Extended Target Tracking Using Gaussian Processes

Niklas Wahlström and Emre Özkan

Linköping University Post Print

Tweet

N.B.: When citing this work, cite the original article.

Extended Target Tracking Using Gaussian Processes

Niklas Wahlström, *Student Member, IEEE,* Emre Özkan, *Member, IEEE,*

Abstract—In this paper, we propose using Gaussian processes to track an extended object or group of objects, that generates multiple measurements at each scan. The shape and the kinematics of the object are simultaneously estimated, and the shape is learned online via a Gaussian process. The proposed algorithm is capable of tracking different objects with different shapes within the same surveillance region. The shape of the object is expressed analytically, with well-defined confidence intervals, which can be used for gating and association. Furthermore, we use an efficient recursive implementation of the algorithm by deriving a state space model in which the Gaussian process regression problem is cast into a state estimation problem.

Index Terms—Extended target tracking, Gaussian processes, star-convex.

I. INTRODUCTION

Target tracking involves estimating the kinematics of an unknown number of objects in a surveillance region based on a set of measurements collected by one or multiple sensors. In the most common formulation of the problem, each object is considered to be a point source, and the measurements are assumed to be generated from the vicinity of the object's center. This assumption will simplify, for example, the computation of the possible association hypothesis between the estimated targets and the available measurements. Thanks to the increasing availability of computational resources, more complex models can now be used for defining object(s), and the inference techniques for such models are drawing considerable interest.

In extended target tracking models, each target is assumed to have an *extent* from which the measurements are generated. The extent of a target can be modeled as a circle, ellipse, rectangle or other simple shapes [1], [2], [3]. Several models have been proposed in the literature for extended object tracking.

A Bayesian approach was first proposed in [1] for elliptical extended targets, where the inverse Wishart distribution is used as a prior for the unknown elliptical target extent. However, in this model, there exists a coupling between the target extent and the target kinematic state, which is restrictive. This issue was later addressed in [2], where an approximate inference method was proposed for using random matrices in extended target tracking. In [4], an approximate measurement update that relies on a variational Bayes approximation was proposed

for the random matrix-based extended target tracking. Two different random matrix-based models were proposed in [5]. The extension of these models to multiple ellipses was presented in [6]. In a recent report [7], a new prediction update for random matrix-based extended target models was proposed. The method proposed in that work focuses on the time update, and the possible rotation of the target extent is taken into account. In [3], rectangular and elliptical extended objects were considered, and an extended Kalman filter was used for inference. The target extent can also be defined as a parametric surface generating measurements [8], [9]. Trackers for the more general class of star-convex shapes were introduced by [8] and further discussed in [10], [11] and [12]. Image-based contour trackers have also been proposed and discussed in the literature. Among the different parameterizations of the contour, B-splines have been the most common. Using Kalman filters for tracking both the shape and position of such contours has been proposed [13], [14], [15]. However, particle filter solutions are preferred to achieve robust trackers [16], [17].

In this article, we propose using Gaussian processes (GPs) to model the boundary of an unknown object. GPs have been widely used by the machine learning, statistics, and signal processing communities for identification, classification, and regression [18] because of their tractable posterior computation and attractive analytical properties. The model proposed herein is flexible enough to represent a large variety of shapes and provides an analytical representation of the objects' extents. The boundaries and the measurement predictions are more precise than rough elliptical approximations. We believe that the assumptions of point targets or elliptical extents are restrictive and that tracking objects with unknown and complex shapes is becoming more important with the increasing accuracy and resolution of the sensors. The ability to learn and track unknown shapes also provides better accuracy and a priori information for the detection algorithms in which the measurements (features of interest) are extracted from raw sensor data, e.g., images. Accurate knowledge of the object's shape, which is summarized into an analytical expression, is also a critical element for target tracking algorithms in which the association between the measurements and the targets has a crucial role on the performance of the tracker. Furthermore, the ability to learn the shapes of the targets can be used for classifying and extracting the attribute information of the targets.

II. TARGET EXTENT MODEL

In conventional tracking methods, targets are considered to be point sources that result in sensor detections. Moreover, in most cases, they are assumed to generate at most one
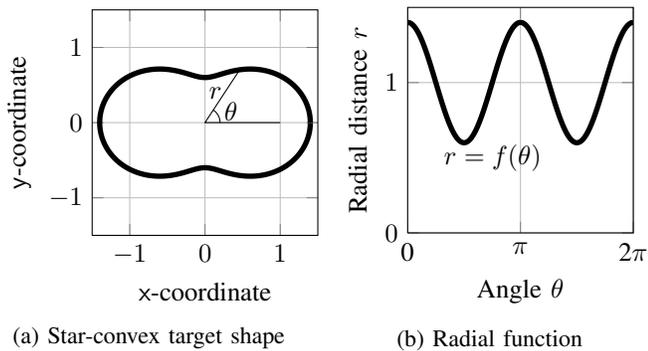
Fig. 1: Description of star-convex shapes using a radial function $r = f(\theta)$.

measurement per scan [19]. In contrast to these traditional assumptions, we will assume that each target can generate multiple measurements from multiple sources on its extent, e.g., multiple reflection points can be detected on the same object by a radar. The object extent can be modeled as a simple geometric primitive, such as a rectangle or an ellipse. In this study, the target extent will be described via *star-convex* shapes. A set $\mathcal{S}(x_k)$ is called star-convex if each line segment from the center to any point is fully contained in $\mathcal{S}(x_k)$. By definition, convex sets are subsets of star-convex sets. The contour of star-convex shapes can be described in polar coordinates with a radial function $r = f(\theta)$ that maps the angle to the radius, which is convenient for representing and learning abstract shapes. An example shape is shown in Fig. 1.

Based on the star-convex description, two different measurement models will be considered in this work

1) *Target contour model:* The measurement equation for the noisy detections originating from the target contour can be written as

$$\mathbf{z}_{k,l} = \mathbf{x}_k^c + \mathbf{p}(\theta_{k,l})f(\theta_{k,l}) + \mathbf{e}_{k,l}, \quad (1)$$

where $\mathbf{x}_k^c$ is the target position at time index k; $\{\mathbf{z}_{k,l}\}_{l=1}^{n_k}$ are the n_k measurements collected at time index k; $\{\theta_{k,l}\}_{l=1}^{n_k}$ denote the angles describing the origin of these measurements on the target contour; $\mathbf{e}_{k,l} \sim \mathcal{N}(0, R)$ represents the zero-mean measurement noise, which is assumed to be Gaussian with covariance R; and $\mathbf{p}(\theta_{k,l})$ is an orientation vector defined as

$$\mathbf{p}(\theta_{k,l}) \triangleq \begin{bmatrix} \cos(\theta_{k,l}) \\ \sin(\theta_{k,l}) \end{bmatrix}. \quad (2)$$

2) *Target surface model:* If the measurements are assumed to originate from the interior of the target region, the model (1) can be extended by considering

$$\mathbf{z}_{k,l} = \mathbf{x}_k^c + s_{k,l}\mathbf{p}(\theta_{k,l})f(\theta_{k,l}) + \mathbf{e}_{k,l}, \quad (3)$$

where the scaling component is $s_{k,l} \in [0, 1]$.

By augmenting a parametrized version of the unknown radial function $r = f(\theta)$ with the target position $\mathbf{x}_k^c$ and its kinematics, in Section V, we will derive a state space model that enables the simultaneous estimation of the target

position and its shape. An orientation state ψ_k will also be added to track orientation changes of the target. The surface model is further described in Section VIII. Additionally, note that by parameterizing star-convex shapes in this manner, the target position $\mathbf{x}_k^c$ and target contour $f(\theta)$ will not be unique. Different combinations of $\mathbf{x}_k^c$ and $f(\theta)$ can provide the same target contour.

Star-convex shapes have been introduced in the context of target tracking by [8] and further elaborated in [10], [11] and [12]. In all these contributions, a Fourier series expansion was used to parametrize the unknown radial function $f(\theta)$. This approach provides great flexibility and is also a standard choice for describing periodic signals. However, in a stochastic setting, this approach has a number of limitations. In this work, we will instead investigate the use of GPs to model the radial function. In contrast to the Fourier series expansion in its basic form, the GP is a probabilistic model that allows the specification of the posterior distribution of the learned function in a natural way. Furthermore, it is defined in the spatial domain rather than in the frequency domain, which enables local learning of the target contour. In other words, while learning the observable parts of the target extent, the uncertainty around the unobserved sections can be maintained, which provides more accurate gates for future observations.

Unfortunately, the GP regression is a batch (off-line) method that requires all the data to be available prior to the inference. To enable the simultaneous estimation of the kinematic state and the extent, we will seek approximations and formulate the GP model as a state space model. By augmenting this state space model with the state space description of the kinematic states (position, orientation and velocities), an extended Kalman filter is used to simultaneously estimate all parameters. Prior to introducing the details of this special inference technique, the standard GP regression and its extension will be explained in the following section.

Throughout this paper, scalars or scalar-valued functions are denoted with non-bold symbols, e.g., θ, vectors or vector-valued functions are denoted with bold symbols, e.g., $\mathbf{z}$, and matrices are denoted with capitalized symbols, e.g., P. Furthermore, Cartesian coordinates are denoted using Sans-serif font, e.g., x and y, to distinguish them from other variables.

III. GAUSSIAN PROCESSES

A Gaussian process (GP) [18] is a stochastic process suitable for modeling spatially correlated measurements. GPs can be considered a distribution over functions. This distribution is uniquely defined with its mean function $\mu(u)$ and covariance function $k(u, u')$ of a function $f(u)$ as

$$\mu(u) = \mathbb{E}[f(u)], \quad (4a)$$

$$k(u, u') = \mathbb{E}[(f(u) - \mu(u))(f(u') - \mu(u'))^\mathsf{T}], \quad (4b)$$

and the Gaussian process is denoted as

$$f(u) \sim \mathcal{GP}\big(\mu(u), k(u, u')\big), \quad (5)$$

where u is the function input.

The GP is a generalization of the multivariate Gaussian probability distribution in the sense that the function values evaluated for a finite number of inputs $u_1, \ldots, u_N$ are normally distributed

$$
\begin{bmatrix} f(u_1) \\ \vdots \\ f(u_N) \end{bmatrix} \sim \mathcal{N}(\boldsymbol{\mu}, K), \quad \text{where} \quad \boldsymbol{\mu} = \begin{bmatrix} \mu(u_1) \\ \vdots \\ \mu(u_N) \end{bmatrix} \tag{6a}
$$

$$
\text{and} \quad K = \begin{bmatrix} k(u_1, u_1) & \cdots & k(u_1, u_N) \\ \vdots & & \vdots \\ k(u_N, u_1) & \cdots & k(u_N, u_N) \end{bmatrix}. \tag{6b}
$$

In this work, our model will be formulated with a zero-valued mean function. The following description will be based on that assumption. However, the formulation can easily be generalized to a non-zero mean function; see [18] for further details.

A. Gaussian process regression

The GP model is primarily used to incorporate training data to learn an unknown function. Consider the following measurement model

$$
z_k = f(u_k) + e_k, \qquad e_k \sim \mathcal{N}(0, R), \tag{7}
$$

where z_k is a noisy measurement of the function $f(\cdot)$ at the training input u_k and e_k is the measurement noise. The objective is to use a set of measurements $\mathbf{z} \triangleq \begin{bmatrix} z_1 & \ldots & z_N \end{bmatrix}^\mathsf{T}$ together with their corresponding inputs $\mathbf{u} \triangleq \begin{bmatrix} u_1 & \ldots & u_N \end{bmatrix}^\mathsf{T}$ to learn the function values $\mathbf{f} \triangleq \begin{bmatrix} f(u_1^\mathbf{f}) & \ldots & f(u_{N^\mathbf{f}}^\mathbf{f}) \end{bmatrix}^\mathsf{T}$ for other test inputs $\mathbf{u}^\mathbf{f} \triangleq \begin{bmatrix} u_1^\mathbf{f} & \ldots & u_N^\mathbf{f} \end{bmatrix}^\mathsf{T}$. In the same manner as in (6), the joint distribution for the measurements $\mathbf{z}$ and the function values $\mathbf{f}$ is

$$
\begin{bmatrix} \mathbf{z} \\ \mathbf{f} \end{bmatrix} \sim \mathcal{N}\left(\mathbf{0}, \begin{bmatrix} K(\mathbf{u}, \mathbf{u}) + I_N \otimes R & K(\mathbf{u}, \mathbf{u}^\mathbf{f}) \\ K(\mathbf{u}^\mathbf{f}, \mathbf{u}) & K(\mathbf{u}^\mathbf{f}, \mathbf{u}^\mathbf{f}) \end{bmatrix} \right), \tag{8}
$$

where $\otimes$ denotes the Kronecker product and

$$
K(\mathbf{u}, \mathbf{u}^\mathbf{f}) = \begin{bmatrix} k(u_1, u_1^\mathbf{f}) & \ldots & k(u_1, u_{N^\mathbf{f}}^\mathbf{f}) \\ \vdots & & \vdots \\ k(u_N, u_1^\mathbf{f}) & \ldots & k(u_N, u_{N^\mathbf{f}}^\mathbf{f}) \end{bmatrix}. \tag{9}
$$

From the joint Gaussian distribution $p(\mathbf{z}, \mathbf{f})$ in (8), the conditional distribution $p(\mathbf{f}|\mathbf{z})$ can easily be computed as

$$
p(\mathbf{f}|\mathbf{z}) = \mathcal{N}(A\mathbf{z}, P), \tag{10a}
$$

where

$$
A = K(\mathbf{u}^\mathbf{f}, \mathbf{u})K_y^{-1}, \tag{10b}
$$

$$
P = K(\mathbf{u}^\mathbf{f}, \mathbf{u}^\mathbf{f}) - K(\mathbf{u}^\mathbf{f}, \mathbf{u})K_y^{-1}K(\mathbf{u}, \mathbf{u}^\mathbf{f}), \tag{10c}
$$

$$
K_y = K(\mathbf{u}, \mathbf{u}) + I_N \otimes R. \tag{10d}
$$

To compute (10b)-(10c), a Cholesky decomposition of K_y is preferred rather than computing its matrix inversion explicitly because it is faster and numerically more stable.

B. Recursive Gaussian process regression

In many applications (for example, target tracking), all the measurements may not be available as a batch, but they might be collected sequentially in time. In such cases, one should aim for recursive solutions for efficient implementation and online inference. In this setting, at each time index k, we are interested in computing the posterior $p(\mathbf{f}|z_{1:k})$ online. For such applications, it is not feasible to use the standard GP regression as presented in the previous section because of the following two reasons:

First, the formulation in (10) is a batch formulation in which all data $z_{1:k}$ are needed to perform inference and cannot be used to update the posterior recursively. Second, the complexity of the regression problem increases cubically with the number of measurements, which is not feasible for an online implementation. We are therefore aiming for an approximate recursive update of the posterior.

In the literature, a few methods have been proposed for recursive GP regression. In [20], the recursive implementation is based on a sequential update of the Cholesky factor of the matrix K_y. In this manner, the full Cholesky decomposition does not have to be recomputed each time a new measurement is received. Hartikainen and Särkkä [21] formulate the GP as a state space model and solve the regression problem with a Kalman filter. This approach only works for one-dimensional inputs, and the measurements are required to be processed in a sequential order with respect to the input dimension. In [22],[23], the GP is approximated with a finite number of basis points, which are updated by further approximations.

In this work, we present a recursive method that resembles the one in [23]. Similar to [22], we explicitly define the recursions as a state space model to which we can apply a Kalman filter. This state space description will be beneficial in extended target tracking because it can be reformulated for augmentation with other state space models, for example, a model that describes the dynamics of target position and target orientation.

We interpret the test inputs $\mathbf{u}^\mathbf{f}$ to be the basis points and $\mathbf{f}$ to be their corresponding outputs. By applying Bayes law consecutively on the posterior $p(\mathbf{f}|z_{1:N})$, we obtain

$$
p(\mathbf{f}|z_{1:N}) \propto p(z_N|\mathbf{f}, z_{1:N-1})p(\mathbf{f}|z_{1:N-1}) \tag{11a}
$$

$$
\propto \cdots \underbrace{p(z_k|\mathbf{f}, z_{1:k-1}) \cdots p(\mathbf{f})}_{\propto p(\mathbf{f}|z_{1:k})}, \tag{11b}
$$

which results in the following recursion

$$
p(\mathbf{f}|z_{1:k}) \propto p(z_k|\mathbf{f}, z_{1:k-1}) \times p(\mathbf{f}|z_{1:k-1}). \tag{12}
$$

$$
\text{posterior} \propto \quad \text{likelihood} \quad \times \quad \text{prior}
$$

We will now approximate $\mathbf{f}$ to be conditionally independent of the past measurements $z_{1:k-1}$, which means that $\mathbf{f}$ will be the sufficient statistic of all the past measurements

$$
p(z_k|\mathbf{f}, z_{1:k-1}) \approx p(z_k|\mathbf{f}). \tag{13}
$$

This approximation would be exact if the input values for $z_{1:k-1}$ were a subset of the input values for $\mathbf{f}$, and it would be a good approximation if the input values for $z_{1:k-1}$ were close to those of $\mathbf{f}$ relative to the characteristic length scale

of the covariance function. In our application, the inputs are angles in the interval $[0, 2\pi]$. Because this interval is bounded, it can be adequately covered by a small number of basis points placed equidistantly within that interval.

As in (8), the measurement z_k and the function values $\mathbf{f}$ are jointly Gaussian

$$\begin{bmatrix} z_k \\ \mathbf{f} \end{bmatrix} \sim \mathcal{N} \left(\mathbf{0}, \begin{bmatrix} k(u_k, u_k) + R & K(u_k, \mathbf{u^f}) \\ K(\mathbf{u^f}, u_k) & K(\mathbf{u^f}, \mathbf{u^f}) \end{bmatrix} \right). \quad (14)$$

Furthermore, we formulate the likelihood and the initial prior in the same manner as (10)

$$p(z_k|\mathbf{f}) = \mathcal{N}\left(z_k;\ H_k^\mathbf{f}\mathbf{f},\ R_k^\mathbf{f}\right), \quad (15a)$$

$$p(\mathbf{f}) = \mathcal{N}\left(0,\ P_0^\mathbf{f}\right), \quad (15b)$$

with

$$H^\mathbf{f}(u_k) = K(u_k, \mathbf{u^f})[K(\mathbf{u^f}, \mathbf{u^f})]^{-1}, \quad (16a)$$

$$R^\mathbf{f}(u_k) = k(u_k, u_k) + R - K(u_k, \mathbf{u^f})[K(\mathbf{u^f}, \mathbf{u^f})]^{-1}K(\mathbf{u^f}, u_k),$$

$$P_0^\mathbf{f} = K(\mathbf{u^f}, \mathbf{u^f}). \quad (16b)$$

By exploiting the structure of this likelihood, the recursive regression can be computed by implementing a Kalman filter on the following state space model

$$\mathbf{x}_{k+1}^\mathbf{f} = \mathbf{x}_k^\mathbf{f}, \quad (17a)$$

$$z_k = H^\mathbf{f}(u_k)\mathbf{x}_k^\mathbf{f} + e_k^\mathbf{f}, \quad e_k^\mathbf{f} \sim \mathcal{N}\left(0,\ R^\mathbf{f}(u_k)\right), \quad (17b)$$

$$\mathbf{x}_0^\mathbf{f} \sim \mathcal{N}\left(0,\ P_0^\mathbf{f}\right), \quad (17c)$$

where $\mathbf{x}_k^\mathbf{f} = \mathbf{f} = \begin{bmatrix} f(u_1^\mathbf{f}) & \dots & f(u_{N^\mathbf{f}}^\mathbf{f}) \end{bmatrix}^\mathsf{T}$ is interpreted as the state.

The state space model not only enables efficient inference but also allows us to design the model for various purposes. First, a dynamical behavior can be added to the extent very easily. In the case where the unknown function changes over time, a process noise term can be added to the dynamics as follows:

$$\mathbf{x}_{k+1}^\mathbf{f} = F^\mathbf{f}\mathbf{x}_k^\mathbf{f} + \mathbf{w}_k, \quad \mathbf{w}_k \sim \mathcal{N}\left(0,\ Q^\mathbf{f}\right), \quad (18)$$

with

$$F^\mathbf{f} = e^{-\alpha T}I, \quad Q^\mathbf{f} = \left(1 - e^{-2\alpha T}\right)K(\mathbf{u^f}, \mathbf{u^f}). \quad (19)$$

The parameter $\alpha \geq 0$ will determine the speed of the dynamics and can be considered a forgetting factor. With $\alpha = 0$, all measurements that have been collected will be of equal importance, and as α increases, less weight is given to older measurements. The choice of dynamics in (19) ensures that the stationary state covariance is $K(\mathbf{u^f}, \mathbf{u^f})$, regardless of the α value, because

$$P = F^\mathbf{f}P(F^\mathbf{f})^\mathsf{T} + Q^\mathbf{f} \quad \Rightarrow \quad P = K(\mathbf{u^f}, \mathbf{u^f}). \quad (20)$$

Furthermore, the state space description will allow us to augment the model with other state space models for joint estimation. This will be presented in Section V, where the kinematic state is augmented with the target extent.

IV. TARGET CONTOUR GP MODEL

As described earlier, our aim is to describe the target contour using a GP model. The input to the model is therefore chosen to be the polar angle $\theta = u$, and the output is the radius $r = y$, as shown in Fig. 1b. The required mean and covariance functions that define the GP model will be described in the following subsections.

A. Mean function

In this work, we will consider a constant but unknown mean function $\mu(\theta) = r$, which is interpreted as being the mean radius of the target contour.

$$f(\theta) \sim \mathcal{GP}\left(r, k(\theta, \theta')\right), \quad \text{where} \quad r \sim \mathcal{N}\left(0,\ \sigma_r^2\right). \quad (21a)$$

By integrating out the parameter r, the same model can be reformulated as a zero mean GP

$$f(\theta) \sim \mathcal{GP}\left(0, k(\theta, \theta') + \sigma_r^2\right). \quad (21b)$$

B. Covariance Function

The covariance function is the crucial component when modeling a GP. This function encodes the assumptions that we make on the functions to be learned. The most common choice is the squared exponential (SE) covariance function [18]

$$k(\theta, \theta') = \sigma_f^2 e^{-\frac{|\theta - \theta'|^2}{2l^2}}, \quad (22)$$

where σ_f^2 is the prior variance of the signal amplitude and l is the length scale of the function that we want to learn. This covariance function provides high correlations for two radial distances $f(\theta)$ and $f(\theta')$ if their corresponding angles θ and θ' are close to each other and less correlation if they are further apart.

To encode the periodicity of $f(\cdot)$ in terms of the angle θ, (22) is slightly modified as

$$k(\theta, \theta') = \sigma_f^2 e^{-\frac{2\sin^2\left(\frac{|\theta - \theta'|}{2}\right)}{l^2}}, \quad (23)$$

which can be derived using a non-linear mapping of the input [24]. With this modification, $f(\theta)$ and $f(\theta + 2\pi)$ will be perfectly correlated as desired because

$$\rho(f(\theta), f(\theta + \pi)) = \frac{k(\theta, \theta + 2\pi)}{\sqrt{k(\theta, \theta)}\sqrt{k(\theta + 2\pi, \theta + 2\pi)}} = 1. \quad (24)$$

Finally, the contribution from the mean function as described in Section IV-A is included in the covariance function, resulting in

$$k_{\text{tot}}(\theta, \theta') = \sigma_f^2 e^{-\frac{2\sin^2\left(\frac{|\theta - \theta'|}{2}\right)}{l^2}} + \sigma_r^2. \quad (25)$$

The final covariance function is shown in Fig. 2. By design, the model function has a periodicity of 2π because $k_{\text{tot}}(\theta + 2\pi, \theta') = k_{\text{tot}}(\theta, \theta')$. Additionally, note that the radius for different angles is always positively correlated, and the correlation increases if the angular distance is shorter. Furthermore, because we treat the radius r as a random variable, this approach will allow us to describe objects of various sizes.
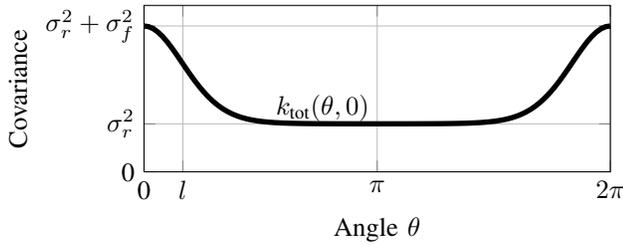
Fig. 2: Periodic covariance function $k_{\text{tot}}(\theta, \theta')$ given in (25). The kernel is used for modeling the covariance of the radial extent.

C. Further extensions

The covariance function (25) will be used throughout the results section in this work. However, to illustrate the flexibility of the GP modeling, some examples will be presented below on how further assumptions on the target shape could be included.

1) Symmetric model: For many targets, there are symmetry assumptions on the shape of the target. Designing the covariance function to have a period of π rather than 2π gives

$$k(\theta, \theta') = \sigma_f^2 e^{-\frac{\sin^2(|\theta - \theta'|)}{2l^2}}. \tag{26}$$

By using this covariance function, only the functions where $f(\theta) = f(\theta + \pi)$ will be learned. Such an assumption will be valid for target shapes that do possess two lines of symmetry (symmetric left-right and back-forward). This would, for example, be valid for shapes S1 and S2 in Fig. 4 but not for shape S3.

2) Conservative model: If we know that the target has a certain predefined radius r_0, such information can be incorporated using a non-zero mean Gaussian prior on $r \sim \mathcal{N}(r_0, \sigma_r^2)$ in (21), giving

$$f(\theta) \sim \mathcal{GP}\big(r_0, k(\theta, \theta') + \sigma_r^2\big). \tag{27}$$

Furthermore, by selecting σ_r and σ_f to be small, the target radius can be kept close to radius r_0, resulting in a conservative and robust version of the algorithm. This property can be desired in high clutter scenarios or when the measurements are coming from the target surface, where the measurements are less informative compared to contour measurements. Moreover, note that to handle a non-zero mean $\mu(\theta) = r_0 \neq 0$, the GP regression presented in Sections III-A and III-A has to be changed accordingly. In particular, this requires performing a non-zero initialization such that $\mathbf{x}_0^{\mathbf{f}} \sim \mathcal{N}(\boldsymbol{\mu}_0^{\mathbf{f}}, P_0)$, where $\boldsymbol{\mu}_0^{\mathbf{f}} = [r_0, \ldots, r_0]^{\mathsf{T}}$.

3) Explicit basis functions: The mean target shape does not necessarily have to be a circle. As a further extension, a set of fixed basis functions $\mathbf{h}(\theta)$ with coefficients $\boldsymbol{\beta}$ can be used to specify the mean function

$$g(\theta) = f(\theta) + \mathbf{h}(\theta)^{\mathsf{T}}\boldsymbol{\beta}, \tag{28a}$$

where

$$f(\theta) \sim \mathcal{GP}\big(0, k(\theta, \theta')\big) \quad \text{and} \quad \boldsymbol{\beta} \sim \mathcal{N}(\mathbf{b}, B). \tag{28b}$$

Such a model indicates that the target shape can be described with a parametric model given by $\mathbf{h}(\theta)^{\mathsf{T}}\boldsymbol{\beta}$, with the residual being modeled with a GP. In a manner similar to that used in Section IV-A, by integrating out the coefficients $\boldsymbol{\beta}$, we obtain

$$g(\theta) \sim \mathcal{GP}\Big(\mathbf{h}(\theta)^{\mathsf{T}}\mathbf{b}, \; k(\theta, \theta') + \mathbf{h}(\theta)^{\mathsf{T}}B\mathbf{h}(\theta')\Big). \tag{28c}$$

V. AUGMENTED STATE SPACE MODEL

In this section, we will derive the augmented state space model, consisting of the dynamic equation, the measurement equation, and the initial state covariance in the form

$$\mathbf{x}_{k+1} = F\mathbf{x}_k + \mathbf{w}_k, \quad \mathbf{w}_k \sim \mathcal{N}(0, Q_k), \tag{29a}$$

$$\mathbf{z}_{k,l} = \mathbf{h}_{k,l}(\mathbf{x}_k) + \mathbf{e}_{k,l}, \quad \mathbf{e}_{k,l} \sim \mathcal{N}(0, R_{k,l}), \tag{29b}$$

$$\mathbf{x}_0 \sim \mathcal{N}(\boldsymbol{\mu}_0, P_0), \tag{29c}$$

for joint estimation of the target extent $\mathbf{x}_k^{\mathbf{f}}$ and the target state $\bar{\mathbf{x}}_k$. We will define the target position $\mathbf{x}_k^c$, the target orientation ψ_k and the optional additional state $\mathbf{x}_k^*$ separately within the target state vector because they are required in the update equations. Consequently, we consider the augmented state vector

$$\mathbf{x}_k \triangleq \begin{bmatrix} \bar{\mathbf{x}}_k^{\mathsf{T}} & (\mathbf{x}_k^{\mathbf{f}})^{\mathsf{T}} \end{bmatrix}^{\mathsf{T}}, \quad \text{where} \tag{30a}$$

$$\bar{\mathbf{x}}_k \triangleq \begin{bmatrix} (\mathbf{x}_k^c)^{\mathsf{T}} & \psi_k & (\mathbf{x}_k^*)^{\mathsf{T}} \end{bmatrix}^{\mathsf{T}}. \tag{30b}$$

The optional additional state $\mathbf{x}_k^*$ denotes the remaining state variables. In this work, it corresponds to the kinematic state of the target (velocity and angular velocity).

A. Measurement model

Each measurement $\mathbf{z}_{k,l}$ is associated with an angle $\theta_{k,l}^G$ that depends on its angular location relative to the target position $\mathbf{x}_k^c$

$$\theta_{k,l}^G(\mathbf{x}_k^c) = \angle(\mathbf{z}_{k,l} - \mathbf{x}_k^c). \tag{31a}$$

This angle can also be described in the local target coordinate frame using the target orientation state ψ_k

$$\theta_{k,l}^L(\mathbf{x}_k^c, \psi_k) = \theta_{k,l}^G(\mathbf{x}_k^c) - \psi_k; \tag{31b}$$

see Fig. 3 for a graphical illustration of this geometry. Note that the angles depend on the unknown target position $\mathbf{x}_k^c$ and the unknown target orientation ψ_k.

The angles can now be used in (1) to describe the relation between the measurement and the state as

$$\mathbf{z}_{k,l} = \mathbf{x}_k^c + \mathbf{p}_{k,l}(\mathbf{x}_k^c)f\Big(\theta_{k,l}^L(\mathbf{x}_k^c, \psi_k)\Big) + \bar{\mathbf{e}}_{k,l}, \tag{32a}$$

where $\bar{\mathbf{e}}_{k,l} \sim \mathcal{N}(0, R)$ is the measurement noise. By combining (2) and (31a), the orientation vector can be expressed as

$$\mathbf{p}_{k,l}(\mathbf{x}_k^c) = \mathbf{p}\Big(\theta_{k,l}^G(\mathbf{x}_k^c)\Big) = \frac{\mathbf{z}_{k,l} - \mathbf{x}_k^c}{\|\mathbf{z}_{k,l} - \mathbf{x}_k^c\|}. \tag{33}$$

Note that the radial function $f(\cdot)$ describes the target extent in the local coordinate frame and the angles $\theta_{k,l}^L(\mathbf{x}_k^c, \psi_k)$ are the input arguments of that function.
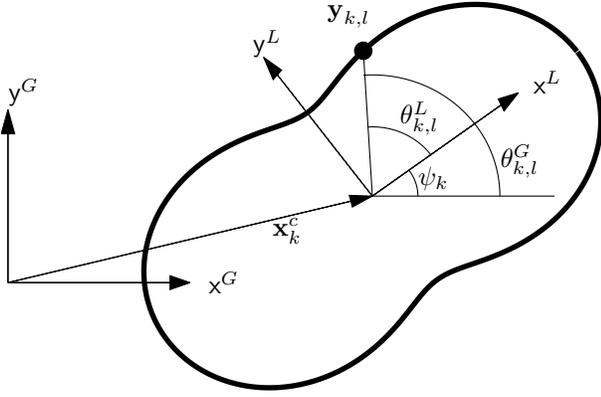
Fig. 3: Each measurement $\mathbf{z}_{k,l}$ is associated with an angle $\theta_{k,l}$ relative to the target center $\mathbf{x}_k^c$. This angle can either be expressed in global coordinates $\theta_{k,l}^G$ or local coordinates $\theta_{k,l}^L$, where $\mathsf{x}^G/\mathsf{y}^G$ and $\mathsf{x}^L/\mathsf{y}^L$ are the basis vectors of the two coordinate systems, respectively, and ψ_k is the target orientation angle.

By using the state space description of the GP as derived in Section III-B, the standard measurement equation can be written as follows:

$$\mathbf{z}_{k,l} = \mathbf{x}_k^c + \mathbf{p}_{k,l}(\mathbf{x}_k^c) \left[H^{\mathbf{f}}\left(\theta_{k,l}^L(\mathbf{x}_k^c, \psi_k) \right) \mathbf{x}_k^{\mathbf{f}} + \mathbf{e}_{k,l}^{\mathbf{f}} \right] + \bar{\mathbf{e}}_{k,l}$$

$$= \underbrace{\mathbf{x}_k^c + \tilde{H}_l(\mathbf{x}_k^c, \psi_k)\mathbf{x}_k^{\mathbf{f}}}_{=\mathbf{h}_{k,l}(\mathbf{x}_k)} + \underbrace{\mathbf{p}_{k,l}(\mathbf{x}_k^c)e_{k,l}^{\mathbf{f}} + \bar{\mathbf{e}}_{k,l}}_{=\mathbf{e}_{k,l}} \quad (34a)$$

$$= \mathbf{h}_{k,l}(\mathbf{x}_k) + \mathbf{e}_{k,l}, \qquad \mathbf{e}_{k,l} \sim \mathcal{N}\left(0, \ R_{k,l}\right), \quad (34b)$$

where the measurement model and the new measurement noise covariance are given by

$$\tilde{H}_l(\mathbf{x}_k^c, \psi_k) = \mathbf{p}_{k,l}(\mathbf{x}_k^c) H^{\mathbf{f}}\left(\theta_{k,l}^L(\mathbf{x}_k^c, \psi_k) \right), \quad (34c)$$

$$\mathbf{h}_{k,l}(\mathbf{x}_k) = \mathbf{x}_k^c + \tilde{H}_l(\mathbf{x}_k^c, \psi_k)\mathbf{x}_k^{\mathbf{f}}, \quad (34d)$$

$$R_{k,l} = \mathbf{p}_{k,l} R_{k,l}^{\mathbf{f}} \mathbf{p}_{k,l}^{\mathsf{T}} + R, \quad (34e)$$

$$\mathbf{p}_{k,l} = \mathbf{p}_{k,l}(\mathbf{x}_k^c), \quad R_{k,l}^{\mathbf{f}} = R^{\mathbf{f}}\left(\theta_{k,l}^L(\mathbf{x}_k^c, \psi_k) \right). \quad (34f)$$

Note that since the measurement noise is already included in (34e), the measurement noise variance in (16b) can be omitted.

B. Motion model

The target state $\bar{\mathbf{x}}_k = [(\mathbf{x}_k^c)^{\mathsf{T}}, \ \psi_k, \ (\mathbf{x}_k^*)^{\mathsf{T}}]^{\mathsf{T}}$ is described with a linear state space model

$$\bar{\mathbf{x}}_{k+1} = \bar{F}\bar{\mathbf{x}}_k + \bar{\mathbf{w}}_k, \quad \bar{\mathbf{w}}_k \sim \mathcal{N}\left(0, \ \bar{Q}\right), \quad (35a)$$

$$\bar{\mathbf{x}}_0 \sim \mathcal{N}\left(\bar{\boldsymbol{\mu}}_0, \ \bar{P}_0\right), \quad (35b)$$

as commonly performed in target tracking applications.

Together with the dynamical description of the target extent in (18), we construct an augmented description of the dynamics.

$$\mathbf{x}_{k+1} = F\mathbf{x}_k + \mathbf{w}_k, \quad \mathbf{w}_k \sim \mathcal{N}\left(0, \ Q\right), \quad (36a)$$

$$\mathbf{x}_0 = \mathcal{N}\left(\boldsymbol{\mu}_0, \ P_0\right), \quad (36b)$$

where

$$\mathbf{x}_k = \begin{bmatrix} \bar{\mathbf{x}}_k \\ \mathbf{x}_k^{\mathbf{f}} \end{bmatrix}, \quad F = \begin{bmatrix} \bar{F} & 0 \\ 0 & F^{\mathbf{f}} \end{bmatrix}, \quad Q = \begin{bmatrix} \bar{Q} & 0 \\ 0 & Q^{\mathbf{f}} \end{bmatrix}, \quad (37a)$$

$$\boldsymbol{\mu}_0 = \begin{bmatrix} \bar{\boldsymbol{\mu}}_0 \\ 0 \end{bmatrix}, \quad P_0 = \begin{bmatrix} \bar{P}_0 & 0 \\ 0 & P_0^{\mathbf{f}} \end{bmatrix}, \quad (37b)$$

where $P_0^{\mathbf{f}}$ is given by (16b), and $F^{\mathbf{f}}$ and $Q^{\mathbf{f}}$ are given by (19). The matrices $\bar{F}$ and $\bar{Q}$ are given later in Section IX-B1.

C. Discussion

In [23], further augmenting the state with hyper-parameters is proposed. Such an approach may be suitable for a vanilla problem because the hyper-parameter tuning of GP models primarily requires batch processing of the measurements and multiple iterations. Unfortunately, in a problem such as online target tracking, neither processing the batch data nor performing iterations is feasible. Furthermore, the uncertainties in a target tracking problem arise from many sources, such as target dynamics model mismatch, false alarms, missed detections, and clutter measurements. Moreover, in ETT, the centroid of the target extent is also unknown and needs to be estimated. Hence, under these uncertainties, we believe that it is more feasible to assume an expected size and length scale for a target rather than online tuning. Later, in the results section, it is shown that both in simulations and real data, targets of different size can be tracked using the same set of hyper-parameters.

VI. INFERENCE

An efficient implementation is essential in many target tracking applications. The state space model (29) derived in the previous section allows us to utilize the standard inference techniques to compute the posterior distribution of the target state vector. In this work, we will use an extended Kalman filter (EKF). To recursively update the posterior, we will construct an augmented model in which all measurements $\{\mathbf{z}_{k,l}\}_{l=1}^{n_k}$ within one scan are augmented

$$\mathbf{z}_k = \begin{bmatrix} \mathbf{z}_{k,1}^{\mathsf{T}}, & \cdots, & \mathbf{z}_{k,n_k}^{\mathsf{T}} \end{bmatrix}^{\mathsf{T}}, \quad (38a)$$

$$R_k = \text{diag}[R_{k,1}, \cdots, R_{k,n_k}], \quad (38b)$$

$$\mathbf{h}_k(\mathbf{x}_k) = \begin{bmatrix} \mathbf{h}_{k,1}(\mathbf{x}_k)^{\mathsf{T}}, & \cdots, & \mathbf{h}_{k,n_k}(\mathbf{x}_k)^{\mathsf{T}} \end{bmatrix}^{\mathsf{T}}. \quad (38c)$$

The corresponding state space description is given as follows:

$$\mathbf{x}_{k+1} = F\mathbf{x}_k + \mathbf{w}_k, \quad \mathbf{w}_k \sim \mathcal{N}\left(0, \ Q\right), \quad (39a)$$

$$\mathbf{z}_k = \mathbf{h}_k(\mathbf{x}_k) + \mathbf{e}_k, \quad \mathbf{e}_k \sim \mathcal{N}\left(0, \ R_k\right), \quad (39b)$$

$$\mathbf{x}_0 \sim \mathcal{N}\left(\boldsymbol{\mu}_0, \ P_0\right). \quad (39c)$$

An estimate $\hat{\mathbf{x}}_k$ can now be computed using a nonlinear filtering technique. In this work we have used an extended Kalman filter, see Appendix A for the required recursions. Note that these recursions require a gradient of the measurement function $\frac{d\mathbf{h}(\mathbf{x}_k)}{d\mathbf{x}_k}$, which can be computed analytically; see Appendix B.

VII. PREDICTIVE LIKELIHOOD AND GATING

In the presence of clutter and/or multiple targets, a gating step could optionally be included to reject the measurements that are unlikely to originate from the target. By using the GP model, the predictive likelihood and gates for the predicted measurements can be computed by using the standard Kalman filter equations [25, Chapter 6.3]. We start by computing the predictive likelihood, which is available in analytical form

$$\mathbf{z}_{k,l} \sim p(\mathbf{z}_{k,l}|\mathbf{z}_{1:k-1}) = \mathcal{N}\left(\hat{\mathbf{z}}_{k|k-1,l}, \ S_{k,l}\right), \qquad (40a)$$

where

$$S_{k,l} = H_{k,l}P_{k|k-1}H_{k,l}^{\mathsf{T}} + R_{k,l}, \qquad (40b)$$

$$H_{k,l} = \frac{d}{d\mathbf{x}_k}\mathbf{h}_{k,l}(\mathbf{x}_k)|_{\mathbf{x}_k = \hat{\mathbf{x}}_{k|k-1}}, \qquad (40c)$$

$$\hat{\mathbf{z}}_{k|k-1,l} = \mathbf{h}_{k,l}(\hat{\mathbf{x}}_{k|k-1}), \qquad (40d)$$

where $\mathbf{h}_{k,l}$ and $R_{k,l}$ are defined in (34d) and (34e). The gating is performed by comparing the statistics of the residual vector, $\tilde{\mathbf{z}}_{1:k-1,l} \triangleq \mathbf{z}_{k,l} - \hat{\mathbf{z}}_{1:k-1,l}$, with a threshold ρ,

$$\tilde{\mathbf{z}}_{1:k-1,l}^{\mathsf{T}} S_{k,l} \tilde{\mathbf{z}}_{1:k-1,l} \gtrless \rho. \qquad (41)$$

The analytical expression of the target extent provides gates for future measurements that incorporate target shape information.

VIII. SURFACE MODEL USING SCALING PARAMETER

The presented model can also be extended to address measurements that originate from an interior point of a target extent rather than the contour. The measurements originating inside the target boundary are modeled by including a scaling parameter $s_{k,l}$ for each measurement, similar to [10],[26]. The measurement equation (34a) can therefore be modified as

$$\mathbf{z}_{k,l} = \mathbf{x}_k^c + s_{k,l}\tilde{H}_l(\mathbf{x}_k^c, \psi_k)\mathbf{x}_k^{\mathbf{f}}, \quad \mathbf{e}_{k,l} \sim \mathcal{N}\left(0, \ R_{k,l}\right), \quad (42)$$

where $s_{k,l}$ is a random variable on the interval [0,1]. With the assumption that the measurement source is uniformly distributed over the star-convex region, Baum and Hanebeck [12] have shown that the squared scaling factor is uniformly distributed

$$s_{k,l}^2 \sim U[0, 1]. \qquad (43)$$

The mean and variance of $s_{k,l}$ can be computed analytically

$$\mu_s = \mathbb{E}(s_{k,l}) = \frac{2}{3}, \qquad (44a)$$

$$\sigma_s^2 = \text{Var}(s_{k,l}) = \mathbb{E}(s_{k,l}^2) - (\mathbb{E}(s_{k,l}))^2 = \frac{1}{18}. \qquad (44b)$$

To be able to use a Kalman filter for inference, the scaling factor is approximated by a Gaussian random variable that has the same first two moments,

$$s_{k,l} \overset{\text{approx.}}{\sim} \mathcal{N}\left(\mu_s, \ \sigma_s^2\right). \qquad (45)$$

The measurement model can now be modified as

$$\mathbf{z}_{k,l} = \underbrace{\mathbf{x}_k^c + \mu_s\tilde{H}_l(\mathbf{x}_k^c, \psi_k)\mathbf{x}_k^{\mathbf{f}}}_{=\tilde{\mathbf{h}}_l(\mathbf{x}_k)} + \underbrace{(s_{k,l} - \mu_s)\tilde{H}_l(\mathbf{x}_k^c, \psi_k)\mathbf{x}_k^{\mathbf{f}} + \mathbf{e}_{k,l}}_{=\tilde{\mathbf{e}}_k}$$

$$= \tilde{\mathbf{h}}_l(\mathbf{x}_k) + \tilde{\mathbf{e}}_{k,l}, \qquad \mathbf{e}_{k,l} \sim \mathcal{N}\left(0, \ \tilde{R}_{k,l}\right), \qquad (46a)$$

where the measurement model and the measurement noise covariance are given by

$$\tilde{\mathbf{h}}_l(\mathbf{x}_k) = \mathbf{x}_k^c + \mu_s\tilde{H}_l(\mathbf{x}_k^c, \psi_k)\mathbf{x}_k^{\mathbf{f}}, \qquad (47a)$$

$$\tilde{R}_{k,l} = R_{k,l} + \sigma_s^2\tilde{H}_{k,l}\mathbf{x}_k^{\mathbf{f}}(\mathbf{x}_k^{\mathbf{f}})^{\mathsf{T}}\tilde{H}_{k,l}^{\mathsf{T}}, \qquad (47b)$$

$$\tilde{H}_{k,l} = \tilde{H}_l(\mathbf{x}_k^c, \psi_k). \qquad (47c)$$

By substituting the measurement function (34d) with (46a), the algorithm can be modified to track extended targets where the measurements are coming from the surface rather than the contour. In addition, note that (34d) will be equal to (46a) for the following choice of parameters, $\mu_s = 1$ and $\sigma_s^2 = 0$.

IX. RESULTS

In this section, we evaluate the performance of the proposed method and compare it with relevant extended target models in the literature. The evaluation is performed through both simulations and real data experiments, and the corresponding results are reported in Section IX-B and Section IX-C, respectively. For the scenarios in which the measurements originate from the target contour, the model described in Section V has been used. For the scenarios in which the measurements originate from the target surface, the extended model introduced in Section VIII has been used.

A. Alternative models

1) Random matrix model: The proposed algorithm is compared with a standard random matrix-based extended target tracker [2], denoted here as RM. RM approximates the posterior distribution of the kinematic state $\bar{\mathbf{x}}_k \in \mathbb{R}^{n_x}$ and the target extent by using the normal inverse Wishart distribution as follows:

$$p(\bar{\mathbf{x}}_k, X_k|\mathbf{z}_{1:k}) \approx p(\bar{\mathbf{x}}_k|\mathbf{z}_{1:k})p(X_k|\mathbf{z}_{1:k}) \qquad (48)$$
$$= \mathcal{N}(\bar{\mathbf{x}}_k; \bar{\mathbf{m}}_{k|k}, \bar{P}_{k|k}) \times \mathcal{IW}(X_k; \nu_{k|k}, V_{k|k}),$$

where $\bar{\mathbf{m}}_{k|k}, \bar{P}_{k|k}$ are the mean vector and the covariance matrix of the estimated kinematic state at time index k. X_k is the symmetric positive definite matrix, which represents the elliptical extent, and $\mathcal{IW}(X; \nu, V)$ denotes an inverse Wishart distribution defined over the matrix $X \in \mathbb{S}_{++}$ with scalar degrees of freedom ν and parameter matrix $V \in \mathbb{S}_{++}$; see [27, Definition 3.4.1]. The corresponding measurement likelihood for the measurements $\mathbf{z}_{k,l} \in \mathbb{R}^{n_y}$ for a linear model is

$$p(\mathbf{z}_{k,l}|\bar{\mathbf{x}}_k, X_k) = \mathcal{N}(\mathbf{z}_{k,l}; C_k\bar{\mathbf{x}}_k, zX_k + R), \qquad (49)$$

where R is the measurement noise covariance matrix and C_k is the $n_x \times n_y$ measurement matrix. The scaling factor z is used in this work to account for the difference between the assumed normal distribution of the measurement sources and the actual uniform distribution over the target region or the target contour.

In the experiments, we employ an exponential forgetting factor in the time update of the sufficient statistics $\nu_{k|k}$ and $V_{k|k}$, which is shown to provide the maximum entropy distribution for the prediction when the transition density for the target extent is unknown but the change in the prediction

density is upper bounded by a Kullback Leibler distance (see [28, Theorem 1]). This will help the elliptical model to adapt itself for possible orientation changes in the examples. The time update for the sufficient statistics of the target extent is performed as follows.

$$\nu_{k|k-1} = \lambda \nu_{k-1|k-1}, \tag{50a}$$
$$V_{k|k-1} = \lambda V_{k-1|k-1}, \tag{50b}$$

where λ is the forgetting factor.

2) Random hypersurface model using Fourier series: The proposed model is also compared with a star-convex model based on a Fourier series expansion of the radial function [8], which was previously described in Section II. Here, the model will be denoted as RHF. An implementation that is available online[1] has been used in the simulations.

B. Simulations

Several simulations have been performed to evaluate the performance of the proposed model in comparison with the RM and RHF models introduced above. In Section IX-B1, moving targets or different shapes are considered, where we compare the proposed model with the RM model, and in Section IX-B2, stationary targets of different sizes are considered, where the proposed model is compared with the RHF model. All simulation experiments were performed 100 times with different realizations of the measurement noise and measurement origin at each simulation. The presented numbers are the average of these 100 Monte Carlo runs.

1) Moving target: We test the algorithm on moving objects with different shapes. A rectangular (S1), a cross-shaped (S2) and a triangular (S3) object are simulated to generate measurements. A trajectory, which is a combination of linear paths and turns, is generated. The objects first move on a linear path then make a turn and again follow a linear path, always with a velocity of $0.1\,\mathrm{m\,s^{-1}}$. Note that the constant velocity assumption is no longer valid during the turns, but the algorithm is required to be robust enough to handle such model mismatches, which are encountered in most tracking applications. Furthermore, the objects make a rotation during the turns, which has to be tracked by the algorithm. The same set of parameters is used for all different objects to illustrate the robustness and flexibility of the GP model.

A constant velocity model is used for both the position and the orientation of moving targets. However, other dynamics, also non-linear, could be considered as well. By using the well-known time-discrete version of the constant velocity model, the state dynamics equation (35a) will be

$$\bar{F} = \begin{bmatrix} 1 & T \\ 0 & 1 \end{bmatrix} \otimes I_3, \ \bar{Q} = \begin{bmatrix} \frac{T^3}{3} & \frac{T^2}{2} \\ \frac{T^2}{2} & T \end{bmatrix} \otimes \begin{bmatrix} \sigma_q^2 & 0 & 0 \\ 0 & \sigma_q^2 & 0 \\ 0 & 0 & \sigma_{q^\psi}^2 \end{bmatrix},$$

where σ_q^2 and $\sigma_{q^\psi}^2$ are the process noise variances for position and angle, respectively. The process noise standard deviations $\sigma_q = 0.01$ and $\sigma_{q^\psi} = 0.001$ have been used for position and angle, respectively, and $\alpha = 0.0001$ has been used for the

[1]www.cloudrunner.eu/algorithm/12/random-hypersurface-model/version/2/

Measurement origin	Target contour			Target surface		
Target shape	S1	S2	S3	S1	S2	S3
Example result (Fig.)	4a	4c	4e	4b	4d	4f
Proposed model	0.13	0.12	0.71	0.76	0.88	1.40
Random matrix (RM)	0.33	0.31	0.76	0.25	0.28	0.28

TABLE I: Root-mean-square error [m] of the target position in the simulated scenario using different shapes and target extent models. The numbers are averaged over 100 MC runs.

target extent dynamics. For the proposed model, the hyper-parameters of the Gaussian process have been set to $\sigma_r = 2$, $\sigma_f = 2$ and $l = \pi/4$; the measurement noise variance has been set to $R = 0.1^2 I_2$; and the sampling time has been set to $T = 1$.

The performance of the target extent estimation is evaluated based on the Intersection-Over-Union (IOU) measure. This is a similarity measure used in, for example, computer vision to compare object shapes for similarity [29]. The IOU measure has also been used for evaluation in an extended target tracking context [3]. If the true target covers region R_0 and the estimated target covers region $\hat{R}$, then the IOU measure is defined as the ratio of the areas for the intersection and the union of these two regions

$$\mathrm{IOU}(\hat{R}, R_0) = \frac{\mathrm{area}(\hat{R} \cap R_0)}{\mathrm{area}(\hat{R} \cup R_0)}. \tag{51}$$

By construction, $\mathrm{IOU}(\hat{R}, R_0) \in [0, 1]$, where the value 1 corresponds to a perfect match between these two regions and the value 0 corresponds to the worst possible match, i.e., the regions are not even overlapping.

The performance of the target position is evaluated using the root-mean-square error

$$\mathrm{RMSE}(\hat{\mathbf{x}}^c, \mathbf{x}^0) = \sqrt{\frac{1}{N} \sum_{k=1}^{N} (\hat{\mathbf{x}}_k^c - \mathbf{x}_{0,k}^c)^2}, \tag{52}$$

where $\mathbf{x}_{0,k}^c$ is the true position at time instance k.

For the RM model, we use the scaling factor $z = 1/4$ in scenarios where the measurements originate from the target surface and $z = 1/2$ if they originate from the target contour to obtain an unbiased estimate of the target extent; see [30] for further details.

The number of measurements at each scan is Poisson distributed. An average of 10 measurements are generated in each scan. In the first set of simulations, the measurements are uniformly distributed over the target contour, and in the second set, the measurements are uniformly distributed over the target surface. The RMSE values for the position estimates are presented in Table I, and the IOU values of the extent estimates are presented in Table II. In Fig. 4, an example of one MC run for each of the setups is presented. The measurements from a few scans are also illustrated.

As shown in Table II, the proposed model is better at estimating the target extent than the RM model. This result is expected because the RM model is restricted to elliptical shapes, whereas the proposed model can handle any star-convex shape. If the measurements originate from the target

(a) Target shape: S1. Measurement origin: Target contour

(b) Target shape: S1. Measurement origin: Target surface

(c) Target shape: S2. Measurement origin: Target contour

(d) Target shape: S2. Measurement origin: Target surface

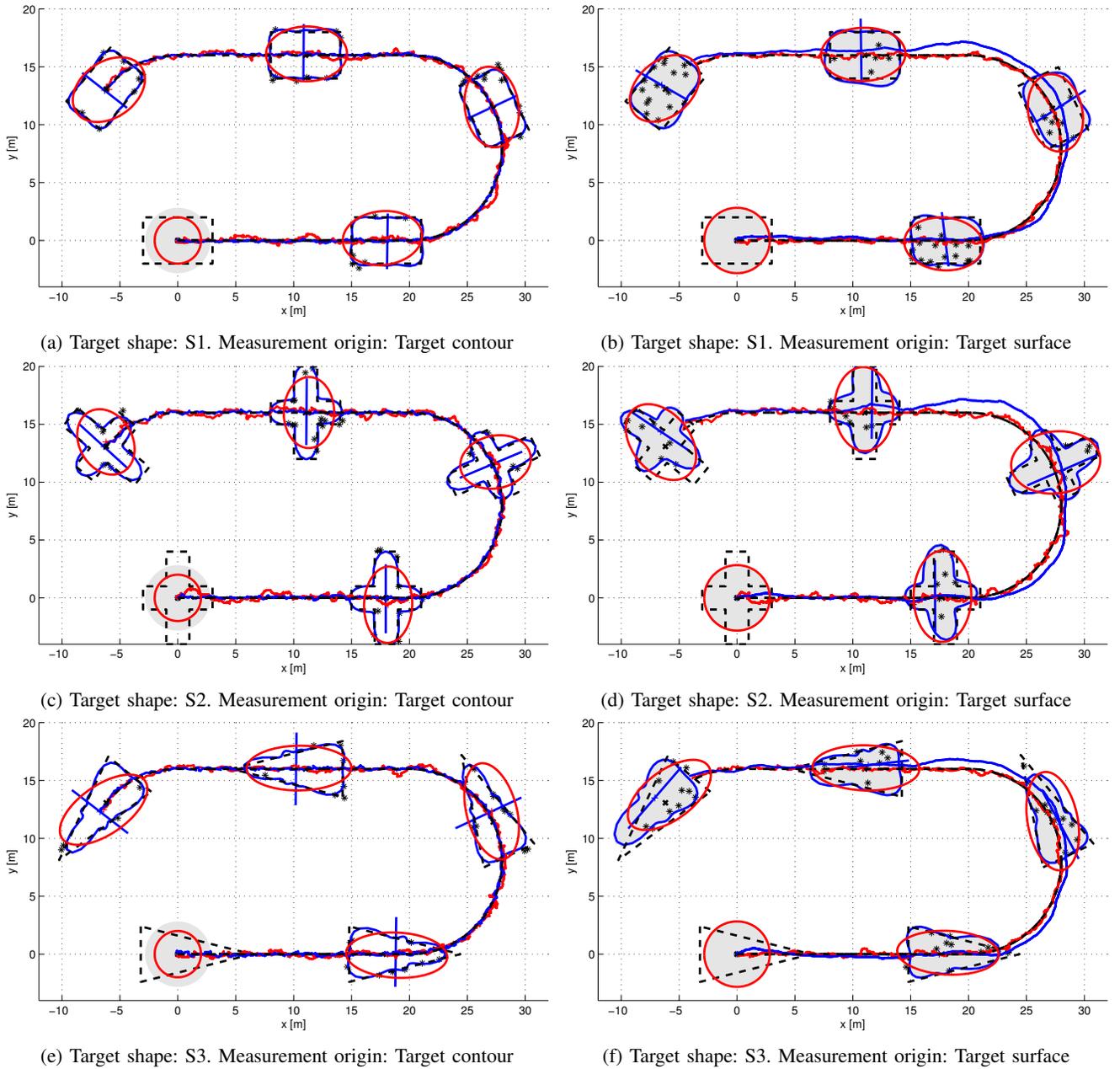(e) Target shape: S3. Measurement origin: Target contour

(f) Target shape: S3. Measurement origin: Target surface

Fig. 4: Example results for tracking different moving targets where the measurements originate from the target contour in the left column (a,c,e) and from the target surface in the right column (b,d,f). True target (dashed black line) is compared with the proposed method (blue contour and with confidence region in gray and blue trajectory) and the RM model (red ellipse and trajectory). The blue line co-centered with the target represents the estimated orientation, and the measurements are depicted as black stars. Five snapshots out of the 500 scans are presented.

Measurement origin	Target contour			Target surface		
Target shape	S1	S2	S3	S1	S2	S3
Example result (Fig.)	4a	4c	4e	4b	4d	4f
Proposed model	0.93	0.86	0.86	0.84	0.73	0.64
Random matrix (RM)	0.79	0.57	0.62	0.80	0.61	0.63

TABLE II: The mean value of Intersection-Over-Union (IOU) between the true and the estimated target regions in the simulated scenario using different shapes and target extent models. The numbers are averaged over 100 MC runs.

surface, the improvement in IOU is rather small. This is particularly the case for shapes S1 and S3, which can be fairly well approximated with an ellipse. However, for shape S2, the improvement is more significant.

If the measurements originate from the target contour, the proposed model outperforms the RM model in estimating both the target extent and target position. The structure of the measurements in this setting can be utilized by the proposed model, which is not the case for the RM model. However, if the measurements originate from the target surface (see right
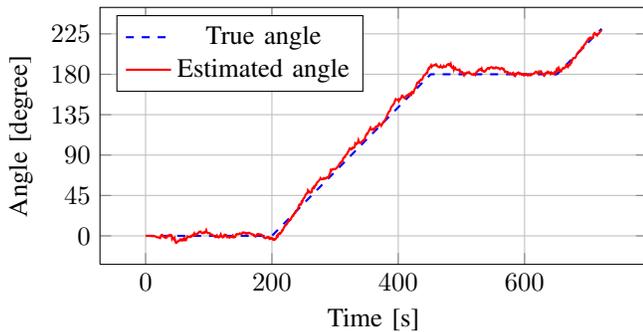
Fig. 5: Target orientation estimate for the scenario illustrated in Fig. 4f.

Target size (width)	0.75 m	3 m	12 m
Example result	Fig. 6a/6d	Fig. 6b/6e	Fig. 6c/6f
Proposed model	0.39	0.66	0.52
Fourier coef. model (RHF)	0.09	0.68	0.25
Fourier coef. model (RHFb)	0.15	0.54	0.36

TABLE III: The Intersection-Over-Union (IOU) between the true and the estimated target regions in the simulated scenario using different targets sizes and extended target models. The numbers are averaged over 100 MC runs.

part of Table II), the RM model performs better in position estimation. This can also be observed in Figs. 4b, 4d and 4f, where the position estimates of the proposed method do not match the true trajectory during the turns. The main reason for the differences in the performances is that there is less structured information that can be exploited by the GP model in the surface measurements than in the contour measurements. The RM model uses the mean of the measurements when updating the target position estimates. In scenarios where the measurement mean contains information about the target centroid, the RM method will have an advantage. Another factor is that the target center is not uniquely defined for a star-convex shape, and this results in a bias in the centroid estimates of the GP model. Therefore, the IOU is a more relevant performance measure because it is invariant to the target center. However, if the target is symmetric, the covariance function presented in Section IV-C1 can be used.

In Fig. 5, the orientation estimate is shown for the target in Fig. 4f. The proposed model is able to estimate not only the target position and extent, but also the target orientation accurately. Such an estimate is not provided by the RM model, and therefore, the proposed model provides a more detailed description of the target trajectory than the RM model.

2) Stationary target: In total, 100 measurements from the surface of a stationary cross-shaped target have been generated. The target shape is the same as the one implemented in the code for the RHF model; see Section IX-A2. To evaluate the ability of the proposed method to estimate targets of different sizes, three different sizes of the target shape have been considered. The tuning parameters for the RHF model, as given in the code, have been tailored for medium-sized targets. For the proposed model, the same hyper-parameters of the Gaussian process have been chosen as in the previous simulation. In Fig. 6a-6c, the results for the three target sizes are presented.

Table III presents the IOU averaged over 100 MC runs. The results indicate that the RHF model performs slightly better for the medium-sized target presented in Fig. 6b. However, the low initial covariance of the Fourier coefficients makes the RHF model unable to estimate the shapes of other target sizes, as shown in Figs. 6a and 6c.

In Figs. 6d-6f, a larger initial covariance has been used to decrease the impact of the prior on the performance for the RHF model, denoted here as RHFb. However, no stable estimates are achieved, as shown in Figs. 6d-6f. Consequently, the authors were not able to find a set of tuning parameters for the RHF model that worked well for data from all these three targets, whereas the proposed model estimates reasonable target shapes in all three of these cases.

In fact, the estimated radius for the RHF model is occasionally negative. The same behavior has also been observed and reported in [10], where inequality constraints on these parameters were suggested. Although this can also theoretically occur for the proposed model, the simulations indicate that this occurs considerably less frequently for the proposed model than for the RHF model.

3) Computation time: Because the proposed solution is implemented with a standard extended Kalman filter, the computational demand is fairly low. The state dimension for 50 basis points is $\dim(\mathbf{x}_k) = \dim(\mathbf{x}_k^c) + \dim(\psi_k) + \dim(\mathbf{x}_k^*) + \dim(\mathbf{x}^\mathbf{f}) = 2 + 1 + 3 + 50 = 56$, and the analytical expressions for the derivatives $\frac{\partial \mathbf{h}(\mathbf{x})}{\partial \mathbf{x}}$ are available. The computational demand does not increase over time because the algorithm is fully recursive. All simulations are run in Matlab(R) R2013b on a standard laptop with an Intel(R) Core(TM) i5-2520M 2.50 GHz platform with 8 GB of RAM. One measurement update and one time update require 2.3 ms for the proposed method, 0.5 ms for RM, and 3.5 ms for the RHF model on average.

C. Real data experiments

In this section, we aim to illustrate the capabilities of the algorithm on real data. We will use a data set presented in [31], [32], where only a single type of object is considered. In this data set, a laser range sensor is used to collect data from multiple objects of different sizes and shapes. The sensor measured the range every $0.5°$ over a $180°$ surveillance region. The sensor is assumed to be located at the origin of a hemisphere, and every detection closer than 15 m is converted to Cartesian positions. The sampling time for a full $180°$ scan is 0.2 s. From the data set, three scenarios have been extracted, and the results are presented in Sections IX-C1 and IX-C2. In all scenarios, the hyper-parameters of the Gaussian process have been set to $\sigma_r = 1$, $\sigma_f = 1$ and $l = \pi/8$, and the measurement noise variance has been set to $R = 0.1 I_2$. The process noise standard deviations $\sigma_q = 2.5$ and $\sigma_{q^\psi} = 0.5$ have been used for position and angle, respectively, as has $\alpha = 0.001$ for the target extent dynamics.

1) Miscellaneous object tracking: In Scenarios 1 and 2, we consider multiple maneuvering objects of different shapes

(a) Target width: 0.75 m (b) Target width: 3 m (c) Target width: 12 m

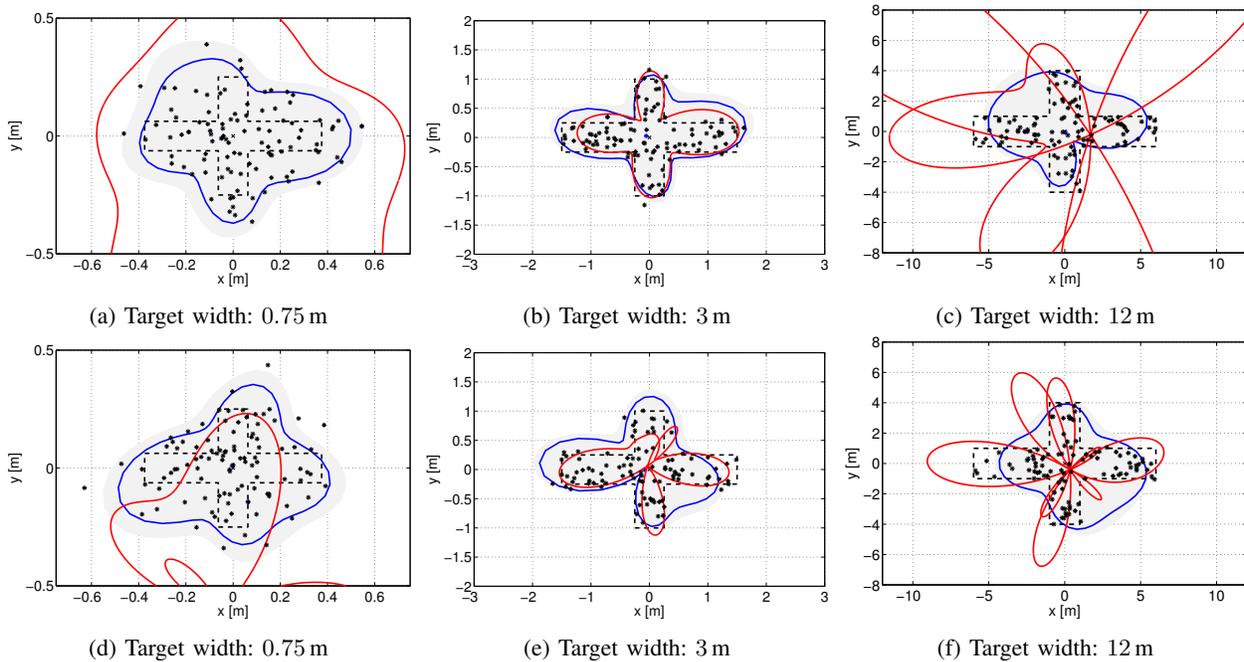(d) Target width: 0.75 m (e) Target width: 3 m (f) Target width: 12 m

Fig. 6: Example results for estimating the extent of stationary targets of different sizes. True target (dashed black line) is compared with the proposed method (blue line and with confidence region in gray) and the RHF model (red line). The measurements are indicated with black stars. In Figures a-c, the same initial standard deviation for the Fourier coefficients in the RHF model has been chosen, as provided in the code available online, whereas in Figures d-f, a four-fold greater initial standard deviation has been chosen.

(cars, bicycles and pedestrians), which enter and exit a surveillance region. The objects of interest are significantly different from each other. A snapshot of measurements originating from the objects are shown in Fig. 7. Note that the problem addressed in this work is tracking a single extended target given the associated measurements. Here, we combined the proposed filter with a simple multi-target tracking approach for illustrating the output of the algorithm for different targets within the same scene without requiring any extra target-dependent parameter tuning.

We use a simple target tracking algorithm, where every measurement is associated with the nearest target if the measurement falls into the gate of the target. We used the gating strategy described in Section VII, where the gate threshold is chosen to ensure that 99 percent of the target-originated measurements fall into the target's gate. The remaining measurements (which are not associated with any of the existing tracks) are simply clustered according to their mutual distances, and a new track is generated from each cluster.

In Figures 8 and 9, we present two different scenarios. In these figures, multiple snapshots of the measurements and the output of the GP tracker are superimposed together. The centroid of the object, its orientation, the estimated target extent and the confidence region of the extent are plotted as the output of the tracker.

In Scenario 1, four targets move in the scene. Multiple snapshots from the scenario are split into two sub-figures and plotted in Fig. 8. In this scenario, one car is moving from left
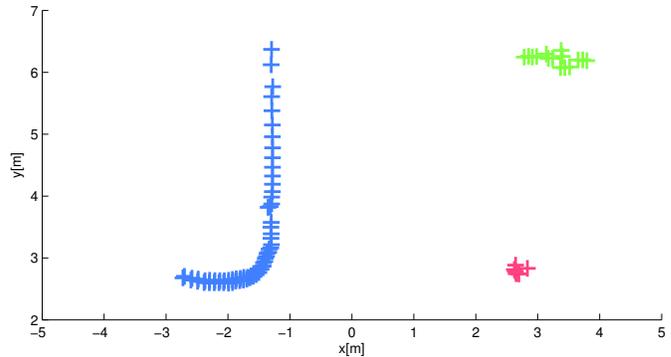

Fig. 7: Laser range sensor measurements corresponding to a car (on the left), a bicycle (top-right corner) and a pedestrian.

to right, one bicycle is moving in the opposite direction, one pedestrian (at the top) is walking in the top to bottom direction, and another pedestrian makes an L-shaped move close to the origin of the scene. The snapshots corresponding to frames {1, 6, 16, 27} are plotted in Fig. 8a, and those corresponding to frames {28, 42, 52} are plotted in Fig. 8b. The snapshots are chosen for the sake of clearer illustration, where the initial and final stages of the tracks are shown and the overlaps between the snapshots are kept to a minimum. In the scenario, the car and the bicycle pass each other while moving in opposite directions. During the transition, the bicycle is fully occluded by the car in multiple consecutive scans, as it blocks the line of sight of the range sensor, thereby preventing any possible detection. The pedestrian moving toward the bottom of the

(a) Scenario 1: Part I
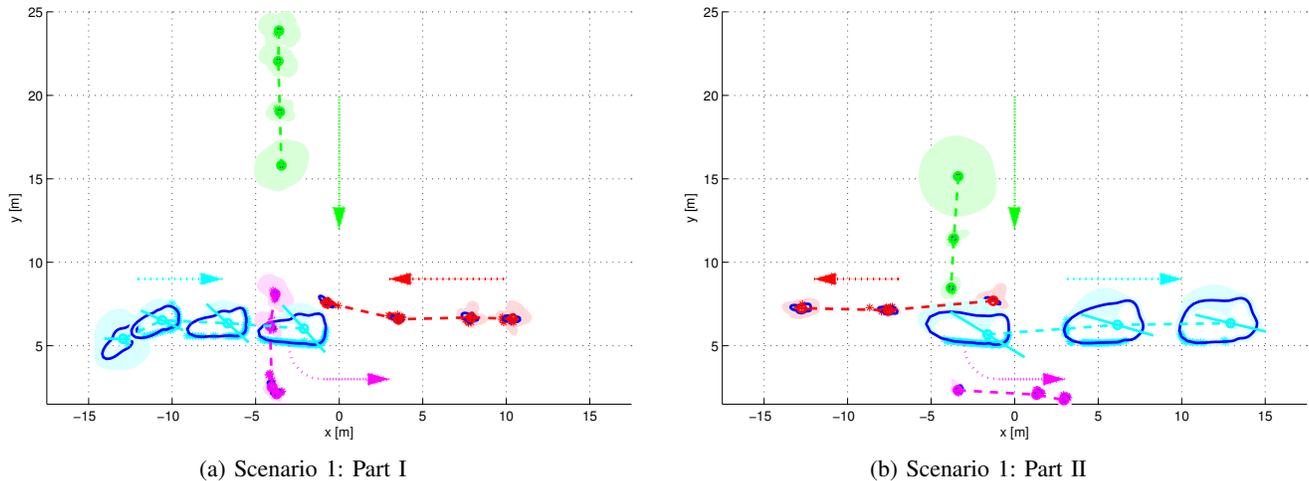

(b) Scenario 1: Part II

Fig. 8: Scenario 1: Four targets moving in the scene. One car, moving from left to right, is shown in blue; one bicycle, moving in the opposite direction, is shown in red; one pedestrian (at the top), walking toward the bottom of the figure, is shown in green; and another pedestrian, who makes an L-shaped move at the bottom of the figure, is shown in magenta. The arrows indicate the direction of the targets' movements. For each target, the estimated contour is plotted with a solid line, the confidence region is plotted as a shaded area, and the measurements are shown as star-shaped markers. Figure (a) and (b) show the first and the second half of the scenario, respectively.

figure is also occluded by the car between scans 20 and 30. The prior for the target extent is the same for all objects at the initialization, which is chosen to be a circle with a radius of 2 m. Note that at the beginning of the scenario, the uncertainty in the target extent is large for all targets, and it decreases in time as more measurements are collected from the targets. Furthermore, one can also observe that the uncertainty region of the car's extent decreases around the observable section of the car where the reflections occur, and the uncertainty for the unobservable section is maintained due to a lack of observations. During the occlusions, the uncertainty regions of the occluded objects increase because the filters are not updated by measurements; only the time update is performed. Once these objects can be detected again, the uncertainty decreases. The GP tracker is successful in tracking all different objects with their extent and orientation. Note that all filters are using the same prior and parameters. No additional tuning of hyper-parameters is required.

In Scenario 2, one car is moving from left to right and two pedestrians walk in the top to bottom direction, as shown in Fig. 9. The pedestrians leave the scene during the scenario. Note that at the end of the scenario, only a few measurements are detected from only one edge of the car, and the algorithm successfully associates those detections with the back of the car and updates its extent and centroid accordingly. A typical approach that would discard the extent structure and attempt to represent the target with an approximate shape would fail to correctly estimate the centroid, which is presented explicitly in the next scenario.

2) Partially observable objects: In Scenario 3, we illustrate the performance of the model in a maneuvering object scenario; see Fig. 10. We compare the output of the proposed method with that from the RM and the RHF models. The
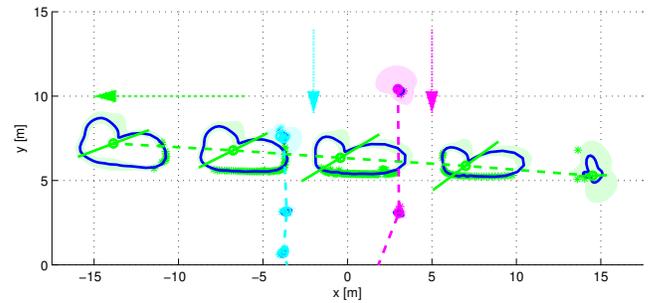

Fig. 9: Scenario 2: Three targets moving in the scene. One car, moving from right to left, is shown in green; one pedestrian (at the top) walking toward the bottom of the figure is shown in blue; and another pedestrian is shown in magenta.

hyper-parameters of the GP are kept the same as in the previous subsection. The forgetting factor for the RM model, λ, is selected to be 0.99. The standard deviation of the Fourier coefficients is chosen as before for the RHF model. Manual tuning is attempted for this model, but no significant performance gain is achieved. This is because the errors arise from the fact that the model does not account for the rotation of the objects. In the scenario, one vehicle makes a right turn starting from the top of the Fig. 10 while moving in the top to bottom direction. Throughout the scenario, the full target extent is never observable; however, the GP model is able to estimate the observable extent and predicts a higher uncertainty around the unobserved section of the target. The RM model tracks the centroid of the measurements rather than the centroid of the target, as expected. The performances of the RM and RHF models are limited because neither of these algorithms is capable of tracking the orientation of a target.
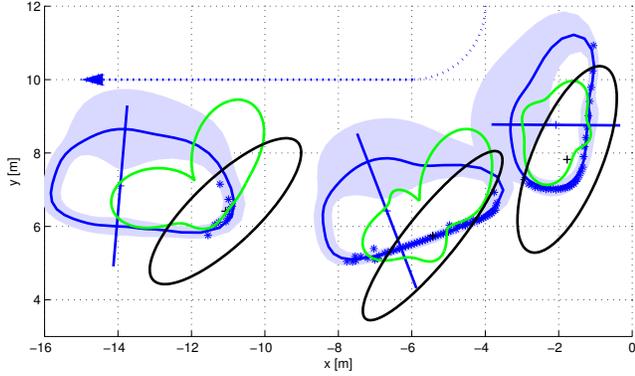
Fig. 10: Scenario 3: A vehicle making a right turn starting from the top of the figure. Outputs of the proposed (blue), the RM (black) and the RHF (green) models are plotted together.
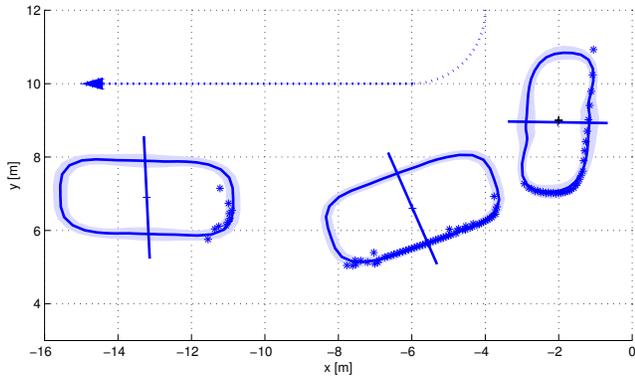

Fig. 11: Scenario 3 (symmetric covariance function): Same data as in Fig. 10, where the symmetric covariance function (26) has been used.

The orientation of the partially observed target, before and after the turn, is successfully tracked by the GP model.

Finally, in Fig. 11, the performance of the proposed method is evaluated on Scenario 3 using the symmetric assumption induced by the covariance function proposed in (26). This covariance function assumes that the radial distance has a period of π, $f(\theta) = f(\theta + \pi)$. This allows us to obtain a low uncertainty of sections of the target contour that has not yet been seen because of the symmetry that we impose. If such assumptions are valid, even better overall tracking performance could be expected. An investigation of such extensions should be addressed in future work.

X. CONCLUSION

In this work, we propose a new approach for extended target tracking. The proposed method uses GPs to model the unknown target extent while simultaneously estimating the kinematic state of the target. We provide an efficient algorithm in which the filter updates are fully recursive and do not suffer from an increase in dimension with each available measurement, unlike the standard GP formulation. The performance and capabilities of the algorithm are demonstrated through simulations and real data experiments. The algorithm

provides an analytical representation of the unknown target extent, which can be used for high-accuracy gating and object classification in future works.

XI. ACKNOWLEDGMENT

APPENDIX A
EXTENDED KALMAN FILTER UPDATE

After initializing the estimate $\hat{\mathbf{x}}_{0|-1} = \boldsymbol{\mu}_0$ and $P_{0|-1} = P_0$, measurement and time update are applied sequentially for each scan.

1) Measurement update: The standard EKF measurement update equations for (39) are

$$\hat{\mathbf{z}}_{k|k-1} = \mathbf{h}_k(\hat{\mathbf{x}}_{k|k-1}), \tag{53a}$$

$$H_k = \frac{d}{d\mathbf{x}_k}\mathbf{h}_k(\mathbf{x}_k)|_{\mathbf{x}_k=\hat{\mathbf{x}}_{k|k-1}}, \tag{53b}$$

$$S_k = H_k P_{k|k-1} H_k^\mathsf{T} + R_k, \tag{53c}$$

$$K_k = P_{k|k-1} H_k^\mathsf{T} S_k^{-1}, \tag{53d}$$

$$\hat{\mathbf{x}}_{k|k} = \hat{\mathbf{x}}_{k|k-1} + K_k \hat{\mathbf{z}}_{k|k-1}, \tag{53e}$$

$$P_{k|k} = P_{k|k-1} + K_k H_k P_{k|k-1}. \tag{53f}$$

These recursions require a gradient of the measurement function $\frac{d\mathbf{h}(\mathbf{x}_k)}{d\mathbf{x}_k}$, which can be computed analytically; see Appendix B. Note that we compute the full derivatives of (34) with respect to $\mathbf{x}_k^c$ and ψ_k, requiring analytical expressions of $\frac{d\theta^G(\mathbf{x}^c)}{d\mathbf{x}^c}$ and $\frac{dk(\theta,\theta')}{d\theta}$ to be combined with the chain rule. This results in a precise and correct state update, enabling estimation of the target orientation, which otherwise would not have been possible.

2) Time update: Furthermore, in accordance with the EKF recursions, the time update equations are as follows:

$$\hat{\mathbf{x}}_{k+1|k} = F\hat{\mathbf{x}}_{k|k-1}, \tag{54a}$$

$$P_{k+1|k} = F P_{k|k-1} F^\mathsf{T} + Q. \tag{54b}$$

APPENDIX B
PARTIAL DERIVATIVES

The derivative in (53b) can be divided into the following submatrices

$$H_k = \frac{d\mathbf{h}_k(\mathbf{x}_k)}{d\mathbf{x}_k} = \frac{d}{d\mathbf{x}_k}\left[\mathbf{h}_{k,1}(\mathbf{x}_k)^\mathsf{T}, \ \ldots \ , \mathbf{h}_{k,n_x}(\mathbf{x}_k)^\mathsf{T}\right]^\mathsf{T}, \tag{55a}$$

$$\frac{d}{d\mathbf{x}_k}\mathbf{h}_{k,l}(\mathbf{x}_k) = \left[\frac{d\mathbf{h}_{k,l}(\mathbf{x}_k)}{d\mathbf{x}_k^c} \quad \frac{d\mathbf{h}_{k,l}(\mathbf{x}_k)}{d\psi_k} \quad \frac{d\mathbf{h}_{k,l}(\mathbf{x}_k)}{d\mathbf{x}_k^{\mathbf{f}}}\right], \tag{55b}$$

where each of them is given by

$$\frac{d\mathbf{h}_{k,l}(\mathbf{x}_k)}{d\mathbf{x}_k^c} = I + \frac{\partial\mathbf{p}_{k,l}(\mathbf{w})}{\partial\mathbf{w}}\bigg|_{\mathbf{w}=\mathbf{x}_k^c} H^{\mathbf{f}}\Big(\theta_{k,l}^L(\mathbf{x}_k^c,\psi_k)\Big)\mathbf{x}_k^{\mathbf{f}}$$
$$+ \mathbf{p}_{k,l}(\mathbf{x}_k^c)\frac{\partial H^{\mathbf{f}}(u)}{\partial u}\bigg|_{u=\theta_{k,l}^L(\mathbf{x}_k^c,\psi_k)}\frac{\partial\theta_{k,l}^G(\mathbf{w})}{\partial\mathbf{w}}\bigg|_{\mathbf{w}=\mathbf{x}_k^c}\mathbf{x}_k^{\mathbf{f}}, \tag{56a}$$

$$\frac{d\mathbf{h}_{k,l}(\mathbf{x}_k)}{d\psi_k} = -\mathbf{p}_{k,l}(\mathbf{x}_k^c)\frac{\partial H^{\mathbf{f}}(u)}{\partial u}\bigg|_{u=\theta_{k,l}^L(\mathbf{x}_k^c,\psi_k)}\mathbf{x}_k^{\mathbf{f}}, \tag{56b}$$

$$\frac{d\mathbf{h}_{k,l}(\mathbf{x}_k)}{d\mathbf{x}_k^{\mathbf{f}}} = \mathbf{p}_{k,l}(\mathbf{x}_k^c)H^{\mathbf{f}}\Big(\theta_{k,l}^L(\mathbf{x}_k^c,\psi_k)\Big), \tag{56c}$$

where

$$\frac{\partial \theta_{k,l}^G(\mathbf{w})}{\partial \mathbf{w}} = \frac{1}{\|\mathbf{z}_{k,l} - \mathbf{w}\|^2} \left[z_{k,l}^{\mathsf{y}} - w^{\mathsf{y}}, \quad -(z_{k,l}^{\mathsf{x}} - w^{\mathsf{x}}) \right], \quad (57a)$$

$$\frac{\partial \mathbf{p}_{k,l}(\mathbf{w})}{\partial \mathbf{w}} = \frac{(\mathbf{z}_{k,l} - \mathbf{w})(\mathbf{z}_{k,l} - \mathbf{w})^{\mathsf{T}}}{\|\mathbf{z}_{k,l} - \mathbf{w}\|^3} - \frac{1}{\|\mathbf{z}_{k,l} - \mathbf{w}\|} I, \quad (57b)$$

$$\frac{\partial H^{\mathbf{f}}(u)}{\partial u} = \frac{\partial K(u, \mathbf{u}^{\mathbf{f}})}{\partial u} [K(\mathbf{u}^{\mathbf{f}}, \mathbf{u}^{\mathbf{f}})]^{-1}, \quad (57c)$$

$$\frac{\partial K(u, \mathbf{u}^{\mathbf{f}})}{\partial u} = \frac{\partial}{\partial u} \left[k_{\text{tot}}(u, u_1^{\mathbf{f}}) \quad \dots \quad k_{\text{tot}}(u, u_{N^{\mathbf{f}}}^{\mathbf{f}}) \right], \quad (57d)$$

$$\frac{\partial k_{\text{tot}}(u, u_i^{\mathbf{f}})}{\partial u} = -\frac{1}{l^2} \sin(u - u_i^{\mathbf{f}}) k(u, u_i^{\mathbf{f}}). \quad (57e)$$

REFERENCES

[1] 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.

[2] M. Feldmann, D. Franken, 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.

[3] 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)*, Chicago, US, Jul. 2011, pp. 592–599.

[4] U. Orguner, "A variational measurement update for extended target tracking with random matrices," *IEEE Transactions on Signal Processing*, vol. 60, no. 7, pp. 3827–3834, Jul. 2012.

[5] J. Lan and X. R. 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)*, Singapore, Jul. 2012, pp. 2177–2184.

[6] ——, "Tracking of extended object or target group using random matrix – Part II: Irregular object," in *Proceedings of the 15th International Conference on Information Fusion (FUSION)*, Singapore, Jul. 2012, pp. 2185–2192.

[7] K. Granström and U. Orguner, "New prediction for extended targets with random matrices," *IEEE Transactions on Aerospace and Electronic Systems*, vol. 50, no. 2, pp. 1577–1589, Apr. 2014.

[8] M. Baum and U. D. Hanebeck, "Random hypersurface models for extended object tracking," in *Proceedings of IEEE International Symposium on Signal Processing and Information Technology (ISSPIT)*, Ajman, United Arab Emirates, Dec. 2009, pp. 178–183.

[9] ——, "Shape tracking of extended objects and group targets with star-convex RHMs," in *Proceedings of the 14th Conference on Information Fusion (FUSION)*, Chicago, US, Jul. 2011, pp. 338–345.

[10] L. Sun, J. Lan, and X. R. Li, "Extended target tracking using star-convex model with nonlinear inequality constraints," in *Proceedings of the 31st Chinese Control Conference (CCC)*. Hefei, China: IEEE, Jul. 2012, pp. 3869–3874.

[11] M. Baum, "Simultaneous tracking and shape estimation of extended objects," Ph.D. dissertation, Karlsruhe Institute of Technology, 2013.

[12] M. Baum and U. D. Hanebeck, "Extended object tracking with random hypersurface models," *IEEE Transactions on Aerospace and Electronic Systems*, vol. 50, no. 1, pp. 149–159, Jan. 2014.

[13] A. Blake, R. Curwen, and A. Zisserma, "A framework for spatio-temporal control in the tracking of visual contours," *International Journal of Computer Vision*, vol. 11, no. 2, pp. 127–145, 1993.

[14] A. Blake, M. Isard, and D. Reynard, "Learning to track the visual motion of contours," *Artificial Intelligence*, vol. 78, no. 1, pp. 179–212, 1995.

[15] P. Li, T. Zhang, and B. Ma, "Unscented Kalman filter for visual curve tracking," *Image and Vision Computing*, vol. 22, no. 2, pp. 157–164, Feb. 2004.

[16] M. Isard and A. Blake, "Condensation - Conditional density propagation for visual tracking," *International Journal of Computer Vision*, vol. 29, no. 1, pp. 5–28, 1998.

[17] P. Li, T. Zhang, and A. E. Pece, "Visual contour tracking based on particle filters," *Image and Vision Computing*, vol. 21, no. 1, pp. 111–123, Jan. 2003.

[18] C. Rasmussen and C. Williams, *Gaussian processes for machine learning*. MIT press Cambridge, MA, 2006.

[19] Y. Bar-Shalom and T. Fortmann, *Tracking and data association*. Mathematics in Science and Engineering, Academic Press Professional, CA, US, 1987.

[20] M. Osborne, "Bayesian Gaussian processes for sequential prediction, optimisation and quadrature," Ph.D. dissertation, University of Oxford, UK, 2010.

[21] J. Hartikainen and S. Särkkä, "Kalman filtering and smoothing solutions to temporal Gaussian process regression models," in *Proceedings of the International Workshop on Machine Learning for Signal Processing (MLSP)*. Kittila, Finland: IEEE, Aug. 2010, pp. 379–384.

[22] M. F. Huber, "Recursive Gaussian process regression," in *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, Vancouver, Canada, May 2013, pp. 3362–3366.

[23] ——, "Recursive Gaussian process: On-line regression and learning," *Pattern Recognition Letters*, vol. 45, pp. 85–91, Aug. 2014.

[24] D. J. C. MacKay, "Introduction to Gaussian processes," in *Neural Networks and Machine Learning*, C. M. Bishop, Ed. Springer Verlag, 1998, pp. 133–165.

[25] S. Blackman and R. Popoli, *Design and Analysis of Modern Tracking Systems*. Artech House, Norwood, MA, 1999.

[26] M. Baum, B. Noack, and U. D. Hanebeck, "Extended object and group tracking with elliptic random hypersurface models," in *Proceedings of 13th Conference on Information Fusion (FUSION)*, Edinburgh, UK, Jul. 2010.

[27] A. Gupta and D. Nagar, *Matrix Variate Distributions*, ser. Monographs and Surveys in Pure and Applied Mathematics. Taylor & Francis, 1999. [Online]. Available: http://books.google.se/books?id=PQOYnT7P1loC

[28] E. Özkan, V. Smidl, S. Saha, C. Lundquist, and F. Gustafsson, "Marginalized adaptive particle filtering for nonlinear models with unknown time-varying noise parameters," *Automatica*, vol. 49, no. 6, pp. 1566–1575, Jun. 2013.

[29] B. Alexe, T. Deselaers, and V. Ferrari, "Measuring the objectness of image windows," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 34, no. 11, pp. 2189–2202, Nov. 2012.

[30] M. Baum, M. Feldmann, D. Fränken, U. D. Hanebeck, and W. Koch, "Extended object and group tracking: A comparison of random matrices and random hypersurface models," in *Proceedings of the IEEE ISIF Workshop on Sensor Data Fusion: Trends, Solutions, Applications (SDF)*, Leipzig, Germany, Oct. 2010.

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

[32] K. Granström, C. Lundquist, and U. Orguner, "Extended target tracking using a Gaussian-mixture PHD filter," *IEEE Transactions on Aerospace and Electronic Systems*, vol. 48, no. 4, pp. 3268–3286, Oct. 2012.

Niklas Wahlström (S'11) received the M.Sc. degree in applied physics and electrical engineering in 2010 and the Tech. Lic. degree in automatic control in 2013, both from Linköping University, Sweden.

Since 2010, he has been working toward his Ph.D. in the Division of Automatic Control, Department of Electrical Engineering, Linköping University. His research interests include sensor fusion, signal processing and machine learning.

Emre Özkan (S'04 - M'10) received his Ph.D. degree in Electrical Engineering from Middle East Technical University, Ankara, Turkey, in 2009. Between 2009 and 2011, he worked as a Postdoctoral Associate in the Division of Automatic Control, Department of Electrical Engineering, Linköping University, Sweden. Currently, he is an Assistant Professor in the same department. His research interests include statistical signal processing, estimation theory, sensor fusion, and target tracking.