Extended Target Tracking With a Lidar Sensor Using Random Matrices and a Virtual Measurement Model

Patrick Hoher, Stefan Wirtensohn, Tim Baur, Johannes Reuter, Felix Govaers, *Senior Member, IEEE*, and Wolfgang Koch, *Fellow, IEEE*

***Abstract*—Random matrices are widely used to estimate the extent of an elliptically contoured object. Usually, it is assumed that the measurements follow a normal distribution, with its standard deviation being proportional to the object's extent. However, the random matrix approach can filter the center of gravity and the covariance matrix of measurements independently of the measurement model. This work considers the whole chain from data acquisition to the linear Kalman Filter with extension estimation as a reference plant. The input is the (unknown) ground truth (position and extent). The output is the filtered center of gravity and the filtered covariance matrix of the measurement distribution. A virtual measurement model emulates the behavior of the reference plant. The input of the virtual measurement model is adapted using the proposed algorithm until the output parameters of the virtual measurement model match the result of the reference plant. After the adaptation, the input to the virtual measurement model is considered an estimation for position and extent. The main contribution of this paper is the reference model concept and an adaptation algorithm to optimize the input of the virtual measurement model.**

***Index Terms*—Extended object tracking, random matrices, lidar, reference model, extension estimation.**

I. Introduction

THE goal of tracking is to determine the spatiotemporal density of an object from the measurement data. A scenario where only one object exists that causes exactly one noisy measurement in each time step is called the standard tracking problem [1]. However, high-resolution sensor technology leads to large objects occupying several sensor grid cells and causing several measurements per time step. This allows for applying Extended Object Tracking (EOT) methods to determine the extension of the object in addition to its kinematic state. A comprehensive overview of EOT (up to 2017) is given in [2]. EOT methods can be classified either by the assumed geometric shape of the object or by how the measurement likelihood is modeled. The most common shapes in 2D-EOT are ellipses [3]–[24], rectangles [17], [25]–[27] and star convex shapes [23], [27]–[29].

In EOT, each target has a finite or infinite number of measurement sources [2, p. 141]. The detection probability p_d describes how likely a measurement source causes a measurement and the measurements are usually affected by noise (e.g. white Gaussian noise). The measurement likelihood $p(\boldsymbol{Z}|\underline{x})$ is used to model this probabilistic relationship and has to consider the number of detections (L) and their spatial distribution across the extent of the target [2, 142–143]. Granström *et al.* describe the three most common ways of modeling the measurement likelihood which can be used to classify EOT methods [2, p. 143–145]:

- set of points on a rigid body (SPRB) [18]
- spatial model [3]–[16], [19], [20], [22]–[29]
- physics-based modeling [17], [21]

For SPRB models, it is assumed that an extended object has L measurement sources or reflection points that can independently cause a measurement with probability p_d [2, p. 143]. A typical approach is to distribute L reflection points on the object's contour [18, p. 2375]. A significant drawback of SPRB is the required data association problem (e.g. via Murty's algorithm [30]) of detections with the L measurement sources or reflection areas [2, p. 143].

A spatial model uses a Poisson distributed number of detections and the detections themselves are spatially distributed across the target [2, p. 144]. An example of a spatial model is the random matrix approach (RM) proposed in [3] and further extended in [4]–[15]. The basic idea of the RM approach is that the measurements are normally distributed across the object's center, with a covariance being proportional to the object's extension [3, p. 1047]. A summary of the RM approach and its underlying models (measurement and dynamic) is given in Section II. Further spatial models are e.g. random hypersurface models [22], [23], where the measurements are distributed uniformly across the extent or contour of the target. The advantage of spatial models compared to SPRB models is that there is no assignment problem of detections to measurement sources [2, p. 144].

Physics-based modeling is done when the model does not fit into an SPRB or a spatial model [2, p. 145]. Using a lidar sensor makes it possible to determine the measurement sources by ray tracing [2, p. 145][17, p. 5]. The result is also a set of points on a rigid body. However, these points have a more profound

Manuscript received June 16, 2021; revised October 13, 2021 and December 10, 2021; accepted December 14, 2021. Date of publication December 23, 2021; date of current version January 7, 2022. The associate editor coordinating the review of this manuscript and approving it for publication was Abla Kammoun. This work was supported by Baden-Württemberg Stiftung gGmbH. *(Corresponding author: Patrick Hoher.)*

Patrick Hoher, Stefan Wirtensohn, Tim Baur, and Johannes Reuter are with the Institute of System Dynamics, University of Applied Sciences, D-78464 Konstanz, Germany (e-mail: phoher@htwg-konstanz.de; stwirten@htwg-konstanz.de; tbaur@htwg-konstanz.de; jreuter@ieee.org).

Felix Govaers and Wolfgang Koch are with the Fraunhofer-FKIE, Sensordata, and Information Fusion, D-53343 Wachtberg, Germany (e-mail: felix.govaers@fkie.fraunhofer.de; wolfgang.koch@fkie.fraunhofer.de).

Digital Object Identifier 10.1109/TSP.2021.3138006

physical background than typical SPRB models. It is easy to derive artificial measurements with ray tracing, but finding a measurement likelihood function in closed form is challenging. The lack of a measurement likelihood function in closed form can make the integration into the Bayesian framework difficult.

In this work, we combine the advantages of physics-based modeling with the benefits of spatial models. An elliptical contour of a 2d-object is assumed. The derived measurement sources are the intersections of the lidar beams with the part of the object's contour facing towards the sensor. The detailed lidar sensor model, including the derivation of a method to calculate the measurement sources, is given in Section III. For the extension estimation, the first step is to use the random matrix approach [3]. Since the basic assumptions for the distribution of the measurements of the random matrix approach do not hold with the physics-based lidar measurement model, it is not possible to estimate position and extension. However, the output of the random matrix approach is the filtered center of gravity and the filtered covariance matrix of the measurements independently of the measurement model.

The whole processing chain from data acquisition to the random matrix approach is denoted as a reference plant. Parallel to the reference plant, a virtual measurement model (VMM) is used. The VMM is a function to generate a large number of artificial measurements. It is physics-based and the measurements are generated using ray tracing. The inputs of the VMM are estimated parameters. The underlying goal is to achieve that mean and covariance of the artificial measurements are equal to the result of the random matrix approach. If these match, the input parameters of the VMM should also match the actual real parameters. This approach will be elaborated in more detail in the further course of this paper (see Fig. 3). An adaptation algorithm is presented in this paper to find the input of the virtual measurement model to achieve this goal. This paper is organized as follows: Section II summarizes the random matrix approach [3] and the underlying dynamic and measurement models. Section III presents the physics-based lidar sensor model with ray tracing. The main contribution of this paper is presented in Section IV and Section V. In Section IV, the basic idea of the reference model concept is given, including an introductory example. Section V explains the adaptation algorithm to find the optimal input of the virtual measurement model. Finally, the whole approach presented in this paper is evaluated in Section VI: In the first evaluation part, the tracking results of a simulation study (Monte Carlo) are shown. Then the results are compared to an extension estimation based on Gaussian Processes. Finally, the procedure is evaluated using real lidar measurements recorded with a Velodyne VLP-16 (Puck) in a vessel tracking application.

II. Background

A. Object Dynamic Model

In this paper, we are using Singer's Object Dynamic Model (ODM, initially presented in [31, p. 475]), which is also used in the Random Matrix approach for Extended Object Tracking [3]. The Singer's Model has a state vector $\underline{x} = (x, v, a)$ consisting of position, speed and acceleration for each dimension. The transition matrix F_k for one dimension is given by [3, p. 1045]

$$F_k = \begin{pmatrix} 1 & \Delta_t & \frac{1}{2}\Delta_t^2 \\ 0 & 1 & \Delta_t \\ 0 & 0 & e^{-\Delta_t/\theta} \end{pmatrix} \tag{1}$$

with sampling period Δ_t and time constant θ. Equation (1) can be expanded into the multiple dimensions d transition matrix Φ_k using the Kronecker product $\otimes$ with an identity matrix $I_d \in \mathbb{R}^{d\times d}$ [3, p. 1045]:

$$\Phi_k = F_k \otimes I_d. \tag{2}$$

The model can be interpreted as a constant velocity (CV) model. Suppose there is an initial acceleration or an acceleration due to process noise. In that case, the acceleration decreases depending on the time constant θ, ensuring the velocity remains bounded.

B. Measurement Model and Measurement Likelihood Function

At each time step, a set $\boldsymbol{Z}_k$ containing n_k measurements is received. A measurement of the object centroid can be generated using the measurement matrix H_k:

$$\underline{z}_k = (H_k \otimes I_d)\underline{x}_k \tag{3}$$

In a 2-dimensional example, a measurement matrix $H_k = (1, 0, 0)$ is used to measure the position in each dimension (x, y) [3, p. 1047]. However, the challenging part is to deal with the extension. As a preliminary for the random matrix approach, it is assumed that the measurements are normally distributed [3, p. 1047]. Usually, measurements are also affected by measurement noise. We assume that the measurement noise is normally distributed with covariance $R_k = diag((\sigma_x^2, \sigma_x^2))$ and zero mean. The likelihood is given by [4, p. 6][5, p. 1031]:

$$p(\boldsymbol{Z}_k|n_k, \underline{x}_k, X_k) = \prod_{j=1}^{n_k} \mathcal{N}(\underline{z}_k^j; (H_k \otimes I_d)\underline{x}_k, X_k + R_k). \tag{4}$$

In this paper, it should also be possible to consider measurement models for which the likelihood can not be represented in closed form. In general, the mean of the measurements (center of gravity) is not necessarily the object's centroid. For example, measurements from a lidar sensor appearing at the object's front side will always show a shift in the center of gravity away from the object's centroid towards the sensor's location. The measurements are generated using a nonlinear function $h(\underline{x}_k, X_k)$ instead of the measurement matrix. The measurements $\underline{z}_k^1, \ldots, \underline{z}_k^{n_k}$ are correlated to each other. However, the measurement noise is white Gaussian with covariance R_k and affects each measurement independently. For arbitrary measurement distributions, it is possible to use the measurement matrix H_k in a recursive filter, even if the measurements are generated by $h(\underline{x}_k, X_k)$.

Note that in this case, the filter will track the measurement's center of gravity and not the object's centroid and that it will track the covariance of the measurements and not the extent. In the

further course of this paper, the lidar sensor model is presented as an example for arbitrary measurement distributions.

C. Random Matrix Approach

The random matrix (RM) approach is a Bayesian recursion to estimate the joint density $p(\underline{x}_k, X_k|\boldsymbol{Z}^k)$ and was initially introduced in [3]. However, assumptions are made to estimate kinematics and extension separately [3, p. 1046]. The density of the kinematical state is Gaussian:

$$p(\underline{x}_k|\boldsymbol{Z}^k) = \mathcal{N}(\underline{x}_k; \underline{x}_{k|k}, P_{k|k}), \tag{5}$$

and the density of the extension is given by an inverted Wishart density described by the parameters $X_{k|k} > 0$ and $\nu_{k|k} > (2d + 2)$:

$$p(X_k|\boldsymbol{Z}^k) = \mathcal{IW}(X_k; \nu_{k|k}, X_{k|k}). \tag{6}$$

In a 2-dimensional scenario, the expected value for the extension matrix is given by:

$$\mathbb{E}(X_{k|k}) = \frac{X_{k|k}}{\nu_{k|k} - 6}, \tag{7}$$

with $X_{k|k} \in \mathbb{R}^{2\times 2}$. It is further assumed that the square root of the largest eigenvalue of $\mathbb{E}(X_{k|k})$ is proportional to half of the object's length and the square root of the smaller one to its width. The object's orientation $\varphi_{k|k}$ can either be given from the extension matrix or the kinematics.

1) Prediction: For the kinematic part, the common equations of the linear Kalman filter are used, considering the Kronecker product model [3, p. 1046]. The prediction equations for the extent must ensure that the expectation stays constant and are given as follows [3, p. 1046]:

$$\nu_{k|k-1} = e^{-\Delta t/\tau}\nu_{k-1|k-1}, \tag{8}$$

$$X_{k|k-1} = \frac{e^{-\Delta t/\tau}\nu_{k-1|k-1} - d - 1}{\nu_{k-1|k-1} - d - 1} X_{k-1|k-1}. \tag{9}$$

To ensure that the expected value (7) is always positive definite, the advanced extent prediction for 2D-CV models given in [10, p. 2409] can be used as an alternative:

$$\nu_{k|k-1} = 6 + \frac{\alpha_k}{1 + \alpha_k}(\nu_{k-1|k-1} - 6), \tag{10}$$

$$X_{k|k-1} = \frac{\alpha_k}{1 + \alpha_k} X_{k-1|k-1}. \tag{11}$$

The parameters τ and α_k enable a trade-off between a small stationary error or a dynamic behavior in a turn.

2) Update: The update step for the kinematics follows the familiar equations of the linear Kalman filter. Minor changes are given due to the use of the Kronecker product and to consider the received number of measurements in the current time step. For further calculations, $\bar{\underline{z}}_k$ is the measurement's mean and Z_k is the covariance matrix. The update equations are given by [3, p. 1048]:

$$\underline{x}_{k|k} = \underline{x}_{k|k-1} + (W_{k|k-1} \otimes I_d)(\bar{\underline{z}}_k - (H_k \otimes I_d)\underline{x}_{k|k-1}), \tag{12}$$

$$P_{k|k} = P_{k|k-1} - W_{k|k-1} S_{k|k-1} W_{k|k-1}^T, \tag{13}$$

with innovation factor $S_{k|k-1}$ and Kalman gain $W_{k|k-1}$:

$$S_{k|k-1} = H_k P_{k|k-1} H_k^T + \frac{\sigma_x^2}{n_k}, \tag{14}$$

$$W_{k|k-1} = P_{k|k-1} H_K^T S_{k|k-1}^{-1}, \tag{15}$$

where n_k is the number of measurements and σ_x^2 is the covariance of the measurement noise. Note that measurement noise was not considered in the original approach [3]. Equation (14) has been modified to include measurement noise. To update the parameters of the inverse Wishart density, the following equations are given [3, p. 1048]:

$$X_{k|k} = X_{k|k-1} + N_{k|k-1} + Z_k, \tag{16}$$

$$\nu_{k|k} = \nu_{k|k-1} + n_k. \tag{17}$$

with innovation matrix [3, p. 1048]:

$$N_{k|k-1} = S_{k|k-1}^{-1}(\bar{\underline{z}}_k - (H_k \otimes I_d)\underline{x}_{k|k-1})(\bar{\underline{z}}_k - (H_k \otimes I_d)\underline{x}_{k|k-1})^T. \tag{18}$$

There are extensions of the random matrix approach which exclude the measurement noise from the filtered covariance matrix [4]–[7]. $\mathbb{E}(X_{k|k})$ (7) then represents only the covariance of the measurement data caused by the object's extension.

3) Applications for the Random Matrix Approach: The random matrix approach was extended in several publications [4]–[10] and is nowadays a state-of-the-art filter for EOT. It was also integrated into the probability hypothesis density (PHD) filter [11] to allow multiple extended object tracking (MEOT). In the result section, we are using real lidar data to estimate the extension of a vessel. In previous works, the random matrix approach has already been applied to ship tracking using radar sensors [19][20]. It has been shown that an ellipse is a suitable shape approximation for ships [19, p. 8][20, p. 6626]. In [12], random matrices are combined with the generalized probabilistic data association (GPDA) filter. The method is also evaluated using vessel tracking with radar data [12, p. 967]. A combination of a lidar sensor with the random matrix approach can be found e.g. in [11, p. 5659]. There, the extensions of a bicyclist and a pedestrian are estimated. However, the surfaces of a bicyclist and a pedestrian are not that homogeneous as the surface of vessels observed from the side. Therefore the measurement distribution in [11, p. 5659, Fig. 1] is closer to the measurement model assumed in [3] than the measurement model we assume for lidar in the next section. Non-uniformly distributed measurements were considered in [9] with a conditional Gaussian mixture (GM) model. The idea is to split the object into several areas with different measurement densities [9, p. 3819]. The extension is then a combination of the GM components [9, p. 3820]. If more than just the present state is of interest (e.g. parameter identification), smoothing methods can be used for random matrices [14].

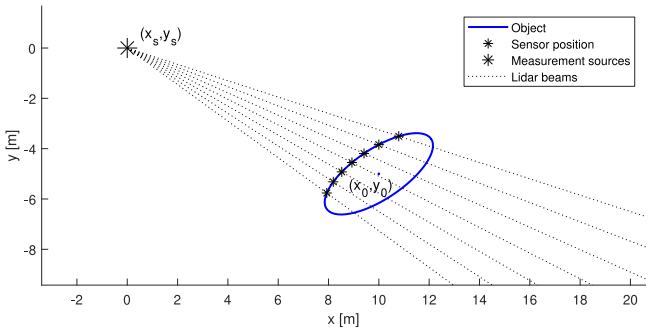


Fig. 1. Measurement generation with a lidar sensor.

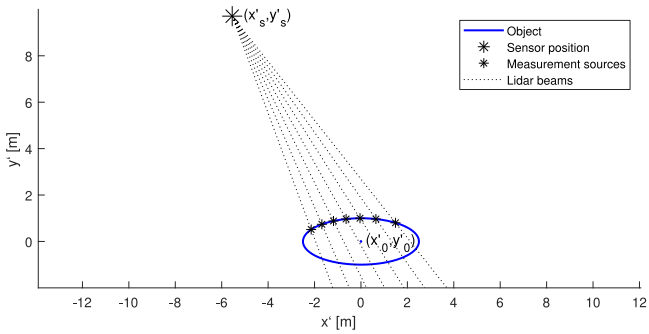


Fig. 2. Ellipse and Sensor in the target coordinate system.

III. Lidar Sensor Model

The measurement generation with a lidar sensor located at (x_s, y_s) for an elliptical-shaped object at (x_0, y_0) is illustrated in Fig. 1. The intersections of the lidar beams with the object's contour are considered as measurement sources and the measurements are generated by adding a measurement error. Since $(p_d < 1)$, not every measurement source will generate a measurement. The measurements do not follow a specific distribution that can be easily described analytically. The lidar measurement model has already been considered in [16]. However, in [16], the measurement sources are distributed uniformly around the contour of the ellipse [16, p. 1]. For an elliptical-shaped model, ray tracing with a lidar sensor was e.g. derived in [17, p. 5] and applied to extension estimation of vessels in [21]. In this section, a method to generate artificial lidar measurements is shown.

A. Ellipse Equations

The equation of an ellipse, centered at the origin and aligned with the coordinate axes, is given by:

$$\frac{x^2}{l^2} + \frac{y^2}{w^2} = \frac{1}{4}. \tag{19}$$

The center of the unrotated ($\varphi = 0°$) ellipse is located at the origin. In the general case, where the center of the ellipse is located at (x_0, y_0) and the ellipse is rotated by φ, the radial function is given by:

$$\begin{pmatrix} x \\ y \end{pmatrix} = \begin{pmatrix} x_0 \\ y_0 \end{pmatrix} + \begin{pmatrix} cos(\varphi) & -sin(\varphi) \\ sin(\varphi) & cos(\varphi) \end{pmatrix} \begin{pmatrix} \frac{l}{2}cos(t) \\ \frac{w}{2}sin(t) \end{pmatrix}. \tag{20}$$

B. Measurement Source (intersection) Calculation

Since the equation for the ellipse in (19) is given for the ellipse placed at the origin without rotation, the intersection calculation must be done in the target coordinate system. The ellipse and the sensor from the example in Fig. 1 are shown in the target coordinate system in Fig. 2.

Note that there is no calculation necessary for the transformed ellipse, since in any case, $x'_0 = 0$, $y'_0 = 0$ and $\varphi' = 0$. However, the sensor position (x'_s, y'_s) in the target coordinate system must be determined. The transformation of the sensor position is given by:

$$\begin{pmatrix} x'_s \\ y'_s \end{pmatrix} = \begin{pmatrix} cos(\varphi) & sin(\varphi) \\ -sin(\varphi) & cos(\varphi) \end{pmatrix} \begin{pmatrix} x_s - x_0 \\ y_s - y_0 \end{pmatrix} \tag{21}$$

With the position of the sensor in the target coordinate system, it is possible to write the equation for the lidar beams:

$$y = tan(\alpha_j)x + (y'_s - tan(\alpha_j)x'_s) \tag{22}$$

where α_j describes the directional angle of the lidar beam. The start point α_0 and the end point α_{max} define the opening angle of the sensor and the step size $\Delta_\alpha = \alpha_{j+1} - \alpha_j$ determines the resolution. The following equations must be calculated for each α_j and the corresponding lidar beam. To find the intersections of the lidar beams with the object's contour, (22) is inserted in (19). After some simplifications and rearrangements, we get:

$$\left(\frac{1}{l^2} + \frac{tan(\alpha_j)^2}{w^2}\right) x^2 + \frac{2tan(\alpha_j)}{w^2}\left(y'_s - tan(\alpha_j)x'_s\right) x + \frac{\left(y'_s - tan(\alpha_j)x'_s\right)^2}{w^2} - \frac{1}{4} = 0. \tag{23}$$

(23) is in the form $ax^2 + bx + c = 0$ and can easily be solved by the quadratic formula. The corresponding y'_m value can be calculated via (22). When there are solutions $x'_{m1,m2} \in \mathbb{R}$, the next step is to determine which solution is on the object's side facing towards the sensor. The euclidean distance to the position of the sensor (x'_s, y'_s) is calculated for both solutions. The solution with the smaller euclidean distance is a measurement source. The measurement source (x'_m, y'_m) must be transformed from the target coordinate system back to the original coordinate system. The transformation is given by:

$$\begin{pmatrix} x_m \\ y_m \end{pmatrix} = \begin{pmatrix} x_0 \\ y_0 \end{pmatrix} + \begin{pmatrix} cos(\varphi) & -sin(\varphi) \\ sin(\varphi) & cos(\varphi) \end{pmatrix} \begin{pmatrix} x'_m \\ y'_m \end{pmatrix} \tag{24}$$

and is the inverse mapping of (21). A similar derivation of the intersection points from the lidar beams with the object's contour is given in [17, p. 5].

C. Measurement Generation

To generate measurements, a random value $u \sim U(0, 1)$ is drawn for each measurement source. If $u < p_d$, then the measurement source will cause a measurement by adding white

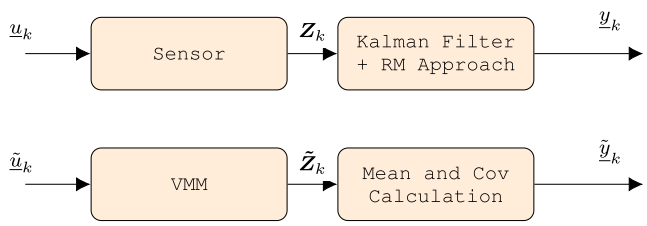


Fig. 3. Reference Model Concept.

Gaussian noise:

$$\underline{z}_j = \begin{pmatrix} x_m \\ y_m \end{pmatrix} + \begin{pmatrix} v_1 \\ v_2 \end{pmatrix} \tag{25}$$

with $v_i \sim \mathcal{N}(v; 0, \sigma_x^2)$.

IV. Reference Model Concept

In EOT, an inverse sensor model is usually set up to estimate the kinematic and extension of the object from the measurements. However, in some applications, it is difficult to set up a measurement likelihood function in closed-form. An example of one of those applications is a lidar measurement model (see Section III). With the reference model concept, it is possible to estimate the position and extension of a target with an elliptical shape without the need for a measurement likelihood function in closed form. This approach's basic idea is to interpret the entire processing chain from reference to estimation with the random matrix approach as a plant model. Fig. 3 shows this plant in its upper part. This plant will be termed the reference model from now on. The input vector $\underline{u}_k$ contains the position and extension parameters of the object, which are unknown.

$$\underline{u}_k = (x_k, y_k, l_k, w_k)^T \tag{26}$$

The sensor generates a set of measurements $\boldsymbol{Z}_k$ depending on the input vector $\underline{u}_k$. The Kalman filter with RM Extension estimation gives the filtered kinematic state $\underline{x}_{k|k}$ and the filtered covariance matrix $\mathbb{E}(X_{k|k})$. We denote the larger eigenvalue of $\mathbb{E}(X_{k|k})$ as e_1 and the smaller one as e_2. It is assumed that the square roots of these eigenvalues have a linear relation to the object's semi-axis. This leads to:

$$l_k \propto 2\sqrt{e_1}, \tag{27}$$

$$w_k \propto 2\sqrt{e_2}. \tag{28}$$

The output vector of the reference plant $\underline{y}_k$ is defined by:

$$\underline{y}_k = \left(x_{k|k}^{(Z)}, y_{k|k}^{(Z)}, 2\sqrt{e_1}, 2\sqrt{e_2}\right)^T, \tag{29}$$

and contains the filtered center of gravity and the extension parameters. With Singer's ODM, the center of gravity corresponds to the first two elements of the kinematic state. Other kinematic states such as velocity or acceleration are assumed to be known from the Kalman filters kinematic since they are almost completely unaffected by different measurement models. In this paper, we assume that the object is aligned according to its direction of motion, assuming no drifting. Therefore, the orientation angle $\varphi_{k|k}$ of the extension is known from the Kalman filters kinematic:

$$\varphi_{k|k} = atan\left(\frac{v_{y_{k|k}}}{v_{x_{k|k}}}\right). \tag{30}$$

This is accurate for simple motion models (like Singer's ODM). However, more complex approaches to estimate $\varphi_{k|k}$ (e.g. [13]) may be necessary for models that drift or whose orientation changes rapidly in time.

Parallel to the reference model, a virtual plant is installed, designed to emulate the reference model. The sensor is replaced by a virtual measurement model (VMM) and the filtering by a mean and covariance calculation. The output vector $\tilde{\underline{y}}_k$ of the VMM plant is in the same form as (29).

Since there is no filtering in the VMM plant, the resolution of the VMM must be sufficiently high to estimate all components of $\tilde{\underline{y}}_k$ precisely. A higher sensor resolution does not influence the center of gravity and the variance of the measurements. The VMM can be interpreted as a function to generate artificial measurements.

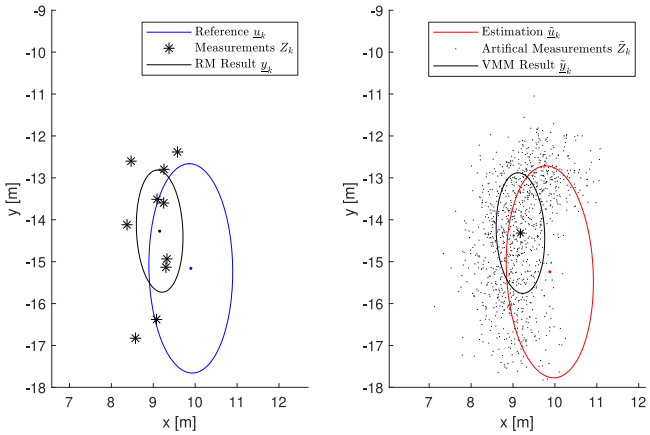
The goal of the reference model concept is to find an input $\tilde{\underline{u}}_k$ for the VMM plant such that the output $\tilde{\underline{y}}_k$ matches the output of the reference plant $\underline{y}$. If the outputs are equal and the inputs are observable, then the inputs must also be identical and $\tilde{\underline{u}}_k$ thus becomes a state estimation. This turns out to be an optimization problem for which the following error vector and error norm are defined:

$$\underline{\epsilon}_k = \underline{y}_k - \tilde{\underline{y}}_k, \tag{31}$$

$$\epsilon_k = \left(\underline{y}_k - \tilde{\underline{y}}_k\right)\left(\underline{y}_k - \tilde{\underline{y}}_k\right)^T. \tag{32}$$

This paper presents an adaptation algorithm for the input $\tilde{\underline{u}}_k$, which is based on an integral adaptation law with variable gain K_i. The adaptation algorithm is shown in Section V. Another way to minimize ϵ could be given by the Nealder-Mead algorithm [32]. The Nealder-Mead algorithm is a numeric optimization method and uses a simplex to find the optimum input for the minimum error. In the following, we will look at an example of a tracking problem to illustrate the reference model concept. A noisy lidar sensor located at the origin is used for tracking an elliptical-shaped object with the random matrix approach. Fig. 4(a) shows the results for the reference plant from the reference object to the random matrix approach. The RM Result is shown as the σ-ellipse (with $l_k = 2\sqrt{e_1}$ and $w_k = 2\sqrt{e_2}$).

Input and output of the virtual plant are shown in Fig. 4(b). We assume that an adaptation algorithm gives a certain $\tilde{\underline{u}}_k$ (Estimation). This $\tilde{\underline{u}}_k$ is used as an input into the VMM to generate a large number of artificial measurements. The mean and covariance of these measurements are shown using the σ-ellipse. To validate the estimation $\tilde{\underline{u}}_k$ (Fig. 4(b)), a comparison with the reference $\underline{u}_k$ (Fig. 4(a)) would be meaningful. In real-world applications, there is no access to the reference. However, we can compare the result of the random matrix approach with the mean and covariance of the artificial measurements. These are the two black ellipses in Fig. 4. In this example, the mean and covariance of the artificial measurements fit well to the result of the random matrix approach. The error ϵ (32) is expected to be



(a) Reference plant (b) Virtual plant

Fig. 4. Introductory example at time step $k = 50$

small. Therefore, $\underline{\tilde{u}}_k$ is a good estimation for the position and extension of the target.

When using the random matrix approach as given in Section II-C based on [3], measurement noise must be considered in the VMM. Either by generating measurements affected by noise or by adding the measurement noise covariance matrix on the empirical covariance matrix of the artificial measurements. However, when using advanced versions like [4]–[7], the expected value of the inverse Wishart contains only the extension part caused by the object's extent. In this case, the VMM must be designed without any noise.

V. Adaptation Algorithm

The goal of the adaptation algorithm is to find the input $\underline{\tilde{u}}_{k,i}$ that minimizes the error $\epsilon_{k,i}$. Fig. 5 shows the structure of the approach. The start of the flow chart from the real-world parameters $\underline{u}_k$ to the result of the random matrix approach $m^{(Z)}_{k|k}, \Sigma^{(Z)}_{k|k}$ is already known from Fig. 3 and the virtual plant in the lower part of the chart. For every time step k, the adaptation algorithm is performed in I iterations. The index for the iteration is $i \in [1, 2, \ldots I]$. In every iteration i, the adaptation algorithm is updating the parameters in $\underline{\tilde{u}}_{k,i}$ using the following adaptation law:

$$\underline{\tilde{u}}_{k,i+1} = \underline{\tilde{u}}_{k,i} - K_i \underline{\epsilon}_{k,i}, \tag{33}$$

with $\underline{\epsilon}_{k,i} = \underline{y}_{k|k} - \underline{\tilde{y}}_{k,i}$.

A. Closed-Loop Stability

The lower half of Fig. 5, consisting of the VMM, the mean, covariance, eigenvalue calculation and the adaptation, forms a closed feedback loop. The most critical requirement for the adaptation law is that this loop is asymptotically stable. This control loop is simplified by combining all components except the adaptation into one plant denoted as A. This leads to the structure given in Fig. 6.

A central challenge in this approach is that the measurement likelihood and consequently the plant A cannot be represented in closed form. However, upper and lower bounds can be assumed to prove stability. For the center of gravity of the measurements, it is assumed that it can shift from the center of the object by a maximum of half of the largest extension in the direction of the sensor. This can be described by:

$$\tilde{z}_x = \tilde{x} - sgn(\tilde{x}) max(\tilde{l}, \tilde{w}) a \tag{34}$$

$$\tilde{z}_y = \tilde{y} - sgn(\tilde{y}) max(\tilde{l}, \tilde{w}) b \tag{35}$$

with $a \in [0; 0.5]$ and $b \in [0; 0.5]$. a and b are considered constant within a time step. Thus, possible variation can be neglected during the adaptation iterations. For the eigenvalues, it is assumed that the root of the larger eigenvalue corresponds linearly to half the length and the root of the smaller eigenvalue corresponds linearly to half the width. The expressions cannot become negative and can be at most as large as the expansion parameters. Therefore, it holds:

$$2\sqrt{\tilde{e}_1} = c\tilde{l}, \tag{36}$$

$$2\sqrt{\tilde{e}_2} = d\tilde{w}, \tag{37}$$

with $c \in [0; 1]$ and $d \in [0; 1]$. For the first quadrant ($\tilde{x}, \tilde{y} > 0$) and with the assumption that the length is per definition larger than the width, (34)–(37) can be rearranged into a system of linear equations in the form $\underline{\tilde{y}}_{k,i} = A \underline{\tilde{u}}_{k,i}$.

$$\begin{pmatrix} \tilde{z}_x \\ \tilde{z}_y \\ 2\sqrt{\tilde{e}_1} \\ 2\sqrt{\tilde{e}_2} \end{pmatrix} = \begin{pmatrix} 1 & 0 & -a & 0 \\ 0 & 1 & -b & 0 \\ 0 & 0 & c & 0 \\ 0 & 0 & 0 & d \end{pmatrix} \begin{pmatrix} \tilde{x} \\ \tilde{y} \\ \tilde{l} \\ \tilde{w} \end{pmatrix}. \tag{38}$$

Using the adaptation law (33) leads to a closed-loop dynamic given by:

$$\underline{\tilde{u}}_{k,i+1} = \underline{\tilde{u}}_{k,i} - K_i A \underline{\tilde{u}}_{k,i}, \tag{39}$$

$$= (I - K_i A)\, \underline{\tilde{u}}_{k,i}. \tag{40}$$

To prove that the closed-loop is asymptotically stable, it must be shown that the absolute values of all four eigenvalues of $(I - K_i A)$ are less than one.

$$(I - K_i A) = \begin{pmatrix} 1 - K_i & 0 & aK_i & 0 \\ 0 & 1 - K_i & bK_i & 0 \\ 0 & 0 & 1 - cK_i & 0 \\ 0 & 0 & 0 & 1 - dK_i \end{pmatrix} \tag{41}$$

Since (41) is upper triangular, the eigenvalues are found on the main diagonal. Therefore, a and b do not influence stability. The first and second eigenvalue lead to $0 < K_i < 2$ to fulfill the condition $|1 - K_i| < 1$. For the third eigenvalue, K_i must fulfill the condition:

$$1 - cK_i < 1, \tag{42}$$

$$cK_i > 0, \tag{43}$$

which is already the case since both c and K_i are positive due to previous restrictions. The third eigenvalue must also fulfill the following condition:

$$1 - cK_i > -1, \tag{44}$$

$$K_i < \frac{2}{c} \tag{45}$$

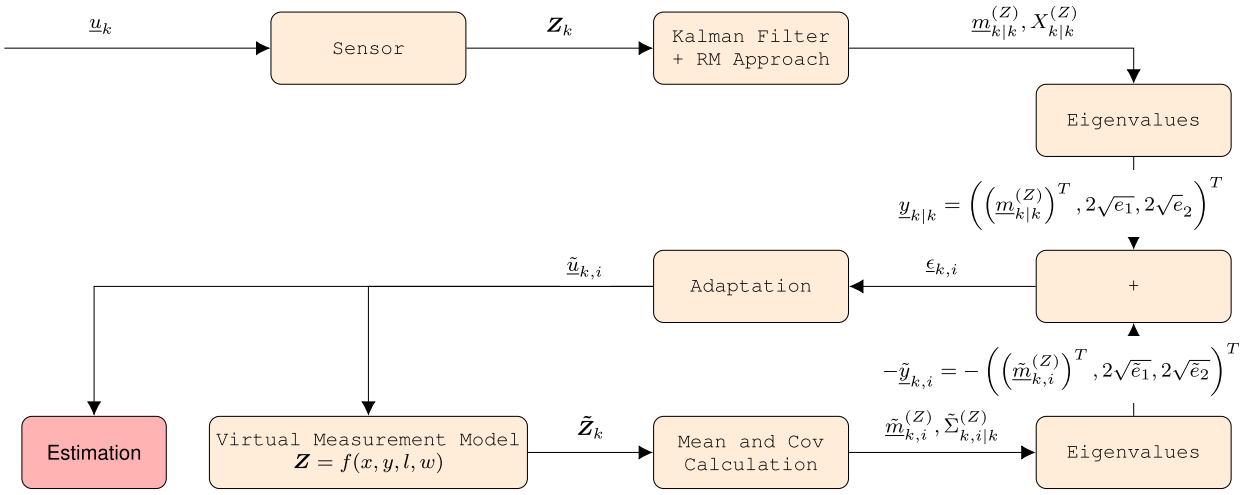


Fig. 5. Reference Model Concept with RM Approach and an Adaptation Algorithm.

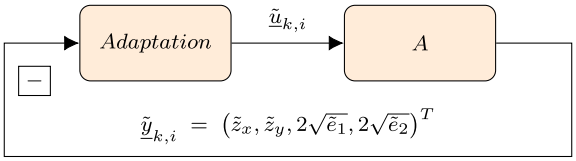


Fig. 6. Adaptation in closed-loop.

Inserting the maximum value $c = 1$ leads to $K_i < 2$. Therefore, no further restrictions for K_i apply from this eigenvalue. Since c and d are defined in the same intervals and due to symmetry, the same applies to the fourth eigenvalue. It could be shown that the adaptation law (41) results in an asymptotically stable closed-loop for $0 < K_i < 2$.

B. Initialization

There are three options to initialize the adaptation algorithm in each time step:

1) Initialize with the result of the Kalman Filter with Random Matrix extension estimation using the following relations (mandatory in the first time step):

$$\tilde{x}_{k,1} = \underline{m}^{(Z)}_{1|1}(1), \tag{46}$$

$$\tilde{y}_{k,1} = \underline{m}^{(Z)}_{1|1}(2), \tag{47}$$

$$\tilde{l}_{k,1} = 2\sqrt{e_1}, \tag{48}$$

$$\tilde{w}_{k,1} = 2\sqrt{e_2}. \tag{49}$$

2) Prediction of the VMM depending on the selected motion model:

$$\tilde{x}_{k,1} = f_x(\tilde{x}_{k-1,I}), \tag{50}$$

$$\tilde{y}_{k,1} = f_y(\tilde{y}_{k-1,I}), \tag{51}$$

$$\tilde{l}_{k,1} = \tilde{l}_{k,I}, \tag{52}$$

$$\tilde{w}_{k,1} = \tilde{w}_{k,I}. \tag{53}$$

3) Continue with the last VMM input of the previous time step:

$$\underline{\tilde{u}}_{k,1} = \underline{\tilde{u}}_{k-1,I}. \tag{54}$$

The most precise option is a prediction from the previous VMM parameters to the current time step. This makes the initial error $\epsilon_{k,1}$ smaller than with the other two methods. Thus, the number of iterations per time step can be kept small.

C. Implementation

It was previously shown that the adaptation algorithm for the VMM input is stable for $0 < K_i < 2$. However, exceptions can occur in an implementation where the assumptions made for stability are not fulfilled. One issue that e.g. can occur is that after the VMM update, the width is larger than the length. This would result in a wrong assignment of the eigenvalues (since the larger one is always assumed to correspond to the length). This can be overcome by switching length and width if necessary after the adaptation or by smaller values for K_i. It must also be ensured that length and width are non-negative after adaptation. Due to stochastic properties, it is still possible in rare situations that parameters (such as c and d) lie outside the intended range. According to (45), an increased c would require a smaller K_i for asymptotical stability.

As a remedy for all these sources of instability, we propose a variable K_i with step-size control since, in those situations, only the range of validity for K_i became smaller. As a criterion to detect the need for a decrease in K_i, we propose the error norm defined in (32). Algorithm 2 shows a suggested implementation of the VMM adaptation with variable K_i.

VI. Results

The proposed approach for extension estimation using a VMM and its adaptation are evaluated in this section, first in simulation studies and then using real lidar data.

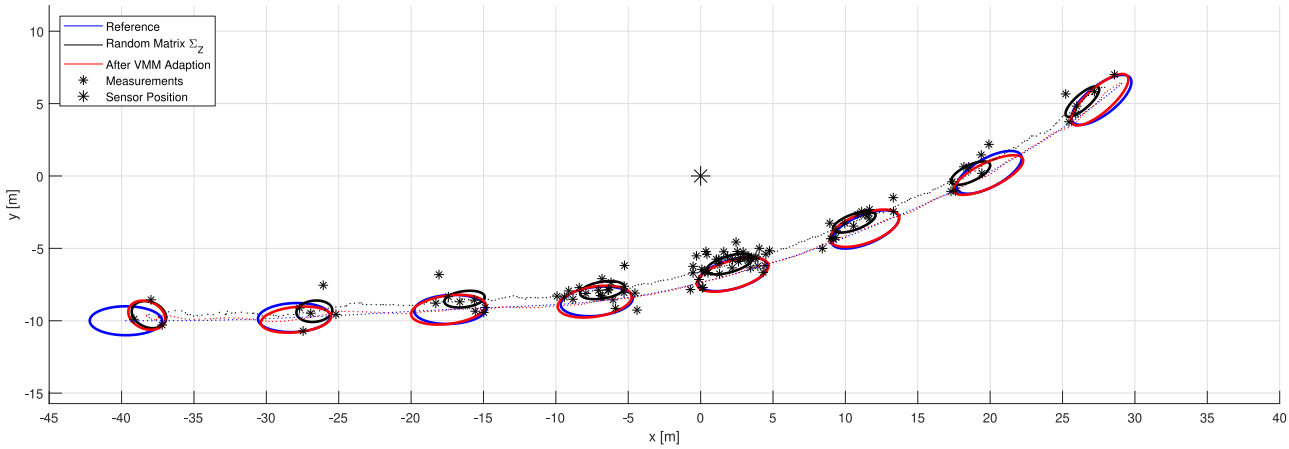


Fig. 7. Tracking results for a Singer's ODM observed by a lidar sensor. Linear Kalman filter implementation. The extension estimation and the measurements are shown every 40 time steps.

Algorithm 1: Random Matrix Approach With VMM Adaptation.

for k=1,...,K **do**
  Kalman filter prediction and update of kinematic and extension using (7)–(10)
  if k==1 **then**
    VMM input initialization (46)–(49).
  else
    VMM input prediction (50)–(53).
  end if
  get initial error ϵ_0 (32).
  initialize K_i ($0 << K_i << 2$).
  for i=1,...,I **do**
    store VMM input
    VMM input adaptation (33) and run VMM
    get error ϵ_i (32).
    if $\epsilon_i > \epsilon_{i-1}$ **then**
      restore previous VMM input parameters.
      decrease K_i (e.g. divide by 2)
    end if
  end for
  provide last VMM input parameters as state estimation
end for

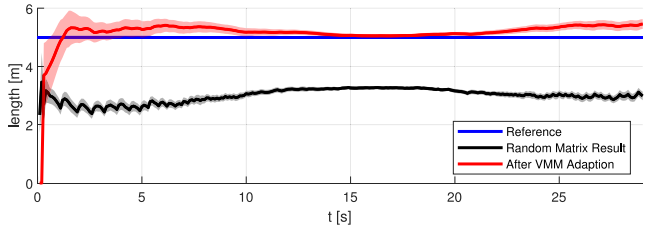
A. Linear Kalman Filter With Lidar Measurement Model

A Singer's ODM is observed by a lidar sensor. The center of gravity and the variance of the measurements are filtered using the random matrix approach as presented in [3]. The filtered center of gravity and the $1\sigma-$ellipses are plotted in Fig. 7. Using the VMM and the proposed adaptation algorithm, the extension estimation is adapted for two iterations in each time step as proposed in this paper using a variable gain starting with $K_i = 0.1$. If the error ϵ increases after an iteration, the previous values are reconstructed and the gain K_i is divided by 2 for the next iteration. In the next time step, K_i is set to 0.1 again.

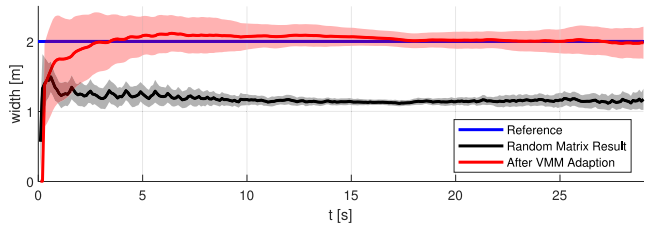
The results after the adaptation are shown in Fig. 7. The scenario is 29 s long with a sampling period $T = 0.1$ s. The kinematic state $(x, y, v_x, v_y, a_x, a_y)'$ is initialized with $(-40\,\mathrm{m}, -10\,\mathrm{m}, 3\,\mathrm{m/s}, 0\,\mathrm{m/s}, 0\,\mathrm{m/s^2}, 0\,\mathrm{m/s^2})$. Acceleration is affected by white Gaussian noise with $\sigma_{ax} = \sigma_{ay} = 0.01\,\mathrm{m/s^2}$. The object has a length $l = 5$ m and a width $w = 2$ m. The Lidar sensor is located in the origin and has a resolution of 1°. At the intersections with lidar beams and the model's contour, the detection probability is 100 % at the front side and 0 % at the backside. Measurements are affected by white Gaussian noise with $\sigma_x = \sigma_y = 0.5$ m. The virtual measurement model has the same properties, only the resolution is set to 0.01° and no measurement noise is assumed. The Kalman filter is initialized by the measurements from the first time step with an initial uncertainty $P = diag((10, 5, 1))$. For the random matrix approach, a parameter of $\tau = 0.5$ and an initial value $\nu = 4$ are used.

In Fig. 7, it is shown that the center of gravity of the measurement data filtered by the random matrix approach has a shift in the direction of the sensor since measurements only appear on the object's contour facing towards the sensor. This results in a bias of the whole trajectory. The filtered 1σ-ellipse is also smaller in both directions than the object's extent and depends on the position of the object in the sensor coordinate frame, while the actual size of the object is constant. Therefore, it would not be appropriate to use a constant scaling factor to estimate the extent from the filtered 1σ-ellipse (respectively the covariance matrix Σ_z). With the adaptation presented in this paper, the bias in the trajectory estimation is removed and the estimated extension matches the reference extension very precisely. However, the accuracy after adaptation depends on the precision of the RM results. In the first estimate (shown at $\approx (-38\,\mathrm{m}| -10\,\mathrm{m})$), the filter has not yet fully converged, so even the adaptation cannot give a very accurate result.

1) Monte Carlo Simulation - Single Trajectory: The scenario from Fig. 7 is evaluated in 100 MC runs. The trajectory is only generated once and the measurements are generated independently in every MC run. Fig. 8 shows the results of the extension

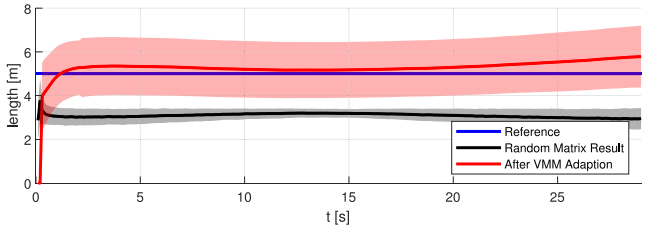


(a) Length

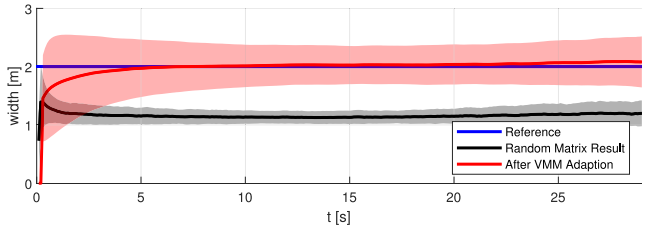


(b) Width

Fig. 8. Extension estimation (mean and standard deviation) averaged over 100 MC runs.



(a) Length



(b) Width

Fig. 9. Extension estimation averaged over 1000 MC runs with random trajectories

estimation. After approx. $15-20$ s, the object passes the sensor. At this time, the highest number of sensor readings is generated. Therefore, the standard deviations of all extension estimations are the smallest. First, the random matrix results are discussed: At the drive-by after $15-20$ s, the entire length of the object can be observed by the sensor. This results in a maximum in the length estimation with RM shown in Fig. 8(a). If the sensor can see the front or the back, the length estimation will be smaller. For the width estimation, this behavior can be observed in Fig. 8(b) in the opposite way. When the object is heading with the front or the back towards the sensor, the sensor can observe the entire width. The width estimation of RM increases at the beginning and the end. After the adaptation using the VMM, the estimated extension matches the reference almost perfectly when the object is close to the sensor. However, if there are not enough measurements, the adaptation tends to overestimate the extension slightly.

2) Monte Carlo Simulation - Random Trajectories: To demonstrate that the VMM and its adaptation does not only work for one trajectory, a Monte Carlo simulation is performed for random trajectories. In each MC run, the initial position is drawn from a Gaussian birth density with expected value $\mathbb{E}(\underline{x}_0) = (-40, -20, 3, 0, 0, 0)^T$ and covariance matrix $Cov(\underline{x}_0) = diag((10, 10, 0.01, 0.01, 0.001, 0.001))$. The process noise of the trajectories is generated independently. Fig. 9 shows the results of the extension estimation.

The extension is slightly overestimated. This bias increases at the end of the scenario, when the target is usually far away from the sensor. The standard deviation has increased compared to the previous simulation with only one trajectory. This was to be expected since some trajectories closely pass the sensor and others do not.

3) Studies on Stability and Computational Cost: Section V-A shows that the adaptation algorithm is asymptotically stable for $0 < K_i < 2$. To demonstrate this, the scenario shown in Fig. 7 is considered for different K_i values. K_i is fixed for the whole scenario and not adjusted, even if the error increases. In each time step, the VMM input is initialized with the Random Matrix result using (46)–(49) and nine iterations are performed. The following Figure shows the error (32) per iteration, averaged over all time steps. Fig. 10 shows a converging error for $K_i < 2$. For $K_i > 2$, the algorithm diverges. The first two eigenvalues from (41) are outside the unit circle. In this case, the third and fourth eigenvalue could sometimes still be stable, depending on whether c and d fully utilize the assumed limits. However, to guarantee stability in all scenarios, we strongly recommend using the variable K_i due to the reasons described in V-C. Using the VMM prediction (50)–(53), a smaller K_i can be used to minimize the error and many iterations can be avoided. This reduces the computational cost, which depends strongly on the number of iterations, shown in Table I.The Kalman Filter with RM extension estimation requires an average of 0.14 ms per

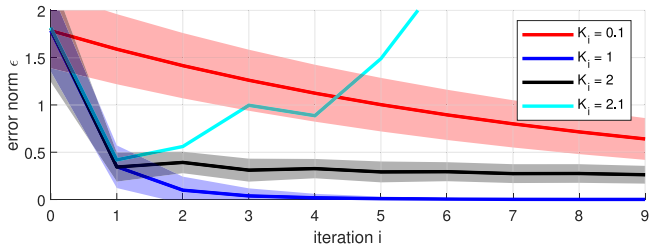


Fig. 10. Error norm (mean and standard deviation), averaged over all time steps. The standard deviation for $K_i = 2.1$ is not shown since its region would cover the whole Figure.

TABLE I
COMPUTATIONAL COST PER TIME STEP, AVERAGED OVER 100 MC RUNS. MATLAB-IMPLEMENTATION ON AN INTEL CORE I7-3720QM CPU @2.6 GHZ

number of iterations	1	2	3	5	10
computational cost [ms]	7.0	15.7	23.5	40.7	52.1

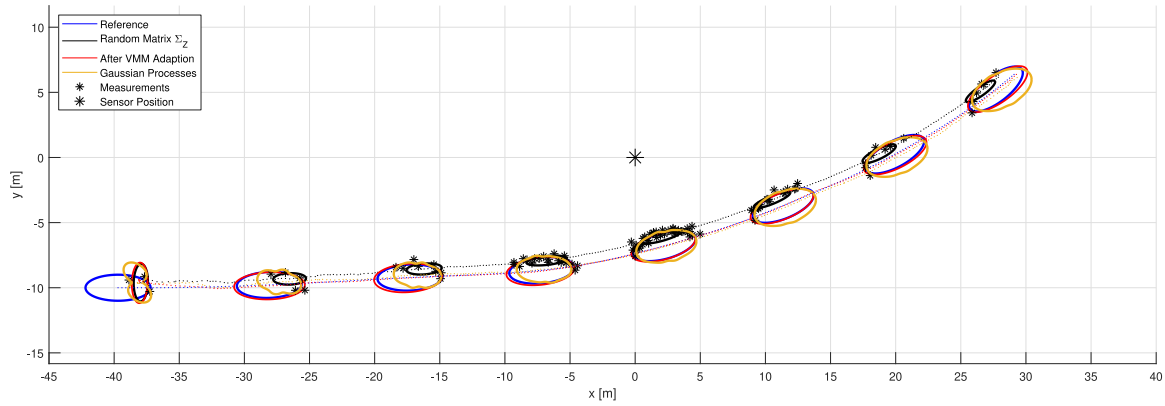


Fig. 11. Tracking results for a Singer's model observed by a lidar sensor. Comparison with Gaussian Processes.

time step. The main reason for the high computational cost is the generation of many artificial measurements. Reducing the resolution of the VMM can reduce the computational cost.

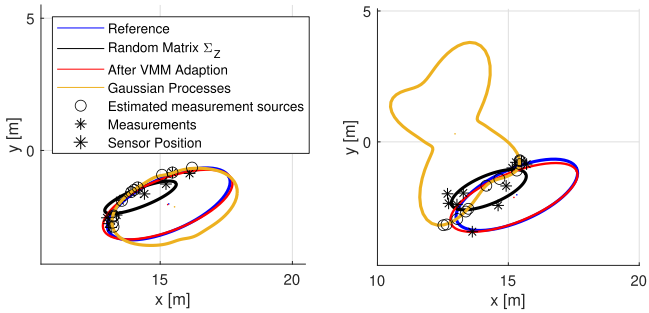
B. Comparison With Gaussian Processes

The results of the extension estimation using a VMM and its adaptation are compared to a shape estimation based on Gaussian Processes (GP) [28]. For the GP algorithm, a nonlinear motion model (Coordinated turn with polar velocity) is used. The kinematic state contains position (x, y), velocity v, orientation angle φ and turn rate ω. The transition is given in [33, p. 2, Equation (4)]. This motion model has the advantage that the orientation φ is directly available. In the GP algorithm, the measurements have to be transformed into the local target coordinate frame using the orientation [28, p. 4169]. A symmetric model is assumed. Therefore, the covariance function has a period of π. The kernel is given by:

$$k(\theta, \theta') = \sigma_f^2 e^{-\frac{sin^2(|\theta-\theta'|)}{2l^2}}, \tag{55}$$

with angle θ. The following parameters are used for the kernel function: $\sigma_f = 2$, $\sigma_r = 2$ and $l = \pi/4$ and a total of 100 radial function samples are used. An extended Kalman filter framework is used to deal with the nonlinear model.

Fig. 11 compares the results of the extension estimation using the VMM with the results of the extension estimation using Gaussian Processes. The measurement noise decreased to $\sigma_x = \sigma_y = 0.2$ m. With large measurement noise (e.g. $\sigma_x = 0.5$ m as used in Section VI-A), the GP algorithm may not converge since measurements are often assigned to the wrong part of the object's contour. Fig. 12 shows the extension estimation at $k = 220$ for different measurement noise levels. In scenarios with low noise (see Fig. 11 and Fig. 12(a)), both algorithms achieve very good results. The GP approach overestimates the width, but unlike the VMM, it does not have the prior information that there is a perfectly elliptical object. This also causes GP to take longer to converge. However, if the shape is unknown and the measurement noise is not too high, GP may be superior. The VMM extension estimation is almost unaffected by the increased measurement noise in Fig. 12(b). Table II provides numerical results. Mean and RMSE are evaluated in 100 MC runs for both algorithms. Since the GP approach may diverge in some



(a) $\sigma_x = 0.2$ m (b) $\sigma_x = 0.5$ m

Fig. 12. Extension estimation at $k = 220$ for different measurement noise.

TABLE II
MEAN AND RMSE AVERAGED OVER 100 MC RUNS

	Ref.	GP mean	VMM mean	GP RMSE	VMM RMSE
length	5 m	4.71 m	5.20 m	1.23 m	0.35 m
width	2 m	2.66 m	2.05 m	1.20 m	0.24 m

Fig. 13. Observed vessel.

situations, results with an error larger than 5 m are neglected for Table II. Since GP has difficulties especially at the beginning and at the end of the scenario with only a few measurements, the VMM approach is superior. The average computational cost per time step of the GP algorithm is 36 ms and is therefore on the same level as the VMM with $4-5$ iterations (see Table I).

C. Kalman Filter With Real Lidar (Velodyne VLP-16) Data

In this scenario, a vessel has been observed on a river using a Velodyne VLP-16 (Puck) Lidar. Fig. 13 shows the observed vessel. With real data, the challenges in extension estimation are different compared to simulation studies. In the previous

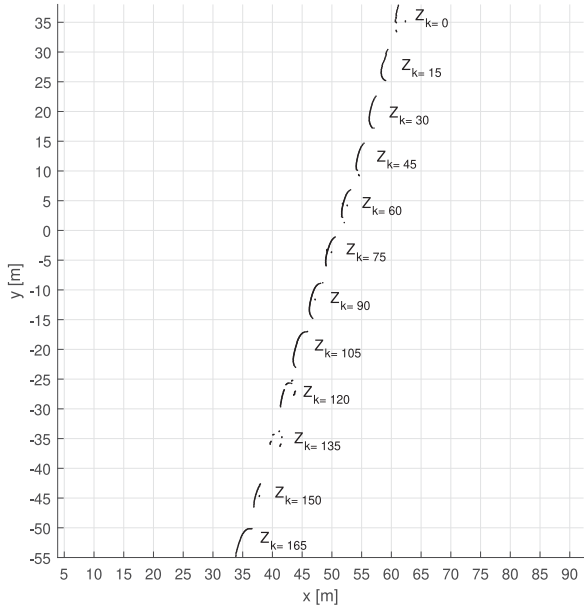
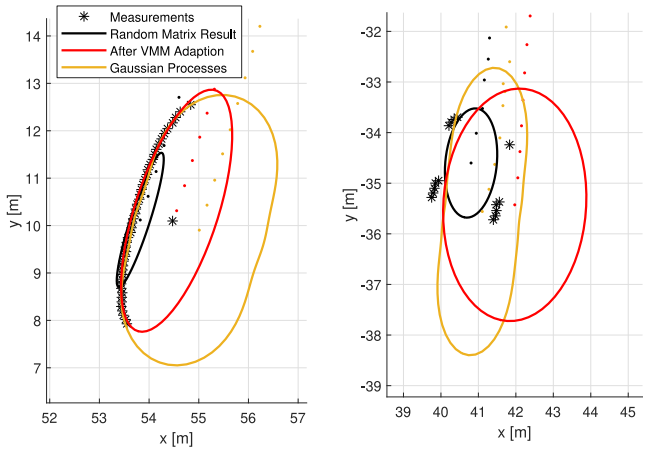
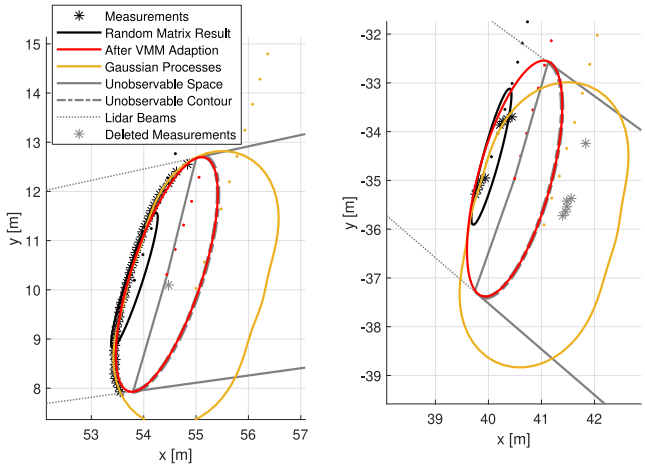


Fig. 14. Lidar measurements shown for every 15 time steps.



(a) Tracking result at $k = 50$ (b) Tracking result at $k = 135$

Fig. 15. Tracking results at selected time steps



(a) $k = 50$ (b) $k = 135$

Fig. 16. Tracking results, increased τ

scenario, a measurement noise with $\sigma_x = \sigma_y = 0.5$ m was used. However, the VLP-16 has a range accuracy of ± 0.03 m. The measurement noise can easily be transformed in Cartesian coordinates using a linear approximation. Since the horizontal resolution $(0.1^\circ - 0.4^\circ)$ is very high, there are numerous measurements in each time step if the target is close enough. The vertical resolution (2.0°) is too low to receive measurements from more than one layer. Therefore, a 2-dimensional tracking method is sufficient. The major challenge is that the assumptions of the measurement model can only be fulfilled to a limited extent: The vessel's contour is not a perfect ellipse and measurements do not always occur only at the target's side faced towards the sensor. Fig. 14 shows the received lidar measurements for every 15th time step. From the beginning until approximately time step $k = 105$, the measurements match almost perfectly to the assumed model with just a few outliers. From $k \approx 120$ until $k \approx 135$, measurements also occur from the side faced away from the sensor. For the filter with VMM-Adaptation, the same parameters were used as in the previous simulation study (Fig. 7), except the measurement noise has been decreased to $\sigma_x = \sigma_y = 0.03$ m. The tracking results at $k = 50$ and $k = 135$ are shown in Fig. 15.

When measurements appear where expected, the extension can be estimated very precisely - see Fig. 15(a). However, if the measurements model does not match the sensor's behavior, the extension estimation becomes significantly worse (see Fig. 15(b)). The extension estimation using Gaussian Processes shows a different behavior: When measurements are only on the object's side facing the sensor, the object is estimated wider than with the VMM. When measurements appear on the other side, the GP algorithm matches them to measurement sources at the backside. The width estimation gets smaller and close to its supposed reference value. There are several options to improve the VMM: The Random Matrix Approach [3] uses a parameter τ which describes how fast the estimated covariance (and therefore the estimated extension) can change. Of course, the length and width of an object do not change, but a rotation of the covariance matrix can be caused by a turn of the object. Another option is to distinguish between measurements in the observable space (according to the assumed model) and those that seem to be outside. The observable space can be determined by the prediction of the VMM once it has converged. Measurements assigned to the non-observable part of the object's contour are deleted. The tracking results for this setting are shown in Fig. 16(a). The parameter τ has been increased from $\tau = 0.5$ up to $\tau = 50$ and only the measurements at the object's side facing towards the sensor are used. The boundary of the unobservable space is also shown. It is defined by the tangential lidar beams and the predicted center of the target. The estimation in Fig. 16(a) has not changed since, in this time step, only a few measurements are affected by the gating. In Fig. 16(b), the estimation of extension and center has been improved. Without measurement gating, a length of ≈ 5.50 m is estimated (see Fig. 15(a)) and with measurement gating, a length of ≈ 5.00 m (see Fig. 16(a)). Both values seem

to be reasonable for the observed vessel. The width is estimated as ≈ 2.00 m, resp. ≈ 1.50 m with measurement gating. When we assume that the measurements shown in Fig. 16(b) are from both sides of the object's contour, a width around 2.00 m seems reasonable. When there are measurements from both sides of the object, the extension estimation using GP is close to this value (see Fig. 15(b)). In this scenario, the object was mainly observed from the side. This made the length estimation easy and the width estimation challenging.

VII. Conclusion

We have proposed a new approach for extended object tracking with a lidar sensor using the random matrix approach [3] to filter the mean and the covariance matrix of measurements. A virtual measurement model with an adaptation algorithm is used to determine which model parameters are most probable to cause this mean and this covariance matrix of the measurements. It could be shown that the adaptation algorithm fulfills the conditions for closed-loop stability and therefore finds the optimal parameters. Simulation studies were used to evaluate the performance and stability of this algorithm and to compare it with an extent estimation using Gaussian Processes (GP) [28]. In the simulations, it was shown that the center and extent could be estimated very precisely. The algorithm presented here is significantly more robust against strong measurement noise than GP. However, the shape (ellipse) is assumed to be a known prior. When this is not given, shape estimators like GP may be the better choice.

Besides the simulation studies, the approach was also evaluated on real lidar (Velodyne VLP-16 Puck) data. At most time steps, the assumptions of the model are held. However, in some time steps, measurements appeared at the side facing away from the sensor, which was not included in the VMM. In these situations, an additional heuristic like measurement gating can significantly improve the tracking results.

References

[1] R. Kalman, "A new approach to linear filtering and prediction problems," *J. Basic Eng.*, vol. 82, no. 1, pp. 35–45, 1960.

[2] K. Granström, M. Baum, and S. Reuter, "Extended object tracking: Introduction, overview, and applications," *J. Adv. Inf. Fusion*, vol. 12, pp. 139–174, 2017.

[3] J. W. Koch, "Bayesian approach to extended object and cluster tracking using random matrices," *IEEE Trans. Aerosp. Electron. Syst.*, vol. 44, no. 3, pp. 1042–1059, Jul. 2008.

[4] M. Feldmann and D. Franken, "Tracking of extended objects and group targets using random matrices—A new approach," in *Proc. 11th Int. Conf. Inf. Fusion*, 2008, pp. 1–8.

[5] M. Feldmann and D. Franken, "Advances on tracking of extended objects and group targets using random matrices," in *Proc. 12th Int. Conf. Inf. Fusion*, 2009, pp. 1029–1036.

[6] M. Feldmann, D. Fränken, and W. Koch, "Tracking of extended objects and group targets using random matrices," *IEEE Trans. Signal Process.*, vol. 59, no. 4, pp. 1409–1420, Apr. 2011.

[7] J. Lan and X. R. Li, "Tracking of extended object or target group using random matrix: New model and approach," *IEEE Trans. Aerosp. Electron. Syst.*, vol. 52, no. 6, pp. 2973–2989, Dec. 2016.

[8] L. Zhang and J. Lan, "Extended object tracking using random matrix with skewness," *IEEE Trans. Signal Process.*, vol. 68, pp. 5107–5121, Aug. 2020.

[9] L. Zhang and J. Lan, "Tracking of extended object using random matrix with non-uniformly distributed measurements," *IEEE Trans. Signal Process.*, vol. 69, pp. 3812–3825, Jun. 2021.

[10] N. J. Bartlett, C. Renton, and A. G. Wills, "A closed-form prediction update for extended target tracking using random matrices," *IEEE Trans. Signal Process.*, vol. 68, pp. 2404–2418, Apr. 2020.

[11] K. Granstrom and U. Orguner, "A PHD filter for tracking multiple extended targets using random matrices," *IEEE Trans. Signal Process.*, vol. 60, no. 11, pp. 5657–5671, Nov. 2012.

[12] M. Schuster, J. Reuter, and G. Wanielik, "Probabilistic data association for tracking extended targets under clutter using random matrices," in *Proc. 18th Int. Conf. Inf. Fusion*, 2015, pp. 961–968.

[13] B. Tuncer and E. Özkan, "Random matrix based extended target tracking with orientation: A new model and inference," *IEEE Trans. Signal Process.*, vol. 69, pp. 1910–1923, Mar. 2021.

[14] K. Granström and J. Bramstång, "Bayesian smoothing for the extended object random matrix model," *IEEE Trans. Signal Process.*, vol. 67, no. 14, pp. 3732–3742, Jul. 2019.

[15] J. Lan and X. R. Li, "Extended-object or group-target tracking using random matrix with nonlinear measurements," *IEEE Trans. Signal Process.*, vol. 67, no. 19, pp. 5130–5142, Oct. 2019.

[16] H. Alqaderi, F. Govaers, and R. Schulz, "Spacial elliptical model for extended target tracking using laser measurements," in *Proc. Sensor Data Fusion: Trends, Solutions, Appl.*, 2019, pp. 1–6.

[17] K. Granström, C. Lundquist, and U. Orguner, "Tracking rectangular and elliptical extended targets using laser measurements," in *Proc. 14th Int. Conf. Inf. Fusion*, 2011, pp. 1–8.

[18] L. Hammarstrand, L. Svensson, F. Sandblom, and J. Sorstedt, "Extended object tracking using a radar resolution model," *IEEE Trans. Aerosp. Electron. Syst.*, vol. 48, no. 3, pp. 2371–2386, Jul. 2012.

[19] K. Granström, A. Natale, P. Braca, G. Ludeno, and F. Serafino, "PHD extended target tracking using an incoherent X-band radar: Preliminary real-world experimental results," in *Proc. 17th Int. Conf. Inf. Fusion*, 2014, pp. 1–8.

[20] K. Granström, A. Natale, P. Braca, G. Ludeno, and F. Serafino, "Gamma Gaussian inverse wishart probability hypothesis density for extended target tracking using X-band marine radar data," *IEEE Trans. Geosci. Remote Sens.*, vol. 53, no. 12, pp. 6617–6631, Dec. 2015.

[21] K. A. Ruud, E. F. Brekke, and J. Eidsvik, "LIDAR extended object tracking of a maritime vessel using an ellipsoidal contour model," in *Proc. Sensor Data Fusion: Trends, Solutions, Appl.*, 2018, pp. 1–6.

[22] M. Baum, B. Noack, and U. D. Hanebeck, "Extended object and group tracking with elliptic random hypersurface models," in *Proc. 13th Int. Conf. Inf. Fusion*, 2010, pp. 1–8.

[23] M. Baum and U. D. Hanebeck, "Extended object tracking with random hypersurface models," *IEEE Trans. Aerosp. Electron. Syst.*, vol. 50, no. 1, pp. 149–159, Jan. 2014.

[24] S. Yang and M. Baum, "Tracking the orientation and axes lengths of an elliptical extended object," *IEEE Trans. Signal Process.*, vol. 67, no. 18, pp. 4720–4729, Sep. 2019.

[25] K. Granström, S. Reuter, D. Meissner, and A. Scheel, "A multiple model PHD approach to tracking of cars under an assumed rectangular shape," in *Proc. 17th Int. Conf. Inf. Fusion*, 2014, pp. 1–8.

[26] M. Baum and U. D. Hanebeck, "Tracking an extended object modeled as an axis-aligned rectangle," in *Proc. 4th German Workshop Sensor Data Fusion: Trends, Solutions, Appl.*, 2009, pp. 1–6.

[27] T. Hirscher, A. Scheel, S. Reuter, and K. Dietmayer, "Multiple extended object tracking using Gaussian processes," in *Proc. 19th Int. Conf. Inf. Fusion*, 2016, pp. 868–875.

[28] N. Wahlström and E. Özkan, "Extended target tracking using Gaussian processes," *IEEE Trans. Signal Process.*, vol. 63, no. 16, pp. 4165–4178, Aug. 2015.

[29] M. Kumru and E. Ozkan, "Three-dimensional extended object tracking and shape learning using Gaussian processes," *IEEE Trans. Aerosp. Electron. Syst.*, vol. 57, no. 5, pp. 2795–2814, Oct. 2021.

[30] K. G. Murty, "An algorithm for ranking all the assignments in order of increasing cost," *Operations Res.*, vol. 16, no. 3, pp. 682–687, 1968. [Online]. Available: http://www.jstor.org/stable/168595

[31] R. A. Singer, "Estimating optimal tracking filter performance for manned maneuvering targets," *IEEE Trans. Aerosp. Electron. Syst.*, vol. AES-6, no. 4, pp. 473–483, Jul. 1970.

[32] J. A. Nelder and R. Mead, "A simplex method for function minimization," *Comput. J.*, vol. 7, pp. 308–313, 1965.

[33] M. Roth, G. Hendeby, and F. Gustafsson, "EKF/UKF maneuvering target tracking using coordinated turn models with polar/Cartesian velocity," in *Proc. 17th Int. Conf. Inf. Fusion*, 2014, pp. 1–8.