A Batch Update Using Multiplicative Noise Modelling for Extended Object Tracking

Christian Gramsch, Shishan Yang and Hosam Alqaderi

***Abstract*—While the tracking of multiple extended targets demands for sophisticated algorithms to handle the high complexity inherent to the task, it also requires low runtime for online execution in real-world scenarios. In this work, we derive a batch update for the recently introduced elliptical-target tracker called MEM-EKF*. The MEM-EKF* is based on the same likelihood as the well-established random matrix approach but is derived from the multiplicative error model (MEM) and uses an extended Kalman filter (EKF) to update the target state sequentially, i.e., measurement-by-measurement. Our batch variant updates the target state in a single step based on straightforward sums over all measurements and the MEM-specific pseudo-measurements. This drastically reduces the scaling constant for typical implementations and indeed we find a speedup of roughly 100x in our numerical experiments. At the same time, the estimation error which we measure using the Gaussian Wasserstein distance stays significantly below that of the random matrix approach in coordinated turn scenarios while being comparable otherwise.**

***Index Terms*—Target tracking, extended object tracking, multiplicative error, Kalman filter, Information filter**

I. Introduction

Autonomous and assisted driving offer safer and more efficient transportation options for our society. With environment perception serving as a key technology enabling the realization of autonomous and assisted driving system, object tracking emerges as one of the most fundamental components of environment perception. Object tracking aims at estimating and predicting the objects states in an area of interest. Traditional object tracking assumes the tracked object as a point and has at most one detection. However, this assumption is challenged by the near-field and high-resolution sensors, which are widely equipped in modern vehicles. For the modern ranged-based sensors, such as RAdio detection and ranging (Radar) and LIght Detection And Ranging (Lidar), one traffic participant emits multiple measurements and its extent cannot be reduced to a point for safety consideration. Extended object tracking estimates and predicts the kinematic and shape of an object given a set of detections. A detailed overview of extended object tracking is given in [1].

The origins of measurements from an extended object can be modelled as reflections from specific positions on the object's surface [2] or an independent random draw from a spatial distribution [3], [4], which avoids the explicit association between measurements and measurement-sources. The shape of an extended object can be interpreted with different levels of complexity depending on the computation and application requirements. For object tracking scenarios in autonomous and assisted driving, the extent of objects is often approximated by simple geometry shapes, such as rectangles and ellipses [5], [6], [7] – the latter being among the most popular due to their statistical convenience in representing Gaussian distributions. Their tracking was pioneered by Random-matrix-based methods [5], [8] which represent elliptical objects using a 2D semi-definite matrix. While elegant, this representation causes an increased error in coordinated turn scenarios due to the implicit coupling of the uncertainty of orientation and size. To address this issue, a non-linear measurement model which relates measurements, position, orientation and axes-lengths was formulated in [9]. There, the key in deriving the measurement equations is the modelling via multiplicative noise – also known as the Multiplicative Error Model (MEM) [10]. Indeed it was found that the MEM formulation improves the performance in coordinate turn scenarios significantly as the dynamics of parameters can be constraint individually. However, one disadvantage of the MEM formulation [9] is its sequential Kalman update, which causes a slower runtime and is sensitive to the measurement order.

In this work, we propose a new filter for extended object tracking, which is derived from the multiplicative error model (MEM) in [9], but uses a batch update based on the extended information filter (EIF). The proposed tracker is named as MEM-EIF with MEM referring to the measurement model and EIF denoting the extended information filter. Our new tracker MEM-EIF: i) decouples the dynamics of object orientation and axes lengths compared with Random Matrix based methods; ii) performs a batch update and thus significantly improves runtime-performance while being robust against changes in the order of processed measurements compared with MEM-EKF*.

Interestingly, the information filter has been combined with the MEM-EKF* before in [11] but in the different context of information fusion. In [11], a distributed tracking system was developed which retains the consensus regarding shape and kinematics among different sensor nodes on a network.

The rest of our paper is structured as follows. The notation is introduced in the next subsection. In section II, we briefly explain the information filter. We derive our batch update using extended information filter in Section III. The numeric evaluation and its result is presented in Section IV. The paper is concluded in Section V.

C. Gramsch and S. Yang work at MicroVision GmbH, Neuer Höltigbaum 6, 22143 Hamburg, Germany (e-mail: gramsch@mailbox.org; yangshishan@gmail.com)). H. Alqaderi is affiliated with the University of Bonn, Germany (e-mail: hosam.alqaderi@uni-bonn.de)

A. Notation

We denote vector-valued quantities in italic, bold (e.g., $\boldsymbol{y} = (y_1, y_2)^{\mathrm{T}}$ for a 2d-measurement). Upright bold quantities, such as the measurement matrix $\mathbf{H}$, denote matrices. Scalars are denoted in non-bold italic, e.g., L for the number of measurements. We indicate the expectation value of random quantities by a bar, e.g., $\bar{\boldsymbol{x}} = \mathbb{E}[\boldsymbol{x}]$ for a vector-valued random variable $\boldsymbol{x}$ where $\mathbb{E}[\cdot]$ denotes the operator taking the expectation value on a suitable probability space. The covariance matrix of a random vector $\boldsymbol{x}$ is denoted as $\mathbf{C}^{\boldsymbol{x}}$ and the matrix of the cross-covariance between two random variables $\boldsymbol{x}$ and $\boldsymbol{q}$ as $\mathbf{C}^{\boldsymbol{xq}}$. Given $\boldsymbol{y}_1, \ldots, \boldsymbol{y}_i$, we indicate the conditional expectation value, the conditional covariance matrix, and conditional cross-covariance using a subscript i, i.e., $\bar{\boldsymbol{x}}_i$, $\mathbf{C}^{\boldsymbol{x}}_i$, $\mathbf{C}^{\boldsymbol{xq}}_i$, respectively. Additionally, $\boldsymbol{x} \sim \mathcal{N}(\bar{\boldsymbol{x}}, \mathbf{C}^{\boldsymbol{x}})$ denotes that $\boldsymbol{x}$ is Gaussian distributed with mean $\bar{\boldsymbol{x}}$ and covariance $\mathbf{C}^{\boldsymbol{x}}$.

II. The Information Filter

To introduce the information filter, we consider L measurements $\boldsymbol{y}_1$, ..., $\boldsymbol{y}_L$ subject to a linear stochastic model:

$$\boldsymbol{y}_i = \mathbf{H}\boldsymbol{x} + \boldsymbol{v}_i, \text{ for } i = 1, \cdots, L, \tag{1}$$

where $\mathbf{H}$ is the measurement matrix. The measurement noise $\boldsymbol{v}_i \sim \mathcal{N}(0, \mathbf{C}^{\boldsymbol{v}})$ and the state $\boldsymbol{x} \sim \mathcal{N}(\bar{\boldsymbol{x}}_0, \mathbf{C}^{\boldsymbol{x}}_0)$ are Gaussian distributed. For the latter, we denote $\bar{\boldsymbol{x}}_0$ and $\mathbf{C}^{\boldsymbol{x}}_0$ as the predicted mean and covariance. Given $\boldsymbol{y}_1, ..., \boldsymbol{y}_i$, for $i \leq L$, $\bar{\boldsymbol{x}}_i$ and covariance $\mathbf{C}^{\boldsymbol{x}}_i$ can be calculated using a sequential Kalman update

$$\bar{\boldsymbol{x}}_i = \bar{\boldsymbol{x}}_{i-1} + \mathbf{C}^{\boldsymbol{xy}}_i \left[\mathbf{C}^{\boldsymbol{y}}_i\right]^{-1} (\boldsymbol{y}_i - \bar{\boldsymbol{y}}_i), \tag{2}$$

$$\mathbf{C}^{\boldsymbol{x}}_i = \mathbf{C}^{\boldsymbol{x}}_{i-1} - \mathbf{C}^{\boldsymbol{xy}}_i \left[\mathbf{C}^{\boldsymbol{y}}_i\right]^{-1} \left(\mathbf{C}^{\boldsymbol{xy}}_i\right)^{\mathrm{T}}, \tag{3}$$

with the innovation covariance $\mathbf{C}^{\boldsymbol{y}}_i = \mathbf{C}^{\boldsymbol{v}} + \mathbf{H}\mathbf{C}^{\boldsymbol{x}}_{i-1}\mathbf{H}^{\mathrm{T}}$ and the cross-covariance $\mathbf{C}^{\boldsymbol{xy}}_i = \mathbf{C}^{\boldsymbol{x}}_{i-1}\mathbf{H}^{\mathrm{T}}$.

As is well-known (see [12]), an equivalent information filter can update with all measurements in a single step. A brief derivation is given in Appendix A. We define $\bar{\boldsymbol{\xi}}_i \equiv [\mathbf{C}^{\boldsymbol{x}}_i]^{-1}\bar{\boldsymbol{x}}_i$ such that

$$\bar{\boldsymbol{\xi}}_i = \bar{\boldsymbol{\xi}}_{i-1} + \mathbf{H}^{\mathrm{T}}[\mathbf{C}^{\boldsymbol{v}}]^{-1}\boldsymbol{y}_i, \tag{4}$$

$$[\mathbf{C}^{\boldsymbol{x}}_i]^{-1} = [\mathbf{C}^{\boldsymbol{x}}_0]^{-1} + \mathbf{H}^{\mathrm{T}}[\mathbf{C}^{\boldsymbol{v}}]^{-1}\mathbf{H}. \tag{5}$$

Iterating the update rule gives us the following batch update:

$$\bar{\boldsymbol{\xi}}_L = \bar{\boldsymbol{\xi}}_0 + \mathbf{H}^{\mathrm{T}}[\mathbf{C}^{\boldsymbol{v}}]^{-1}\sum_{i=1}^{L}\boldsymbol{y}_i, \tag{6}$$

$$[\mathbf{C}^{\boldsymbol{x}}_L]^{-1} = [\mathbf{C}^{\boldsymbol{x}}_0]^{-1} + L\mathbf{H}^{\mathrm{T}}[\mathbf{C}^{\boldsymbol{v}}]^{-1}\mathbf{H}, \tag{7}$$

where $\bar{\boldsymbol{x}}_L = \mathbf{C}^{\boldsymbol{x}}_L\bar{\boldsymbol{\xi}}_L$ recovers the original state.

The main characteristic of this batch form is the simple dependence on the outcomes $\boldsymbol{y}_i$ which enter via their sum only. This summation can typically be implemented in a very efficient manner, e.g., by employing vectorization or even parallelization. In addition, only three matrix inversions are required to obtain the final outcome compared to L such inversions in the standard Kalman form.

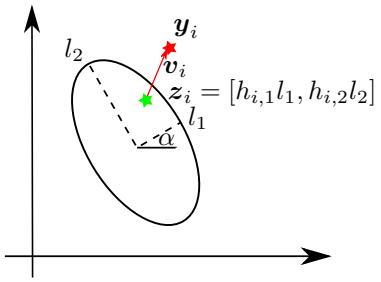

Fig. 1. This figure depicts the multiplication error measurement model for extended object tracking. A measurement $\boldsymbol{y}_i$ is the measurement source $\boldsymbol{z}_i$ corrupted with sensor noise $\boldsymbol{v}_i$. The measurement source $\boldsymbol{z}_i$ relates object orientation α and axes-lengths $(l_1, l_2)^{\mathrm{T}}$ by a multiplicative noise $\boldsymbol{h}_i$, which are uniformly distributed on a unit circle.

III. Application to the MEM-EKF

Elliptical extend-object tracking focuses on updating the state of a target that has produced L measurements. In a recent work [9], this setting has been modeled using the multiplicative error model [10] to which we give a short introduction in subsection III-A. The closed form solution derived in [9] is based on a problem-tailored sequential extended Kalman filter (EKF). The EKF update formulæ consist of two parts: one update for kinematics and another for the shape. Similarly, our batch update consists of a kinematic and a shape update step which are derived in the subsections III-B and III-C. In contrast to [9], however, all the measurements are processed within a single step.

A. Multiplicative error model (MEM)

We consider an elliptical target moving in the 2d-plane with unknown kinematic state $\boldsymbol{r} = (r_x, r_y, r_{\dot{x}}, r_{\dot{y}}, r_{\ddot{x}}, r_{\ddot{y}})^{\mathrm{T}}$, where $(r_x, r_y)^{\mathrm{T}}$ describes the target's 2d position, $(r_{\dot{x}}, r_{\dot{y}})^{\mathrm{T}}$ its velocity and $(r_{\ddot{x}}, r_{\ddot{y}})^{\mathrm{T}}$ its acceleration. The shape of the target is assumed to be that of an ellipse with unknown parameters denoted as $\boldsymbol{p} = (\alpha, l_1, l_2)$, where α is the orientation of the ellipse while l_1 and l_2 denote the lengths of its major and minor semi-axes, respectively. The elliptical target is assumed to give rise to L measurements $\boldsymbol{y}_1, ..., \boldsymbol{y}_L$ via the measurement model

$$\boldsymbol{y}_i = \mathbf{H}\boldsymbol{r} + \mathbf{S}\boldsymbol{h}_i + \boldsymbol{v}_i, \tag{8}$$

where

$$\mathbf{H} = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \end{bmatrix} \equiv \begin{bmatrix} \mathbf{H}_1 \\ \mathbf{H}_2 \end{bmatrix} \tag{9}$$

projects the target position into the 2d-plane, $\boldsymbol{h}_i$ is uniformly distributed on a unit circle, while $\boldsymbol{v}_i \sim \mathcal{N}(0, \mathbf{C}^{\boldsymbol{v}})$. $\mathbf{S} \equiv \mathbf{S}(\boldsymbol{p})$ describes the shape of the target, where

$$\mathbf{S}(\boldsymbol{p}) = \begin{bmatrix} l_1\cos(\alpha) & -l_2\sin(\alpha) \\ l_1\sin(\alpha) & l_2\cos(\alpha) \end{bmatrix} \equiv \begin{bmatrix} \mathbf{S}_1(\boldsymbol{p}) \\ \mathbf{S}_2(\boldsymbol{p}) \end{bmatrix}. \tag{10}$$

The model is illustrated in Fig. 1. One of the main challenges in deriving a closed-form solution to MEM is the high non-linearity in $\mathbf{S}\boldsymbol{h}_i$. This hinders a simultaneous estimation of kinematics and shape using a linear minimum mean squared

error (LMMSE) estimator [10] and motivates the introduction of a second-order pseudo-measurement

$$\boldsymbol{Y}_i = \mathbf{F}\left(\boldsymbol{y}_i - \bar{\boldsymbol{y}}_i\right) \otimes \left(\boldsymbol{y}_i - \bar{\boldsymbol{y}}_i\right), \tag{11}$$

which is used to update the estimate for the shape state. Here $\boldsymbol{u} \otimes \boldsymbol{w} = (u_1 w_1, u_1 w_2, u_2 w_1, u_2 w_2)^{\mathrm{T}}$ denotes the Kronecker product. The sole purpose of the matrix $\mathbf{F}$ (and its sibling $\tilde{\mathbf{F}}$ which will come in handy later) is to remove one of the identical off-diagonals

$$\mathbf{F} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 1 & 0 & 0 \end{bmatrix}, \quad \tilde{\mathbf{F}} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{bmatrix}. \tag{12}$$

It is worthwhile to note that the introduction of a second-order pseudo-measurement to account for the non-linearity can be put on solid stochastical ground by taking a measure-theoretic viewpoint (cf. [13]). For a more intuitive treatment on the basis of uncorrelated transforms we refer to [14].

As we focus on deriving a batch measurement update, the time index k is omitted for the seek of compactness.[1] Given the measurement outcomes $\boldsymbol{y}_1, ..., \boldsymbol{y}_L$ at our current time, the prior distributions as $\boldsymbol{r} \sim \mathcal{N}(\bar{\boldsymbol{r}}_0, \mathbf{C}_0^{\boldsymbol{r}})$ and $\boldsymbol{p} \sim \mathcal{N}(\bar{\boldsymbol{p}}_0, \mathbf{C}_0^{\boldsymbol{p}})$, our goal is to obtain a batch update along the lines of Eqs. (6-7) for both kinematics and shape.

B. Kinematics

To apply the information filter to the kinematic update of the MEM-EKF*, it is necessary to identify the linearized measurement model that has been used to derive the Kalman update given by Eqs. (11-13) in [9]. With

$$\bar{\boldsymbol{y}}_i = \mathbf{H}\bar{\boldsymbol{r}}_{i-1}, \tag{13}$$

the Kalman update reads

$$\bar{\boldsymbol{r}}_i = \bar{\boldsymbol{r}}_{i-1} + \mathbf{C}_i^{\boldsymbol{ry}} \left[\mathbf{C}_i^{\boldsymbol{y}}\right]^{-1} \left(\boldsymbol{y}_i - \bar{\boldsymbol{y}}_i\right), \tag{14}$$

$$\mathbf{C}_i^{\boldsymbol{r}} = \mathbf{C}_{i-1}^{\boldsymbol{r}} - \mathbf{C}_i^{\boldsymbol{ry}} \left[\mathbf{C}_i^{\boldsymbol{y}}\right]^{-1} \left(\mathbf{C}_i^{\boldsymbol{ry}}\right)^{\mathrm{T}}. \tag{15}$$

According to [9], $\mathbf{C}_i^{\boldsymbol{y}}$ and $\mathbf{C}_i^{\boldsymbol{ry}}$ are given as follows

$$\mathbf{C}_i^{\boldsymbol{ry}} = \mathbf{C}_{i-1}^{\boldsymbol{r}} \mathbf{H}^{\mathrm{T}}, \tag{16}$$

$$\mathbf{C}_i^{\boldsymbol{y}} = \mathbf{H}\mathbf{C}_{i-1}^{\boldsymbol{r}} \mathbf{H}^{\mathrm{T}} + \mathbf{C}_i^{\mathrm{I}} + \mathbf{C}_i^{\mathrm{II}} + \mathbf{C}^{\boldsymbol{v}}, \tag{17}$$

where $\mathbf{C}^{\boldsymbol{v}}$ is the covariance of the measurement noise. Furthermore

$$\mathbf{C}_i^{\mathrm{I}} = \mathbf{S}(\bar{\boldsymbol{p}}_{i-1}) \mathbf{C}^{\boldsymbol{h}} \mathbf{S}(\bar{\boldsymbol{p}}_{i-1})^{\mathrm{T}}, \tag{18}$$

with $\mathbf{C}^{\boldsymbol{h}}$ being the covariance of the multiplicative noise and finally:

$$[\mathbf{C}_i^{\mathrm{II}}]_{mn} = \mathrm{tr}\left\{\mathbf{C}_0^{\boldsymbol{p}} \mathbf{J}_{n,i-1}^{\mathrm{T}} \mathbf{C}^{\boldsymbol{h}} \mathbf{J}_{m,i-1}\right\} \text{ for } m, n = 1, 2, \tag{19}$$

$$\mathbf{J}_{n,i-1} \equiv \mathbf{J}_n(\boldsymbol{p}_{i-1}) \equiv \left.\frac{\partial \mathbf{S}_n(\boldsymbol{p})}{\partial \boldsymbol{p}}\right|_{\boldsymbol{p}=\bar{\boldsymbol{p}}_{i-1}}. \tag{20}$$

Regarding the derivation of these quantities, we refer to [9].

The Kalman update Eqs. (14-15) can be derived from the following linearized measurement model

$$\boldsymbol{y}_i = \mathbf{H}\boldsymbol{r} + \boldsymbol{s}_i, \tag{21}$$

[1] For example, $\boldsymbol{y}_i$ corresponds to $\boldsymbol{y}_k^{(i)}$ in [9].

where $\boldsymbol{s}_i \sim \mathcal{N}\left(0, \mathbf{C}_i^{\mathrm{I}} + \mathbf{C}_i^{\mathrm{II}} + \mathbf{C}^{\boldsymbol{v}}\right) \equiv \mathcal{N}(0, \mathbf{C}_i^{\boldsymbol{s}})$. In contrast to the original model, Eq. (8), the non-linearity $\mathbf{S}\boldsymbol{h}_i$ is accounted for only effectively via a renormalization of the measurement noise. The validity of the model is easily verified by plugging the definition of $\mathbf{H}$ from Eq. (9) and the measurement noise $\mathbf{C}_i^{\boldsymbol{s}}$ into the formulæ of the Kalman update, Eqs. (2-3).

To obtain a batch update, we approximate $\mathbf{C}_i^{\boldsymbol{s}} \approx \mathbf{C}_1^{\boldsymbol{s}} \equiv \mathbf{C}^{\boldsymbol{s}}$. This implies that we are ignoring any feedback of changes to the object shape estimate during the kinematic update. Iterating the information update yields

$$\bar{\boldsymbol{\xi}}_L^{\boldsymbol{r}} = \bar{\boldsymbol{\xi}}_0^{\boldsymbol{r}} + \mathbf{H}^{\mathrm{T}} [\mathbf{C}^{\boldsymbol{s}}]^{-1} \sum_{i=1}^{L} \boldsymbol{y}_i, \tag{22}$$

$$[\mathbf{C}_L^{\boldsymbol{r}}]^{-1} = [\mathbf{C}_0^{\boldsymbol{r}}]^{-1} + L\mathbf{H}^{\mathrm{T}} [\mathbf{C}^{\boldsymbol{s}}]^{-1} \mathbf{H}, \tag{23}$$

$$\bar{\boldsymbol{r}}_L = \mathbf{C}_L^{\boldsymbol{r}} \bar{\boldsymbol{\xi}}_L^{\boldsymbol{r}}, \quad \bar{\boldsymbol{\xi}}_0^{\boldsymbol{r}} = [\mathbf{C}_0^{\boldsymbol{r}}]^{-1} \bar{\boldsymbol{r}}_0. \tag{24}$$

where L is the number of measurements.

C. Shape Update

The shape update goes along similar lines. First of all, the expectation value of the pseudo-measurement is given as

$$\bar{\boldsymbol{Y}}_i = \mathbf{F}\mathrm{vect}\left\{\mathbf{C}_i^{\boldsymbol{y}}\right\}, \tag{25}$$

where the $\mathrm{vect}\{\cdot\}$ operator transforms a matrix into a column vector as follows

$$\mathrm{vect}\left\{\begin{bmatrix} m_{11} & m_{12} \\ m_{21} & m_{22} \end{bmatrix}\right\} = \begin{bmatrix} m_{11} & m_{12} & m_{21} & m_{22} \end{bmatrix}^{\mathrm{T}}. \tag{26}$$

The Kalman update for the shape parameters now reads (cf. Eqs. (25-26) in [9])

$$\bar{\boldsymbol{p}}_i = \bar{\boldsymbol{p}}_{i-1} + \mathbf{C}_i^{\boldsymbol{pY}} \left[\mathbf{C}_i^{\boldsymbol{Y}}\right]^{-1} \left(\boldsymbol{Y}_i - \bar{\boldsymbol{Y}}_i\right), \tag{27}$$

$$\mathbf{C}_i^{\boldsymbol{p}} = \mathbf{C}_{i-1}^{\boldsymbol{p}} - \mathbf{C}_i^{\boldsymbol{pY}} \left[\mathbf{C}_i^{\boldsymbol{Y}}\right]^{-1} \left(\mathbf{C}_i^{\boldsymbol{pY}}\right)^{\mathrm{T}}, \tag{28}$$

with

$$\mathbf{C}_i^{\boldsymbol{pY}} = \mathbf{C}_{i-1}^{\boldsymbol{p}} \mathbf{M}_{i-1}^{\mathrm{T}} \tag{29}$$

and (cf. Eqs. (10, 20) for the definitions of $\mathbf{S}_n(\boldsymbol{p})$ and $\mathbf{J}_n(\boldsymbol{p}_i)$)

$$\mathbf{M}_i \equiv \mathbf{M}(\boldsymbol{p}_i) \equiv \begin{bmatrix} 2\mathbf{S}_1(\boldsymbol{p}_i) \mathbf{C}^{\boldsymbol{h}} \mathbf{J}_1(\boldsymbol{p}_i) \\ 2\mathbf{S}_2(\boldsymbol{p}_i) \mathbf{C}^{\boldsymbol{h}} \mathbf{J}_2(\boldsymbol{p}_i) \\ \mathbf{S}_1(\boldsymbol{p}_i) \mathbf{C}^{\boldsymbol{h}} \mathbf{J}_2(\boldsymbol{p}_i) + \mathbf{S}_2(\boldsymbol{p}_i) \mathbf{C}^{\boldsymbol{h}} \mathbf{J}_1(\boldsymbol{p}_i) \end{bmatrix}^{\mathrm{T}}. \tag{30}$$

The covariance of the pseudo-measurement is given as

$$\mathbf{C}_i^{\boldsymbol{Y}} = \mathbf{F}\left(\mathbf{C}_i^{\boldsymbol{y}} \otimes \mathbf{C}_i^{\boldsymbol{y}}\right)\left(\mathbf{F} + \tilde{\mathbf{F}}\right). \tag{31}$$

Again, we refer to [9] for the details of the derivation.

To derive a batch update based on the update formulæ of the information filter, we identify the linearized measurement model as

$$\boldsymbol{Y}_i \approx \bar{\boldsymbol{Y}}_i + \mathbf{M}_{i-1}(\boldsymbol{p} - \bar{\boldsymbol{p}}_{i-1}) + \boldsymbol{t}_i, \tag{32}$$

where $\boldsymbol{t}_i \sim \mathcal{N}\left(0, \mathbf{C}_i^{\boldsymbol{Y}} - \mathbf{M}_{i-1}^{\mathrm{T}} \mathbf{C}_{i-1}^{\boldsymbol{p}} \mathbf{M}_{i-1}\right) \equiv \mathcal{N}\left(0, \mathbf{C}_i^{\boldsymbol{t}}\right)$. Verification of correctness is again as straightforward as checking that the resulting Kalman update for this model exactly matches Eqs. (27-28). Similar to the kinematic case, additional

approximations are necessary to enable a batch update. We approximate $\mathbf{M}_i \approx \mathbf{M}_0$, $\mathbf{C}_i^{\boldsymbol{t}} \approx \mathbf{C}_1^{\boldsymbol{t}} \equiv \mathbf{C}^{\boldsymbol{t}}$, $\bar{\boldsymbol{p}}_{i-1} \approx \bar{\boldsymbol{p}}_0$ and

$$\mathbf{C}_L^{\boldsymbol{y}} \approx \mathbf{H}\mathbf{C}_L^{\boldsymbol{r}}\mathbf{H}^{\mathrm{T}} + \mathbf{C}_1^{\mathrm{I}} + \mathbf{C}_1^{\mathrm{II}} + \mathbf{C}^{\boldsymbol{v}}, \tag{33}$$

such that $\bar{\boldsymbol{Y}}_i = \mathbf{F}\mathrm{vect}\{\mathbf{C}_i^{\boldsymbol{y}}\} \approx \bar{\boldsymbol{Y}}_1 \equiv \bar{\boldsymbol{Y}}$. Note here that $\mathbf{C}_L^{\boldsymbol{r}}$ is already available since we perform the kinematic update first. The measurement model thus simplifies to

$$\boldsymbol{Y}_i \approx \bar{\boldsymbol{Y}} + \mathbf{M}_0(\boldsymbol{p} - \bar{\boldsymbol{p}}_0) + \boldsymbol{t}_i, \tag{34}$$

with $\boldsymbol{t}_i \sim \mathcal{N}\left(0, \mathbf{C}^{\boldsymbol{t}}\right)$. Also we approximate the i-th outcome of the pseudo-measurement as

$$\boldsymbol{Y}_i \approx \mathbf{F}\left(\boldsymbol{y}_i - \bar{\boldsymbol{y}}_L\right) \otimes \left(\boldsymbol{y}_i - \bar{\boldsymbol{y}}_L\right) \tag{35}$$

where $\bar{\boldsymbol{y}}_L$ is again known from the previously-run kinematic update. Iterating the information filter update yields

$$\bar{\boldsymbol{\xi}}_L^{\boldsymbol{p}} = \bar{\boldsymbol{\xi}}_0^{\boldsymbol{p}} + \mathbf{M}_0^{\mathrm{T}}[\mathbf{C}^{\boldsymbol{t}}]^{-1} \sum_{i=1}^{L} \left[\boldsymbol{Y}_i - \bar{\boldsymbol{Y}} + \mathbf{M}_0\bar{\boldsymbol{p}}_0\right], \tag{36}$$

$$[\mathbf{C}_L^{\boldsymbol{p}}]^{-1} = [\mathbf{C}_0^{\boldsymbol{p}}]^{-1} + L\mathbf{M}_0^{\mathrm{T}}[\mathbf{C}^{\boldsymbol{t}}]^{-1}\mathbf{M}_0, \tag{37}$$

$$\bar{\boldsymbol{p}}_L = \mathbf{C}_L^{\boldsymbol{p}}\bar{\boldsymbol{\xi}}_L^{\boldsymbol{p}}, \quad \bar{\boldsymbol{\xi}}_0^{\boldsymbol{p}} = [\mathbf{C}_0^{\boldsymbol{p}}]^{-1}\bar{\boldsymbol{p}}_0. \tag{38}$$

As approximating $\bar{\boldsymbol{y}}_i \approx \bar{\boldsymbol{y}}_L$ in Eq. (35) and $\mathbf{C}_i^{\boldsymbol{r}} \approx \mathbf{C}_L^{\boldsymbol{r}}$ in Eq. (33) is not the only possible choice to avoid a dependency on intermediate results from the kinematic update, we refer to update equations Eqs. (36-38) as MEM-EIF[$\boldsymbol{y}_L$]. Another natural choice is $\bar{\boldsymbol{y}}_i \approx \bar{\boldsymbol{y}}_0, \mathbf{C}_i^{\boldsymbol{r}} \approx \mathbf{C}_0^{\boldsymbol{r}}$ which we refer to as MEM-EIF[$\boldsymbol{y}_0$]. In this case, the proposed MEM-EIF[$\boldsymbol{y}_0$] is equivalent to the MEM-EKF* if we avoid the approximations by performing a sequential information update, i.e., one information update per measurement. This is further discussed in the numerics part Sec. IV.

D. Discussion of the approximations

1) Overview: As we have seen, additional approximations on top of the original update equations from [9] are required to allow for a batch update. In case of the kinematic update, these are quite intuitive: As the shape enters the update only via a renormalization of the measurement noise $\mathbf{R}_{i,\boldsymbol{r}}$, approximating $\mathbf{R}_{i,\boldsymbol{r}} \approx \mathbf{R}_{0,\boldsymbol{r}}$ should not impact the quality of the final estimate $\bar{\boldsymbol{r}}_L$ too badly as long as the initial shape estimate is reasonably close to the true value. Consequently, approximating the expected measurement $\bar{\boldsymbol{y}}_i \approx \bar{\boldsymbol{y}}_L = \mathbf{H}\bar{\boldsymbol{r}}_L$ and using $\mathbf{C}_i^{\boldsymbol{r}} \approx \mathbf{C}_L^{\boldsymbol{r}}$ for the calculation of $\mathbf{C}_i^{\boldsymbol{y}}$ (cf. Eq. (33)) within the shape update seems quite reasonable. After all, $\bar{\boldsymbol{r}}_L$ should be the more precise estimate as is reflected in its lower covariance $\mathbf{C}_L^{\boldsymbol{r}}$ after L updates.

However, regarding the remaining approximations – $\mathbf{M}_i \approx \mathbf{M}_0$, $\mathbf{C}_i^{\boldsymbol{t}} \approx \mathbf{C}_1^{\boldsymbol{t}}$, $\bar{\boldsymbol{p}}_{i-1} \approx \bar{\boldsymbol{p}}_0$, $\mathbf{C}_i^{\mathrm{I}} \approx \mathbf{C}_1^{\mathrm{I}}$ and $\mathbf{C}_i^{\mathrm{II}} \approx \mathbf{C}_1^{\mathrm{II}}$ – things are less obvious. Assuming the validity of the just mentioned approximation regarding the expected measurement and its covariance, we justify the remaining approximations via an order analysis. The idea is to apply the information update before applying any approximations, i.e., directly to the model (32). By construction, the resulting information update is then equivalent to the original Kalman update Eqs. (27-28). We proceed to show that by neglecting higher order corrections in the shape, the batch variant of the information update as stated in Eqs. (36-38) is obtained. This proves that in lowest order the batch variant of the shape update is equivalent to its Kalman counterpart.

2) Order analysis: Consider the first two information updates based on the model Eq. (32) where it already shows that the i-dependence of $\bar{\boldsymbol{Y}}_i$, $\mathbf{M}_{i-1}$, $\bar{\boldsymbol{p}}_{i-1}$ and $\boldsymbol{t}_i$ seems to hinder the derivation of a batch update:

$$\begin{aligned}\bar{\boldsymbol{\xi}}_2^{\boldsymbol{p}} = {} & [\mathbf{C}_0^{\boldsymbol{p}}]^{-1}\bar{\boldsymbol{p}}_0 + \mathbf{M}_0^{\mathrm{T}}[\mathbf{C}_1^{\boldsymbol{t}}]^{-1}[\boldsymbol{Y}_1 - \bar{\boldsymbol{Y}}_1 + \mathbf{M}_0\bar{\boldsymbol{p}}_0] \\ & + \mathbf{M}_1^{\mathrm{T}}[\mathbf{C}_2^{\boldsymbol{t}}]^{-1}[\boldsymbol{Y}_2 - \bar{\boldsymbol{Y}}_2 + \mathbf{M}_1\bar{\boldsymbol{p}}_1].\end{aligned} \tag{39}$$

If our prior is informative enough, i.e., the covariance $\mathbf{C}_0^{\boldsymbol{p}}$ is small and $\bar{\boldsymbol{p}}_0$ is close to the true state, we can expand as follows

$$\mathbf{M}_1 = \mathbf{M}(\bar{\boldsymbol{p}}_1) \approx \mathbf{M}(\bar{\boldsymbol{p}}_0) + \Delta\mathbf{M}, \tag{40}$$

where we defined (with $(\boldsymbol{u})_k = u_k$ being the k-th component of the vector $\boldsymbol{u} = (u_1, u_2, u_3)^{\mathrm{T}}$)

$$[\Delta\mathbf{M}]_{mn} \equiv \sum_{k=1}^{3} \left.\frac{\partial[\mathbf{M}(\boldsymbol{p})]_{ij}}{\partial p_k}\right|_{\boldsymbol{p}=\bar{\boldsymbol{p}}_0} (\bar{\boldsymbol{p}}_1 - \bar{\boldsymbol{p}}_0)_k. \tag{41}$$

Assuming $\mathbf{C}_0^{\boldsymbol{p}}$ is small enough, we have that (with $|\cdot|$ denoting the absolute value)

$$|([\mathbf{C}_0^{\boldsymbol{p}}]^{-1}\boldsymbol{p})_k| \gg |(\mathbf{M}_0[\mathbf{C}_2^{\boldsymbol{t}}]^{-1}[\boldsymbol{Y}_2 - \bar{\boldsymbol{Y}}_2 + \mathbf{M}_1\bar{\boldsymbol{p}}_1])_k| \tag{42}$$

$$\gg |(\Delta\mathbf{M}[\mathbf{C}_2^{\boldsymbol{t}}]^{-1}[\boldsymbol{Y}_2 - \bar{\boldsymbol{Y}}_2 + \mathbf{M}_1\bar{\boldsymbol{p}}_1])_k|. \tag{43}$$

The first inequality above is large because the eigenvalues of $[\mathbf{C}_0^{\boldsymbol{p}}]^{-1}$ tend to infinity in the limit $\mathbf{C}_0^{\boldsymbol{p}} \to 0$, while for the second we have $\bar{\boldsymbol{p}}_1 - \bar{\boldsymbol{p}}_0 \to 0$ in this limit. Keeping the lowest order corrections only gives us

$$\begin{aligned}\bar{\boldsymbol{\xi}}_2^{\boldsymbol{p}} \approx {} & [\mathbf{C}_0^{\boldsymbol{p}}]^{-1}\bar{\boldsymbol{p}}_0 + \mathbf{M}_0^{\mathrm{T}}[\mathbf{C}_1^{\boldsymbol{t}}]^{-1}[\boldsymbol{Y}_1 - \bar{\boldsymbol{Y}}_1 + \mathbf{M}_0\bar{\boldsymbol{p}}_0] \\ & + \mathbf{M}_0^{\mathrm{T}}[\mathbf{C}_2^{\boldsymbol{t}}]^{-1}[\boldsymbol{Y}_2 - \bar{\boldsymbol{Y}}_2 + \mathbf{M}_1\bar{\boldsymbol{p}}_1].\end{aligned} \tag{44}$$

Furthermore

$$\mathbf{M}_1\bar{\boldsymbol{p}}_1 \approx (\mathbf{M}_0 + \Delta\mathbf{M})(\bar{\boldsymbol{p}}_0 + [\bar{\boldsymbol{p}}_1 - \bar{\boldsymbol{p}}_0]) \tag{45}$$

such that the same idea can be applied again to get

$$\begin{aligned}\bar{\boldsymbol{\xi}}^{\boldsymbol{p}_2} \approx {} & [\mathbf{C}_0^{\boldsymbol{p}}]^{-1}\bar{\boldsymbol{p}}_0 + \mathbf{M}_0^{\mathrm{T}}[\mathbf{C}_1^{\boldsymbol{t}}]^{-1}[\boldsymbol{Y}_1 - \bar{\boldsymbol{Y}}_1 + \mathbf{M}_0\bar{\boldsymbol{p}}_0] \\ & + \mathbf{M}_0^{\mathrm{T}}[\mathbf{C}_2^{\boldsymbol{t}}]^{-1}[\boldsymbol{Y}_2 - \bar{\boldsymbol{Y}}_2 + \mathbf{M}_0\bar{\boldsymbol{p}}_0].\end{aligned} \tag{46}$$

On a meta level, the important insight is that terms proportional to $[\mathbf{C}_0^{\boldsymbol{p}}]^{-1}$ are large, and therefore we can drop any corrections proportional to $\bar{\boldsymbol{p}}_1 - \bar{\boldsymbol{p}}_0$ or also $\mathbf{C}_i^{\boldsymbol{p}}$ itself as the leading order corrections are found in zeroth-order. The other problematic quantities, $\bar{\boldsymbol{Y}}_i$ and $\mathbf{C}_2^{\boldsymbol{t}}$, are a bit more tedious but otherwise straightforward as well. We have

$$\bar{\boldsymbol{Y}}_2 = \mathbf{F}\mathrm{vect}\{\mathbf{C}_2^{\boldsymbol{y}}\}, \quad \mathbf{C}_2^{\boldsymbol{t}} = \mathbf{C}_2^{\boldsymbol{Y}} - \mathbf{M}_1^{\mathrm{T}}\mathbf{C}_1^{\boldsymbol{p}}\mathbf{M}_1 \tag{47}$$

where

$$\mathbf{C}_2^{\boldsymbol{Y}} = \mathbf{F}(\mathbf{C}_2^{\boldsymbol{y}} \otimes \mathbf{C}_2^{\boldsymbol{y}})(\mathbf{F} + \tilde{\mathbf{F}}), \tag{48}$$

$$\mathbf{C}_2^{\boldsymbol{y}} \approx \mathbf{H}\mathbf{C}_L^{\boldsymbol{r}}\mathbf{H}^{\mathrm{T}} + \mathbf{C}_2^{\mathrm{I}} + \mathbf{C}_2^{\mathrm{II}} + \mathbf{C}^{\boldsymbol{v}}. \tag{49}$$

The quantity $\mathbf{C}_L^{\boldsymbol{r}}$ is already known from the kinematic update which is run first. We approximate

$$\mathbf{C}_2^{\mathrm{I}} \approx \mathbf{S}(\bar{\boldsymbol{p}}_0)\mathbf{C}^{\boldsymbol{h}}\mathbf{S}(\bar{\boldsymbol{p}}_0)^{\mathrm{T}} + \Delta\mathbf{S}\mathbf{C}^{\boldsymbol{h}}\mathbf{S}(\bar{\boldsymbol{p}}_0)^{\mathrm{T}} + \mathbf{S}(\bar{\boldsymbol{p}}_0)\mathbf{C}^{\boldsymbol{h}}\Delta\mathbf{S}^{\mathrm{T}} \tag{50}$$

$$\approx \mathbf{S}(\bar{\boldsymbol{p}}_0)\mathbf{C}^{\boldsymbol{h}}\mathbf{S}(\bar{\boldsymbol{p}}_0)^{\mathrm{T}} = \mathbf{C}_1^{\mathrm{I}}, \tag{51}$$

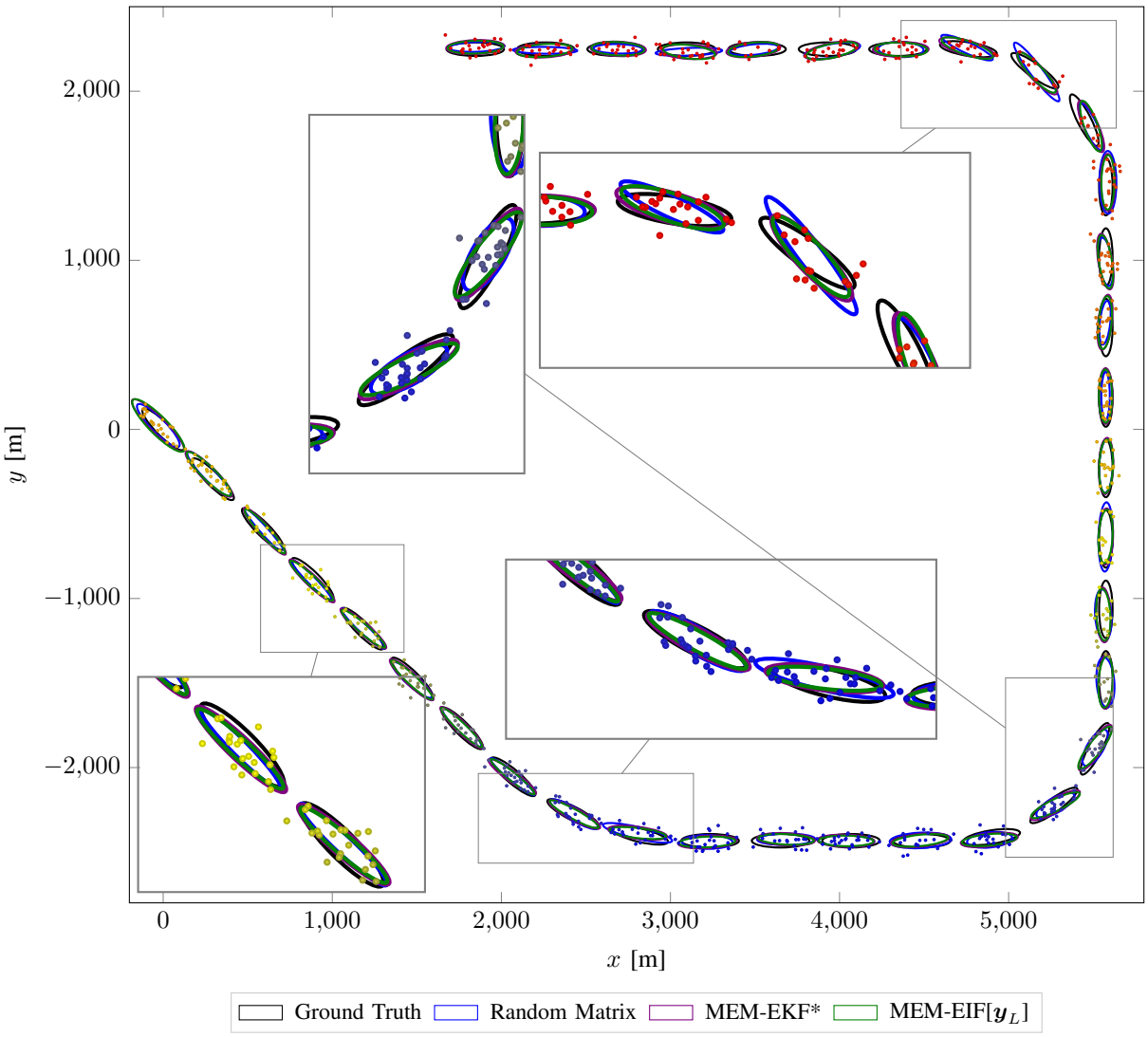

Fig. 2. This figure shows one example run with three trackers: random matrix (blue), MEM-EKF* (red) and MEM-EIF (green). The ground truth object is black ellipses and measurements are colored dots.

with

$$[\Delta\mathbf{S}(\bar{\boldsymbol{p}}_1)]_{mn} \equiv \sum_{k=1}^{3} \frac{\partial[\mathbf{S}(\boldsymbol{p})]_{mn}}{\partial p_k}\bigg|_{\boldsymbol{p}=\bar{\boldsymbol{p}}_0} (\bar{\boldsymbol{p}}_1 - \bar{\boldsymbol{p}}_0)_k, \quad (52)$$

by keeping only zeroth-order terms as discussed above. The approximation $\mathbf{C}_2^{\mathrm{II}} \approx \mathbf{C}_1^{\mathrm{II}}$ is even more straightforward, as $\mathbf{C}_2^{\mathrm{II}}$ is directly proportional to $\mathbf{C}_1^{\boldsymbol{p}}$ and thus a negligible term. Therefore

$$\mathbf{C}_2^{\boldsymbol{y}} \approx \mathbf{H}\mathbf{C}_L^{\boldsymbol{r}}\mathbf{H}^{\mathrm{T}} + \mathbf{C}_1^{\mathrm{I}} + \mathbf{C}_1^{\mathrm{II}} + \mathbf{C}^{\boldsymbol{v}} = \mathbf{C}_1^{\boldsymbol{y}}, \quad (53)$$

such that $\bar{\boldsymbol{Y}}_2 \approx \bar{\boldsymbol{Y}}_1 \equiv \bar{\boldsymbol{Y}}$ and

$$\mathbf{C}_2^{\boldsymbol{t}} = \mathbf{C}_2^{\boldsymbol{Y}} - \mathbf{M}_1^{\mathrm{T}}\mathbf{C}_1^{\boldsymbol{p}}\mathbf{M}_1 \approx \mathbf{C}_1^{\boldsymbol{Y}} - \mathbf{M}_0^{\mathrm{T}}\mathbf{C}_0^{\boldsymbol{p}}\mathbf{M}_0 = \mathbf{C}_1^{\boldsymbol{t}}. \quad (54)$$

Putting it all together we find

$$\bar{\boldsymbol{\xi}}^{\boldsymbol{p}_2} \approx [\mathbf{C}_0^{\boldsymbol{p}}]^{-1}\bar{\boldsymbol{p}}_0 + \sum_{i=1}^{2} \mathbf{M}_0^{\mathrm{T}}[\mathbf{C}_1^{\boldsymbol{t}}]^{-1}[\boldsymbol{Y}_i - \bar{\boldsymbol{Y}} + \mathbf{M}_0\bar{\boldsymbol{p}}_0]. \quad (55)$$

The same analysis applies to all further update steps $i = 3, ..., L$ as well which yields the final result Eqs. (36)-(38) by induction.

IV. Numerical Evaluation

In this section, we investigate the performance of the proposed information-filter-based batch update for the multiplicative error model (MEM-EIF). After some remarks regarding tuning and initialization, we compare the proposed MEM-EIF with the sequential update tracker MEM-EKF*, and another batch update algorithm: Random Matrix (RM) [8]. Next, we investigate the effect of different approximation variations in MEM-EIF as explained in Section III-C. Finally, we highlight the benefit of a batch update by comparing the runtime of MEM-EIF and its reference algorithm, MEM-EKF*.

A. Tuning and Initialization

While tuning the MEM-EIF (and to some lesser extent the MEM-EKF*) it is important to recall that the original derivation (cf. [9]) as well as our additional approximations (cf. Sec. III-D) rely on a Taylor expansion around the shape estimate. It is therefore mandatory to restrict the shape covariance to reasonable small values. In addition, an

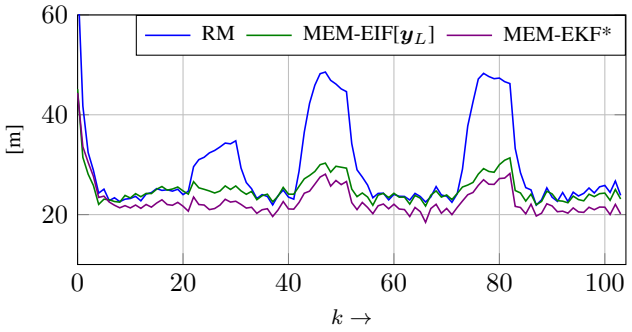

Fig. 3. This figure depicts the average error for 100 simulated runs using the simulation shown in Fig. 2.

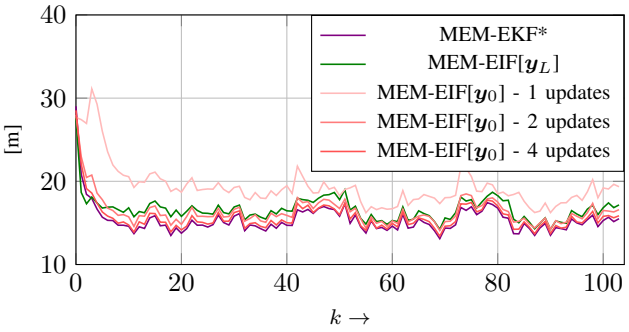

Fig. 4. Estimation Error of the MEM-EIF[$\bar{\boldsymbol{y}}_0$] for different batch sizes (U updates is equivalent to a batch size of L/U) versus MEM-EKF* and MEM-EIF[$\boldsymbol{y}_L$]. We used a Poisson rate $\lambda = 50$ for the simulation, otherwise the simulation is the same as in Fig. 2. The results are averaged over 100 runs.

unreasonable large covariance for the shape parameters would put a significant amount of weight toward the unphysical negative values of length and width. As a rule of thumb for practical applications, we therefore recommend keeping the shape covariance below $\mathbf{C}^{\boldsymbol{l}_1} \leq (0.4 \times \bar{l}_1)^2, \mathbf{C}^{\boldsymbol{l}_2} \leq (0.4 \times \bar{l}_2)^2$ based on the current estimates for length and width $\bar{l}_1, \bar{l}_2$.

For the numerical experiments detailed below, the tracker is initialized by calculating sample mean and covariance of the measurements at time $k = 0$. The sample mean serves as the initial position estimate while the shape estimate is obtained as the 95% confidence ellipse corresponding to the sample covariance. The initial estimates for velocity and acceleration are set to zero.

B. Performance along reference trajectory

In this simulation, we consider a single extended target that moves in the 2d plane along a reference trajectory and generates a random number of measurements at every time step according to a Poisson distribution. The simulated elliptical extended object has diameters of 340 and 80 meters and moves with a constant velocity of 50km/h. The measurements are Poisson distributed with a Poisson rate $\lambda = 20$ which is the same as in [9] and [8]. We compare our proposed MEM-EIF with the reference MEM-EKF* and the random matrix method [5], [8]. As it features the better performance-cost ratio among the two variants, we picked the MEM-EIF[$\boldsymbol{y}_L$] variant over the MEM-EIF[$\boldsymbol{y}_0$] (see IV-D for a comparison of the two variants). We use a constant-acceleration white-noise jerk model [15] for the prediction between the time steps. The simulated ground truth objects, trajectory, measurements, and estimates of three trackers from one example run are depicted in Fig. 2. Consistent with the observations in [9], we find an increase in error of the random random matrix approach around the three coordinated turns. The trackers based on the multiplicative-error measurement model, on the other hand, can constrain the size change while allowing for the necessary flexibility in the orientation through the process noise.

We use the Gaussian Wasserstein distance [16] as the metric for evaluating the estimation error of elliptical shapes. The Gaussian Wasserstein distance incorporates the estimation error on both shape and location, can be calculated in closed-form, and is a true metric. As the center $\mathbf{H}\boldsymbol{r}$ and shape matrix $\boldsymbol{\Sigma} = \mathbf{S}\mathbf{S}^{\mathrm{T}}$ of an ellipse can be interpreted as the mean and covariance matrix of a Gaussian distribution, the Wasserstein distance between two elliptical objects is

$$d^2_{GW}(\boldsymbol{x}_1, \boldsymbol{x}_2) = ||\mathbf{H}\boldsymbol{r}_1 - \mathbf{H}\boldsymbol{r}_2||^2 + \operatorname{Tr}\left(\boldsymbol{\Sigma}_1 + \boldsymbol{\Sigma}_2 - 2\sqrt{\sqrt{\boldsymbol{\Sigma}_1}\boldsymbol{\Sigma}_2\sqrt{\boldsymbol{\Sigma}_1}}\right). \quad (56)$$

As shown in Fig. 3, among the batch updates the MEM-EIF[$\boldsymbol{y}_L$] outperforms the random matrix approach at the coordinated turns while showing a similar error otherwise. Compared to its reference MEM-EKF*, we see an increase in error of roughly ten percent due to the additional approximations.

C. Runtime performance

The main motivation for employing the batch update, i.e., MEM-EIF[$\boldsymbol{y}_L$], instead of the sequential update is an improved runtime performance. Therefore, we compare the scaling of our Python implementation of the batch update with the sequential MEM-EKF* updates. The result is shown in Fig. 5. As we can see, both update schemes scale linearly in the number of measurements L, but the scaling constant of the batch update is roughly a hundred times smaller than that of the batch update. The big performance win for the batch update stems from eliminating the explicit main loop of the sequential update which performs one Kalman update of kinematics and shape per measurement (Eqs. (14-15,27-28) above; see TABLE I in [9] for pseudo code). This loop in the sequential update cannot be easily vectorized due to the dependence of the n-th update on the $(n-1)$-th update and causes a huge performance penalty, while the batch update can profit from an implicit loop via vectorized summation over all measurements and pseudo-measurements. In principle, even a parallelization of this summation (e.g., on a GPU) would be possible here. Despite there being a difference in scaling constant only, we therefore expect a similarly significant speedup in general (i.e., independent of our particular Python implementation) if one chooses to switch from a sequential to a batch update. Finally, while we have not measured explicit performance metrics for our implementation of the random matrix approach, it should sit close the MEM-EIF[$\boldsymbol{y}_L$] as it

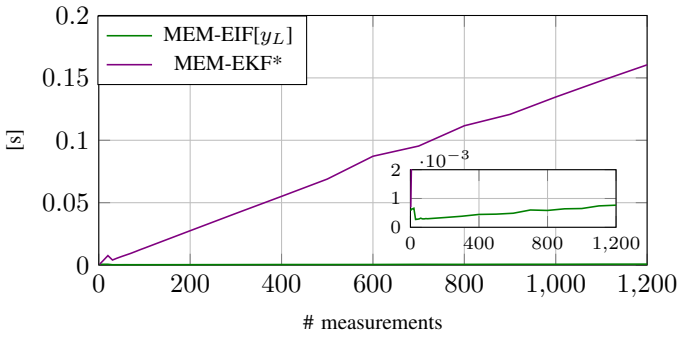

Fig. 5. Comparison of the runtime performance of the sequential MEM-EKF* (purple line) and the batch MEM-EIF[$\boldsymbol{y}_L$] (green line) update based on our python implementation. The number of seconds necessary to perform the update (y-axis) is plotted against the number of measurements which are part of the update (x-axis). Due to the large difference in the scaling constant, the green line representing the scaling of the MEM-EIF[$\boldsymbol{y}_L$] update is hugging the x-axis. In the inset, the same data is shown such that the linear scaling of the MEM-EIF[$\boldsymbol{y}_L$] update in the number of measurements becomes visible as well.

is also a batch update, i.e., it works based on an implicit vectorized loop over the measurements.

D. Convergence of MEM-EIF[$\boldsymbol{y}_0$] against MEM-EKF*

As explained in III-C, approximating $\bar{\boldsymbol{y}}_i \approx \bar{\boldsymbol{y}}_L, \mathbf{C}_i^{\boldsymbol{r}} \approx \mathbf{C}_L^{\boldsymbol{r}}$ in Eq. (33) is not the only natural choice. An interesting alternative is given by the approximation $\bar{\boldsymbol{y}}_i \approx \bar{\boldsymbol{y}}_0, \mathbf{C}_i^{\boldsymbol{r}} \approx \mathbf{C}_0^{\boldsymbol{r}}$ (referred to as MEM-EIF[$\boldsymbol{y}_0$]) since it reduces to the MEM-EKF* if we perform one information update per measurement. This interesting property suggests the batch size, which we recall to be L (the number of measurements) for the MEM-EIF[$\boldsymbol{y}_L$], as a tuning parameter of the MEM-EIF[$\boldsymbol{y}_0$] approximation. It allows for trading accuracy versus performance, with the maximal accuracy given for batch size equal to one while the maximal performance is obtained for batch size equal to L. In Fig. 4 we compare the estimation error of the MEM-EIF[$\boldsymbol{y}_0$] with different batch sizes against the MEM-EKF* and the MEM-EIF[$\boldsymbol{y}_L$]. As is apparent from the plot, the quality of the MEM-EIF[$\boldsymbol{y}_0$] is significantly inferior if only a single update step is performed, i.e., if the batch size is taken to be L. However, by increasing the batch size to $L/2$ (implying two update steps) and $L/4$ (implying four update steps) we see a quick improvement in the quality and, as to be expected, convergence against the MEM-EKF*. The drawback is of course an increase in runtime and therefore it needs to be decided per application if the increase in quality is worth the cost. In general, we recommend the MEM-EIF[$\boldsymbol{y}_L$] over the MEM-EIF[$\boldsymbol{y}_0$]. In this regard, it is worthwhile to note that varying the batch size of the MEM-EIF[$\boldsymbol{y}_L$] did not show any quality improvements in our experiments. This is due to the fact that there is no limit in which the MEM-EIF[$\boldsymbol{y}_L$] becomes equivalent to the MEM-EKF*. Therefore, the MEM-EIF[$\boldsymbol{y}_L$] should always be run with batch size equal to L.

V. Remarks & Conclusion

The need for an elliptical-target tracker typically arises in the context of a larger system which also features a computationally demanding association step. One common way to reduce complexity is to work with the marginal association probabilities for each track which in the context of point-target tracking is often combined with the probability density approximation (PDA) [17]. Two extensions of this idea to extended target tracking that were recently proposed in [18], [19] involved a sequential PDA update based on the MEM-EKF*. Our proposed batch update straightforwardly generalizes to this setting as the summation over measurements (Eq. (22)) and pseudo-measurements (Eq. (36)) turn into weighted summations – the weights being the association probabilities of each measurement. This way, the sequential PDA update can be replaced by a weighted batch update – thus offering the performance benefit to this important setting.

Another interesting remark concerns the observation made in [9] that the sequential update of the MEM-EKF* depends on the order of the measurements. For the proposed batch update this is clearly not the case: the sums in Eqs. (22-24,36-38) are invariant under changes to the measurement order. Based on our analysis in Sec. III-D we conclude that the order dependence stems from the higher-order corrections in the shape update, as well as the feedback of the shape update to the kinematics. Giving up on both yields the order-invariant batch update – the price being a slightly increased estimation error and the benefit a significantly reduced runtime.

To conclude this work, we like to highlight again that the newly-introduced tracker MEM-EIF[$\boldsymbol{y}_L$] combines two important strengths of the random matrix and the MEM-EKF* approach: Similar to the random matrix approach it is a batch update, i.e., the target state is updated in a single step using only the sums over all measurements and the MEM-specific pseudo-measurements. Compared to a sequential measurement-by-measurement update, the scaling constant is drastically reduced leading to a strong increase in performance. At the same time, as the tracker is based on the MEM-EKF*, it offers direct access to the tracked ellipse parameters and their covariance thus easing the application of relevant motion models.

Acknowledgments

The authors would like to thank Marcus Baum for helpful suggestions on an early draft of this manuscript. Christian Gramsch thanks Clemens Wuth for pair-debugging sessions and Stephan Timmer for proof reading the manuscript.

Appendix A
Derivation of the Information Filter

For completeness, we give a possible derivation of the information filter in this appendix. We assume a linear measurement model as stated in Eq. (1). Based on the Kalman update in Eqs. (2-3) we define $\bar{\boldsymbol{\xi}}_i = [\mathbf{C}_i^{\boldsymbol{x}}]^{-1}\bar{\boldsymbol{x}}_i$ such that

$$\begin{aligned}
\bar{\boldsymbol{x}}_i &= \bar{\boldsymbol{x}}_{i-1} + \mathbf{C}_i^{\boldsymbol{xy}}[\mathbf{C}_i^{\boldsymbol{y}}]^{-1}(\boldsymbol{y}_i - \mathbf{H}\bar{\boldsymbol{x}}_{i-1}) && (57)\\
&= \mathbf{C}_{i-1}^{\boldsymbol{x}}\bar{\boldsymbol{\xi}}_{i-1} + \mathbf{C}_i^{\boldsymbol{xy}}[\mathbf{C}_i^{\boldsymbol{y}}]^{-1}(\boldsymbol{y}_i - \mathbf{H}\mathbf{C}_{i-1}^{\boldsymbol{x}}\bar{\boldsymbol{\xi}}_{i-1}) && (58)\\
&= \left[\mathbf{C}_{i-1}^{\boldsymbol{x}} - \mathbf{C}_{i-1}^{\boldsymbol{x}}\mathbf{H}^{\mathrm{T}}[\mathbf{C}_{i-1}^{\boldsymbol{y}}]^{-1}\mathbf{H}\mathbf{C}_{i-1}^{\boldsymbol{x}}\right]\bar{\boldsymbol{\xi}}_{i-1} && (59)\\
&\quad + \mathbf{C}_{i-1}^{\boldsymbol{x}}\mathbf{H}^{\mathrm{T}}[\mathbf{C}_{i-1}^{\boldsymbol{y}}]^{-1}\boldsymbol{y}_i\\
&= \mathbf{C}_i^{\boldsymbol{x}}\bar{\boldsymbol{\xi}}_{i-1} + \mathbf{C}_{i-1}^{\boldsymbol{x}}\mathbf{H}^{\mathrm{T}}[\mathbf{C}_{i-1}^{\boldsymbol{y}}]^{-1}\boldsymbol{y}_i && (60)
\end{aligned}$$

which gives the intermediate result

$$\bar{\boldsymbol{\xi}}_i = \bar{\boldsymbol{\xi}}_{i-1} + [\mathbf{C}_i^{\boldsymbol{x}}]^{-1}\mathbf{C}_{i-1}^{\boldsymbol{x}}\mathbf{H}^{\mathrm{T}}[\mathbf{C}_{i-1}^{\boldsymbol{y}}]^{-1}\boldsymbol{y}_i. \tag{61}$$

From the matrix inversion lemma we have the relation

$$[\mathbf{C}_i^{\boldsymbol{x}}]^{-1} = \left[\mathbf{C}_{i-1}^{\boldsymbol{x}} - \mathbf{C}_{i-1}^{\boldsymbol{x}}\mathbf{H}^{\mathrm{T}}[\mathbf{C}_{i-1}^{\boldsymbol{y}}]^{-1}\mathbf{H}\mathbf{C}_{i-1}^{\boldsymbol{x}}\right]^{-1} \tag{62}$$

$$= [\mathbf{C}_{i-1}^{\boldsymbol{x}}]^{-1} + \mathbf{H}^{\mathrm{T}}[\mathbf{C}^{\boldsymbol{v}}]^{-1}\mathbf{H}, \tag{63}$$

which gives us

$$[\mathbf{C}_i^{\boldsymbol{x}}]^{-1}\mathbf{C}_{i-1}^{\boldsymbol{x}}\mathbf{H}^{\mathbf{T}}[\mathbf{C}_{i-1}^{\boldsymbol{y}}]^{-1}$$

$$= \left[\mathbf{C}_{i-1}^{\boldsymbol{x}} + \mathbf{H}^{\mathrm{T}}[\mathbf{C}^{\boldsymbol{v}}]^{-1}\mathbf{H}\right]\mathbf{C}_{i-1}^{\boldsymbol{x}}\mathbf{H}^{\mathrm{T}}[\mathbf{C}_{i-1}^{\boldsymbol{y}}]^{-1} \tag{64}$$

$$= \left[\mathbf{H}^{\mathrm{T}} + \mathbf{H}^{\mathrm{T}}[\mathbf{C}^{\boldsymbol{y}}]^{-1}\mathbf{H}\mathbf{C}_{i-1}^{\boldsymbol{x}}\mathbf{H}^{\mathrm{T}}\right][\mathbf{C}_{i-1}^{\boldsymbol{y}}]^{-1} \tag{65}$$

$$= \mathbf{H}^{\mathrm{T}}[\mathbf{C}^{\boldsymbol{v}}]^{-1}\underbrace{\left[\mathbf{C}^{\boldsymbol{v}} + \mathbf{H}\mathbf{C}_{i-1}^{\boldsymbol{x}}\mathbf{H}^{\mathrm{T}}\right]}_{=\mathbf{C}_{i-1}^{\boldsymbol{y}}}[\mathbf{C}_{i-1}^{\boldsymbol{y}}]^{-1} \tag{66}$$

$$= \mathbf{H}^{\mathrm{T}}[\mathbf{C}^{\boldsymbol{v}}]^{-1} \tag{67}$$

The information filter update is thus found to be

$$\bar{\boldsymbol{\xi}}_i = \bar{\boldsymbol{\xi}}_{i-1} + \mathbf{H}^{\mathrm{T}}[\mathbf{C}^{\boldsymbol{v}}]^{-1}\boldsymbol{y}_i, \tag{68}$$

$$[\mathbf{C}_i^{\boldsymbol{x}}]^{-1} = [\mathbf{C}_{i-1}^{\boldsymbol{x}}]^{-1} + \mathbf{H}^{\mathrm{T}}[\mathbf{C}^{\boldsymbol{v}}]^{-1}\mathbf{H}. \tag{69}$$

TABLE I
MEASUREMENT UPDATE OF THE MEM-EIF[$\boldsymbol{y}_L$] ALGORITHM

Input: Measurement outcomes $\{\boldsymbol{y}_1, \ldots, \boldsymbol{y}_L\}$, predicted estimates $\bar{\boldsymbol{r}}_0$, $\bar{\boldsymbol{p}}_0$, $\mathbf{C}_0^{\boldsymbol{r}}$, $\mathbf{C}_0^{\boldsymbol{p}}$, measurement noise covariance $\mathbf{C}^{\boldsymbol{v}}$, $\mathbf{H}$ as defined in (8), multiplicative noise covariance $\mathbf{C}^{\boldsymbol{h}}$
Output: updated estimates $\bar{\boldsymbol{r}}_L$, $\bar{\boldsymbol{p}}_L$ and $\mathbf{C}_L^{\boldsymbol{r}}$, $\mathbf{C}_L^{\boldsymbol{p}}$

Helper quantities:

$$\begin{bmatrix}\alpha & l_1 & l_2\end{bmatrix}^{\mathrm{T}} = \bar{\boldsymbol{p}}_0$$

$$\mathbf{S} = \begin{bmatrix}\mathbf{S_1}\\ \mathbf{S_2}\end{bmatrix} = \begin{bmatrix}\cos\alpha & -\sin\alpha\\ \sin\alpha & \cos\alpha\end{bmatrix}\begin{bmatrix}l_1 & 0\\ 0 & l_2\end{bmatrix}$$

$$\mathbf{J_1} = \begin{bmatrix}-l_1\sin\alpha & \cos\alpha & 0\\ -l_2\cos\alpha & 0 & -\sin\alpha\end{bmatrix}$$

$$\mathbf{J_2} = \begin{bmatrix}l_1\cos\alpha & \sin\alpha & 0\\ -l_2\sin\alpha & 0 & \cos\alpha\end{bmatrix}$$

$$\mathbf{C}^{\mathrm{I}} = \mathbf{S}\mathbf{C}^{\boldsymbol{h}}\mathbf{S}^{\mathrm{T}}$$

$$\mathbf{C}^{\mathrm{II}} = [\epsilon_{mn}] = \mathrm{tr}\left\{\mathbf{C}_0^{\boldsymbol{p}}\mathbf{J}_n^{\mathrm{T}}\mathbf{C}^{\boldsymbol{h}}\mathbf{J}_m\right\} \text{ for } m,n = 1,2$$

$$\mathbf{M} = \begin{bmatrix}2\mathbf{S_1}\mathbf{C}^h\mathbf{J_1}\\ 2\mathbf{S_2}\mathbf{C}^h\mathbf{J_2}\\ \mathbf{S_1}\mathbf{C}^h\mathbf{J_2} + \mathbf{S_2}\mathbf{C}^h\mathbf{J_1}\end{bmatrix}$$

$$\mathbf{F} = \begin{bmatrix}1 & 0 & 0 & 0\\ 0 & 0 & 0 & 1\\ 0 & 1 & 0 & 0\end{bmatrix}, \quad \widetilde{\mathbf{F}} = \begin{bmatrix}1 & 0 & 0 & 0\\ 0 & 0 & 0 & 1\\ 0 & 0 & 1 & 0\end{bmatrix}$$

Kinematic Update:

$$\mathbf{C}^{\boldsymbol{s}} = \mathbf{C}^{\mathrm{I}} + \mathbf{C}^{\mathrm{II}} + \mathbf{C}^{v}$$

$$\mathbf{C}_L^{\boldsymbol{r}} = \left[\mathbf{C}_0^{\boldsymbol{r}} + L\mathbf{H}^{\mathrm{T}}[\mathbf{C}^{\boldsymbol{s}}]^{-1}\mathbf{H}\right]^{-1}$$

$$\bar{\boldsymbol{\xi}}_0^{\boldsymbol{r}} = [\mathbf{C}_0^{\boldsymbol{r}}]^{-1}\bar{\boldsymbol{r}}_0$$

$$\bar{\boldsymbol{\xi}}_L^{\boldsymbol{r}} = \bar{\boldsymbol{\xi}}_0^{\boldsymbol{r}} + \mathbf{H}^{\mathrm{T}}[\mathbf{C}^{\boldsymbol{s}}]^{-1}\sum_{i=1}^{L}\boldsymbol{y}_i$$

$$\bar{\boldsymbol{r}}_L = \mathbf{C}_L^{\boldsymbol{r}}\bar{\boldsymbol{\xi}}_L^{\boldsymbol{r}}$$

Shape Update:

$$\mathbf{C}^{\boldsymbol{y}} = \mathbf{H}\mathbf{C}_L^{\boldsymbol{r}}\mathbf{H}^{\mathrm{T}} + \mathbf{C}^{\mathrm{I}} + \mathbf{C}^{\mathrm{II}} + \mathbf{C}^{\boldsymbol{v}}$$

$$\mathbf{C}^{\boldsymbol{Y}} = \mathbf{F}(\mathbf{C}^{\boldsymbol{y}} \otimes \mathbf{C}^{\boldsymbol{y}})(\mathbf{F} + \widetilde{\mathbf{F}})^{\mathrm{T}}$$

$$\mathbf{C}^{\boldsymbol{t}} = \mathbf{C}^{\boldsymbol{Y}} - \mathbf{M}^{\mathrm{T}}\mathbf{C}_0^{\boldsymbol{p}}\mathbf{M}$$

$$\mathbf{C}_L^{\boldsymbol{p}} = \left[\mathbf{C}_0^{\boldsymbol{p}} + L\mathbf{M}^{\mathrm{T}}[\mathbf{C}^{\boldsymbol{t}}]^{-1}\mathbf{M}\right]^{-1}$$

$$\boldsymbol{Y}_i = \mathbf{F}\left((\boldsymbol{y}_i - \mathbf{H}\bar{\boldsymbol{r}}_L) \otimes (\boldsymbol{y}_i - \mathbf{H}\bar{\boldsymbol{r}}_L)\right)$$

$$\bar{\boldsymbol{Y}} = \mathbf{F}\mathrm{vect}\left\{\mathbf{C}^{\boldsymbol{y}}\right\}$$

$$\bar{\boldsymbol{\xi}}_0^{\boldsymbol{p}} = [\mathbf{C}_0^{\boldsymbol{p}}]^{-1}\bar{\boldsymbol{p}}_0$$

$$\bar{\boldsymbol{\xi}}_L^{\boldsymbol{p}} = \bar{\boldsymbol{\xi}}_0^{\bar{\boldsymbol{p}}} + \mathbf{M}^{\mathrm{T}}[\mathbf{C}^{\boldsymbol{t}}]^{-1}\left(\sum_{i=1}^{L}\boldsymbol{Y}_i - L\bar{\boldsymbol{Y}} + L\mathbf{M}\boldsymbol{p}_0\right)$$

$$\bar{\boldsymbol{p}}_L = \mathbf{C}_L^{\boldsymbol{p}}\bar{\boldsymbol{\xi}}_L^{\boldsymbol{p}}$$

REFERENCES

[1] K. Granström, M. Baum, and S. Reuter, "Extended Object Tracking: Introduction, Overview, and Applications," *Journal of Advances in Information Fusion*, vol. 12, Dec. 2017.

[2] J. Vermaak, N. Ikoma, and S. Godsill, "Sequential Monte Carlo Framework for Extended Object Tracking," *IEE Proceedings on Radar, Sonar and Navigation*, vol. 152, no. 5, pp. 353–363, October 2005.

[3] K. Gilholm, S. Godsill, S. Maskell, and D. Salmond, "Poisson Models for Extended Target and Group Tracking," in *SPIE: Signal and Data Processing of Small Targets*, 2005.

[4] K. Gilholm and D. Salmond, "Spatial Distribution Model for Tracking Extended Objects," *IEE Proceedings on Radar, Sonar and Navigation*, vol. 152, no. 5, pp. 364–371, Oct. 2005.

[5] J. W. Koch, "Bayesian Approach to Extended Object and Cluster Tracking Using Random Matrices," *IEEE Transactions on Aerospace and Electronic Systems*, vol. 44, no. 3, pp. 1042–1059, Jul. 2008.

[6] K. Granström, C. Lundquist, and U. Orguner, "Tracking Rectangular and Elliptical Extended Targets using Laser Measurements," in *Proceedings of the 14th International Conference on Information Fusion (Fusion 2011)*, Chicago, Illinois, USA, Jul. 2011.

[7] J. Lan and X. Rong Li, "Tracking of Extended Object or Target Group Using Random Matrix - Part I: New Model and Approach," in *Proceedings of the 15th International Conference on Information Fusion (Fusion 2012)*, Jul. 2012, pp. 2177–2184.

[8] M. Feldmann, D. Fränken, and W. Koch, "Tracking of Extended Objects and Group Targets Using Random Matrices," *IEEE Transactions on Signal Processing*, vol. 59, no. 4, pp. 1409–1420, Apr. 2011.

[9] S. Yang and M. Baum, "Tracking the Orientation and Axes Lengths of an Elliptical Extended Object," *IEEE Transactions on Signal Processing*, vol. 67, no. 18, pp. 4720–4729, Sep. 2019.

[10] M. Baum, F. Faion, and U. D. Hanebeck, "Modeling the Target Extent with Multiplicative Noise," in *2012 15th International Conference on Information Fusion*, Jul. 2012, pp. 2406–2412.

[11] Z. Li, Y. Liang, L. Xu, and S. Ma, "Distributed Extended Object Tracking Information Filter Over Sensor Networks," Oct. 2022.

[12] S. Thrun, W. Burgard, and D. Fox, *Probabilistic Robotics*, illustrated edition ed. Cambridge, Mass: The MIT Press, Aug. 2005.

[13] A. De Santis, A. Germani, and M. Raimondi, "Optimal Quadratic Filtering of Linear Discrete-Time Non-Gaussian Systems," *IEEE Transactions on Automatic Control*, vol. 40, no. 7, pp. 1274–1278, Jul. 1995.

[14] J. Lan and X. Li, "Nonlinear Estimation by LMMSE-Based Estimation With Optimized Uncorrelated Augmentation," *IEEE Transactions on Signal Processing*, vol. 63, pp. 1–1, Aug. 2015.

[15] X. Rong Li and V. Jilkov, "Survey of Maneuvering Target Tracking. Part I. Dynamic Models," *IEEE Transactions on Aerospace and Electronic Systems*, vol. 39, no. 4, pp. 1333–1364, Oct. 2003.

[16] S. Yang, M. Baum, and K. Granström, "Metrics for Performance Evaluation of Elliptic Extended Object Tracking Methods," in *2016 IEEE International Conference on Multisensor Fusion and Integration for Intelligent Systems (MFI)*, Sep. 2016, pp. 523–528.

[17] Y. Bar-Shalom and L. A. U. E. U. of California, *Multitarget-Multisensor Tracking: Advanced Applications*. Boston: Artech House Publishers, Feb. 1990.

[18] S. Yang, K. Thormann, and M. Baum, "Linear-Time Joint Probabilistic Data Association for Multiple Extended Object Tracking," Jul. 2018, pp. 6–10.

[19] S. Yang, L. M. Wolf, and M. Baum, "Marginal Association Probabilities for Multiple Extended Objects without Enumeration of Measurement Partitions," in *2020 IEEE 23rd International Conference on Information Fusion (FUSION)*, Rustenburg, South Africa, Jul. 2020, pp. 1–8.