description of the target trajectory is given. Section 4 outlines the steps of the proposed estimation algorithm. Four simulation results and analysis are presented in Section 5. Concludes are given in Section 6.

The notations used in this paper are as follows. Superscript "′" denotes the transpose of a matrix or vector.

Superscript "−1" denotes the matrix inverse. diag(·) denotes the diagonal matrix. Superscripts "a" and "$*$" represent the augmented vector or matrix and the reset of the parameter vector or coefficient vector or coefficients in this paper, respectively. "max" and "min" denote the maximum value and the minimum value, respectively.

2. Problem formulation

In mileage coordinates, the motion model is considered as

$$\boldsymbol{x}_{k+1} = \boldsymbol{F}_k \boldsymbol{x}_k + \boldsymbol{\Gamma}_k \boldsymbol{v}_k \tag{1}$$

where $\boldsymbol{x}_k$ is the state vector consisting of mileage (arc length) s_k and its derivatives, and $\boldsymbol{v}_k$ is the zero-mean Gaussian white process noise with covariance q^2. The terms $\boldsymbol{x}_k$, $\boldsymbol{F}_k$, and $\boldsymbol{\Gamma}_k$ will be specified later for different motion models.

(i) NCV model: For the NCV model to describe the nonmaneuvering mode, the state vector is defined by

$$\boldsymbol{x}_k = [\ s_k \quad \dot{s}_k\ ]'. \tag{2}$$

The state transition matrix $\boldsymbol{F}_k$ and noise gain $\boldsymbol{\Gamma}_k$ are given by

$$\boldsymbol{F}_k = \boldsymbol{F}_{\mathrm{NCV},k} = \begin{bmatrix} 1 & T \\ 0 & 1 \end{bmatrix} \tag{3}$$

and

$$\boldsymbol{\Gamma}_k = \boldsymbol{\Gamma}_{\mathrm{NCV},k} = \begin{bmatrix} \dfrac{T^2}{2} \\ T \end{bmatrix}, \tag{4}$$

respectively.

(ii) NCA model: For the NCA model to describe the maneuvering mode, the second derivative of mileage is often augmented into the state vector of the constant velocity model given by

$$\boldsymbol{x}_k = [\ s_k \quad \dot{s}_k \quad \ddot{s}_k\ ]'. \tag{5}$$

The state transition matrix $\boldsymbol{F}_k$ and noise gain $\boldsymbol{\Gamma}_k$ can be written as

$$\boldsymbol{F}_k = \boldsymbol{F}_{\mathrm{NCA},k} = \begin{bmatrix} 1 & T & \dfrac{T^2}{2} \\ 0 & 1 & T \\ 0 & 0 & 1 \end{bmatrix} \tag{6}$$

and

$$\boldsymbol{\Gamma}_k = \boldsymbol{\Gamma}_{\mathrm{NCA},k} = \begin{bmatrix} \dfrac{T^2}{2} \\ T \\ 1 \end{bmatrix}, \tag{7}$$

respectively.

(iii) Multiple model (MM): The IMM to describe target maneuvers [6−8] assumes that the model sequence satisfies a homogeneous Markov chain $m_k \in \{M_1,\ M_2\}$, M_1 is the nonmaneuvering NCV model, M_2 is the maneuvering NCA model, and m_k denotes a mode at time k. With a known mode transition probability matrix $\boldsymbol{\Pi} = [p_{ij}]_{i,j=1}^2$ and an initial probability vector $\boldsymbol{\mu}_0 = [\mu_{0,1}, \mu_{0,2}]'$, the mode transition probabilities and the initial probabilities are given by

$$p_{ij} = P\{m_k = M_j | m_{k-1} = M_i\}, \quad i, j = 1, 2, \tag{8}$$

$$\mu_{0,i} = P\{m_0 = M_i\},\ \ i = 1, 2. \tag{9}$$

The hybrid system model can be given by

$$\boldsymbol{x}_{k+1} = \boldsymbol{F}_k(m_k)\boldsymbol{x}_k + \boldsymbol{\Gamma}_k(m_k)\boldsymbol{v}_k. \tag{10}$$

Based on the NCV and NCA models in mileage coordinates, an MM configuration for maneuvering target tracking is selected in the following: NCV-NCA.

It is assumed that measurements only depend on target position in Cartesian coordinates. The measurement model is given by

$$\boldsymbol{z}_k = \begin{bmatrix} x_k^m \\ y_k^m \end{bmatrix} = \boldsymbol{h}(\boldsymbol{x}_k) + \boldsymbol{w}_k = \begin{bmatrix} x_k \\ y_k \end{bmatrix} + \begin{bmatrix} w_k^x \\ w_k^y \end{bmatrix} \tag{11}$$

where the process noise $\boldsymbol{v}_k$ and the measurement noise w_k are mutually independent white noise. The measurement noise components w_k^x and w_k^y are assumed to be mutually independent zero-mean white Gaussian with variances of σ_x^2 and σ_y^2, respectively. The covariance $\boldsymbol{R}_k$ of the measurement noise can be written as

$$\boldsymbol{R}_k = \begin{bmatrix} \sigma_x^2 & 0 \\ 0 & \sigma_y^2 \end{bmatrix}.$$

In some practical tracking applications, the evolution of the trajectory shape of a target is independent of target dynamic characteristics. For example, when a vehicle moves along a straight-line road, the trajectory of the vehicle is a straight line,which is irrelevant to the values of speed. When the trajectory and dynamic characteristics of a target are described separately by the 1-D modeling method, the situation where the prior information about the target trajectory such as the map is unavailable may occur. Therefore, in order to improve tracking performance, how to describe, determine, and integrate the target trajectory into the framework of the nonlinear state estimation is a problem that deserves investigation.

3. Description of target trajectory

The target trajectory is described by

$$\boldsymbol{x}_c = \boldsymbol{f}(s) \Rightarrow \begin{cases} x = f_x(s) \\ y = f_y(s) \end{cases} \tag{12}$$

where s is the arc length (mileage) from the reference starting point of the trajectory and is referred to as the mileage coordinate [21]. The mapping f_x and f_y are functions of the arc length s. The two functions achieve the

transformation from mileage coordinates to Cartesian coordinates.

Generally, given the arc length s, the arbitrary point $\boldsymbol{x}_c=(x,y)$ on the trajectory is uniquely determined by the known arc length function (12). However, in this paper, what is lacking is prior information about the target trajectory such as the map; this is needed for determination of the arc length function (12). In this case, the arc length s in mileage coordinates is not transformed into the position $\boldsymbol{x}_c=(x,y)$ in Cartesian coordinates because the arc length function (12) is unknown. However, the independence of the on-road target trajectory shape and target velocity properties requires that the target trajectory is described by the arc length function. Therefore, the target trajectory over a sliding window of the latest k_m measurements is defined without knowing prior information about the target trajectory in this paper.

The target trajectory over a sliding window is specified as

$$\begin{cases} x=f_x(s,\boldsymbol{\psi}_x)=x_l+(s-s_l)r_{x,l} \\ y=f_y(s,\boldsymbol{\psi}_y)=y_l+(s-s_l)r_{y,l} \end{cases} \tag{13}$$

where f_x and f_y are not only functions of the mileage s, but also functions of the coefficient vector components $\boldsymbol{\psi}_x$ and $\boldsymbol{\psi}_y$ on the x and y axes, respectively, and $\boldsymbol{\psi}=[\boldsymbol{\psi}_x',\boldsymbol{\psi}_y']'$ with $\boldsymbol{\psi}_x=[x_l,r_{x,l}]'$ and $\boldsymbol{\psi}_y=[y_l,r_{y,l}]'$. Each window is defined by the above three attributes, starting point, direction vector, and starting arc length.

Since (13) does not involve target velocity properties, it can be used for road-target, sea-route-target or flight route-target tracking, where target trajectory shape is independent of target velocity properties. In addition, unlike [31,32] that the shape of a target trajectory in constrained state estimation is needed to be known, the low-order polynomial over a sliding window fills the gap without shape information about the target trajectory in the framework of mileage coordinate state estimation. Actually, the low-order polynomial (13) over a sliding window of the latest k_m measurements can be used to approximate target trajectories with arbitrary shapes.

The parameter vector $\boldsymbol{\eta}=[\boldsymbol{\psi}_x',\boldsymbol{\psi}_y',s_l]'=[x_l,r_{x,l},y_l,r_{y,l},s_l]'$ is unknown, it needs to be estimated together with the target state. In tracking systems based on (13), in order to determine the target trajectory, the state augmentation approach is adopted and the augmented state vector $\boldsymbol{x}_k^a$ is given by

$$\boldsymbol{x}_k^a=[\boldsymbol{x}_k',\boldsymbol{\eta}']'=\ [s_k,\dot{s}_k,\ddot{s}_k,x_l,r_{x,l},y_l,r_{y,l},s_l]'. \tag{14}$$

The measurement equation with the augmented state vectoris obtained by using (13) to substitute for (11). This implies that

$$\begin{aligned} \boldsymbol{z}_k=\begin{bmatrix} x_k^m \\ y_k^m \end{bmatrix}=\boldsymbol{h}(\boldsymbol{x}_k,\boldsymbol{\eta}_l)+\boldsymbol{w}_k= \\ \boldsymbol{h}(\boldsymbol{x}_k^a)+\boldsymbol{w}_k=\begin{bmatrix} x_k \\ y_k \end{bmatrix}+\begin{bmatrix} w_k^x \\ w_k^y \end{bmatrix}= \\ \begin{bmatrix} x_l+(s_k-s_l)r_{x,l} \\ y_l+(s_k-s_l)r_{y,l} \end{bmatrix}+\begin{bmatrix} w_k^x \\ w_k^y \end{bmatrix}. \end{aligned} \tag{15}$$

4. An estimation algorithm based on SM

As shown in (14), the base state $\boldsymbol{x}_k$ is augmented by the parameter vector $\boldsymbol{\eta}$ that needs to be estimated, and thus the evolution of the augmented state from sampling step k to $k+1$ can be written as

$$\begin{aligned} \boldsymbol{x}_{k+1}^a=\boldsymbol{F}_k^a\boldsymbol{x}_k^a+\boldsymbol{\Gamma}_k^a\boldsymbol{v}_k^a= \\ \begin{bmatrix} \boldsymbol{F}_k & \boldsymbol{0}_{\lambda_1\times\lambda_2} \\ \boldsymbol{0}_{\lambda_2\times\lambda_1} & \boldsymbol{I}_{\lambda_2\times\lambda_2} \end{bmatrix}\begin{bmatrix} \boldsymbol{x}_k \\ \boldsymbol{\eta}_k \end{bmatrix}+\begin{bmatrix} \boldsymbol{\Gamma}_k & \boldsymbol{0}_{\lambda_1\times\lambda_2} \\ \boldsymbol{0}_{\lambda_2\times1} & \boldsymbol{I}_{\lambda_2\times\lambda_2} \end{bmatrix}\begin{bmatrix} \boldsymbol{v}_k \\ \boldsymbol{\varphi}_k \end{bmatrix} \end{aligned} \tag{16}$$

where $\boldsymbol{F}_k^a$ and $\boldsymbol{\Gamma}_k^a$ denote the augmented state transition matrix and augmented noise gain, respectively. $\boldsymbol{v}_k^a$ refers to the augmented process noise. $\boldsymbol{0}_{\lambda_1\times\lambda_2}$ denotes the zero matrix with the dimension $\lambda_1\times\lambda_2$. $\boldsymbol{I}_{\lambda_2\times\lambda_2}$ is the identity matrix with the dimension $\lambda_2\times\lambda_2$. λ_1 and λ_2 refer to the dimension of the base state vector and parameter vector, respectively. The artificial process noise $\boldsymbol{\varphi}_k$ is used for error compensation of low-order polynomials, when describing a curve trajectory. It is assumed to be zero-mean white Gaussian distribution with covariance $\boldsymbol{\Phi}_k$.

The covariance of the augmented process noise term is given by

$$\boldsymbol{Q}_k^a=\begin{bmatrix} \boldsymbol{Q}_k & \boldsymbol{0}_{\lambda_1\times\lambda_2} \\ \boldsymbol{0}_{\lambda_2\times\lambda_1} & \boldsymbol{\Phi}_k \end{bmatrix}$$

where $\boldsymbol{Q}_k=\boldsymbol{\Gamma}_k q^2\boldsymbol{\Gamma}_k'$.

To estimate the augmented state in (16) based on the measurement equation (15), the SM-based estimation algorithm, which implements the IMM for NCV-NCA MM in mileage coordinates defined by (2)−(10), is developed. The flowchart of every cycle of the proposed SM algorithm is shown in Fig. 1, where N_t denotes the total sampling step. Due to the nonlinearity of the measurement equation (15), the SM-based estimation algorithm uses an EKF [9] for the model-conditional filtering (under M1 and M2, respectively). Note the unscented KF (UKF) [33,34], the cubature KF (CKF) [35,36] can be also used to handle the nonlinearity here.

(i) Initialization of the augmented model

An initial estimate of the augmented state and the corresponding initial covariance are approximated by unscented transformation (UT) [37,38]. The sigma points of $\bar{\boldsymbol{z}}=[x_{k-1}^m,y_{k-1}^m,\ x_k^m,y_k^m,x_{k+1}^m,y_{k+1}^m]'$ with dimension n can be selected as

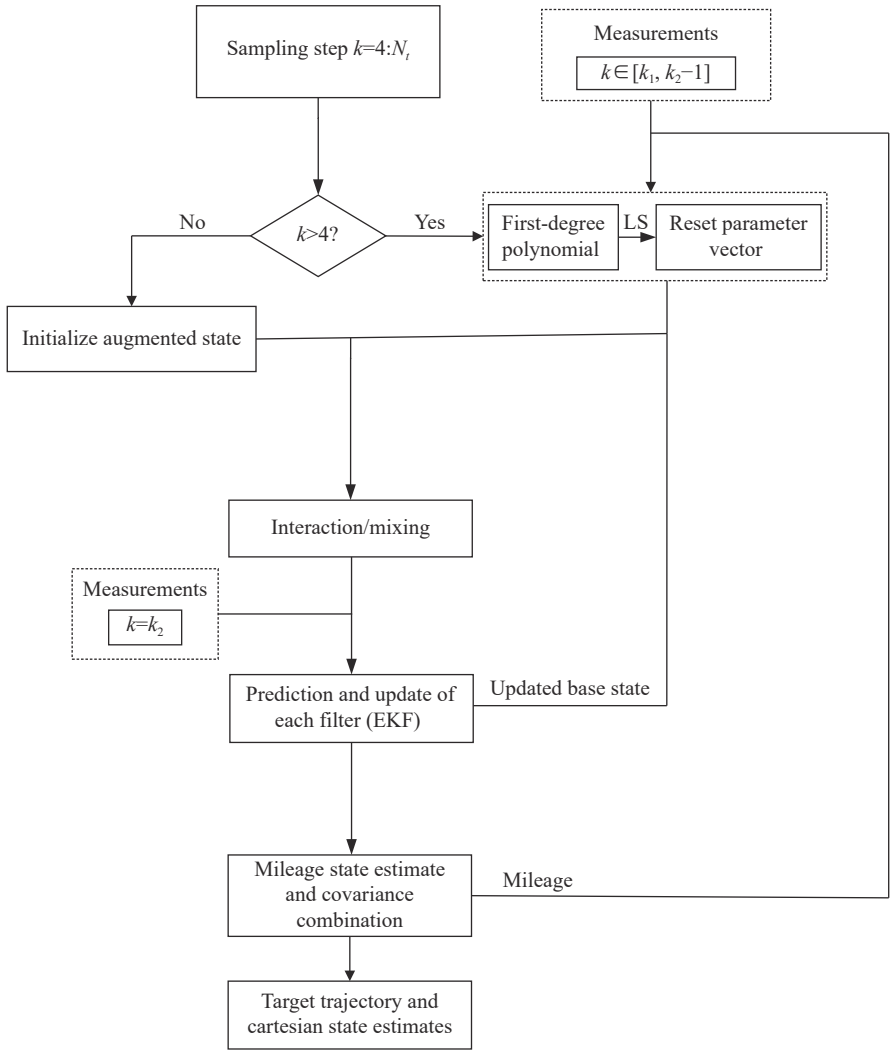


Fig. 1 Flowchart of the proposed tracking algorithm

$$\begin{cases} z^0 = \bar{z} \\ z^i = \bar{z} + (\sqrt{(n+\kappa)\tilde{\boldsymbol{R}}_k})_i \\ z^{i+n} = \bar{z} - (\sqrt{(n+\kappa)\tilde{\boldsymbol{R}}_k})_i \\ i = 1, 2, \cdots, n \end{cases} \tag{17}$$

where the covariance of $\bar{z}$ is the diagonal matrix $\tilde{\boldsymbol{R}}_k = \mathrm{diag}(\sigma_x^2, \sigma_y^2, \sigma_x^2, \sigma_y^2, \sigma_x^2, \sigma_y^2)$, and $(\sqrt{(n+\kappa)\tilde{\boldsymbol{R}}_k})_i$ denotes the ith row or column of the matrix $\sqrt{(n+\kappa)\tilde{\boldsymbol{R}}_k}$.

The $2n+1$ weighting coefficients are given by

$$\begin{cases} W_0^m = \dfrac{\kappa}{n+\kappa}, W_0^c = \dfrac{\kappa}{n+\kappa} + (1-\alpha^2+\beta) \\ W_i^m = W_i^c = \dfrac{\kappa}{2(n+\kappa)}, \quad i = 1, 2, \cdots, 2n \end{cases}. \tag{18}$$

The chord length between position measurements at two adjacent times is used to approximate the target traveling distance, due to without additional prior information about the state except the measurements. The base state is formulated according to the two-point differencing procedure [4]. Using the nonlinear equations (19)

$$\begin{cases} s_{k+1} = s_k + \sqrt{(x_{k+1}^m - x_k^m)^2 + (y_{k+1}^m - y_k^m)^2} \\ \dot{s}_{k+1} = \dfrac{s_{k+1} - s_k}{T} \\ \ddot{s}_{k+1} = \dfrac{\dfrac{s_{k+1} - s_k}{T} - \dfrac{s_k - s_{k-1}}{T}}{T} \\ x_l = x_{k_1}^m \\ r_{x,l} = \dfrac{x_{k_2} - x_{k_1}}{\left\| \boldsymbol{x}_{c,k_2} - \boldsymbol{x}_{c,k_1} \right\|} = \dfrac{x_{k_2}^m - x_{k_1}^m}{\sqrt{(x_{k_2}^m - x_{k_1}^m)^2 + (y_{k_2}^m - y_{k_1}^m)^2}} \\ y_l = y_{k_1}^m \\ r_{y,l} = \dfrac{y_{k_2} - y_{k_1}}{\left\| \boldsymbol{x}_{c,k_2} - \boldsymbol{x}_{c,k_1} \right\|} = \dfrac{y_{k_2}^m - y_{k_1}^m}{\sqrt{(x_{k_2}^m - x_{k_1}^m)^2 + (y_{k_2}^m - y_{k_1}^m)^2}} \\ s_l = s_0 \end{cases} \tag{19}$$

to transform the sigma points generated by (17) into $\boldsymbol{x}_k^{a,i}$. Here $s_0 = 0$, $k_1 = 1$, $k_2 = 3$, and $\boldsymbol{x}_{c,k_1}$ and $\boldsymbol{x}_{c,k_2}$ are the target positions at time steps k_1 and k_2, respectively.

The approximated covariance is given by

$$\bar{\boldsymbol{P}}_k^a = \sum_{i=0}^{2n} W_i^c (\boldsymbol{x}_k^{a,i} - \bar{\boldsymbol{x}}_k^a)(\boldsymbol{x}_k^{a,i} - \bar{\boldsymbol{x}}_k^a)' \tag{20}$$

where the approximated mean is determined by

$$\bar{\boldsymbol{x}}_k^a = \sum_{i=0}^{2n} W_i^m \boldsymbol{x}_k^{a,i}. \tag{21}$$

For the NCV model, the acceleration component in the augmented state $\bar{\boldsymbol{x}}_k^a$ is zero and the elements related to the acceleration in the covariance $\bar{\boldsymbol{P}}_k^a$ are zero accordingly. Since the first three measurements are used to determine the initialization of the augmented model, the filter starts running at time $k = 4$.

(ii) Interaction/mixing ($i, j = 1,2,\cdots,r$)

The mixing probabilities $\boldsymbol{\mu}_{k-1,\,k-1}^{i|j}$ are calculated by

$$\boldsymbol{\mu}_{k-1,k-1}^{i|j} = \boldsymbol{p}_{ij}\boldsymbol{\mu}_{k-1}^i \Big/ \sum_i \boldsymbol{p}_{ij}\boldsymbol{\mu}_{k-1}^i. \tag{22}$$

Starting with $\bar{\boldsymbol{x}}_{k-1}^{a,i}$ and $\bar{\boldsymbol{P}}_{k-1}^{a,i}$, the mixed initial condition $\bar{\boldsymbol{x}}_{k-1,k-1}^{0j}$ for each filter is given by

$$\bar{\boldsymbol{x}}_{k-1,k-1}^{0j} = \sum_i \bar{\boldsymbol{x}}_{k-1}^{a,i}\boldsymbol{\mu}_{k-1,k-1}^{i|j} \tag{23}$$

and the mixed initial covariance $\bar{\boldsymbol{P}}_{k-1,k-1}^{0j}$ for each filter is computed by

$$\bar{\boldsymbol{P}}_{k-1,k-1}^{0j} = \sum_i \boldsymbol{\mu}_{k-1,k-1}^{i|j}[\bar{\boldsymbol{P}}_{k-1}^{a,i} + (\bar{\boldsymbol{x}}_{k-1}^{a,i} - \bar{\boldsymbol{x}}_{k-1,k-1}^{0j})(\bar{\boldsymbol{x}}_{k-1}^{a,i} - \bar{\boldsymbol{x}}_{k-1,k-1}^{0j})']. \tag{24}$$

(iii) Prediction and update ($j = 1,2,\cdots,r$)

The augmented state prediction $\hat{\boldsymbol{x}}_{k,k-1}^{a,j}$ and prediction covariance $\boldsymbol{P}_{k,k-1}^{a,j}$ of each filter can be written as

$$\begin{cases} \hat{\boldsymbol{x}}_{k,k-1}^{a,j} = \boldsymbol{F}_{k-1}^{a,j}\bar{\boldsymbol{x}}_{k-1,k-1}^{0j} \\ \boldsymbol{P}_{k,k-1}^{a,j} = \boldsymbol{F}_{k-1}^{a,j}\bar{\boldsymbol{P}}_{k-1,k-1}^{0j}(\boldsymbol{F}_{k-1}^{a,j})' + \boldsymbol{Q}_{k-1}^{a,j} \end{cases}. \tag{25}$$

The measurement prediction covariance is defined as

$$\boldsymbol{S}_k^j = \boldsymbol{H}_k^j \boldsymbol{P}_{k,k-1}^{a,j}(\boldsymbol{H}_k^j)' + \boldsymbol{R}_k^j. \tag{26}$$

The covariance $\boldsymbol{P}_{\boldsymbol{x}^a z,k}^j$ between the augmented state vector and the measurement vector can be computed by

$$\boldsymbol{P}_{x^a z,k}^j = \boldsymbol{P}_{k,k-1}^{a,j}(\boldsymbol{H}_k^j)'. \tag{27}$$

The filter gain $\boldsymbol{K}_k^j$ is given by

$$\boldsymbol{K}_k^j = \begin{bmatrix} \tilde{\boldsymbol{K}}_k^{x,j} \\ \tilde{\boldsymbol{K}}_k^{\eta,j} \end{bmatrix} = \boldsymbol{P}_{\boldsymbol{x}^a z,k}^j (\boldsymbol{S}_k^j)^{-1} = \boldsymbol{P}_{k,k-1}^{a,j}(\boldsymbol{H}_k^j)'[\boldsymbol{H}_k^j \boldsymbol{P}_{k,k-1}^{a,j}(\boldsymbol{H}_k^j)' + \boldsymbol{R}_k^j]^{-1}. \tag{28}$$

The updated augmented state of each filter is

$$\hat{\boldsymbol{x}}_{k,k}^{a,j} = [(\hat{\boldsymbol{x}}_{k,k}^j)', (\hat{\boldsymbol{\eta}}_{k,k}^j)']' = \hat{\boldsymbol{x}}_{k,k-1}^{a,j} + \boldsymbol{K}_k^j[z_k - \boldsymbol{h}(\hat{\boldsymbol{x}}_{k,k-1}^{a,j})]. \tag{29}$$

The associated updated augmented state covariance can be written as

$$\boldsymbol{P}_{k,k}^{a,j} = \begin{bmatrix} \boldsymbol{P}_{k,k}^{\boldsymbol{x}\boldsymbol{x},j} & \boldsymbol{P}_{k,k}^{\boldsymbol{x}\eta,j} \\ \boldsymbol{P}_{k,k}^{\eta\boldsymbol{x},j} & \boldsymbol{P}_{k,k}^{\eta\eta,j} \end{bmatrix} = \boldsymbol{P}_{k,k-1}^{a,j} - \boldsymbol{K}_k^j \boldsymbol{H}_k^j \boldsymbol{P}_{k,k-1}^{a,j} \tag{30}$$

where $\boldsymbol{P}_{k,k}^{\eta\boldsymbol{x},j} = (\boldsymbol{P}_{k,k}^{\boldsymbol{x}\eta,j})'$.

The Jacobian matrix, based on the measurement model (15), is

$$\boldsymbol{H}_k^j = \left.\frac{\partial \boldsymbol{h}(\boldsymbol{x}^a)}{\partial \boldsymbol{x}^a}\right|_{\boldsymbol{x}^a = \hat{\boldsymbol{x}}_{k,k-1}^{a,j}} = \begin{bmatrix} \hat{r}_{x,l}^j & 0 & 0 & 1 & \hat{s}_{k,k-1}^j - \hat{s}_l^j & 0 & 0 & -\hat{r}_{x,l}^j \\ \hat{r}_{y,l}^j & 0 & 0 & 0 & 0 & 1 & \hat{s}_{k,k-1}^j - \hat{s}_l^j & -\hat{r}_{y,l}^j \end{bmatrix} \tag{31}$$

where since the coefficients $\hat{r}_{x,l}^j$, $\hat{r}_{y,l}^j$, and $\hat{s}_l^j$ are assumed to be constant in a window, the time subscript k is omitted.

The likelihood function corresponding to the jth filter is

$$\boldsymbol{\Lambda}_k^j = \frac{\exp\left[-\frac{1}{2}(\boldsymbol{\upsilon}_k^j)'(\boldsymbol{S}_k^j)^{-1}\boldsymbol{\upsilon}_k^j\right]}{\sqrt{|2\pi S_k^j|}} \tag{32}$$

where the innovation is $\boldsymbol{\upsilon}_k^j = \boldsymbol{z}_k - \boldsymbol{h}(\hat{\boldsymbol{x}}_{k,k-1}^{a,j})$.

Mode probability update is given by

$$\boldsymbol{\mu}_k^j = \frac{1}{c_1}\boldsymbol{\Lambda}_k^j \sum_{i=1}^r \boldsymbol{p}_{ij}\boldsymbol{\mu}_{k-1}^i \tag{33}$$

where $c_1 = \sum_{j=1}^r \boldsymbol{\Lambda}_k^j \sum_{i=1}^r \boldsymbol{p}_{ij}\boldsymbol{\mu}_{k-1}^i$.

(iv) State estimate and covariance combination

The combined state estimate $\hat{\boldsymbol{x}}_{k,k}^a$ and the corresponding covariance $\boldsymbol{P}_{k,k}^a$ are given by

$$\begin{cases} \hat{\boldsymbol{x}}_{k,k}^a = \sum_{j=1}^r \hat{\boldsymbol{x}}_{k,k}^{a,j}\boldsymbol{\mu}_k^j \\ P_{k,k}^a = \sum_{j=1}^r \boldsymbol{\mu}_k^j[\boldsymbol{P}_{k,k}^{a,j} + (\hat{\boldsymbol{x}}_{k,k}^{a,j} - \hat{\boldsymbol{x}}_{k,k}^a)(\hat{\boldsymbol{x}}_{k,k}^{a,j} - \hat{\boldsymbol{x}}_{k,k}^a)'] \end{cases}. \tag{34}$$

(v) Reset parameter vector

Generally, the parameter vector $\boldsymbol{\eta}_k$ estimated at time k is set as the initial parameter vector for estimating $\boldsymbol{\eta}_{k+1}$ at time $k+1$. However, as mentioned above, the proposed SM-based estimation algorithm describes the target trajectory over a sliding window of the latest k_m measurements. If the latest parameter estimate is regarded as the initial estimate for the next sampling time, then the proposed SM-based algorithm will obviously introduce errors in the filtering process. This is because the parameter components estimated at time $k_2 - 1$ are the starting position and starting arc length of the current window $[k_1 - 1, k_2 - 1]$, rather than the starting position and starting arc length of the next window $[k_1, k_2]$, which are needed to be determined to estimate the parameter vector at time k_2. In addition, although the direction of straight-

line trajectory is constant, the circular arc/curve trajectory of a target is approximated by straight line over a sliding window of the latest k_m measurements. This leads to different directions of two linear segments over two adjacent windows. To avoid the errors, at every estimation cycle except the first one, the interaction stage of the filters starts with the reset parameter vector and the latest updated base state. To this end, the cost function can be specified as

$$J_k(\boldsymbol{\eta}) = \sum_{k=k_1}^{k_2-1} \left\| \boldsymbol{z}_k - \boldsymbol{h}(\boldsymbol{x}_k, \boldsymbol{\eta}) \right\| \tag{35}$$

where $k \in [k_1, k_2-1]$, and $\boldsymbol{J}_k = [J_{x,k}, J_{y,k}]'$.

In the LS, the estimation problem can be solved by minimizing the following cost functions (36) with respect to the polynomial coefficients x_l and $r_{x,l}$, and y_l and $r_{y,l}$, respectively,

$$\begin{cases} J_{x,k}(x_l, r_{x,l}) = \sum_{k=k_1}^{k_2-1} [x_k^m - (x_l + (s_k - s_l) r_{x,l})]^2 \\ J_{y,k}(y_l, r_{y,l}) = \sum_{k=k_1}^{k_2-1} [y_k^m - (y_l + (s_k - s_l) r_{y,l})]^2 \end{cases}. \tag{36}$$

The reset values $\boldsymbol{\psi}_x^* = [x_l^*, r_{x,l}^*]'$ and $\boldsymbol{\psi}_y^* = [y_l^*, r_{y,l}^*]'$ of the coefficient vector components $\boldsymbol{\psi}_x = [x_l, r_{x,l}]'$ and $\boldsymbol{\psi}_y = [y_l, r_{y,l}]'$ are given by

$$\begin{cases} \boldsymbol{\psi}_x^* = [x_l^*, r_{x,l}^*]' = \\ \quad \arg \min_{x_l, r_{x,l}} J_{x,k}(x_l, r_{x,l}) = \boldsymbol{G}^{-1}(\boldsymbol{U}' \boldsymbol{X}^m) \\ \boldsymbol{\psi}_y^* = [y_l^*, r_{y,l}^*]' = \\ \quad \arg \min_{y_l, r_{y,l}} J_{y,k}(y_l, r_{y,l}) = \boldsymbol{G}^{-1}(\boldsymbol{U}' \boldsymbol{Y}^m) \end{cases}, \tag{37}$$

where

$$\boldsymbol{X}^m = [x_{k_1}^m, x_{k_1+1}^m, \cdots, x_{k_2-1}^m]', \tag{38}$$

$$\boldsymbol{Y}^m = [y_{k_1}^m, y_{k_1+1}^m, \cdots, y_{k_2-1}^m]', \tag{39}$$

$$\boldsymbol{V} = \begin{bmatrix} 1 & s_{k_1} - s_l \\ 1 & s_{k_1+1} - s_l \\ \vdots & \vdots \\ 1 & s_{k_2-1} - s_l \end{bmatrix}. \tag{40}$$

An orthogonal-triangular (or QR) decomposition of the matrix $\boldsymbol{V}$ satisfies $\boldsymbol{V} = \boldsymbol{U}\boldsymbol{G}$ and $\boldsymbol{U}'\boldsymbol{U} = \boldsymbol{I}$. Here, $s_l = s_{k_1}$.

Thus, the parameter vector is reset to

$$\boldsymbol{\eta}_k^* = [(\boldsymbol{\psi}_k^*)', \hat{s}_{k_1,k_1}]' = [x_l^*, r_{x,l}^*, y_l^*, r_{y,l}^*, \hat{s}_{k_1,k_1}]'.$$

At every estimation cycle k except the first one (i.e., $k = 4$), the interaction stage starts with

$$\overline{\boldsymbol{x}}_k^{a,j} = [(\hat{\boldsymbol{x}}_{k,k}^j), \boldsymbol{\eta}_k^*]' = [(\hat{\boldsymbol{x}}_{k,k}^j,)(\boldsymbol{\psi}_k^*)', \hat{s}_{k_1,k_1}]', \tag{41}$$

and

$$\bar{\boldsymbol{P}}_k^{a,j} = \begin{bmatrix} \boldsymbol{P}_{k,k}^{xx,j} & \boldsymbol{P}_k^{x\psi,j} & \boldsymbol{0}_{\lambda_1 \times 1} \\ \boldsymbol{P}_k^{\psi x,j} & \boldsymbol{P}_k^{\psi\psi} & \boldsymbol{0}_{(\lambda_2-1)\times 1} \\ \boldsymbol{0}_{1\times\lambda_1} & \boldsymbol{0}_{1\times(\lambda_2-1)} & \boldsymbol{P}_{k_1,k_1}^{(1,1)} \end{bmatrix} \tag{42}$$

where $\hat{\boldsymbol{x}}_{k,k}^j$ and $P_{k,k}^{\boldsymbol{xx},j}$ are given by (29) and(30), respectively. The reset coefficient vector $\boldsymbol{\psi}_k^*$ consists of the reset polynomial coefficients x_l^*, $r_{x,l}^*$, y_l^*, and $r_{y,l}^*$, and the corresponding covariance $\boldsymbol{P}_k^{\psi\psi}$ is determined by the LS. The starting arc length s_l of the lth window is approximated by the mileage component $\hat{s}_{k_1,k_1}$ of the combined state estimate $\hat{\boldsymbol{x}}_{k_1,k_1}^a$ at time k_1 given by (34), and its variance is provided by $\boldsymbol{P}_{k_1,k_1}^{(1,1)}$, which is the element at the first-row first-column of the combined state estimate covariance $\boldsymbol{P}_{k_1,k_1}^a$ presented in (34). The cross-covariance $\boldsymbol{P}_k^{\psi\boldsymbol{x},j}$ between the reset coefficient vector $\boldsymbol{\psi}_k^*$ and the updated base state $\hat{\boldsymbol{x}}_{k,k}^j$ of the jth filter is given by

$$\begin{cases} \boldsymbol{P}_k^{\psi\boldsymbol{x},j} = \begin{bmatrix} \boldsymbol{P}_k^{\psi_x\boldsymbol{x},j} \\ \boldsymbol{P}_k^{\psi_y\boldsymbol{x},j} \end{bmatrix} \\ \boldsymbol{P}_k^{\boldsymbol{x}\psi,j} = (\boldsymbol{P}_k^{\psi\boldsymbol{x},j})' \end{cases} \tag{43}$$

with

$$\boldsymbol{P}_k^{\psi_x\boldsymbol{x},j} = \boldsymbol{G}^{-1}\boldsymbol{U}_{k_2-1}'\sigma_x^2(\boldsymbol{K}_k^{x,j})', \tag{44}$$

$$\boldsymbol{P}_k^{\psi_y\boldsymbol{x},j} = \boldsymbol{G}^{-1}\boldsymbol{U}_{k_2-1}'\sigma_y^2(\boldsymbol{K}_k^{y,j})', \tag{45}$$

where $\boldsymbol{U}_{k_2-1}$ is the last row of the unitary matrix $\boldsymbol{U}$, and $\tilde{\boldsymbol{K}}_k^{\boldsymbol{x},j} = [\boldsymbol{K}_k^{x,i}, \boldsymbol{K}_k^{y,j}]$ is given by (28). The sliding window $[k_1, k_2]$ of the latest k_m measurements is supposed to move forward with time k and is performed from $k_1 = \max(1, k_2 - k_l)$, where k_l is the window length and $k_m = k_2 - k_1$.

5. Numerical experiments and analysis

In this section, the proposed SM-based estimation algorithm is compared against the KF, IMM-KFs, and online fitting by four simulations. The first is about a linear trajectory, the second is the case with a circular arc trajectory, and the last two are with respect to curve trajectories. KF(CA) and EKF(CT) denote that the KF and EKF

implement state estimates based on the NCA and NCT models in Cartesian coordinates, respectively. The IMM-KF(CVCA) and IMM-EKF(CVCACT) implement the IMM algorithm for NCV-NCA and NCV-NCA-NCT multiple models in Cartesian coordinates, respectively. The NCT model being estimated using the EKF is a nonlinear one as the turn rate is not a known constant. Online fitting refers to that the target dynamics are modeled by a continuous time function that is a first-degree polynomial determined by the LS.

5.1 Linear trajectory

(i) Scenario setting

In the first simulation, assume that the target moves along a linear road and its trajectory is a straight line. The target without turn maneuvers starts at (200 m, 300 m) and its traveling direction is $\theta = 65^{\circ}$, an angle measured counterclockwise from due x axis. The target maintains a constant velocity of 1 m/s from the start until a sampling step of 40, and then it speeds up with an acceleration of 5 m/s^2 starting from 41 to 70 and keeps a constant velocity from 71 to 120, and also has an acceleration from 121 to 150 with 5 m/s^2, after acceleration, it remains a constant velocity until the end. The target trajectory is determined by

$$\begin{cases} x = f_x(s) = x_1 + s\cos\theta \\ y = f_y(s) = y_1 + s\sin\theta \end{cases} \tag{46}$$

where (x_1, y_1) denotes the starting point of the target trajectory. The measurements are generated every T=0.3 s according to (11) and (46) models with $\sigma_x = \sigma_y = 2$ m.

(ii) Estimator parameters

The proposed SM-based estimation algorithm is compared with the KF(CA), IMM-KF(CVCA), and online fitting in the scenario with a linear trajectory. The standard deviations of the process noises of NCV and NCA models are 0.008 m/s^2. The standard deviation of the artificial process noise is 10^{-5}. The performances of these algorithms are evaluated over 100 Monte Carlo runs. The window length is 16. The initial probability vectors of the IMM-KF(CVCA) and the proposed SM-based algorithm are $\boldsymbol{\mu}_0 = [0.5, 0.5]'$, and the mode transition probability matrix is given by

$$\boldsymbol{\Pi} = \begin{bmatrix} 0.95 & 0.05 \\ 0.05 & 0.95 \end{bmatrix}.$$

(iii) Results

Fig. 2 shows the root mean square errors (RMSEs) of position and velocity estimates of the four algorithms, respectively. When the target maneuvers occur, the proposed SM-based estimation algorithm significantly outperforms the KF(CA), IMM-KF(CVCA), and online fitting (see Fig. 2(a)). This is because the on-road target trajectory shape is independent of target dynamic characteristics. The independence can be regarded as prior information, which is used by the proposed SM-based algorithm and can improve estimation performance. However, other estimation algorithms do not consider the independence of target trajectory shape and dynamic characteristics. This leads to loss of prior information about the independence that is not incorporated into other algorithms, which yield higher estimation errors. In Fig. 2(b), it can be seen that the proposed SM-based algorithm performs better than the KF(CA), IMM-KF(CVCA), and online fitting. The occurrence of the divergence of the online fitting is due to the utilization of a mismatched model, which assumes that the target remains a constant speed in a window. The RMSEs of the proposed SM algorithm falls within the 95% probability interval.

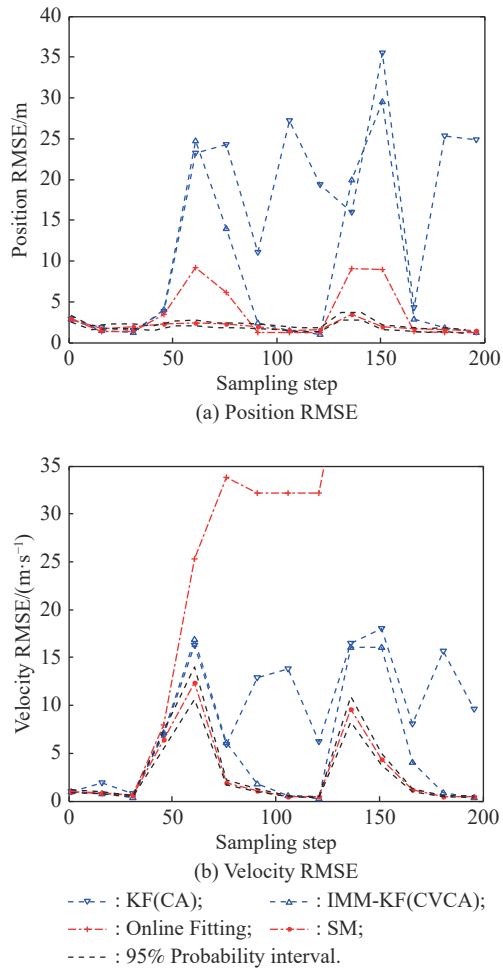


Fig. 2 RMSEs of state estimates and 95% probabilityinterval for the linear trajectory case

5.2 Circular arc trajectory

(i) Scenario setting

A target is simulated to move on a circular arc road with the starting position of (100 m, 30 m), the radius of $R = 50$ m, and the center of (60 m, 60 m). The target maintains a constant speed at time 0−40, 71−110, and 151−200, and then it speeds up with an acceleration from a sampling step of 41 to 70 with 3 m/s^2 and from 111 to 150 with 2 m/s^2. The initial speed is of 5 m/s. The on-road target trajectory is given by

$$\begin{cases} x = f_x(s) = x_m + R\cos\left(\theta_0 + \dfrac{s}{R}\right) \\ y = f_y(s) = y_m + R\sin\left(\theta_0 + \dfrac{s}{R}\right) \end{cases} \tag{47}$$

where $\theta_0 = \arctan\dfrac{y_m - y_0}{x_m - x_0}$. The center of the circle is (x_m, y_m) and the starting point is (x_0, y_0).The measurements are generated according to the models (11) and (47). The standard deviations of the measurement noises are set as $\sigma_x = \sigma_y = 0.4$ m.The sampling interval is T=0.1 s. The geometry of the simulation scenario is shown in Fig. 3.

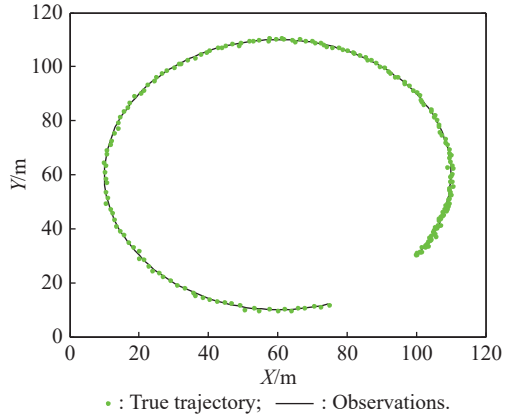


Fig. 3 Circular arc trajectory

(ii) Estimator parameters

The proposed SM-based estimation algorithm, EKF(CT), IMM-KF(CVCA), IMM-EKF(CVCACT), and online fitting are compared for the scenario with the circular arc trajectory. The standard deviations of the process noises of NCV and NCA models are 0.008 m/s^2. The standard deviation of the artificial process noise is 0.001. The simulation runs 200 with 100 Monte Carlo. The window length is 12. The initial model probability vectors of the IMM-KF(CVCA) and the proposed SM-based estimation algorithm are designed as $\boldsymbol{\mu}_0 = [0.5, 0.5]'$ and the corresponding mode transition probability matrix is given by

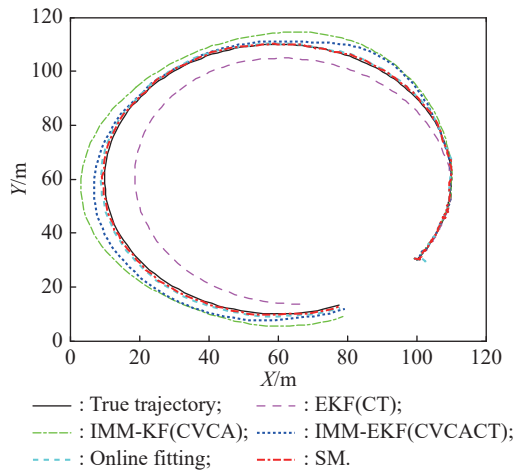
$$\boldsymbol{\Pi} = \begin{bmatrix} 0.9 & 0.1 \\ 0.1 & 0.9 \end{bmatrix}.$$

The IMM-EKF(CVCACT) uses $\boldsymbol{\mu}_0 = [0.3, 0.3, 0.4]'$ and

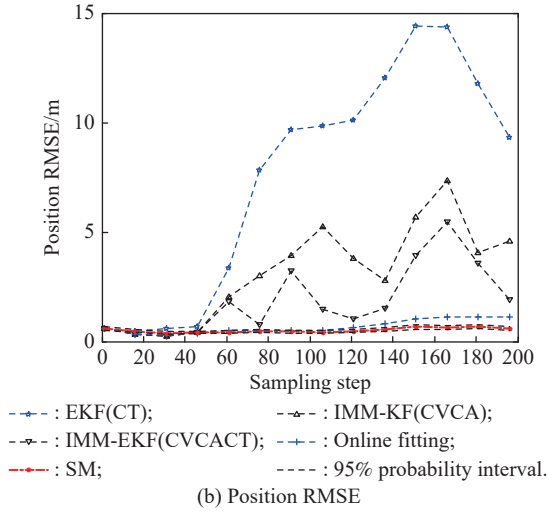
$$\boldsymbol{\Pi} = \begin{bmatrix} 0.90 & 0.05 & 0.05 \\ 0.05 & 0.90 & 0.05 \\ 0.05 & 0.05 & 0.90 \end{bmatrix}.$$

(iii) Simulation results

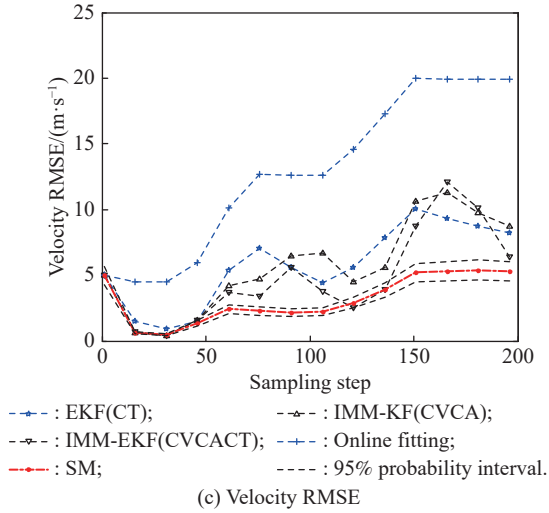
Fig. 4(a) shows the target trajectories obtained by the proposed SM-based estimation algorithm, EKF(CT), IMM-KF(CVCA), IMM-EKF(CVCACT), and online fitting. It can be seen that the SM-based estimation algorithm is better than the EKF(CT), IMM-KF(CVCA), IMM-EKF(CVCACT), and online fitting. This is because the occurrence of maneuvers has no effect on the target trajectory estimated by the SM-based algorithm due to SM, while the performances of other algorithms are degraded due to coupled models, which determine the target trajectory shape by target dynamic characteristics. Fig. 4(b) plots the position RMSEs of the five algorithms. When the target maneuvers appear, the proposed SM-based algorithm performs better than the EKF(CT), IMM-KF (CVCA), and IMM-EKF(CVCACT). Although the position RMSE differences between the SM-based algorithm and the online fitting algorithm are not large due to the small window length, the SM-based algorithm is more accurate than the online fitting. This demonstrates the effectiveness of SM, in reducing estimation errors. Fig. 4(c) compares the velocity RMSEs of the five algorithms. Although the performance of the SM-based algorithm in some sampling steps slightly declines, it is still better than that of other algorithms. This is because the EKF(CT), IMM-KF(CVCA), and online fitting use the motion models that are mismatched with the target maneuvers with both turning and speed-up. As shown in the figure, using the motion models in mileage coordinates to describe the target dynamics in the SM-based algorithm is better than Cartesian coordinate models, which are used by the IMM-EKF(CVCACT).



(a) Estimated target trajectory



(b) Position RMSE

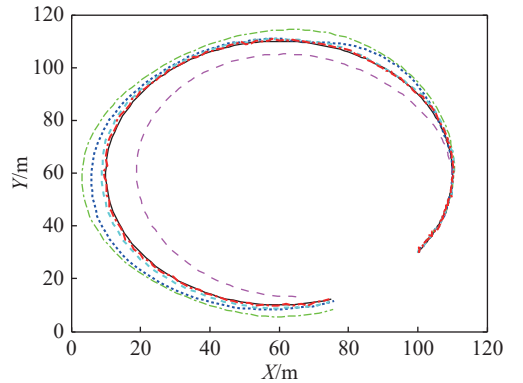


(c) Velocity RMSE

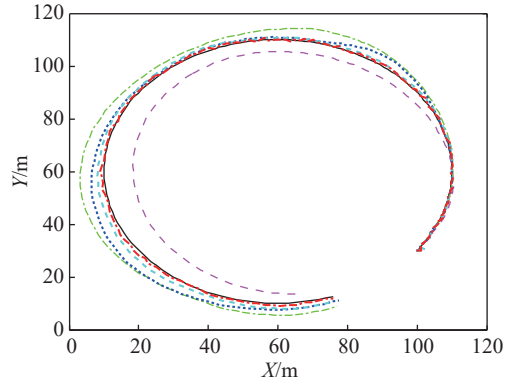
Fig. 4 Target trajectory and RMSEs of state estimates and 95% probability interval

In order to test the effect of the window length on the algorithms, the window length used in the proposed SM-based estimation algorithm and online fitting is set to 14 and 16.

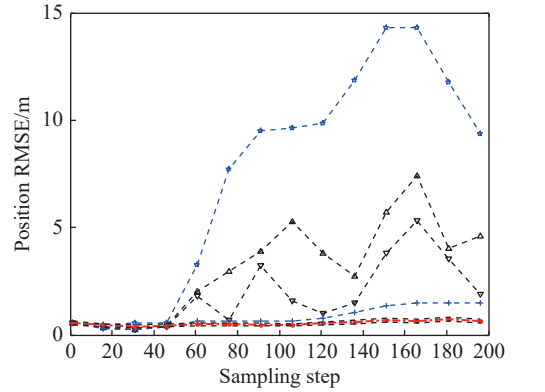
When increasing the window length to 14 and 16, the target trajectories and the position RMSEs of the five algorithms are shown in Fig. 5. From these figures, it can be seen that the proposed SM-based estimation algorithm performs better again than the EKF(CT), IMM-KF(CVCA), IMM-EKF(CVCACT), and online fitting when the target maneuvers occur. The performance of the online fitting deteriorates in comparison with the choice of a smaller window length value 12, in terms of both the trajectory and position estimation. The differences between the SM-based algorithm and the online fitting become larger compared with Fig. 4(b). In Fig. 5(c) and Fig. 5(d), the proposed SM-based algorithm exhibits robustness compared with other KF-based algorithms, suggesting that the RMSEs of the proposed SM-based algorithm are insensitive to the presence of the target maneuvers and the value of the window length.



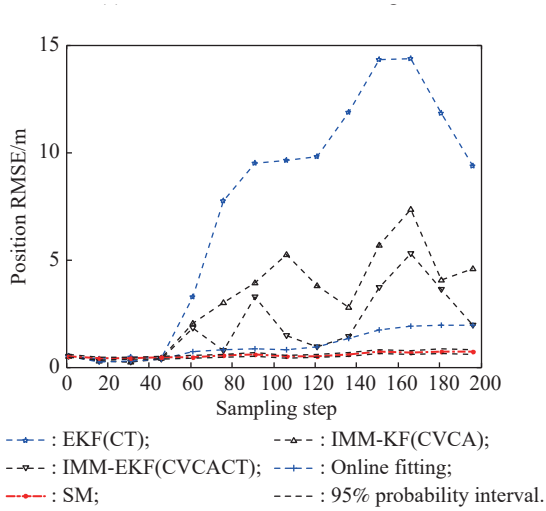
(a) Estimated target trajectory in window length 14



(b) Estimated target trajectory in window length 16



(c) Position RMSE in window length 14



(d) Position RMSE in window length 16

Fig. 5 Target trajectories and position RMSEs and 95% probability interval corresponding to window length 14 and 16

5.3 Curve trajectory in Scenario I

(i) Scenario setting

In this scenario, a target is simulated to move along a curve road. It starts at (−400 m, 0) and moves at a constant speed of 2.3 m/s from the start until 40. Then it speeds up with an acceleration from sampling step 41 to 70 with 3.7 m/s^2 and from 111 to 150 with 2.2 m/s^2, and maintains a constant speed at time 71−110 and 151−200. The curve trajectory is approximated with cubic splines which reduce the order of a polynomial because of the piecewise interpolation [29]. It is given by

$$\begin{cases} x = f_x(s) = a_{x,i} + b_{x,i}(s-s_i) + c_{x,i}(s-s_i)^2 + d_{x,i}(s-s_i)^3 \\ y = f_y(s) = a_{y,i} + b_{y,i}(s-s_i) + c_{y,i}(s-s_i)^2 + d_{y,i}(s-s_i)^3 \end{cases}. \tag{48}$$

The supporting points $P_i = \{p_{x,i}, p_{y,i}\}$ ($i = 1,2,\cdots,\rho$) and the second derivatives $M_{x,i}$ and $M_{y,i}$ of the spline curves at these points can be utilized to calculate the parameter vectors $[a_{x,i}, b_{x,i}, c_{x,i}, d_{x,i}]'$ and $[a_{y,i}, b_{y,i}, c_{y,i}, d_{y,i}]'$ for the x- and y-components, respectively. The term s_i denotes the starting arc length of the ith segment ($i = 1,2,\cdots,\rho-1$). The coefficients are calculated for the x-component as

$$\begin{cases} a_{x,i} = p_{x,i} \\ b_{x,i} = \dfrac{p_{x,i+1} - p_{x,i}}{\varepsilon_i} - \left(\dfrac{M_{x,i+1}}{6} + \dfrac{M_{x,i}}{3}\right)\varepsilon_i \\ c_{x,i} = \dfrac{M_{x,i}}{2} \\ d_{x,i} = \dfrac{M_{x,i+1} - M_{x,i}}{6\varepsilon_i} \end{cases} \tag{49}$$

where

$$\begin{cases} s_{i+1} = s_i + \displaystyle\int_{u_i}^{u_{i+1}} \|\dot{l}(\mu)\| \mathrm{d}\mu \\ \varepsilon_i = s_{i+1} - s_i \\ u_{i+1} = u_i + \sqrt{(p_{x,i+1} - p_{x,i})^2 + (p_{y,i+1} - p_{y,i})^2} \end{cases}, \tag{50}$$

and $\dot{l}(\mu)$ is the first derivative of $l(\mu)$. The term $l(\mu)$ is the cubic spline parameterized by the chord length. Its representation is the same as (48) except the arc length s and s_i are substituted with the chord length u and u_i. Related coefficients of y-component in (48) are deduced in an analogous manner with x-component. The detailed deductions of the coefficients can be found in [28].

Position measurements are generated by the models (11) and (48) every 0.2 s with the measurement noise variances being $\sigma_x^2 = 0.8$ m and $\sigma_y^2 = 0.8$ m along the x and y axes, respectively. The target trajectory is shown in Fig. 6.

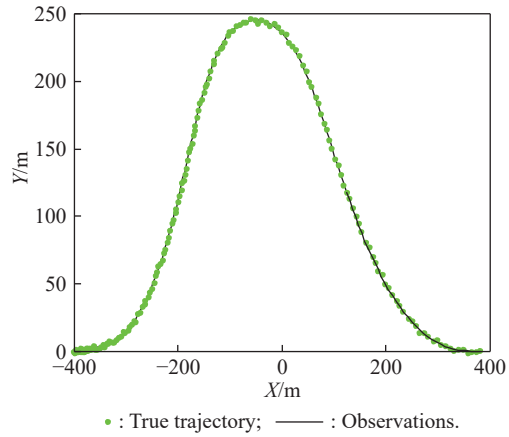


Fig. 6 Target trajectory in curve Scenario I

(ii) Estimator parameters

The proposed SM-based algorithm for curve trajectory estimation is compared with the KF(CA), IMM-KF (CVCA), IMM-EKF(CVCACT), and online fitting. The standard deviations of the process noises of NCV and NCA models are set as 0.01 m/s^2. The standard deviation of the artificial process noise is 0.002. The window length is 12. The performance of the five algorithms is evaluated from 100 Monte Carlo runs. The initial model probability vectors of the IMM-KF(CVCA) and SM algorithms are both $\boldsymbol{\mu}_0 = [0.5, 0.5]'$ and the corresponding model transition probability matrices are given by

$$\boldsymbol{\Pi} = \begin{bmatrix} 0.95 & 0.05 \\ 0.05 & 0.95 \end{bmatrix}.$$

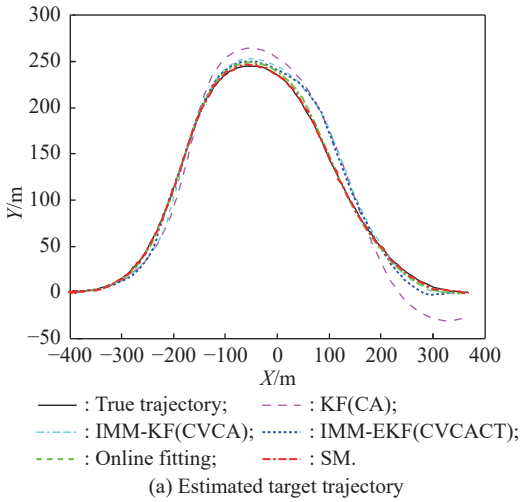
The IMM-EKF(CVCACT) uses $\boldsymbol{\mu}_0 = [0.4, 0.4, 0.2]'$ and

$$\boldsymbol{\Pi} = \begin{bmatrix} 0.94 & 0.03 & 0.03 \\ 0.03 & 0.94 & 0.03 \\ 0.05 & 0.05 & 0.90 \end{bmatrix}.$$

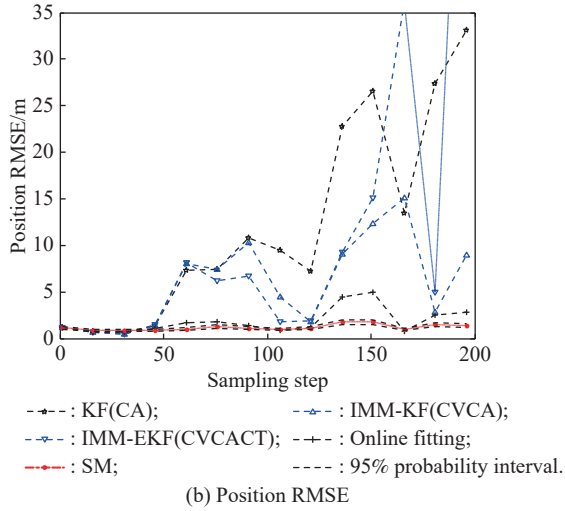
(iii) Results

Fig. 7(a) compares the target trajectories obtained by the proposed SM-based estimation algorithm, KF(CA), IMM-KF(CVCA), IMM-EKF(CVCACT), and online fit-

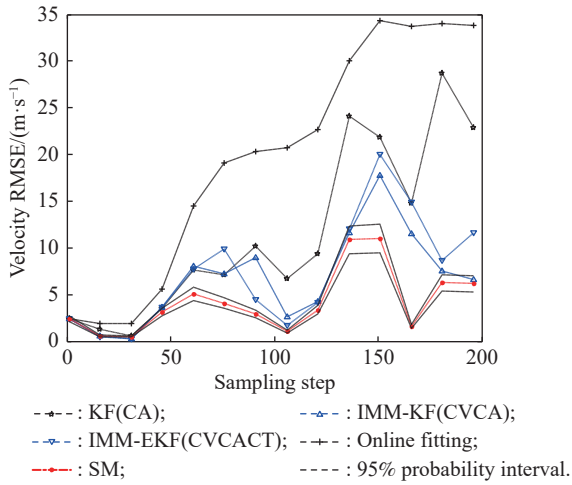
ting. It can be seen that the proposed SM-based algorithm outperforms other algorithms, which use mismatched coupled models to describe the target trajectory. The position and velocity RMSEs are given in Fig. 7(b) and Fig. 7(c), respectively. The proposed SM-based estimation algorithm performs better than the KF(CA), IMM-KF(CVCA), IMM-EKF(CVCACT), and online fitting in the presence of target maneuvers, in terms of both position and velocity estimation. This is because the KF(CA), IMM-KF(CVCA), and IMM-EKF(CVCACT) use the Cartesian coordinate models to describe target maneuvers and the online fitting keeps a constant dynamic by a continuous first-degree time function, while the proposed SM-based algorithm uses the mileage coordinate models. It suggests that the mileage coordinate models are better than both the Cartesian coordinate models and the continuous time function for describing target maneuvers, in the case of the trajectory shape of a target is independent of its dynamic characteristics.



(a) Estimated target trajectory



(b) Position RMSE



(c) Velocity RMSE

Fig. 7 Target trajectory and RMSEs of state estimates and 95% probability interval in curve Scenario I

5.4 Curve trajectory in Scenario II

In order to test the ability of the tracking algorithms to adapt to more complex scenarios, a more complex trajectory is assumed as shown in Fig. 8 compared with Scenario I.

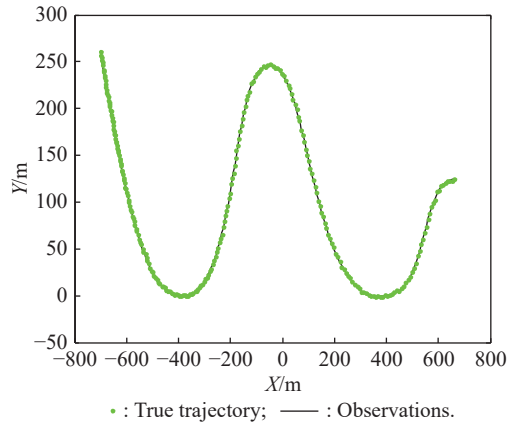


Fig. 8 Target trajectory in curve Scenario II

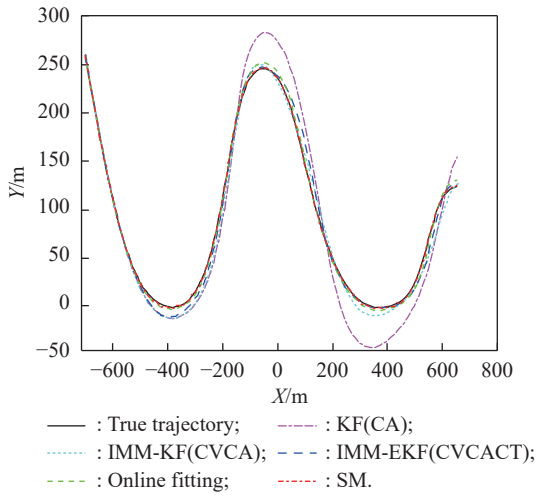
(i) Scenario setting

A target starts at (−700 m, 260 m) and its initial speed is 16.5 m/s. The target maintains a constant speed at time 0−40, 81−110, 151−180, and 211−250, and it speeds up at time 41−80, 111−150, and 181−210 with an acceleration of 1.9 m/s^2, 1.6 m/s^2, and 0.1 m/s^2, respectively. The rest parameters maintain unchanged as designed in Subsection 5.3 except the sampling step is 250. In order to evaluate the effect of the window length, the sensitivity of algorithms to different choices of the window length is analyzed by the following simulations.

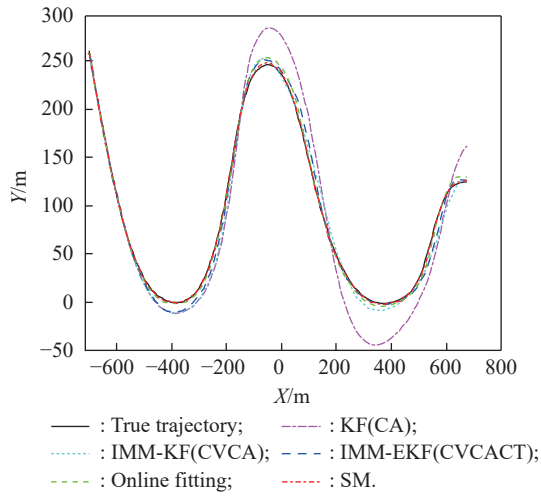
(ii) Results

Fig. 9 plots the target trajectories and position RMSEs obtained by the KF(CA), IMM-KF(CVCA), IMM-EKF(CVCACT), online fitting, and the proposed SM-based estimation algorithms, in the case of three window length values 10, 12, and 14. In the absence of turn maneuvers, the five estimators have close estimation performance for the target trajectory. The proposed SM-based algorithm can correctly estimate the target trajectory in the presence oftarget maneuvers, while the KF(CA), IMM-KF(CVCA), IMM-EKF(CVCACT), and online fitting have large error. The window length has no effect on Cartesian-coordinate-based KF algorithms that do not use the window length information. When the target maneuvers occur, the RMSEs of the proposed SM-based algorithm do not rise as sharply as those KF algorithms, suggesting that the proposed SM-based algorithm has relatively more robust performance than those KF algorithms during the target maneuvers. At a larger window length value (at choice of 14), the performance of the SM-based algorithm becomes remarkably better than the online fitting (see Fig. 9).

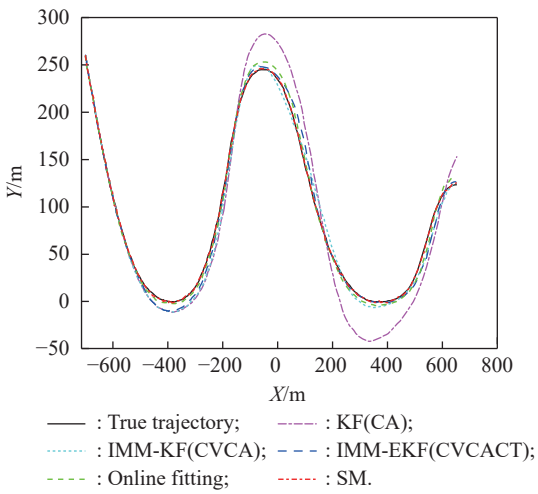
Fig. 10 presents average RMSEs of position and velocity estimates of the online fitting and proposed SM algorithms, in the case of four window length values. It is seen that the window length has a much greater impact on the position RMSE of the online fitting compared with the proposed SM algorithm (as the figure shown on the left-hand side). The figure on the right-hand side indicates that the performance of the velocity of the proposed SM algorithm has significant improvement in comparison with the online fitting.



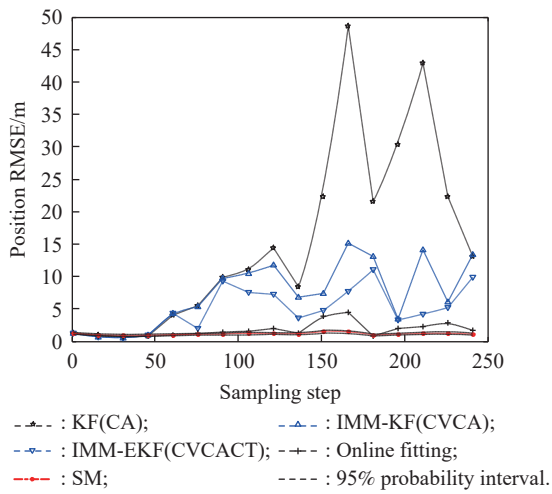
(a) Estimated trajectory in window length 10



(b) Estimated trajectory in window length 12



(c) Estimated trajectory in window length 14



(d) Position RMSE in window length 10

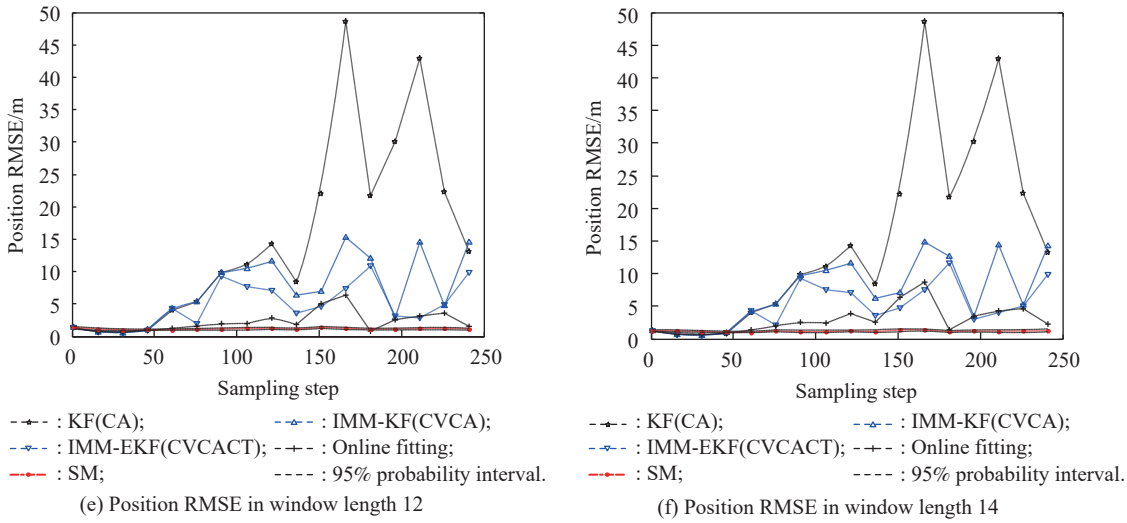


Fig. 9 Target trajectories and position RMSEs and 95% probability interval corresponding to window length 10, 12, and 14

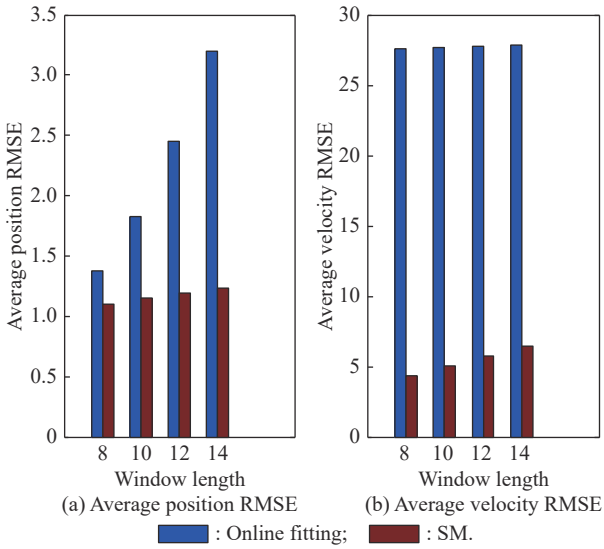


Fig. 10 Histogram of average RMSEs of position and velocity of the online fitting and the proposed SM-based algorithms in curve Scenario II

6. Conclusions

A new estimation algorithm is proposed to estimate the trajectory and state of a target which is subject to various constraints imposed by external environments such as roads, terrains, buildings, trees, and sea-routes. Based on the fact that the trajectory shape of a target subject to external environment constraints is independent of target dynamic characteristics, SM of target trajectory shape and target dynamic characteristics is developed. The target trajectory over a sliding window of the latest k_m measurements is described by a function of the arc length, which is determined by three attributes: starting point, direction vector, andstarting arc length. To determine the target trajectory, an augmented system is generated by using the function coefficients to augment the base state. The proposed algorithm is used to estimate simultaneously the target trajectory and target velocity properties. At every estimation cycle except the first one, the interaction stage of the proposed algorithm starts with the latest updated base state and the reset parameter vector. The parameter vector is resetby the least squares.The proposed algorithm is compared with the KF-based and online fitting algorithms with coupled systems using numerical experiments. Simulation results indicate that the proposed SM-based algorithm performs better with respect to both the trajectory and state estimates when target maneuvers occur, due to SM of target trajectory shape and target dynamic characteristics.

References

[1] KIRUBARAIAN T, BAR-SHALOM Y, PATTIPATI K, et al. Ground target tracking with variable structure IMM estimator. IEEE Trans. on Aerospace and Electronic Systems, 2000, 36(1): 26–46.
[2] ZHAO J, ZHOU R, JIN X L. Progress in reentry trajectory planning for hypersonic vehicle. Journal of Systems Engineering and Electronics, 2014, 25(4): 101–113.
[3] SIMON D. Optimal state estimation: Kalman, H infinity, and nonlinear approaches. New York: John Wiley & Sons, 2006.
[4] BAR-SHALOM Y, LI X R, KIRUBARAIAN T. Estimation with applications to tracking and navigation: theory, algorithms, and software. New York: Wiley, 2001.
[5] ZHANG, Z B, JI H B, YANG J. Autonomous optical navigation of Mars probe aided by one-way Doppler measurements in capture stage. Journal of Systems Engineering and Electronics, 2020, 31(3): 160–169.
[6] LI X R, JILKOV V P. Survey of maneuvering target track-

ing. Part I: dynamic models. IEEE Trans. on Aerospace and Electronic Systems, 2003, 39(4): 1333–1364.
[7] LI X R, JILKOV V P. Survey of maneuvering target tracking. Part V: multiple-model methods. IEEE Trans. on Aerospace and Electronic Systems, 2005, 41(4): 1255–1321.
[8] LI X R, JILKOV V P. A survey of maneuvering target tracking. Part IV: decision-based methods. Proc. of SPIE- the International Society for Optical Engineering, 2002: 511–534.
[9] HARTIKAINEN J, SOLIN A, SARKKA S. Optimal filtering with Kalman filters and smoothers: a manual for the Matlab toolbox EKF/UKF. https://www.researchgate.net/publication/228683456.
[10] LI X R. Multiple-model estimation with variable structure. Part II: model-set adaptation. IEEE Trans. on Automatic Control, 2000, 45(11): 2047–2060.
[11] LI X R, ZHAO Z L, LI X B. General model-set design methods for multiple-model approach. IEEE Trans. on Automatic Control, 2005, 50(9): 1260–1276.
[12] LI X R, BAR-SHALOM Y. Design of an interacting multiple model algorithm for air traffic control tracking. IEEE Trans. on Control Systems Technology, 1993, 1(3): 186–194.
[13] ZDZISLAW K, MIROSLAW S. Soft- and hard-decision multiple-model estimators for air traffic control. IEEE Trans. on Aerospace and Electronic Systems, 2010, 46(4): 2056–2065.
[14] SINGER R A. Estimating optimal tracking filter performance for manned maneuvering targets. IEEE Trans. on Aerospace and Electronic Systems, 1970, 6(4): 473–483.
[15] ZHOU H, KUMAR K S P. A current statistical model and adaptive algorithm for estimating maneuvering targets. Journal of Guidance, Control, and Dynamics, 1984, 7(5): 596–602.
[16] QIAN X D, WANG B S. A motion model for tracking highly maneuvering targets. Proc. of the IEEE Radar Conference, 2002. DOI: 10.1109/NRC.2002.999767.
[17] SONG D, THARMARASA R, ZHOU G J, et al. Multi-vehicle tracking using microscopic traffic models. IEEE Trans. on Intelligent Transportation Systems, 2019, 20(1): 149–161.
[18] JO K, LEE M, KIM J, et al. Tracking and behavior reasoning of moving vehicles based on roadway geometry constraints. IEEE Trans. on Intelligent Transportation Systems, 2017, 18(2): 460–476.
[19] JO K, LEE M, SUNWOO M. Track fusion and behavioral reasoning for moving vehicles based on curvilinear coordinates of roadway geometries. IEEE Trans. on Intelligent Transportation Systems, 2018, 19(9): 3068–3074.
[20] YANG C, BAKICH M, BLASCH E. Nonlinear constrained tracking of targets on roads. Proc. of the 8th International Conference on Information Fusion, 2005: 235–242.
[21] CHEN Y S, JILKOV V, LI X R. Multilane-road target tracking using radar and image sensors. IEEE Trans. on Aerospace and Electronic Systems, 2015, 51(1): 65–80.
[22] ULMKE M. Improved GMTI-tracking using road-maps and topographic information. Proc. of SPIE-the International Society for Optical Engineering, 2003: 143–154.
[23] ULMKE M, KOCH W. Road-map assisted ground moving target tracking. IEEE Trans. on Aerospace and Electronic Systems, 2006, 42(4): 1264–1274.
[24] MERTENS M, ULMKE M. Precision GMTI tracking using road constraints with visibility information and a refined sensor model. Proc. of the IEEE Radar Conference, 2008: 1–6.
[25] KOCH W, KOLLER J, ULMKE M. Ground target tracking and road map extraction. Journal of Photogrammetry and Remote Sensing, 2006, 61(3): 197–208.
[26] SONG D, THARMARASA R, KIRUBARAIAN T, et al. Multi-vehicle tracking with road maps and car-following models. IEEE Trans. on Intelligent Transportation Systems, 2018, 19(5): 1375–1386.
[27] TIAN Z, LI Y G, CEN M, et al. Multi-vehicle tracking using an environment interaction potential force model. IEEE Sensors Journal, 2020, 20(20): 12282–12294.
[28] HASBERG C, HENSEL S, STILLER C. Simultaneous localization and mapping for path-constrained motion. IEEE Trans. on Intelligent Transportation Systems, 2012, 13(2): 541–552.
[29] WANG H L, KEARNEY J, ATKINSON K. Arc-length parameterized spline curves for real-time simulation. Proc. of the 5th Conference Curve and Surface Design, 2002: 387–396.
[30] LI T C, CHEN H M, SUN S D, et al. Joint smoothing, tracking, and forecasting based on continuous-time target trajectory fitting. IEEE Trans.on Automation Science and Engineering, 2017, 16(3): 1476−1483.
[31] LI K Y, KIRUBARAIAN T, ZHOU G J. State estimation with implicit constraints of circular trajectory using pseudo-measurements. IEEE Trans. on Aerospace and Electronic Systems, 2020. DOI:10.1109/TAES.2020.2988894.
[32] ZHOU G J, LI K Y, KIRUBARAIAN T. State estimation with trajectory shape constraints using pseudo-measurements. IEEE Trans. on Aerospace and Electronic Systems, 2019, 55(5): 2395–2407.
[33] LIU C Y, SHUI P L, WEI G, et al. Modified unscented Kalman filter using modified filter gain and variance scale factor for highly maneuvering target tracking. Journal of Systems Engineering and Electronics, 2014, 25(3): 380–385.
[34] DENG F, CHEN J, CHEN C. Adaptive unscented Kalman filter for parameter and state estimation of nonlinear high-speed objects. Journal of Systems Engineering and Electronics, 2013, 24(4): 655–665.
[35] ARASARATNAM T, HAYKIN S. Cubature Kalman filter. IEEE Trans. on Automatic Control, 2009, 54(6): 1254–1269.
[36] NIE X, ZHANG F M. Adaptive tracking algorithm based on 3D variable turn model. Journal of Systems Engineering and Electronics, 2017, 28(5): 851–860.
[37] JULIER S J. A new method for nonlinear transformation of means and covariances in filters and estimates. IEEE Trans. on Automatic Control, 2000, 45(3): 477–482.
[38] WAN E A, VAN D M R. The unscented Kalman filter for nonlinear estimation. Proc. of the IEEE Adaptive Systems for Signal Processing, Communication and Control Symposium, 2000: 153–158.

Biographies

ZHANG Zhuanhua was born in 1989. She received her M.E. degree from Northwest Normal University. She is currently pursuing her Ph.D. degree in Harbin Institute of Technology. Her research interests include estimation and tracking.
E-mail: zhuanhuazhang@163.com

ZHOU Gongjian was born in 1979. He is a Ph.D. and a professor with the Department of Electronic Engineering, Harbin Institute of Technology. He is also a Longjiang Young Scholar. His research interests include estimation, tracking, detection, information fusion and signal processing.
E-mail: zhougj@hit.edu.cn