Spatio-temporal Gaussian Process Models for Extended and Group Object Tracking with Irregular Shapes

Waqas Aftab, Roland Hostettler, *Member, IEEE*, Allan De Freitas, Mahnaz Arvaneh, and Lyudmila Mihaylova, *Senior Member, IEEE*

***Abstract*—Extended object tracking has become an integral part of many autonomous systems during the last two decades. For the first time, this paper presents a generic spatio-temporal Gaussian process (STGP) for tracking an irregular and non-rigid extended object. The complex shape is represented by key points and their parameters are estimated both in space and time. This is achieved by a factorization of the power spectral density function of the STGP covariance function. A new form of the temporal covariance kernel is derived with the theoretical expression of the filter likelihood function. Solutions to both the filtering and the smoothing problems are presented. A thorough evaluation of the performance in a simulated environment shows that the proposed STGP approach outperforms the state-of-the-art GP extended Kalman filter approach [1], with up to 90% improvement in the accuracy in position, 95% in velocity and 7% in the shape, while tracking a simulated asymmetric non-rigid object. The tracking performance improvement for a non-rigid irregular real object is up to 43% in position, 68% in velocity, 10% in the recall and 115% in the precision measures.**

***Index Terms*—Extended Object Tracking, Spatio-temporal Gaussian Process, Rauch-Tung-Streibel Smoother.**

I. Introduction

EXTENDED object tracking (EOT) includes the process of estimating the kinematic states and the shape parameters of the objects of interest using a sequence of noisy sensor measurements. Group object tracking (GOT) involves state estimation of closely spaced objects moving with similar dynamics. Often, the average dynamics and the shape of the group are estimated over time. Since the state estimation requirements are comparable, hence the methods for EOT and GOT are similar. EOT and GOT are an integral part of various autonomous systems[1], for example: robot navigation [2], people tracking using depth sensors such as Microsoft Kinect [3]. Other applications include crowd analysis [4], tracking of chemical, biological, radiological and nuclear (CBRN) pollutant clouds [5] and sea surveillance [6]. In all these systems, specialized sensors are used to collect measurements. Although the states are interpreted according to the application, the measurement processing and the state estimation requirements are similar. The object kinematic states such as the position, velocity or other higher order time derivatives and shape parameters are estimated based on sensor data. The shape is generally represented by the size or volume parameters.

The EOT methods are also applied in self-driving cars navigating through the urban traffic [7]. Although the technology of autonomous vehicles has many advantages, the main focus is on improving safety and resilience of this technology. Various types of sensors, such as a camera, radar and LiDAR (light detection and ranging), are installed for detection and tracking of roads and obstacles [8]–[10], around the driverless car. The EOT techniques are an inherent part of the cars obstacle avoidance and navigation system.

Tracking multiple point objects (commonly called targets) is known as multi-target tracking (MTT) [11], [12] and requires non-linear estimation methods that successfully deal with data association and clutter. In contrast to tracking point targets [13]–[16], which has been a widely researched area, EOT is a relatively new area and has seen an increase in real-world applications during the last two decades. In point objects tracking, the object kinematics is estimated and most methods assume single measurement per object per time sample. In a similar way, in EOT, the kinematics as well as the shape, extent, and size of the object are estimated. Moreover, multiple measurements per object per time sample are received. The already developed kinematics models for point objects [17], [18] have been typically used in the EOT kinematic state estimation. Hence, the focus of the EOT research has been on the measurement models, the shape estimation and data association.

A typical approach to an EOT problem is to estimate the kinematics of the center of the object (CoO) and model the extent as a parametric or non-parametric function [16], which is unknown and nonlinear as shown in Fig. 1. The extended objects can be classified as either *rigid* if the shape does not change over time, or *non-rigid* if the shape changes over time. The existing methods for EOT use simple models for the temporal changes in shape and as a result, the performance is compromised while tracking non-rigid extended objects. In this paper, an advanced shape model is proposed to improve the shape estimation of non-rigid extended objects. The improved estimates give better tracking performance [16] especially in challenging environments such as low SNR [19].

W. Aftab, M. Arvaneh, L. Mihaylova and A. De Freitas are at the Department of Automatic Control and Systems Engineering (ACSE), The University of Sheffield, Sheffield, UK. waftab1,m.arvaneh,L.S.Mihaylova@sheffield.ac.uk. allandefreitas1@gmail.com

R. Hostettler is at the Department of Electrical Engineering and Automation, Aalto University, Espoo, Finland. roland.hostettler@aalto.fi.

[1]From here on, the term EOT is used only and everything proposed in this paper is applicable to the GOT as well.

A Gaussian Process (GP) [20] can also be employed for estimation of the extent. A GP is a distribution over an unknown and nonlinear function, in the continuous domain. The observed values of these functions can be used to predict the values at unobserved points. Traditionally, the GP is a batch processing method and cannot be used for real-time applications. A GP based recursive filter for real-time EOT has been proposed in [1], [21]. This GP based EOT method models the correlations in a single input domain. The extent states consist of radial values of the object extent from the CoO at different angles. The GP is used to model the nonlinear mapping of the spatial input (angle) to the spatial output (radius). However, the spatial output is correlated in both the spatial and the temporal domain. In [22], it has been shown that if one of the input dimensions is stationary (and some other conditions), then it has an equivalent state space representation which can be solved using Rauch-Tung-Streibel smoothing. The model is termed as spatio-temporal GP (STGP). A recursive equivalent of a temporal GP is proposed in [23], [24] and of an STGP is proposed in [22]. The filter requires a forward pass and is suitable for a real-time implementation. The smoother requires a forward and backward pass and the increase in computational expense with time makes it unsuitable for real-time processing. In this paper, a new model based on the recursive equivalent of the STGP is proposed for real-time tracking of a non-rigid extended object.

A. Related Work

Recent methods for EOT have been comprehensively summarized in two overview papers [15] and [16]. The EOT research can be divided into two parts as shown in Fig. 1: tracking of the CoO and the shape. The CoO kinematics models have been inherited and are similar to the point object kinematics models. Object shapes have been estimated using basic geometric shapes based models for example stick [25], circle [26], [27], rectangle [28] or ellipse [13], [29]. Although the real world objects do not typically have such basic geometrical shapes, these models have been shown to perform satisfactorily well in some applications such as tracking boats with elliptical shapes [30] or cars with rectangular shapes [31]. In the presence of sensor clutter and multiple objects or for tracking irregularly shaped objects, a more detailed shape estimation not only improves the data association performance by providing more accurate confidence regions but also gives better kinematics states estimation [16]. Relatively complex shape models have been proposed using a mixture of ellipsoids [32] or star-convex shape models such as random hypersurface model (RHM) [33] and GP model [1]. The shape changes have been estimated well using basic geometrical shape models. These models are however insufficient for the tracking of irregularly shaped objects. The shape changes in the complex shape models, which perform better for irregular shapes, and some other basic shape models such as the random matrix approach the shape is assumed constant (rigid) and the changes in the shape are incorporated using motion process noise (random walk) [13], [14], [34]. A different approach is taken in the GP extended Kalman filter (GP-EKF) [1], where the shape changes are modeled using a forgetting factor. When tracking non-rigid extended objects, the extent changes over time. Tracking performance is degraded in such scenarios when modeled using the random walk or a forgetting factor shape change model. Additionally, in [1] the GP based approach has been proposed equivalent to a batch GP regression without giving the theoretical explanation and the necessary conditions for the equivalence. Additionally, the measurement noise is ignored at some points during the derivation of the measurement likelihood.

B. Contributions

The contribution area of the proposed model in the EOT literature is depicted in Fig. 1. The key contributions of this work are as follows; (i) A novel interpretation of the center of an asymmetric extended object is presented (given in Subsection IV-A). (ii) A novel non-rigid extended object tracker is proposed based on an STGP model, which includes both the spatial and the temporal correlations of the extent (See Subsection IV-C). (iii) Based on the theoretical results of [22], the full GP regression is proposed to be approximated using a fixed-lag Rauch-Tung-Streibel smoother to obtain quasi-real time approach. This is the first time in the literature of EOT that the theoretical fundamentals of the equivalence between a batch and the recursive GP regression are described for deep understanding (See Subsection IV-H). (iv) A real-time fixed lag smoother based on the STGP model is proposed, which improves upon the accuracy of the filter estimates (See Subsections IV-H and VI-B). (v) The measurement likelihood is derived considering all noises. Due to the complex relationship between the states and the measurements, the previous GP based implementations of the EOT ignored part of the noise (given in Subsection IV-E). (vi) The performance validation of the proposed approach is provided on real and simulated data. The computational complexity and the effect of smoother lag is also evaluated (given in Section VI).

The remaining part of the paper is structured as follows. The theoretical background of the GP and the STGP is covered in Section II and that of inference is covered in Section III. The proposed model of the EOT is explained in Section IV, an example is given in Section V and the evaluation is presented in Section VI followed by conclusions. The sensor measurements coordinate conversions are given in Appendix A and the transformed sensor noise pdfs are derived in Appendix B.

II. Gaussian Process Regression Models

A. Gaussian Processes

A Gaussian Process (GP) [20] is a stochastic process that models a nonlinear function from an input to an output space. A GP is defined by a mean and a covariance kernel. The mean models the mean of the GP output whereas the covariance kernel models the correlations among the inputs of the GP. The parameters of the mean and the covariance kernel are called hyperparameters. The optimal values of the hyperparameters can be determined for instance by maximizing the likelihood of the GP on a given set of input-output data. This process is also called *learning*. A GP with learned hyperparameters

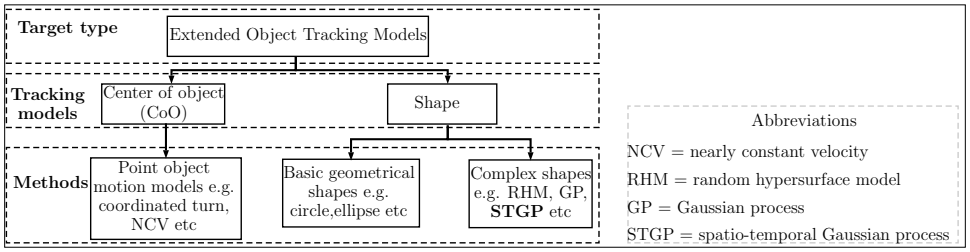


Fig. 1. **The proposed innovation**. The figure gives a hierarchical representation of single EOT research and highlights the contribution of this paper within this paradigm. A complex extent model namely STGP (bold) has been proposed in this paper.

can predict the mean output and its uncertainty at new input locations.

Suppose a GP models the nonlinear function g from a random input θ to an output γ given below:

$$\gamma = g(\theta), \quad g(\theta) \sim GP(\mu(\theta), k(\theta, \theta')), \tag{1}$$

where $\mu(\theta)$ represents the mean and $k(\theta, \theta')$ represents the covariance kernel of the GP. The output γ is observed at n different input values and modeled using the measurement equation given below:

$$\boldsymbol{z} = \boldsymbol{g}(\boldsymbol{\theta}) + \boldsymbol{v}, \tag{2}$$

where $\boldsymbol{z} = [z_1, z_2, ..., z_n]^T$ represents the measurement vector corresponding to the input vector $\boldsymbol{\theta} = [\theta_1, \theta_2, ..., \theta_n]^T$, $\boldsymbol{g}(\boldsymbol{\theta}) = [g(\theta_1), g(\theta_2), ..., g(\theta_n)]^T$ represents the function values vector, $\boldsymbol{v} \sim \mathcal{N}(0, \sigma^2 \boldsymbol{I}_n)$ represents the additive independent identically distributed (i.i.d.) measurement noise vector with variance σ^2 and $\boldsymbol{I}_n$ represents an n-dimensional identity matrix. Several works on GP regression for non-i.i.d. Gaussian measurement noise assumption can be found at [35]–[37]. The GP covariance matrix $\boldsymbol{\Sigma}_{\boldsymbol{\theta}\boldsymbol{\theta}}$ and the GP prediction at the new input vector $\boldsymbol{\theta}^\star$ is given below [20]:

$$\boldsymbol{\Sigma}_{\boldsymbol{\theta}\boldsymbol{\theta}} = \begin{bmatrix} k(\theta_1, \theta_1') & k(\theta_1, \theta_2') & \cdots & k(\theta_1, \theta_n') \\ \vdots & \vdots & \ddots & \vdots \\ k(\theta_n, \theta_1') & k(\theta_n, \theta_2') & \cdots & k(\theta_n, \theta_n') \end{bmatrix}, \tag{3}$$

$$\boldsymbol{\mu}(\boldsymbol{\theta}^\star) = \boldsymbol{\mu}(\boldsymbol{\theta}) + \boldsymbol{\Sigma}_{\boldsymbol{\theta}^\star\boldsymbol{\theta}}(\boldsymbol{\Sigma}_{\boldsymbol{\theta}\boldsymbol{\theta}} + \sigma^2 \boldsymbol{I}_n)^{-1}[\boldsymbol{g}(\boldsymbol{\theta}) - \boldsymbol{\mu}(\boldsymbol{\theta})], \tag{4}$$

$$\boldsymbol{C}(\boldsymbol{\theta}^\star) = \boldsymbol{\Sigma}_{\boldsymbol{\theta}^\star\boldsymbol{\theta}^\star} - \boldsymbol{\Sigma}_{\boldsymbol{\theta}^\star\boldsymbol{\theta}}(\boldsymbol{\Sigma}_{\boldsymbol{\theta}\boldsymbol{\theta}} + \sigma^2 \boldsymbol{I}_n)^{-1}\boldsymbol{\Sigma}_{\boldsymbol{\theta}\boldsymbol{\theta}^\star}, \tag{5}$$

where $\boldsymbol{\mu}(\boldsymbol{\theta})$ represents the mean vector of GP at $\boldsymbol{\theta}$, $\boldsymbol{\mu}(\cdot)$ and $\boldsymbol{C}(\cdot)$ represent, respectively, the mean vector and the error covariance matrix of the GP prediction.

B. Spatio-Temporal Gaussian Processes

An STGP is a stochastic process model for systems evolving in both space and time [22]. Let the spatial input be represented by θ and the temporal input is represented by t, then an STGP can be used to model a functional mapping from the input to the output r of the form given below:

$$r = f(\theta, t), \quad f(\theta, t) \sim STGP(\mu(\theta, t), k(\theta, \theta'; t, t')), \tag{6}$$

where $\mu(\theta, t)$ and $k(\theta, \theta'; t, t')$ represent, respectively, the mean and the covariance kernel of the STGP model. The STGP regression can be determined in the same way as the GP regression explained in Subsection II-A. The time complexity of determining an STGP regression on a model trained at T time steps for N input locations is $\mathcal{O}(N^3T^3)$. As the time progresses the computational expense increases beyond desired for most applications that require real-time processing. In [22] it has been shown that under some conditions, the STGP regression is equivalent to an infinite dimensional state space model. An infinite dimensional recursive filter and a smoother can then be used to perform the inference instead of using the batch processing method. An additional separability assumption, given below, simplifies the resulting model:

$$k(\theta, \theta'; t, t') = k_\theta(\theta, \theta')k_t(t, t'),$$

where $k_\theta(\cdot, \cdot)$ and $k_t(\cdot, \cdot)$ represent the spatial and temporal covariance kernels, respectively. The conditions are given below:

(C_1) The temporal (process) covariance is stationary

$$k_t(t, t') = k_t(t - t').$$

(C_2) The power spectral density (PSD) of the process is rational

$$S(\omega_\theta, \omega_t) = \mathcal{F}[k(\theta, \theta'; t, t')] = \frac{\text{constant w.r.t } \omega_t}{\text{polynomial in } \omega_t^2},$$

where $S(\cdot)$ represents the PSD of the process, ω_θ and ω_t represent the Fourier frequency in the θ and t domains, respectively and $\mathcal{F}[\cdot]$ denotes the Fourier transform.

(C_3) The order of the temporal PSD is a multiple of 2

$$S(\omega_\theta, \omega_t) = \frac{q_t S(\omega_\theta)}{S(\omega_t^2)},$$

where q_t denotes the spectral density of a white noise process driving the temporal dynamics.

(C_4) The spectral factorization of PSD gives a stable transfer function i.e.

$$S(\omega_\theta, \omega_t) = G(\iota\omega_t)S(\omega_\theta)G(-\iota\omega_t),$$

where $G(\iota\omega_t)$ and $G(-\iota\omega_t)$ represent the unstable and the stable transfer function components, respectively, and $\iota\omega_t$ represents the complex Fourier frequency.

As a result, the corresponding GP covariance matrices are also separable. Under the above conditions, the spatio-temporal stochastic process can be equivalently represented by an infinite dimensional dynamic system given below:

$$\frac{\partial \boldsymbol{f}(\boldsymbol{\theta}, t)}{\partial t} = \boldsymbol{A}\boldsymbol{f}(\boldsymbol{\theta}, t) + \boldsymbol{L}\boldsymbol{w}(\boldsymbol{\theta}, t), \tag{7}$$

where $\boldsymbol{f}(\boldsymbol{\theta}, t)$ is a function of the spatial input θ at time t, $\boldsymbol{A}$ is the state transition matrix, $\boldsymbol{L}$ represents the noise effect and $\boldsymbol{w}(\cdot, \cdot)$ represents a zero mean continuous time white noise process.

The measurements are assumed to be arriving at discrete time. The equivalent discrete time model is given below:

$$\boldsymbol{f}(\boldsymbol{\theta}, t_k) = \boldsymbol{F}_k \boldsymbol{f}(\boldsymbol{\theta}, t_{k-1}) + \boldsymbol{w}_k(\boldsymbol{\theta}), \tag{8}$$

$$\boldsymbol{z}_k = \boldsymbol{H}_k \boldsymbol{f}(\boldsymbol{\theta}, t_k) + \boldsymbol{v}_k, \tag{9}$$

where k denotes the discrete time step, $\boldsymbol{F}_k$ is the state transition matrix, $\boldsymbol{w}_k(\boldsymbol{\theta}) \sim \mathcal{N}(0, \boldsymbol{Q}(\boldsymbol{\theta}, \boldsymbol{\theta}'; T_s = t_k - t_{k-1}))$ represents the zero mean white process noise with corresponding covariance matrix $\boldsymbol{Q}(\cdot, \cdot; \cdot)$, T_s represents the sampling time, $\boldsymbol{H}_k$ is the measurement matrix and $\boldsymbol{v}_k$ represents the measurement noise vector.

Given a system model of the form (8)–(9), recursive Bayesian filtering and smoothing solutions can be developed to estimate the function $f(\boldsymbol{\theta}, t_k)$. As a result, the computational complexity of the STGP regression is reduced to $\mathcal{O}(N^3 T)$ and becomes linear in time.

III. Bayesian Inference

The state estimation for the model defined by (8) and (9) can be done using Bayesian inference methods. Bayesian inference relies on belief propagation using a prior density and the measurements. The standard Bayesian inference is done in two steps namely the prediction and the update step. The prediction step uses the prior density and the system dynamics model to determine a predictive density. The update step is performed once the measurements have been received. This step uses the predictive distribution and the measurement likelihood to determine the posterior density. All the information regarding the state is encapsulated in the posterior density. Consider the system dynamics and the measurement model given below:

$$\boldsymbol{x}_k = \boldsymbol{f}(\boldsymbol{x}_{k-1}, \boldsymbol{w}_k), \tag{10}$$

$$\boldsymbol{z}_k = \boldsymbol{h}(\boldsymbol{x}_k, \boldsymbol{v}_k), \tag{11}$$

where $\boldsymbol{x}$ and $\boldsymbol{z}$ represent the state and measurement vectors, respectively, $\boldsymbol{f}$ and $\boldsymbol{h}$ represent the nonlinear state dynamics and measurement functions, respectively, and $\boldsymbol{w}$ and $\boldsymbol{v}$ represent the process and measurement noise vectors, respectively.The Chapman-Kolmogorov equation given below describes the Bayesian prediction:

$$p(\boldsymbol{x}_k|\boldsymbol{z}_{1:k-1}) = \int_{\mathbb{R}^{n_{\boldsymbol{x}}}} p(\boldsymbol{x}_k|\boldsymbol{x}_{k-1}) p(\boldsymbol{x}_{k-1}|\boldsymbol{z}_{1:k-1}) d\boldsymbol{x}_{k-1}, \tag{12}$$

where $p(\boldsymbol{x}_{k-1}|\boldsymbol{z}_{1:k-1})$ denotes the prior and $p(\boldsymbol{x}_k|\boldsymbol{z}_{1:k-1})$ denotes the predictive density, $p(\boldsymbol{x}_k|\boldsymbol{x}_{k-1})$ denotes the one step state prediction and $\boldsymbol{z}_{1:k-1}$ represents all measurements from beginning up to time $k-1$. Under the Markovian assumption, the posterior density is determined using the following recursion:

$$p(\boldsymbol{x}_k|\boldsymbol{z}_{1:k}) = \frac{p(\boldsymbol{z}_k|\boldsymbol{x}_k) p(\boldsymbol{x}_k|\boldsymbol{z}_{1:k-1})}{\int_{\mathbb{R}^{n_{\boldsymbol{x}}}} p(\boldsymbol{z}_k|\boldsymbol{x}_k) p(\boldsymbol{x}_k|\boldsymbol{z}_{1:k-1}) d\boldsymbol{x}_k}, \tag{13}$$

where $p(\boldsymbol{x}_k|\boldsymbol{z}_{1:k})$ is the posterior density and $p(\boldsymbol{z}_k|\boldsymbol{x}_k)$ is the measurement likelihood. For a linear Gaussian system dynamics and measurement model, the Kalman filter [38] is the closed form optimal solution to the Bayes recursion given above. For nonlinear Gaussian models various nonlinear filtering techniques such as extended Kalman filter can be used while for nonlinear non-Gaussian models sequential Monte Carlo methods have been proposed, some of which have been studied in the survey paper [39].

In this paper, an EKF is derived for recursive filtering and Rauch-Tung-Streibel smoother (RTSS) for smoothing.

A. Extended Kalman Filter

The EKF provides a recursive solution to the model given by (10)–(11) under additional assumptions of additive, i.i.d. Gaussian noises (both process and measurement). The model under these assumptions is given below:

$$\boldsymbol{x}_k = \boldsymbol{f}(\boldsymbol{x}_{k-1}) + \boldsymbol{w}_k, \quad \boldsymbol{w}_k \sim \mathcal{N}(0, \boldsymbol{Q}_k), \tag{14}$$

$$\boldsymbol{z}_k = \boldsymbol{h}(\boldsymbol{x}_k) + \boldsymbol{v}_k, \quad \boldsymbol{v}_k \sim \mathcal{N}(0, \boldsymbol{R}_k). \tag{15}$$

The time update equations are given below;

$$\boldsymbol{x}_{k+1|k} = \boldsymbol{f}_{k+1}(\boldsymbol{x}_{k|k}), \boldsymbol{F}_{k+1} = \frac{\partial \boldsymbol{f}}{\partial \boldsymbol{x}}\Big|_{\boldsymbol{x}_k = \boldsymbol{x}_{k|k}}, \tag{16}$$

$$\boldsymbol{P}_{k+1|k} = \boldsymbol{F}_{k+1} \boldsymbol{P}_{k|k} (\boldsymbol{F}_{k+1})^T + \boldsymbol{Q}_{k+1}, \tag{17}$$

where $\boldsymbol{P}$ represents the state error covariance matrix, $(\cdot)_{k|k}$ represents the estimate and $(\cdot)_{k+1|k}$ represents one-step prediction. The measurement update is given below:

$$\boldsymbol{z}_{k+1|k} = \boldsymbol{h}(\boldsymbol{x}_{k+1|k}), \boldsymbol{H}_{k+1} = \frac{\partial \boldsymbol{h}}{\partial \boldsymbol{x}}\Big|_{\boldsymbol{x}_k = \boldsymbol{x}_{k+1|k}}, \tag{18}$$

$$\boldsymbol{S}_{k+1} = \boldsymbol{H}_{k+1} \boldsymbol{P}_{k+1|k} \boldsymbol{H}_{k+1}^T + \boldsymbol{R}_{k+1}, \tag{19}$$

$$\boldsymbol{K}_{k+1} = \boldsymbol{P}_{k+1|k} \boldsymbol{H}_{k+1}^T \boldsymbol{S}_{k+1}^{-1}, \tag{20}$$

$$\boldsymbol{x}_{k+1|k+1} = \boldsymbol{x}_{k+1|k} + \boldsymbol{K}_{k+1}[\boldsymbol{z}_{k+1} - \boldsymbol{z}_{k+1|k}], \tag{21}$$

$$\boldsymbol{P}_{k+1|k+1} = \boldsymbol{P}_{k+1|k} - \boldsymbol{K}_{k+1} \boldsymbol{H}_{k+1} \boldsymbol{P}_{k+1|k}. \tag{22}$$

B. Fixed Lag Smoother

Given a smoothing length k_s, the smoothed state $\tilde{\boldsymbol{x}}_k$ and the state error covariance $\tilde{\boldsymbol{P}}_k$ are recursively estimated using the following recursion [40], which is performed for the time-steps $\{k-1, k-2, \cdots, k-k_s\}$;

$$\boldsymbol{G}_k = \boldsymbol{P}_{k|k} \boldsymbol{F}^T (\boldsymbol{P}_{k+1|k})^{-1}, \tag{23}$$

$$\tilde{\boldsymbol{x}}_k = \hat{\boldsymbol{x}}_{k|k} + \boldsymbol{G}_k[\tilde{\boldsymbol{x}}_{k+1} - \hat{\boldsymbol{x}}_{k+1|k}], \tag{24}$$

$$\tilde{\boldsymbol{P}}_k = \boldsymbol{P}_{k|k} + \boldsymbol{G}_k[\tilde{\boldsymbol{P}}_{k+1} - \boldsymbol{P}_{k+1|k}] \boldsymbol{G}_k^T. \tag{25}$$

The smoother is initialized at the current time step k as $\tilde{\boldsymbol{x}}_k = \hat{\boldsymbol{x}}_{k|k}$ and $\tilde{\boldsymbol{P}}_k = \boldsymbol{P}_{k|k}$.

IV. The Proposed Extended Object Model

In this section, the proposed model for EOT and the associated multiple measurements likelihood is derived. A novel extent dynamical model based on a spatio-temporal GP (STGP) is used. A GP based model is preferred as it is a non-parametric method and can model complex shapes. Additionally, the STGP includes both the spatial and the temporal dynamics to give better shape estimation. The inference can be done using STGP batch regression, however in this work real-time EOT / GOT using an STGP model is presented. The real-time processing requires modeling the STGP in a state space form and deriving a recursive filtering and smoothing solution to the STGP state space model. The object is modeled as a star convex [1] shape as shown in Figs. 2 and 3.

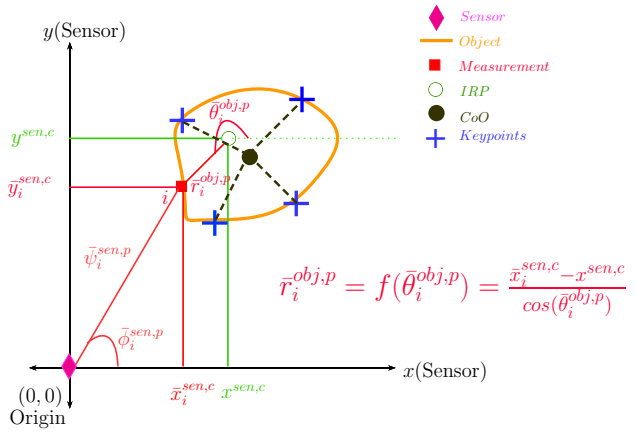


Fig. 2. **An example illustrating sensor and object frames.** The figure shows a sensor, an extended object, the CoO, two (Cartesian and polar) sensor frames with origin at the sensor and an object (polar) frame (origin at IRP). A superscript with the coordinates represents the frame it belongs to e.g. the coordinates of the IRP in Cartesian sensor frame are $(x^{sen,c}, y^{sen,c})$. The state vector consists of radial extent values at equidistant points in the angle domain. A sensor measurement i is reported at $\bar{z}_i^{sen,p} = [\bar{\psi}_i^{sen,p}, \bar{\phi}_i^{sen,p}]^T$ in polar sensor frame. The coordinates of i in Cartesian sensor frame are $\bar{z}_i^{sen,c} = [\bar{x}_i^{sen,c}, \bar{y}_i^{sen,c}]^T$ and in object polar frame are $\bar{z}_i^{obj,p} = [\bar{r}_i^{obj,p}, \bar{\theta}_i^{obj,p}]^T$.

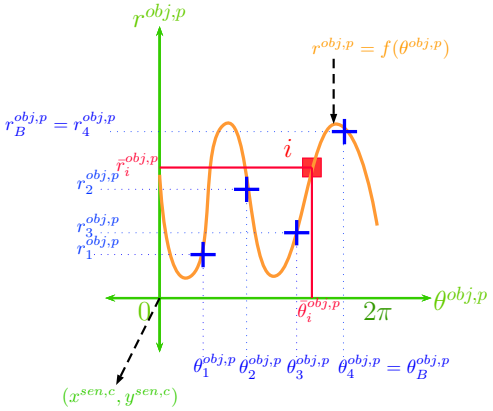


Fig. 3. **Visualization of the nonlinear function estimated using the STGP.** The figure shows the radial function $r^{obj,p} = f(\theta^{obj,p})$ (orange) on polar axis (green). The origin corresponds to the IRP i.e. $(x^{sen,c}, y^{sen,c})$ given in Fig. 2. The measurement i (red square) is shown for comparison with Fig. 2. The extent state vector (blue plus) consists of radial values (shown on y-axis), which are equidistant points on the $\theta^{obj,p}$ axis. In this figure, the number of extent states is $B = 4$. The function is periodic with period equal to 2π. An STGP, trained on coordinates of i and other measurements reported by the sensor, can model the extent at complete angle domain.

A. Center of a Non-rigid Asymmetric Extended Object

The definition of the CoO depends on the application, for example, for uniformly dense objects the geometric center (centroid) of the object shape is considered as the CoO. For non-uniformly dense objects, it can be defined as the center of gravity or the center of mass. In this paper, objects with uniform density are considered. The CoO of rigid objects is assumed to lie on the same position relative to the object extent at all times. In such cases, a filter with nearly accurate initialization and appropriately modeled dynamics can provide efficient CoO estimates. In contrast, the CoO of a non-rigid asymmetric extended object can shift relative to the object extent. This displacement of the CoO needs to be considered in the CoO kinematics model. In this paper, it is proposed that the estimator does not model the CoO kinematics. Instead, the kinematics of a reference point and the extent states relative to this reference point are modeled. This point lies anywhere inside the object boundary and is called the Internal Reference Point (IRP). The CoO kinematic parameters are determined from the IRP and the extent estimates.

B. Sensor and Object Reference Frames

The extended object tracking problem is modeled in two frames, the sensor (global), and the object (local) frames. The sensor measurements are reported in the sensor (polar or Cartesian) frame. The kinematics of the CoO parameters and the IRP states are modeled in the Cartesian sensor frame whereas the extent states and their kinematics are modeled in the object (polar) frame as shown in Fig. 2. The extent states are radial values of the object extent at an angle from the IRP that is $r^{obj,p} = f(\theta^{obj,p})$, where $(\cdot)^{obj,p}$ denotes the variable is in the polar object frame, $r^{obj,p}$ represents the radial extent and $\theta^{obj,p}$ represents the angle from the IRP. This is shown in Fig. 3. The frames and coordinates superscripts are omitted from hereon for brevity.

C. Dynamic Model

The IRP dynamics are modeled using point object motion models [17], [18]. The extent dynamics are designed as separable kernels, which satisfy (C_1) to (C_4), as given below:

$$k^E(\theta, \theta'; t, t') = k_\theta^E(\theta, \theta') k_t^E(t, t'), \tag{26}$$

where $k^E(\cdot)$ represents the spatio-temporal covariance kernel, $k_\theta^E(\cdot)$ represents the spatial and $k_t^E(\cdot)$ represents the temporal covariance kernel. A periodic [20] or Von-Mises [41] covariance kernel can be used to model $k_\theta^E(\cdot)$. $k_t^E(\cdot)$ can be modeled in a number of ways, e.g. squared exponential or Whittle-Matèrn, which shows the generality of the proposed method. The proposed model is converted to a transfer function form and subsequently to an equivalent state space representation using steps given in Subsection II-B. The dynamics of the IRP and the extent states are assumed independent of each other. The dynamical models are given below:

$$\boldsymbol{x}_k^I = \boldsymbol{f}_k^I(\boldsymbol{x}_{k-1}^I) + \boldsymbol{w}_k^I, \quad \boldsymbol{w}_k^I \sim \mathcal{N}(0, \boldsymbol{Q}_k^I), \tag{27}$$

$$\boldsymbol{x}_k^E = \boldsymbol{F}_k^E \boldsymbol{x}_{k-1}^E + \boldsymbol{w}_k^E, \quad \boldsymbol{w}_k^E \sim \mathcal{N}(0, \boldsymbol{Q}_k^E), \tag{28}$$

where $(\cdot)^I$ and $(\cdot)^E$ denote the vector or matrix corresponds, respectively, to the IRP and the extent, $\boldsymbol{f}^I$ represents the nonlinear IRP dynamics function, $\boldsymbol{F}^E$ represents the linear extent dynamics function as derived in (8), $\boldsymbol{w}$ and $\boldsymbol{Q}$ denote the corresponding process noise and process noise covariance matrix, respectively. The IRP state transition models the object motion and determines the maximum velocity or acceleration limits. For slowly moving objects, simple motion models can be used. However, for fast maneuvering objects, complex motion

models [17], [18] can be adopted. The resulting state vector at time k is given below:

$$\boldsymbol{x}_k = \begin{bmatrix} (\boldsymbol{x}_k^I)^T & (\boldsymbol{x}_k^E)^T \end{bmatrix}^T, \tag{29}$$

where $\boldsymbol{x}_k \in \mathbb{R}^{n^{\boldsymbol{x}}}$ represents the overall state vector, $\boldsymbol{x}_k^I = \left[(\boldsymbol{p}_k)^T, (\boldsymbol{p}'_k)^T\right]^T \in \mathbb{R}^{n^I}$ denotes the IRP kinematic states, $\boldsymbol{x}_k^E = \left[(\boldsymbol{r}_k)^T, (\boldsymbol{r}'_k)^T\right]^T \in \mathbb{R}^{n^E}$ represents the extent dynamics states, $\boldsymbol{p}_k$ and $\boldsymbol{p}'_k$ denote, respectively, the position and its higher order time derivatives, $\boldsymbol{r}_k$ and $\boldsymbol{r}'_k$ represent, respectively, the radial extent and its higher order time derivatives. The spatial input of the STGP model is denoted as $\boldsymbol{\theta} = [\theta_1, \theta_2, ..., \theta_B]^T$ which consists of B keypoints in the angle domain between 0 and 2π, as shown in Fig. 3.

D. Measurement Model

N_k measurements are received from the object boundary at time k. The coordinates of the sensor measurements can be either polar $\tilde{\boldsymbol{z}}_k^{sen,p}$ or Cartesian $\tilde{\boldsymbol{z}}_k^{sen,c}$. The polar measurement vector is represented as $\tilde{\boldsymbol{z}}_k^{sen,p} = \left[\tilde{\boldsymbol{z}}_{1,k}^{sen,p}, \cdots, \tilde{\boldsymbol{z}}_{N_k,k}^{sen,p}\right]^T$. Each measurement is modeled as an i.i.d. Gaussian $\tilde{\boldsymbol{z}}_{i,k}^{sen,p} \sim \mathcal{N}(\boldsymbol{\mu}_{i,k}^{sen,p}, \boldsymbol{R}_{i,k}^{sen,p})$. The coordinate converted measurement vector in sensor (Cartesian) frame is represented by $\tilde{\boldsymbol{z}}_k^{sen,c} = \left[\tilde{\boldsymbol{z}}_{1,k}^{sen,c}, \cdots, \tilde{\boldsymbol{z}}_{N_k,k}^{sen,c}\right]^T$ and the corresponding pdf of the i^{th} measurement at time k is approximated to a correlated Gaussian $\tilde{\boldsymbol{z}}_{i,k}^{sen,c} \sim \mathcal{N}(\boldsymbol{\mu}_{i,k}^{sen,c}, \boldsymbol{R}_{i,k}^{sen,c})$. For the Cartesian sensor measurements case, this approximation is not required. After translating $\tilde{\boldsymbol{z}}_k^{sen,c}$ to the IRP and converting the coordinates to polar, the measurement vector $\tilde{\boldsymbol{z}}_k^{obj,p} = \left[\tilde{\boldsymbol{z}}_{1,k}^{obj,p}, \cdots, \tilde{\boldsymbol{z}}_{N_k,k}^{obj,p}\right]^T$ is obtained. The corresponding pdf of the i^{th} measurement at time k is approximated to a Gaussian $\tilde{\boldsymbol{z}}_{i,k}^{obj,p} \sim \mathcal{N}(\boldsymbol{\mu}_{i,k}^{obj,p}, \boldsymbol{R}_{i,k}^{obj,p})$. The relationship among $\tilde{\boldsymbol{z}}_k^{sen,p}, \tilde{\boldsymbol{z}}_k^{sen,c}$ and $\tilde{\boldsymbol{z}}_k^{obj,p}$ is explained in Fig. 2 and given in Appendix A. The resulting measurement model is given below:

$$\boldsymbol{z}_k^{sen,c} = \boldsymbol{h}(\boldsymbol{x}_k^C, \boldsymbol{x}_k^E, \tilde{\boldsymbol{z}}_k^{obj,p}, \boldsymbol{v}_k), \tag{30}$$

where $\boldsymbol{h}(\cdot)$ is a generic measurement function (linear / nonlinear) and $\boldsymbol{v}_k$ is the measurement noise.

E. Derivation of the Measurement Likelihood Function

The measurement likelihood is derived in this subsection assuming contour measurements. For the surface measurements case, the model derived in this section and a GP convolution particle filter [41] can be used. Alternatively, Kalman filter based approach, given in this paper, can be adopted using a modified spatial covariance kernel as proposed in [1].

1) Likelihood function of a single measurement: The likelihood function is derived for the i^{th} measurement. Refer to Fig. 2 and consider the following vectors:

$$x_i^{obj,c} = \bar{x}_i^{sen,c} - x^{sen,c}, \quad y^{sen,c} = \bar{y}_i^{sen,c} - y^{sen,c}, \tag{31}$$

where $a_1 - a_2$ represents the vector difference of a_2 from a_1, $(\bar{x}_i^{sen,c}, \bar{y}_i^{sen,c})$ represents the coordinates of the i^{th} measurement and $(x^{sen,c}, y^{sen,c})$ represents the coordinates of the IRP. Assuming a noise free environment and using vector algebra the measurement vectors are related to the IRP as given below:

$$\bar{x}_i^{sen,c} = x^{sen,c} + x_i^{obj,c} = x^{sen,c} + \bar{r}_i^{obj,p}\cos(\bar{\theta}_i^{obj,p}), \tag{32}$$

$$\bar{y}_i^{sen,c} = y^{sen,c} + y_i^{obj,c} = y^{sen,c} + \bar{r}_i^{obj,p}\sin(\bar{\theta}_i^{obj,p}), \tag{33}$$

$$\boldsymbol{z}_i^{sen,c} = \boldsymbol{p} + \mu_i^{GP}\bar{\boldsymbol{\zeta}}_i, \tag{34}$$

where $(\bar{r}_i^{obj,p}, \bar{\theta}_i^{obj,p})$ represent the i^{th} measurement predicted coordinates, $\boldsymbol{z}_i^{sen,c}$ represents the i^{th} sensor measurement vector, $\boldsymbol{p} = [\bar{x}_i^{sen,c}, \bar{y}_i^{sen,c}]^T$ represents the coordinates of the IRP, $\mu_i^{GP} = \bar{r}_i^{obj,p}$ represents the mean of the STGP model at the i^{th} measurement angle and $\bar{\boldsymbol{\zeta}}_i = [\cos(\bar{\theta}_i^{obj,p}), \sin(\bar{\theta}_i^{obj,p})]^T$ represents the i^{th} measurement transformation vector mean. $\bar{r}_i^{obj,p}$ is determined using the STGP model prediction and has an associated error represented by the STGP covariance matrix. $\bar{\theta}_i^{obj,p}$ is calculated using coordinates transform between the sensor and the object frames (Appendix A) and has an associated uncertainty for the noisy measurement case, represented by the pdf $p(\tilde{r}, \tilde{\theta})$. The sensor and the object frames and coordinates superscripts are omitted from the right hand side of the measurement equation and the time step subscript is added from here on for clarity. The measurement equation with the noise terms is given below:

$$\boldsymbol{z}_{i,k}^{sen,c} = \boldsymbol{p}_k + (\bar{\boldsymbol{\zeta}}_{i,k} + \boldsymbol{e}_{\zeta_{i,k}})(\mu_{i,k}^{GP} + e_i^{GP}) + \boldsymbol{e}_{i,k} \tag{35}$$

where $e_i^{GP} \sim \mathcal{N}(0, c_{i,k}^{GP})$ represents the error in the GP prediction on the i^{th} noisy input angle, $c_{i,k}^{GP}$ represents the corresponding error variance, $\boldsymbol{e}_{i,k} \sim \mathcal{N}(0, \boldsymbol{R}_{i,k}^{sen,c})$ represents the i^{th} measurement noise vector and $\boldsymbol{R}_{i,k}^{sen,c}$ represents the corresponding sensor error covariance matrix. The pdf of the i^{th} transformed vector $\tilde{\boldsymbol{\zeta}}_{i,k} = \left[\cos(\tilde{\theta}_{i,k}), \sin(\tilde{\theta}_{i,k})\right]^T$ (Appendix B) is approximated to a Gaussian $\tilde{\boldsymbol{\zeta}}_{i,k} \sim \mathcal{N}(\bar{\boldsymbol{\zeta}}_{i,k}, \boldsymbol{R}_{i,k}^{\tilde{\boldsymbol{\zeta}}})$ where

$$\bar{\boldsymbol{\zeta}}_{i,k} = \begin{bmatrix} \mu_{\mathcal{C}_{i,k}} \\ \mu_{\mathcal{S}_{i,k}} \end{bmatrix}, \boldsymbol{R}_{i,k}^{\tilde{\boldsymbol{\zeta}}} = \begin{bmatrix} \sigma_{\mathcal{C}_{i,k}}^2 & \sigma_{\mathcal{C}_{i,k}\mathcal{S}_{i,k}}^2 \\ \sigma_{\mathcal{S}_{i,k}\mathcal{C}_{i,k}}^2 & \sigma_{\mathcal{S}_{i,k}}^2 \end{bmatrix}. \tag{36}$$

The i^{th} measurement equation can be written as:

$$\begin{aligned} \boldsymbol{z}_{i,k}^{sen,c} &= \boldsymbol{p}_k + \bar{\boldsymbol{\zeta}}_{i,k}\mu_{i,k}^{GP} + \boldsymbol{e}_{\zeta_{i,k}}\mu_{i,k}^{GP} + \bar{\boldsymbol{\zeta}}_{i,k}e_i^{GP} + \boldsymbol{e}_{\zeta_{i,k}}e_i^{GP} \\ &\quad + \boldsymbol{e}_{i,k} \\ &= \boldsymbol{p}_k + \bar{\boldsymbol{\zeta}}_{i,k}\mu_{i,k}^{GP} + \boldsymbol{e}_{i,k}^1 + \boldsymbol{e}_{i,k}^2 + \boldsymbol{e}_{i,k}^3 + \boldsymbol{e}_{i,k} \\ &= \boldsymbol{p}_k + \bar{\boldsymbol{\zeta}}_{i,k}\mu_{i,k}^{GP} + \boldsymbol{v}_{i,k}, \end{aligned} \tag{37}$$

where $\boldsymbol{v}_{i,k}$ represents the cumulative measurement error vector consisting of four error vector components $\boldsymbol{e}_{i,k}$, $\boldsymbol{e}_{i,k}^1$, $\boldsymbol{e}_{i,k}^2$ and $\boldsymbol{e}_{i,k}^3$. The components of the noise term $\boldsymbol{v}_{i,k}$ are derived below

$$\boldsymbol{e}_{i,k}^1 \sim \mathcal{N}(0, (\mu_{i,k}^{GP})^2\boldsymbol{R}_{i,k}^{\tilde{\boldsymbol{\zeta}}}) = \mathcal{N}(0, \boldsymbol{R}_{i,k}^{\bar{G}\bar{\zeta}}), \tag{38}$$

$$\boldsymbol{e}_{i,k}^2 \sim \mathcal{N}(0, \bar{\boldsymbol{\zeta}}_{i,k}c_{i,k}^{GP}\bar{\boldsymbol{\zeta}}_{i,k}^T) = \mathcal{N}(0, \boldsymbol{R}_{i,k}^{\bar{\zeta}\bar{G}}), \tag{39}$$

$$\boldsymbol{e}_{i,k}^3 \sim \mathcal{N}(0, \boldsymbol{R}_{i,k}^{\tilde{\zeta}\tilde{G}}), \tag{40}$$

$$\boldsymbol{R}_{i,k}^{\tilde{\zeta}\tilde{G}} = \mathrm{diag}\left(\frac{\sigma_{\mathcal{C}_{i,k}}^2 c_{i,k}^{GP}}{2\pi(\sigma_{\mathcal{C}_{i,k}}^2 + c_{i,k}^{GP})^2}, \frac{\sigma_{\mathcal{S}_{i,k}}^2 c_{i,k}^{GP}}{2\pi(\sigma_{\mathcal{S}_{i,k}}^2 + c_{i,k}^{GP})^2}\right), \tag{41}$$

where $\boldsymbol{R}_{i,k}^{\bar{G}\bar{\zeta}}$,$\boldsymbol{R}_{i,k}^{\bar{\zeta}\bar{G}}$ and $\boldsymbol{R}_{i,k}^{\tilde{\zeta}\tilde{G}}$ represent the noise covariance matrices corresponding to the error terms $\boldsymbol{e}_{i,k}^1$, $\boldsymbol{e}_{i,k}^2$ and $\boldsymbol{e}_{i,k}^3$, respectively, and $\mathrm{diag}(\cdot)$ represents a diagonal matrix. The sum of independent Gaussian random variables is a Gaussian given below:

$$\boldsymbol{v}_{i,k} \sim \mathcal{N}(0, \boldsymbol{\Lambda}_{i,k}), \boldsymbol{\Lambda}_{i,k} = \boldsymbol{R}_{i,k}^{\bar{G}\bar{\zeta}} + \boldsymbol{R}_{i,k}^{\bar{\zeta}\bar{G}} + \boldsymbol{R}_{i,k}^{\tilde{\zeta}\tilde{G}} + \boldsymbol{R}_{i,k}^{sen,c} \tag{42}$$

where $\boldsymbol{\Lambda}_{i,k}$ represents the i^{th} measurement noise covariance matrix. The likelihood function is given below:

$$p(\tilde{\boldsymbol{z}}_{i,k}^{sen,c}|\boldsymbol{x}_k) = \mathcal{N}(\Upsilon_{i,k}, \boldsymbol{\Lambda}_{i,k}), \tag{43}$$

$$\Upsilon_{i,k} = \bar{\boldsymbol{z}}_{i,k}^{sen,c} - (\boldsymbol{p}_k + \bar{\boldsymbol{\zeta}}_{i,k}\mu_{i,k}^{GP}). \tag{44}$$

2) Likelihood Function for Multiple Measurements: In this section, the multiple measurement likelihood is given for N_k measurements using the single measurement likelihood. The measurement equation is given below:

$$\bar{\boldsymbol{z}}_k^{sen,c} = \boldsymbol{\mathcal{H}}(\tilde{\boldsymbol{z}}_k^{obj,p})\boldsymbol{x}_k + \boldsymbol{v}_k, \tag{45}$$

where $\boldsymbol{\mathcal{H}}(\tilde{\boldsymbol{z}}_k^{obj,p})$ represents the measurement function and is given below:

$$\boldsymbol{\mathcal{H}}(\tilde{\boldsymbol{z}}_k^{obj,p}) = \boldsymbol{\mathcal{H}}_1(\tilde{\boldsymbol{z}}_k^{obj,p})\boldsymbol{\mathcal{C}}_1(\tilde{\boldsymbol{z}}_k^{obj,p})\boldsymbol{C}_2, \tag{46}$$

where $\boldsymbol{\mathcal{H}}_1, \boldsymbol{\mathcal{C}}_1$ and $\boldsymbol{C}_2$ represent the sub-functions of $\boldsymbol{\mathcal{H}}$. The matrix multiplication of $\boldsymbol{\mathcal{C}}_1$ and $\boldsymbol{C}_2$ with the state vector $\boldsymbol{x}_k$ gives a matrix consisting of the IRP states and the prediction of the object's extent at the angles defined by N_k measurements with respect to the IRP. Subsequent multiplication with $\boldsymbol{\mathcal{H}}_1$ performs the coordinate frame conversion of the predicted measurements from polar local to Cartesian local and further to Cartesian global. These matrices and the measurement noise are:

$$\boldsymbol{\mathcal{H}}_1(\tilde{\boldsymbol{z}}_k^{obj,p}) = \begin{bmatrix} \boldsymbol{H}_2 & \bar{\boldsymbol{\zeta}}_{1,k} & \underline{\boldsymbol{o}}_2 & \cdots & \underline{\boldsymbol{o}}_2 \\ \boldsymbol{H}_2 & \underline{\boldsymbol{o}}_2 & \bar{\boldsymbol{\zeta}}_{2,k} & \cdots & \underline{\boldsymbol{o}}_2 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ \boldsymbol{H}_2 & \underline{\boldsymbol{o}}_2 & \underline{\boldsymbol{o}}_2 & \cdots & \bar{\boldsymbol{\zeta}}_{N_k,k} \end{bmatrix}, \tag{47}$$

$$\boldsymbol{H}_2 = \begin{bmatrix} 1 & 0 & \overline{\boldsymbol{o}}_{(n^I-2)} \\ 0 & 1 & \overline{\boldsymbol{o}}_{(n^I-2)} \end{bmatrix}, \tag{48}$$

$$\boldsymbol{\mathcal{C}}_1(\tilde{\boldsymbol{z}}_k^{obj,p}) = \begin{bmatrix} \boldsymbol{I}_{n^I} & \boldsymbol{O}_{n^I \times n^E} \\ \boldsymbol{O}_{N_k \times n^I} & \boldsymbol{C}_{\bar{\boldsymbol{\theta}}_k\boldsymbol{\theta}} + \frac{\boldsymbol{\Sigma}_{\tilde{\boldsymbol{\theta}}_k}\boldsymbol{C}''_{\tilde{\boldsymbol{\theta}}_k\boldsymbol{\theta}}}{2} \end{bmatrix}, \tag{49}$$

$$\bar{\boldsymbol{\theta}}_k = \begin{bmatrix} \bar{\theta}_{1,k} & \bar{\theta}_{2,k} & \cdots & \bar{\theta}_{N_k,k} \end{bmatrix}^T \tag{50}$$

$$\boldsymbol{\Sigma}_{\tilde{\boldsymbol{\theta}}_k} = \mathrm{diag}(\sigma^2_{\tilde{\theta}_{1,k}}, \sigma^2_{\tilde{\theta}_{2,k}}, \cdots, \sigma^2_{\tilde{\theta}_{N_k,k}}), \tag{51}$$

$$\boldsymbol{C}_2 = \begin{bmatrix} \boldsymbol{I}_{n^I} & \boldsymbol{O}_{n^I \times n^E} \\ \boldsymbol{O}_{n^E \times n^I} & \boldsymbol{C}^{-1}_{\boldsymbol{\theta}\boldsymbol{\theta}} \end{bmatrix}, C_{\boldsymbol{\theta}\boldsymbol{\theta}} = \boldsymbol{\Sigma}_{\boldsymbol{\theta}\boldsymbol{\theta}} \otimes \boldsymbol{I}_{\frac{n^E}{B}}, \tag{52}$$

$$\boldsymbol{v}_k \sim \mathcal{N}(0, \boldsymbol{\Lambda}_k = \boldsymbol{\Omega}_k^{\bar{G}\bar{\zeta}} + \boldsymbol{\Omega}_k^{\bar{\zeta}\tilde{G}} + \boldsymbol{\Omega}_k^{\tilde{\zeta}\tilde{G}} + \boldsymbol{\Omega}_k^{\tilde{\tilde{\boldsymbol{z}}}^G}), \tag{53}$$

$$\boldsymbol{\Omega}_k^{\bar{G}\bar{\zeta}} = \mathrm{blkdiag}\begin{pmatrix} \boldsymbol{R}_{1,k}^{\bar{G}\bar{\zeta}} & \boldsymbol{R}_{2,k}^{\bar{G}\bar{\zeta}} & \cdots & \boldsymbol{R}_{N_k,k}^{\bar{G}\bar{\zeta}} \end{pmatrix}, \tag{54}$$

$$\boldsymbol{\Omega}_k^{\bar{\zeta}\tilde{G}} = \mathrm{blkdiag}\begin{pmatrix} \boldsymbol{R}_{1,k}^{\bar{\zeta}\tilde{G}} & \boldsymbol{R}_{2,k}^{\bar{\zeta}\tilde{G}} & \cdots & \boldsymbol{R}_{N_k,k}^{\bar{\zeta}\tilde{G}} \end{pmatrix}, \tag{55}$$

$$\boldsymbol{\Omega}_k^{\tilde{\zeta}\tilde{G}} = \mathrm{blkdiag}\begin{pmatrix} \boldsymbol{R}_{1,k}^{\tilde{\zeta}\tilde{G}} & \boldsymbol{R}_{2,k}^{\tilde{\zeta}\tilde{G}} & \cdots & \boldsymbol{R}_{N_k,k}^{\tilde{\zeta}\tilde{G}} \end{pmatrix}, \tag{56}$$

$$\boldsymbol{\Omega}_k^{sen,c} = \mathrm{blkdiag}\begin{pmatrix} \boldsymbol{R}_{1,k}^{sen,c} & \boldsymbol{R}_{2,k}^{sen,c} & \cdots & \boldsymbol{R}_{N_k,k}^{sen,c} \end{pmatrix}, \tag{57}$$

where $\boldsymbol{O}_{m\times n}$ represents an m by n zero matrix, $\overline{\boldsymbol{o}}_m$ represents an m-dimensional zero row vector, $\underline{\boldsymbol{o}}_m$ represents an m-dimensional zero column vector, $\otimes$ represents the Kronecker product and $\mathrm{blkdiag}[\cdot]$ represents a block diagonal matrix. The measurements are assumed independent which gives the structure of $\boldsymbol{\Sigma}_{\tilde{\boldsymbol{\theta}}_k}$,$\boldsymbol{\Omega}_k^{\bar{G}\bar{\zeta}}$, $\boldsymbol{\Omega}_k^{\bar{\zeta}\tilde{G}}$, $\boldsymbol{\Omega}_k^{\tilde{\zeta}\tilde{G}}$ and $\boldsymbol{\Omega}_k^{sen,c}$ as block diagonal. The GP expression appearing in $\boldsymbol{\mathcal{C}}_1(\boldsymbol{z}_k^{obj,p})$ and $\boldsymbol{C}_2$ and the GP covariance $\boldsymbol{C}_k^{GP}$ are derived in Subsection IV-F. The multiple measurements likelihood function is given below:

$$p(\tilde{\boldsymbol{z}}_k^{sen,c}|\boldsymbol{x}_k) = \mathcal{N}(\Upsilon_k, \boldsymbol{\Lambda}_k), \tag{58}$$

$$\Upsilon_k = \bar{\boldsymbol{z}}_k^{sen,c} - \left([\boldsymbol{p}_k]_{\times N_k} + \bar{\boldsymbol{\zeta}}_k \odot \left(\boldsymbol{\mu}_k^{GP} \otimes \begin{bmatrix} 1 \\ 1 \end{bmatrix} \right) \right),$$

where $[\boldsymbol{a}]_{\times n}$ represents a column vector with n times repetition of the vector $\boldsymbol{a}$, $\odot$ represents element-wise product, $\bar{\boldsymbol{\zeta}}_k = [\bar{\boldsymbol{\zeta}}_{1,k}, \cdots, \bar{\boldsymbol{\zeta}}_{N_k,k}]^T$ and $\boldsymbol{\mu}_k^{GP} = [\mu_{1,k}^{GP}, \cdots, \mu_{N_k,k}^{GP}]^T$.

F. GP Prediction at Noisy Input Locations

The input locations $\bar{\boldsymbol{\theta}}_k$ in (50) are corrupted by the sensor noise. This gives a non-Gaussian posterior, which is approximated to a Gaussian. The GP prediction given in (4)–(5) is valid for noise-free inputs. The GP prediction for noisy training input locations and non-noisy predicted locations is derived in [20]. In (45), the GP prediction is required at noisy locations using data of non-noisy input locations. This has been derived in [42], [43] for different covariance kernels. Exact first and second moments of the posterior are derived for linear or Gaussian covariance kernels. For remaining covariance kernels (like the spatial covariance kernel), using a Taylor series expansion, the approximate moments are derived. For a given input with distribution $\tilde{\boldsymbol{\theta}}_k \sim \mathcal{N}(\bar{\boldsymbol{\theta}_k}, \boldsymbol{\Sigma}_{\tilde{\boldsymbol{\theta}}_k})$, the predictive mean and covariance are given below:

$$\boldsymbol{\mu}_k^{GP} = \boldsymbol{\mu}(\tilde{\boldsymbol{\theta}_k}) + \frac{1}{2}\sum_{i=1}^{B} \beta_i Tr[\boldsymbol{C}''_{\tilde{\boldsymbol{\theta}}_k\boldsymbol{\theta}_i}\boldsymbol{\Sigma}_{\tilde{\boldsymbol{\theta}}_k}],$$

$$\boldsymbol{C}_k^{GP} = \boldsymbol{C}(\tilde{\boldsymbol{\theta}}_k) + Tr\Big[\Big(\frac{1}{2}\boldsymbol{C}''(\tilde{\boldsymbol{\theta}}_k) + \boldsymbol{\mu}'(\tilde{\boldsymbol{\theta}}_k)\boldsymbol{\mu}'(\tilde{\boldsymbol{\theta}}_k)^T\Big)\boldsymbol{\Sigma}_{\tilde{\boldsymbol{\theta}}_k}\Big],$$

where $\boldsymbol{\mu}_k^{GP}$ and $\boldsymbol{C}_k^{GP}$ represent the mean and the covariance of the GP prediction at the noisy input angle measurements, $\boldsymbol{\mu}(\tilde{\boldsymbol{\theta}_k})$ and $\boldsymbol{C}(\tilde{\boldsymbol{\theta}}_k)$ represent the noise-free GP prediction mean and covariance, respectively, and $Tr[\cdot]$ is the trace function. The terms on the right side of the summation in both equations can be seen as the correction of the noise free GP mean and covariance values. These are explained in the following equations:

$$\boldsymbol{\beta} = C_{\boldsymbol{\theta}\boldsymbol{\theta}}^{-1}\boldsymbol{x}_k^E, \boldsymbol{\mu}'(\tilde{\boldsymbol{\theta}}_k) = \boldsymbol{C}'_{\tilde{\boldsymbol{\theta}}_k\boldsymbol{\theta}}\boldsymbol{C}^{-1}_{\boldsymbol{\theta}\boldsymbol{\theta}}\boldsymbol{x}_k^E, \tag{59}$$

$$\boldsymbol{C}''(\tilde{\boldsymbol{\theta}}_k) = \boldsymbol{C}''_{\tilde{\boldsymbol{\theta}}_k\bar{\boldsymbol{\theta}}_k} - \boldsymbol{C}''_{\tilde{\boldsymbol{\theta}}_k\boldsymbol{\theta}}\boldsymbol{C}^{-1}_{\boldsymbol{\theta}\boldsymbol{\theta}}\boldsymbol{C}''_{\boldsymbol{\theta}\bar{\boldsymbol{\theta}}_k}, \tag{60}$$

$$\boldsymbol{C}''_{\tilde{\boldsymbol{\theta}}_k\boldsymbol{\theta}} = \boldsymbol{\Sigma}''_{\tilde{\boldsymbol{\theta}}_k\boldsymbol{\theta}} \otimes \begin{bmatrix} 1 & \overline{\boldsymbol{o}}_{\left(\frac{n^E}{B}-1\right)} \end{bmatrix}, \tag{61}$$

$$\boldsymbol{C}''_{\tilde{\boldsymbol{\theta}}_k\bar{\boldsymbol{\theta}}_k} = \boldsymbol{\Sigma}''_{\tilde{\boldsymbol{\theta}}_k\bar{\boldsymbol{\theta}}_k} \otimes \begin{bmatrix} 1 & \overline{\boldsymbol{o}}_{\left(\frac{n^E}{B}-1\right)} \end{bmatrix}, \tag{62}$$

$$\boldsymbol{C}'_{\tilde{\boldsymbol{\theta}}_k\boldsymbol{\theta}} = \boldsymbol{\Sigma}'_{\tilde{\boldsymbol{\theta}}_k\boldsymbol{\theta}} \otimes \begin{bmatrix} 1 & \overline{\boldsymbol{o}}_{\left(\frac{n^E}{B}-1\right)} \end{bmatrix}, \tag{63}$$

where $\boldsymbol{\Sigma}'_{\cdot}$ and $\boldsymbol{\Sigma}''_{\cdot}$ represent the first and second differential of the corresponding noise free GP covariance matrices.

G. CoO Parameter Estimates

The parameters of the CoO kinematics are the position and the higher order time derivatives of the position. These parameters are calculated from the estimated shape (polygon) in the sensor frame at each time step. Consider $\{(x_1^{\hat{P}}, y_1^{\hat{P}}), (x_2^{\hat{P}}, y_2^{\hat{P}}), \cdots, (x_B^{\hat{P}}, y_B^{\hat{P}})\}$ represents the coordinates of the estimated polygon. The positional [44] (x_k^C, y_k^C) and the velocity $(\dot{x}_k^C, \dot{y}_k^C)$ parameters are determined as given below:

$$A = \frac{1}{2}\sum_{i=1}^{B}(x_i^{\hat{P}}y_{i+1}^{\hat{P}} - x_{i+1}^{\hat{P}}y_i^{\hat{P}}), \tag{64}$$

$$x_k^C = \frac{1}{6A}\sum_{i=1}^{B}(x_i^{\hat{P}} + x_{i+1}^{\hat{P}})(x_i^{\hat{P}} y_{i+1}^{\hat{P}} - x_{i+1}^{\hat{P}} y_i^{\hat{P}}), \tag{65}$$

$$y_k^C = \frac{1}{6A}\sum_{i=1}^{B}(y_i^{\hat{P}} + y_{i+1}^{\hat{P}})(x_i^{\hat{P}} y_{i+1}^{\hat{P}} - x_{i+1}^{\hat{P}} y_i^{\hat{P}}), \tag{66}$$

$$\dot{x}_k^C = \frac{x_k^C - x_{k-1}^C}{T_s}, \quad \dot{y}_k^C = \frac{y_k^C - y_{k-1}^C}{T_s}, \tag{67}$$

where A represents the area of the polygon.

H. Real-time Inference

The inference can be done using an STGP batch regression. As most of the EOT applications require real-time processing, the estimation of the state space model and the measurement likelihood derived above is done recursively. A real-time recursive filter equivalent to a full GP regression has also been proposed in [1], [21]. The mathematical equivalence of a full GP regression is a smoother rather than a filter [22]. Given a model of the form (27), (28) and (45) a recursive (nonlinear) Kalman filtering and smoothing solution is developed, to estimate the states at each time step. In high nonlinearity scenarios, advance nonlinear filtering and smoothing methods such as sequential Monte Carlo (SMC), Markov chain Monte Carlo (MCMC) are preferred [39].

The processing time of the smoother increases with time and the computation becomes non-real time as more measurement samples are reported. A fixed lag RTSS is proposed for real-time smoothing. It is further proposed to set the lag of the RTSS equal to as long as the states are correlated in time. In short, the real-time inference is achieved using a fixed lag RTSS with lag value set equal to l_t.

V. Extended Object Tracking using Whittle-Matèrn Temporal Covariance

Section V demonstrates the proposed method using different models. A block diagram of the proposed method is given in Fig. 4.

A. Extent Evolution Model

1) Spatial Covariance Kernel: The periodic spatial covariance kernel [20] is illustrated in Fig. 5 and is given below:

$$k_\theta^E(\theta,\theta') = \sigma_f^2 e^{-\frac{2\sin^2\left(\frac{\theta-\theta'}{p}\right)}{l_\theta^2}} + \sigma_r^2, \tag{68}$$

where $\sigma_f^2, \sigma_r^2, p$ and l_θ are hyperparameters. σ_f^2 controls the correlation magnitude, σ_r^2 is the prior radial variance, p is periodicity and l_θ controls correlation length-scale. This kernel is generic and can be used for various real-world extended objects. Given $\vartheta = \cos\left(\frac{2\varepsilon}{p}\right)$ and $\varepsilon = \bar{\theta} - \theta'$, the derivatives of the covariance kernel are given below:

$$k_\theta^{'}(\bar{\theta},\theta') = \frac{d}{d\bar{\theta}}\left[k_\theta(\bar{\theta},\theta')\right] = -\frac{\sigma_f^2 e^{\frac{\vartheta-1}{4l_\theta^2}}\sin\left(\frac{2\varepsilon}{p}\right)}{2l_\theta^2 p}, \tag{69}$$

$$k_\theta^{''}(\bar{\theta},\theta') = \frac{d}{d\bar{\theta}}\left[k^{'}(\bar{\theta},\theta')\right] = -\frac{\sigma_f^2 e^{\frac{\vartheta-1}{4l_\theta^2}}(4l_\theta^2\vartheta + \vartheta^2) - 1)}{4l_\theta^4 p^2}, \tag{70}$$

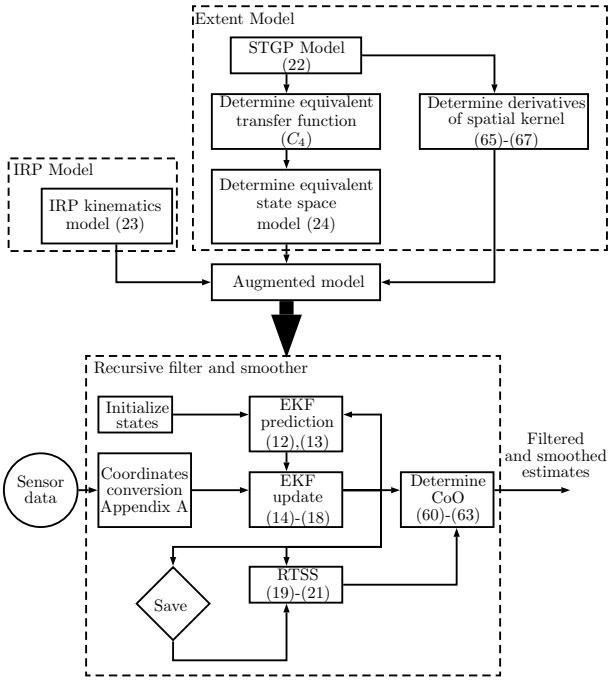


Fig. 4. **Proposed method**. The figure shows the proposed method. The top half (blue) of the figure shows the modeling part whereas the bottom half shows the recursive filtering and smoothing solution. The diamond shape represents a memory storage, required by the RTSS.

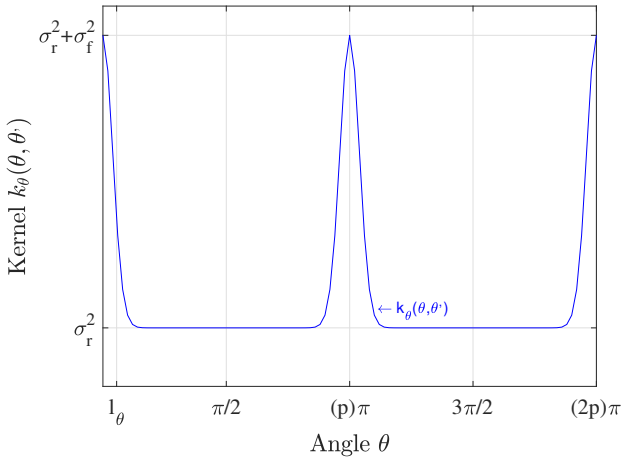


Fig. 5. **Spatial (Periodic) covariance kernel**.

$$\begin{aligned} k_\theta^{''}(\bar{\theta},\theta') &= \frac{\partial}{\partial\bar{\theta}\partial\theta'}\left[\frac{\partial}{\partial\bar{\theta}\partial\theta'}\left[k(\bar{\theta},\theta')\right]\right] \\ &= \frac{\sigma_f^2}{l_\theta^8}\exp\left(\frac{\cos(\varepsilon)-1.0}{l_\theta^2}\right) \\ &\quad\left(\frac{-\cos(2\varepsilon)}{2} + \frac{\cos(4\varepsilon)}{8} - \frac{3l_\theta^2\cos(\varepsilon)}{2} + l_\theta^6\cos(\varepsilon)\right. \\ &\quad\left. + \frac{3l_\theta^2\cos(3\varepsilon)}{2} + \frac{7l_\theta^4\cos(2\varepsilon)}{2} - \frac{l_\theta^4}{2} + \frac{3}{8}\right). \end{aligned} \tag{71}$$

2) Temporal Covariance Kernel: A Whittle-Matèrn temporal covariance kernel [45], [46] is chosen and is given below:

$$k_t^E(t,t') = \sigma_t^2\frac{2^{1-\nu}}{\Gamma(\nu)}\left(\frac{\sqrt{2\nu}}{l_t}\tau\right)^\nu K_\nu\left(\frac{\sqrt{2\nu}}{l_t}\tau\right), \tag{72}$$

where $\tau = t' - t$, $\Gamma(\cdot)$ is the gamma function and σ_t^2 represents the correlation magnitude and l_t represents the temporal length-scale. K_ν is modified Bessel function of the second kind and

smoothness of process is determined by the kernel parameter ν. The corresponding spectral density $S(\omega_t)$ is given below:

$$S(\omega_t) = \sigma_t^2 \frac{2\pi^{\frac{1}{2}}\Gamma(\nu+\frac{1}{2})}{\Gamma(\nu)}\lambda^{2\nu}(\lambda^2+\omega_t^2)^{-(\nu+\frac{1}{2})}, \tag{73}$$

where ω_t represents frequency and $\lambda = \frac{\sqrt{2\nu}}{l_t}$. As the spectral density is a function of ω_t^2, a stable transfer function $G(\iota\omega_t) = (\lambda+\iota\omega)^{-(p+1)}$ can be obtained after spectral factorization of the given kernel where $p = \nu - \frac{1}{2}$ and:

$$q_t = \frac{2\sigma_t^2\pi^{\frac{1}{2}}\lambda^{(2p+1)}\Gamma(p+1)}{\Gamma(p+\frac{1}{2})}, \tag{74}$$

where q_t is the spectral density of the white noise process driving the temporal evolution of the states.

Remark 1: Choosing the order of the Whittle-Matern covariance function, $\nu = \frac{1}{2}$, yields the Ornstein-Uhlenbeck function [47]. This in turn has the same state-space representation as the model used in GP-EKF [1]. Hence, the GP-EKF can be seen as a special case of the method proposed herein.

3) Extent State Space Model: The system matrix and the noise effect vector of the corresponding state space model for $\nu = \frac{5}{2}$ are derived in [23] and given below:

$$\boldsymbol{A} = \begin{pmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ -\lambda^3 & -3\lambda^2 & -3\lambda \end{pmatrix}, \quad \boldsymbol{L} = \begin{pmatrix} 0 \\ 0 \\ 1 \end{pmatrix}. \tag{75}$$

Using above a multidimensional discrete time state space model for B keypoints is derived and given below:

$$\boldsymbol{x}_k^E = \boldsymbol{F}^E\boldsymbol{x}_{k-1}^E + \boldsymbol{w}_k^E, \boldsymbol{F}^E = \boldsymbol{I}_B \otimes e^{\boldsymbol{A}T_s} \tag{76}$$

$$\boldsymbol{w}_k^E \sim \mathcal{N}(0, \boldsymbol{Q}^E(\boldsymbol{\theta}, \boldsymbol{\theta}'; T_s)), \tag{77}$$

$$\boldsymbol{Q}^E(\boldsymbol{\theta}, \boldsymbol{\theta}'; T_s) = \boldsymbol{C}_{\boldsymbol{\theta}\boldsymbol{\theta}}[\boldsymbol{I}_B \otimes \tilde{\boldsymbol{Q}}(T_s)],$$

$$\tilde{\boldsymbol{Q}}(T_s) = \int_0^{T_s} \boldsymbol{F}^E(T_s-\tau)\boldsymbol{L}q_t\boldsymbol{L}^T\boldsymbol{F}^E(T_s-\tau)^T d\tau, \tag{78}$$

where $\boldsymbol{F}^E$ and $\boldsymbol{Q}^E(\boldsymbol{\theta}, \boldsymbol{\theta}'; T_s)$ represent the discrete time state transition matrix and the process noise covariance matrix for B keypoints, respectively.

B. IRP Kinematics Model

The IRP kinematics are modeled using a nearly constant velocity (NCV) [17] motion model as given below:

$$\boldsymbol{x}_k^I = \boldsymbol{F}^I\boldsymbol{x}_{k-1}^I + \boldsymbol{w}_k^I, \quad \boldsymbol{w}_k^I \sim \mathcal{N}(0, \boldsymbol{Q}^I), \tag{79}$$

$$\boldsymbol{F}^I = \mathrm{diag}(\tilde{\boldsymbol{F}}^I, \tilde{\boldsymbol{F}}^I), \quad \boldsymbol{Q}^I = \mathrm{diag}(q_x\tilde{\boldsymbol{Q}}^I, q_y\tilde{\boldsymbol{Q}}^I), \tag{80}$$

$$\tilde{\boldsymbol{F}}^I = \begin{bmatrix} 1 & T_s \\ 0 & 1 \end{bmatrix}, \quad \tilde{\boldsymbol{Q}}^I = \begin{bmatrix} \frac{T_s^3}{3} & \frac{T_s^2}{2} \\ \frac{T_s^2}{2} & T_s \end{bmatrix}, \tag{81}$$

where q_x and q_y represent the process noise variances.

C. State Vector

The corresponding state vectors are given below:

$$\boldsymbol{x}_k^I = \begin{bmatrix} x_k & \dot{x}_k & y_k & \dot{y}_k \end{bmatrix}^T, \tag{82}$$

$$\boldsymbol{x}_k^E = \begin{bmatrix} r_k^1 & \dot{r}_k^1 & \ddot{r}_k^1 & \cdots & r_k^B & \dot{r}_k^B & \ddot{r}_k^B \end{bmatrix}^T, \tag{83}$$

where the location of the IRP is represented by x_k, y_k and the velocity of the IRP is represented by $\dot{x}_k, \dot{y}_k$. The extent states consist of B radial values from the IRP and its first and second time derivatives.

VI. Performance Validation

The performance of the proposed method is validated using simulated and real data. The estimates of the proposed method are compared with the GP-EKF estimates [1] over 100 Monte Carlo runs for the simulated experiments. The performance evaluation parameters are the positional and velocity root mean square errors (RMSE) of the CoO, the mean shape precision P_μ and the mean shape recall R_μ. These are defined below:

$$RMSE_a = \sqrt{\frac{1}{K}\sum_{j=1}^{K}\frac{1}{N_{MC}}\sum_{i=1}^{N_{MC}}(a_j^i - \hat{a}_j^i)^2}, \tag{84}$$

$$R_\mu = \frac{1}{K}\sum_{j=1}^{K}\frac{1}{N_{MC}}\sum_{i=1}^{N_{MC}}\frac{Area(T_j^i \cap E_j^i)}{Area(T_j^i)}, \tag{85}$$

$$P_\mu = \frac{1}{K}\sum_{j=1}^{K}\frac{1}{N_{MC}}\sum_{i=1}^{N_{MC}}\frac{Area(T_j^i \cap E_j^i)}{Area(E_j^i)}, \tag{86}$$

where $RMSE_a$ represents the RMSE of the parameter a, a_j^i represents the true and $\hat{a}_j^i$ represents the estimated value, T_j^i represents the true shape, E_j^i represents the estimated shape, $\cap$ represents the intersection of two star-convex polygons and $Area(p)$ represents the area of the polygon p. The recall specifies how much of the true shape has been recalled while the precision evaluates the false (not belonging to true object) area. These parameters have been used to evaluate estimators in computer vision for rectangular objects estimation problems [48]. The percentage improvement compared to GP-EKF is also given in the results section. If $RMSE_a$, R_μ or P_μ of the GP-EKF is represented by vector $\boldsymbol{b}$ and those of STGP-EKF and STGP-RTSS by $\boldsymbol{c}$, then the corresponding percentage improvement $\boldsymbol{d}$ and the mean percentage improvement $\boldsymbol{d}_\mu$ are given below:

$$\boldsymbol{d} = \frac{\boldsymbol{b}-\boldsymbol{c}}{\boldsymbol{b}}, \quad \boldsymbol{d}_\mu = \frac{\boldsymbol{d}}{K} \times 100. \tag{87}$$

A. Simulation Results

The IRP motion model of the simulated object and the estimators is CV with matched process noise variance $q_x = q_y = 1$. Five different shape evolutions are simulated using two shape models active at different time samples for $K = 250$ time samples. These are the Singer acceleration model [49] and a constant shape model. The shape of the object does not change and the time derivatives of the radial states are zero when the constant shape model is active. The Singer model is active for the time samples in the range $k = [(1-50), (80-130), (180-230)]$ and the constant model is active at all other times. The parameters of the Singer model are maneuver variance $\sigma_m^2 = 12$ and maneuver time constant $\tau = 1s$. The shape model and the parameters for simulation are different from the model in filter and smoother. The switching and mismatched shape models further validate the robustness of the proposed method. The different shape evolutions simulated are explained in Table I.

The number of keypoints is $B = 24$, the sample time is $T_s = \frac{1}{30}s$, the spatial length-scale is $l_\theta = 15°$, the prior radial variance is $\sigma_r^2 = 1$, the spatial correlation magnitude variance is $\sigma_f^2 = 1$, the periodicity is $p = 2$, the temporal length-scale is

TABLE I
NON-RIGID SHAPE MODELS

Non-rigid object model	Description
S1 Triangle	A triangular object randomly changes its size.
S2 Quadrilateral	A quadrilateral object randomly changes its size.
S3 Regular Dodecagon	An almost circular object (regular 12-gon) randomly changes its size.
S4 Axis-symmetric Dodecagon	A 12-gon object adapts random axis-symmetric shapes.
S5 Asymmetric Dodecagon	A 12-gon object adapts random asymmetric shapes.

TABLE II
RMSE FROM SIMULATED DATA

States	Methods	Shape Models				
		S1	S2	S3	S4	S5
x (m)	STGP-EKF	0.23	0.12	0.08	0.14	0.12
	STGP-RTSS	0.11	0.06	0.04	0.07	0.07
y (m)	STGP-EKF	0.34	0.14	0.09	0.16	0.13
	STGP-RTSS	0.17	0.06	0.04	0.09	0.07
$\dot{x}$ (m/s)	STGP-EKF	1.11	0.84	0.75	0.91	0.90
	STGP-RTSS	0.35	0.29	0.25	0.29	0.35
$\dot{y}$ (m/s)	STGP-EKF	1.07	0.82	0.74	0.89	0.90
	STGP-RTSS	0.35	0.29	0.25	0.30	0.34
P	STGP-EKF	0.98	0.99	1.00	0.99	0.99
	STGP-RTSS	0.97	0.99	0.99	0.99	0.99
R	STGP-EKF	0.96	0.98	0.99	0.98	0.98
	STGP-RTSS	0.98	0.99	0.99	0.99	0.99

$l_t = 2s$, the temporal correlation magnitude is $\sigma_t^2 = 1$. The GP-EKF forgetting factor is tuned to $\alpha = 0.001$. The sensor error standard deviations are $\sigma_r = 0.25m$ for range and $\sigma_\theta = 0.25^\circ$ for angle. The number of measurements is Poisson distributed with mean $\lambda_m = 20$. The measurements are located randomly over the contour of the object using a uniform distribution.

a) Results: The RMSE values and the percentage improvement from 100 Monte Carlo runs for the five scenarios is given in the Tables II and III. The tables show that the performance of the STGP-EKF and STGP-RTSS is improved in all five cases. Fig. 6 shows the snapshots of tracking of a single simulation run of the five scenarios at the selected time steps. It can again be observed that the GP-EKF shape estimates are less accurate as compared to both the STGP-EKF and the STGP-RTSS estimates except for S3 (simplest shape model), where they are comparable.

TABLE III
MEAN PERCENTAGE IMPROVEMENT (SIMULATIONS)

States	Methods compared	Shape Models				
		S1	S2	S3	S4	S5
x	STGP-EKF	85.81	86.87	79.85	83.49	85.76
	STGP-RTSS	92.09	93.47	89.66	90.89	91.47
y	STGP-EKF	77.91	84.74	73.82	83.90	84.17
	STGP-RTSS	87.55	92.47	87.15	91.50	90.86
$\dot{x}$	STGP-EKF	91.98	90.23	61.90	88.40	87.96
	STGP-RTSS	97.08	96.99	87.53	96.42	95.53
$\dot{y}$	STGP-EKF	91.79	90.43	62.69	88.43	87.85
	STGP-RTSS	96.60	96.90	87.40	96.50	95.51
P	STGP-EKF	33.27	14.71	2.01	9.33	8.51
	STGP-RTSS	32.55	14.12	1.70	8.90	8.13
R	STGP-EKF	18.02	11.21	0.69	7.20	6.59
	STGP-RTSS	19.62	12.28	1.52	8.20	7.56

B. Effect of the STGP-RTSS Lag Value

The performance of the fixed-lag smoother is evaluated using the shape model S5. The performance is evaluated at different lag values for 100 Monte Carlo runs. The smoother lag k_s is chosen less than, equal to and more than the true temporal correlation length-scale l_t. The results are given in Fig. 7. It can be observed that the performance of the smoother is degraded for $k_s < l_t$. However, the smoother performance is comparable for the cases $k_s >= l_t$. Keeping in mind the computational advantage gained by keeping the lag smaller, as proposed, the $k_s = l_t$ is a reasonable trade-off value for the smoother lag. The peaks in the graphs are observed at time samples when the shape model switches between the Singer and the constant model.

C. Computational Complexity

The computational complexity of the STGP-EKF and STGP-RTSS scale as $\mathcal{O}(N_k^3 B + N_k^2 B^3)$and $\mathcal{O}(k_s B^3)$, respectively. The empirical results with respect to the three variables B, N_k and k_s are shown in Figs. 8, 9 and 10, respectively. The program was run on MATLAB R2016b on a Windows 10 (64 bit) Desktop computer installed with an Intel(R) Core(TM) i5-6500 CPU @ 3.20 GHz (4 CPUs) and 8 GB RAM. B and k_s are the model parameters and can be managed during the design phase. The number of extent states, B, can be decreased in the model according to the available computational resources. At the end of each time-step, the object shape can be constructed as per the requirement using the standard GP prediction (4) and (5). If the object shape is constructed at B_0 angles, the increase in computational expense due to this operation is $B_0 B$. Similarly, k_s can be reduced according to the available computational resources. The third variable, N_k, is dependent on the sensor, the object and other environmental conditions. The processing time can be further reduced through faster code implementation in C++.

D. Real Data

In this section, the proposed method is evaluated on real data presented in [50]. This is a thermal video data of pedestrians and vehicles sampled at 10Hz obtained using a fixed camera in an open environment. Three different video samples are chosen for evaluation which are a motorcycle, a rickshaw and a pedestrian. The rickshaw appears as a regular rigid object, the motorcycle as an irregular rigid object and the pedestrian as an irregular non-rigid object. The ground truth data is not available and is manually generated by marking the object contour (as precisely as possible) in each frame, calculating the CoO location in each frame and the CoO kinematics are determined using the CoO locations of consecutive frames. The video frames are pre-processed using frame differencing and median filtering to generate contour measurements. The following parameters are changed for the real data experiments; $B = 48$, $l_t = 1$, $\sigma_f^2 = 30$ and $q_x = q_y = 50$ for the STGP-EKF / STGP-RTSS and $B = 48$, $\sigma_f^2 = 2$ and $q_x = q_y = 10$ for the GP-EKF.

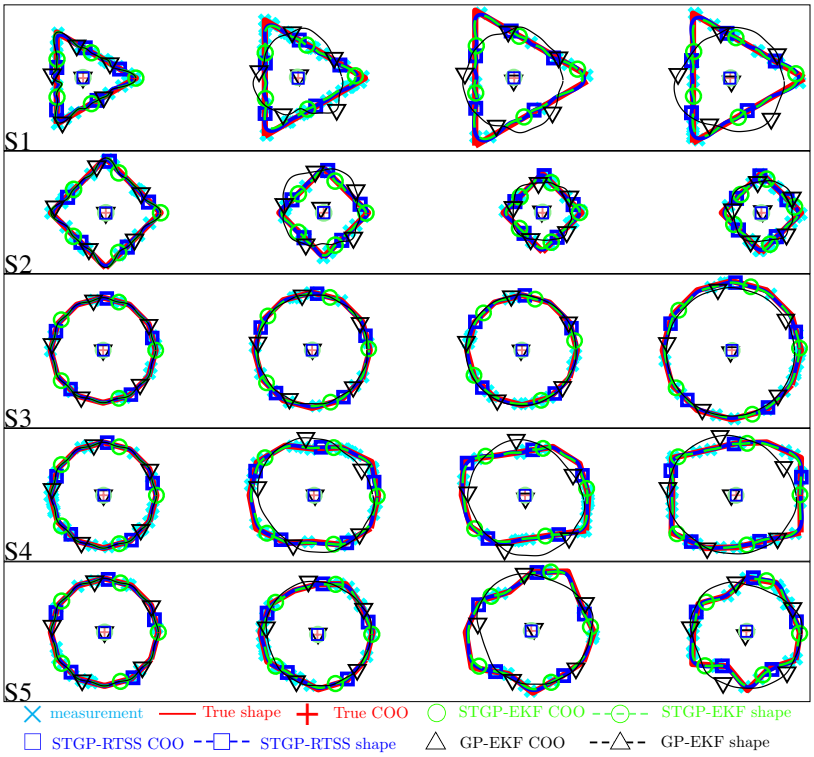


Fig. 6. Simulated shapes at $k = 1, 50, 150, 230$. The figure shows snapshots at selected time samples of the five different shape evolutions. The true CoO (red plus) and the shape (red solid line) along with the corresponding estimates are also presented in the figure. It can be observed that except for the S3 the shape estimates of the proposed method are improved as compared to the method [1].

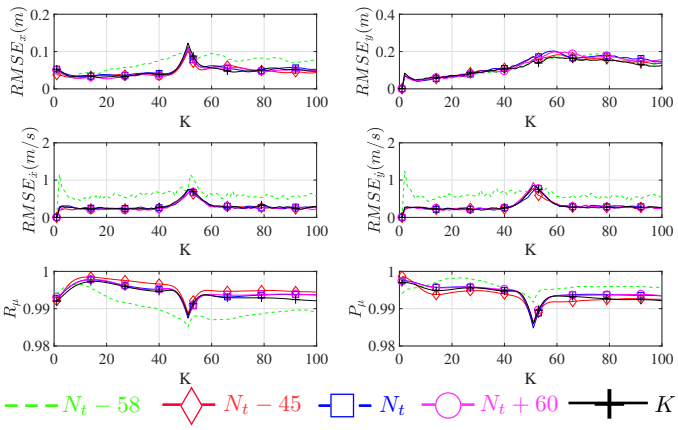


Fig. 7. **STGP-RTSS performance at different lag values**. The figure shows the comparison of smoother performance at different lag values compared to the true temporal correlation l_t. The best performance is given by the smoother with lag $k_s = K$ that is a full STGP regression. It can be observed that values less than l_t provide degraded performance while the performance is almost similar for $k_s >= l_t$.

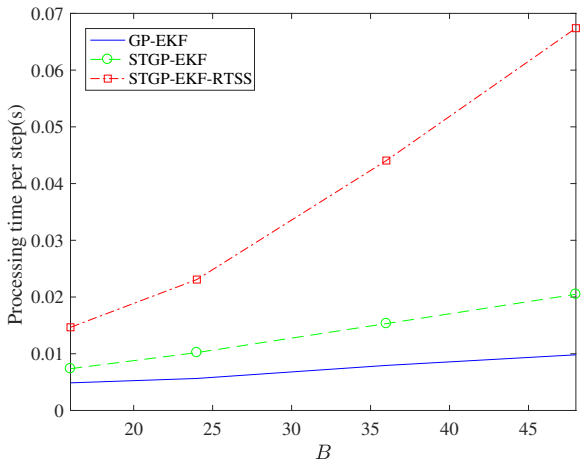


Fig. 8. Effect of B on the processing time. The figure shows a comparison of time taken per time sample by increasing B. The extent state vector of the STGP model is three times the size of the GP-EKF vector. Hence, the STGP-EKF and the STGP-EKF-RTSS require more processing time. As B increases, the processing time of the STGP-EKF-RTSS rises at a much faster rate compared to the filters due to the RTSS recursion.

Results: The RMSE and the percentage improvement of all three scenarios are given in Tables IV and V. It can be observed that the performance of the proposed approach is comparable to the reference (GP-EKF) method while tracking a regularly shaped rigid object (rickshaw). As observed in the simulated experiments, there is a significant improvement in performance while tracking irregularly shaped objects, especially when the object shape is also changing (pedestrian). The snapshots at three different samples is given in Fig. 11. It can again be observed that the shape estimates (especially the precision) are significantly improved for non-rigid objects.

VII. Conclusions

A novel and generic model has been proposed to track a non-rigid extended or group object based on STGP. Real-time filtering and smoothing STGP approaches are presented, along with the theoretical derivations. An improved tracking efficiency is demonstrated compared to the reference method [1] using simulated data with more than 90% improvement in

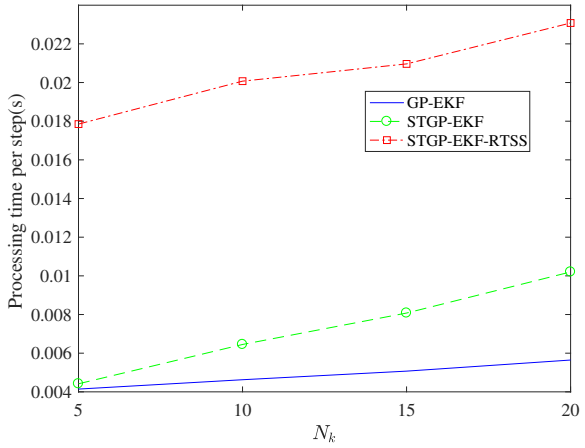


Fig. 9. Effect of the N_k on the processing time. The figure shows a comparison of time taken per time sample by increasing N_k. The computational cost of the STGP-EKF increases at a higher rate as compared to the GP-EKF. The STGP-EKF-RTSS computational cost is not dependent on the N_k and hence the plot follows the a similar slope to STGP-EKF with a vertical shift equal to the time required for RTSS recursion.

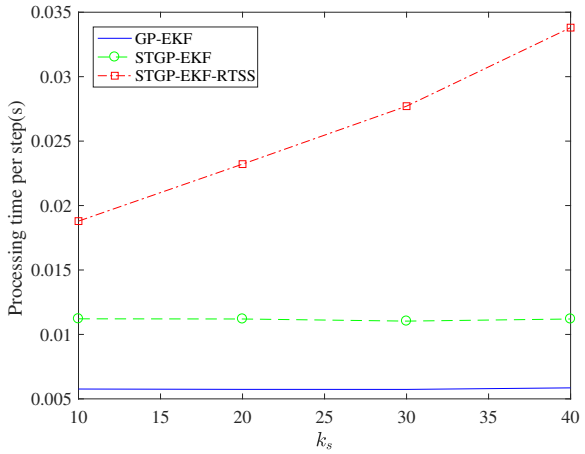


Fig. 10. Effect of the k_s on the processing time. The figure shows a comparison of time taken per time sample by increasing k_s. The computational complexity of the filter is independent of k_s while that of RTSS increases exponentially.

TABLE IV
RMSE FROM REAL DATA

States	Methods	Scenarios		
		Rickshaw	Motorcycle	Pedestrian
x (p)	STGP-EKF	1.68	1.74	1.63
	STGP-RTSS	1.11	1.05	1.47
y (p)	STGP-EKF	2.31	1.50	3.95
	STGP-RTSS	0.84	1.08	4.00
$\dot{x}$ (p/s)	STGP-EKF	9.13	11.43	6.31
	STGP-RTSS	7.79	7.87	5.51
$\dot{y}$ (p/s)	STGP-EKF	8.35	10.43	14.87
	STGP-RTSS	6.08	8.19	13.41
P	STGP-EKF	0.99	0.96	0.83
	STGP-RTSS	0.97	0.93	0.76
R	STGP-EKF	0.81	0.81	0.84
	STGP-RTSS	0.89	0.89	0.91

TABLE V
MEAN PERCENTAGE IMPROVEMENT (REAL DATA)

States	Methods compared	Scenarios		
		Rickshaw	Motorcycle	Pedestrian
x	STGP-EKF	30.37	89.18	89.56
	STGP-RTSS	54.19	93.45	90.58
y	STGP-EKF	-22.16	88.29	43.96
	STGP-RTSS	55.82	91.53	43.32
$\dot{x}$	STGP-EKF	27.11	82.52	91.11
	STGP-RTSS	37.78	87.96	92.23
$\dot{y}$	STGP-EKF	37.84	83.04	64.81
	STGP-RTSS	54.75	86.69	68.26
P	STGP-EKF	7.58	75.81	135.06
	STGP-RTSS	5.95	71.67	115.54
R	STGP-EKF	-8.94	16.23	0.20
	STGP-RTSS	0.60	27.90	9.23

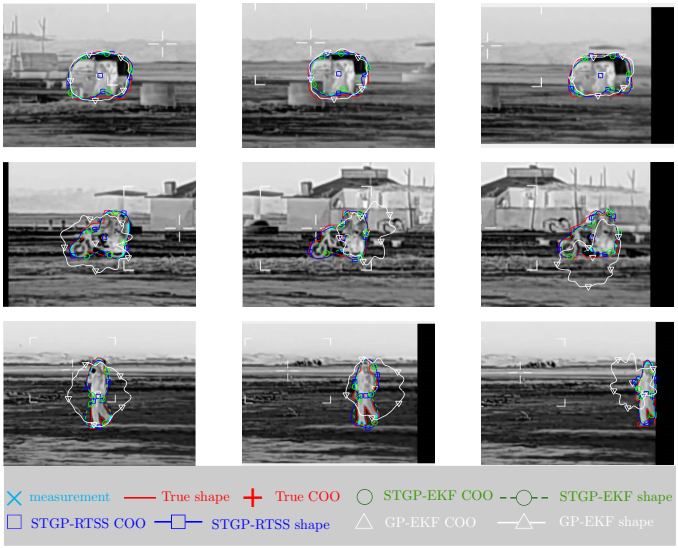


Fig. 11. Snapshots of three time samples. The figure shows snapshots at selected time samples of the three scenarios, that is rickshaw (top), motorcyclist (middle) and the pedestrian (bottom). The ground truth and the estimates from the STGP-EKF, STGP-RTSS and GP-EKF are also shown. It can be observed that the shape estimates of the STGP based models are improved as compared to the GP-EKF.

the accuracy in position, 95% in velocity and 7% in the shape for the tracking of an asymmetric non-rigid object. The performance improvement to track a non-rigid real object (pedestrian) is up to 43% in position, 68% in velocity, 10% in the recall and 115% in the precision. For complicated nonlinear scenarios, advanced nonlinear filters and smoothers can be derived for the same model using similar steps. Being a general model, it can be applied to solve various real-world problems. The model can also be extended to 3D scenarios.

APPENDIX A
SENSOR MEASUREMENTS COORDINATE CONVERSIONS

This Appendix presents the coordinate converted measurement pdfs as derived in [51]. It presents a geometrical approximation to an i.i.d. Gaussian pdf which undergoes the following transformations; polar to Cartesian, translation, and then back from the translated Cartesian to polar. Another approximate approach is presented using unscented transforms for the above mentioned transformations in [52]. In the Appendix of [51], the author derives the pdf for polar to Cartesian case using mathematical identities, which is exact.

A biased conversion degrades the filter performance [53]. The unbias coordinate conversion has also been proposed namely unbiased converted measurement (UCM) in [54], for polar to Cartesian conversion. In [55], an incompatibility in the UCM derivation was highlighted and removed. The corrected conversion was named modified UCM (MUCM), which was later verified through experiments in [56], [57]. The MUCM conversion is exactly same as proposed in the Appendix of [51]. In this paper, we use the geometric approximate conversion proposed in [51], as the approximation is valid for low sensor noise, which is often the case in EOT / GOT applications.

The sensor measurement noise is modeled i.i.d. Gaussian with variances $\sigma^2_{\tilde{\psi}_{i,k}}$ and $\sigma^2_{\tilde{\phi}_{i,k}}$. The sensor measurement pdf in polar global frame is $\tilde{\boldsymbol{z}}^{sen,p}_{i,k} = \mathcal{N}(\boldsymbol{\mu}^{sen,p}_{i,k}, \boldsymbol{R}^{sen,p}_{i,k})$ with $\tilde{\boldsymbol{z}}^{sen,p}_{i,k} = [\tilde{\psi}_{i,k}, \tilde{\phi}_{i,k}]^T$, $\boldsymbol{\mu}^{sen,p}_{i,k} = [\bar{\psi}_{i,k}, \bar{\phi}_{i,k}]^T$ and $\boldsymbol{R}^{sen,p}_{i,k} = \text{diag}(\sigma^2_{\tilde{\psi}_{i,k}}, \sigma^2_{\tilde{\phi}_{i,k}})$. The corresponding sensor measurement pdf in Cartesian global frame is $\tilde{\boldsymbol{z}}^{sen,c}_{i,k} = \mathcal{N}(\boldsymbol{\mu}^{sen,c}_{i,k}, \boldsymbol{R}^{sen,c}_{i,k})$ with $\tilde{\boldsymbol{z}}^{sen,c}_{i,k} = [\tilde{x}_{i,k}, \tilde{y}_{i,k}]^T$, $\boldsymbol{\mu}^{sen,c}_{i,k} = [\bar{x}_{i,k}, \bar{y}_{i,k}]^T$, $\boldsymbol{R}^{sen,c}_{i,k} = \begin{bmatrix} \sigma^2_{\tilde{x}_{i,k}} & \sigma^2_{\tilde{x}_{i,k}\tilde{y}_{i,k}} \\ \sigma^2_{\tilde{y}_{i,k}\tilde{x}_{i,k}} & \sigma^2_{\tilde{y}_{i,k}} \end{bmatrix}$, $\lambda_b = \exp\left(\frac{-\tilde{\phi}^2_{i,k}}{2}\right)\bar{\psi}_{i,k}$ and

$$\bar{x}_{i,k} = \lambda_b \cos\bar{\phi}_{i,k}, \bar{y}_{i,k} = \lambda_b \sin\bar{\phi}_{i,k}, \tag{88}$$

$$\sigma^2_{\tilde{x}_{i,k}} = \frac{1}{2}(\bar{\psi}^2_{i,k} + \sigma^2_{\tilde{\psi}_{i,k}})[1 + \cos(2\bar{\phi}_{i,k})\exp(-2\sigma^2_{\tilde{\phi}_{i,k}})] - \exp(\sigma^2_{\tilde{\phi}_{i,k}})\bar{\psi}^2_{i,k}\cos^2\bar{\phi}_{i,k} \tag{89}$$

$$\sigma^2_{\tilde{y}_{i,k}} = \frac{1}{2}(\bar{\psi}^2_{i,k} + \sigma^2_{\tilde{\psi}_{i,k}})[1 - \cos(2\bar{\phi}_{i,k})\exp(-2\sigma^2_{\tilde{\phi}_{i,k}})] - \exp(\sigma^2_{\tilde{\phi}_{i,k}})\bar{\psi}^2_{i,k}\sin^2\bar{\phi}_{i,k} \tag{90}$$

$$\sigma^2_{\tilde{x}_{i,k}\tilde{y}_{i,k}} = \frac{1}{2}(\bar{\psi}^2_{i,k} + \sigma^2_{\tilde{\psi}_{i,k}})[\sin(2\bar{\phi}_{i,k})\exp(-2\sigma^2_{\tilde{\phi}_{i,k}})] - \exp(\sigma^2_{\tilde{\phi}_{i,k}})\bar{\psi}^2_{i,k}\cos\bar{\phi}_{i,k}\sin\bar{\phi}_{i,k} \tag{91}$$

Suppose a $[-x_k, -y_k]^T$ translation is applied to $\tilde{\boldsymbol{z}}^{sen,c}_{i,k}$ to obtain $\tilde{\boldsymbol{z}}^{obj,c}_{i,k} = [\tilde{x}^t_{i,k}, \tilde{y}^t_{i,k}]^T$ where $\tilde{x}^t_{i,k} = \bar{x}_{i,k} - x_k + \nu_{\tilde{x}_{i,k}} = \breve{x}_{i,k} + \nu_{\tilde{x}_{i,k}}$, $\tilde{y}^t_{i,k} = \bar{y}_{i,k} - y_k + \nu_{\tilde{y}_{i,k}} = \breve{y}_{i,k} + \nu_{\tilde{y}_{i,k}}$ and $\boldsymbol{\nu}_{i,k} = [\nu_{\tilde{x}_{i,k}}, \nu_{\tilde{y}_{i,k}}]^T = \mathcal{N}(0, \boldsymbol{R}^{sen,c}_{i,k})$ represents measurement noise. The measurement pdf after converting the translated vector to the polar coordinates is approximated to a Gaussian $\tilde{\boldsymbol{z}}^{obj,p}_{i,k} = \mathcal{N}(\boldsymbol{\mu}^{obj,p}_{i,k}, \boldsymbol{R}^{obj,p}_{i,k})$ with $\tilde{\boldsymbol{z}}^{obj,p}_{i,k} = [\tilde{r}_{i,k}, \tilde{\theta}_{i,k}]^T$, $\boldsymbol{\mu}^{obj,p}_{i,k} = [\bar{r}_{i,k}, \bar{\theta}_{i,k}]^T$ and $\boldsymbol{R}^{obj,p}_{i,k} = \begin{bmatrix} \sigma^2_{\tilde{r}_{i,k}} & \sigma^2_{\tilde{r}_{i,k}\tilde{\theta}_{i,k}} \\ \sigma^2_{\tilde{\theta}_{i,k}\tilde{r}_{i,k}} & \sigma^2_{\tilde{\theta}_{i,k}} \end{bmatrix}$ where

$$\sigma^2_{\tilde{r}_{i,k}} = \sigma^2_{\tilde{x}_{i,k}}\cos^2(\bar{\theta}_{i,k}) + \varrho_{i,k} + \sigma^2_{\tilde{y}_{i,k}}\sin^2(\bar{\theta}_{i,k}) \tag{92}$$

$$\sigma^2_{\tilde{\theta}_{i,k}} = \frac{\sigma^2_{\tilde{x}_{i,k}}\sin^2(\bar{\theta}_{i,k}) - \varrho_{i,k} + \sigma^2_{\tilde{y}_{i,k}}\cos^2(\bar{\theta}_{i,k})}{\bar{r}^2_{i,k}} \tag{93}$$

$$\varrho_{i,k} = 2\sigma_{\tilde{x}_{i,k}\tilde{y}_{i,k}}\cos(\bar{\theta}_{i,k})\sin(\bar{\theta}_{i,k}) \tag{94}$$

$$\rho_{\tilde{r}_{i,k}\tilde{\theta}_{i,k}} = \frac{(-\sigma^2_{\tilde{x}_{i,k}} + \sigma^2_{\tilde{y}_{i,k}})\sin(2\bar{\theta}_{i,k}) + 2\sigma_{\tilde{x}_{i,k}\tilde{y}_{i,k}}\cos(2\bar{\theta}_{i,k})}{2\sigma_{\tilde{r}_{i,k}}\sigma_{\tilde{\theta}_{i,k}}\bar{r}_{i,k}} \tag{95}$$

$$\sigma_{\tilde{r}_{i,k}\tilde{\theta}_{i,k}} = \sigma_{\tilde{\theta}_{i,k}\tilde{r}_{i,k}} = \rho_{\tilde{r}_{i,k}\tilde{\theta}_{i,k}}\sigma_{\tilde{r}_{i,k}}\sigma_{\tilde{\theta}_{i,k}}, \tag{96}$$

$$\bar{r}_{i,k} = \sqrt{\breve{x}^2_{i,k} + \breve{y}^2_{i,k}}, \quad \bar{\theta}_{i,k} = \tan^{-1}\left(\frac{\breve{y}_{i,k}}{\breve{x}_{i,k}}\right) \tag{97}$$

The above conversions are approximate and this approximation is valid in the central and near central regions. If the angular error is $\sigma_{\tilde{\phi}} = 0.5\,\text{deg}$, then the approximation becomes invalid at $10\sigma_{\tilde{\phi}}$. Similarly, if $\frac{\sigma_{\tilde{\psi}}}{\psi} = 0.01$, then 5% error occurs at $5\sigma_{\tilde{\psi}}$. The sensor errors in the EOT / GOT applications are generally lower and the above approximation remains valid.

Appendix B
Probability Density Function of $\tilde{\boldsymbol{\zeta}}_{i,k}$

Given $\boldsymbol{\zeta}_{i,k} = \left[\cos(\tilde{\theta}_{i,k}), \sin(\tilde{\theta}_{i,k})\right]^T$, the Gaussian approximation of the pdf of $\boldsymbol{\zeta}_{i,k}$ is derived in this Appendix. Suppose, a cosine transformation is applied to a standard normal distribution $\beta \sim \mathcal{N}(0, \sigma^2_\beta)$. According to Euler's formula $\exp(\iota\beta) = \cos\beta + \iota\sin\beta$ and $\mathbb{E}[\exp(\iota\beta)] = \exp\left(-\frac{\sigma^2_\beta}{2}\right)$ where $\mathbb{E}[\cdot]$ represents the mathematical expectation operator. Also $\mathbb{E}[e^{\iota\beta}] = \mathbb{E}[\cos\beta + \iota\sin\beta] = \mathbb{E}[\cos\beta] + \iota\mathbb{E}[\sin\beta]$. As a result, the real and imaginary parts can be equated as $\Re\{\mathbb{E}[e^{\iota\beta}]\} = \exp\left(-\frac{\sigma^2_\beta}{2}\right) = \mathbb{E}[\cos\beta]$ and $\Im\{\mathbb{E}[e^{\iota\beta}]\} = 0 = \mathbb{E}[\sin\beta]$, respectively, where $\Re\{.\}$ and $\Im\{.\}$ represent the real and imaginary parts of the variable. Now consider cosine and sine transformations applied to $\tilde{\theta}_{i,k} \sim \mathcal{N}(\bar{\theta}_{i,k}, \sigma^2_{\tilde{\theta}_{i,k}})$ with $\sigma^2_{\tilde{\theta}_{i,k}} = \sigma^2_\beta$, $\mathcal{C}_{i,k} = \cos(\tilde{\theta}_{i,k})$ and $\mathcal{S}_{i,k} = \sin(\tilde{\theta}_{i,k})$. Given that $\beta = \tilde{\theta}_{i,k} - \bar{\theta}_{i,k}$, the mean and variances are approximated as follows:

$$\mu_{\mathcal{C}_{i,k}} = \mathbb{E}[\cos(\tilde{\theta}_{i,k})] = e^{-\frac{\sigma^2_{\tilde{\theta}_{i,k}}}{2}}\cos\bar{\theta}_{i,k}, \tag{98}$$

$$\mu_{\mathcal{C}^2_{i,k}} = \mathbb{E}[\cos^2(\tilde{\theta}_{i,k})] = \frac{1}{2} + \frac{1}{2}e^{-2\sigma^2_{\tilde{\theta}_{i,k}}}\cos 2\bar{\theta}_{i,k}, \tag{99}$$

$$\sigma^2_{\mathcal{C}_{i,k}} = \mathbb{E}[\cos^2\tilde{\theta}_{i,k}] - (\mathbb{E}[\tilde{\theta}_{i,k}])^2 = \frac{1}{2} + \frac{1}{2}e^{-2\sigma^2_{\tilde{\theta}_{i,k}}}\cos 2\bar{\theta}_{i,k} - e^{-\sigma^2_{\tilde{\theta}_{i,k}}}\cos^2\bar{\theta}_{i,k},$$

$$\mu_{\mathcal{S}_{i,k}} = e^{-\frac{\sigma^2_{\tilde{\theta}_{i,k}}}{2}}\sin\bar{\theta}_{i,k}, \tag{100}$$

$$\sigma^2_{\mathcal{S}_{i,k}} = \frac{1}{2} - \frac{1}{2}e^{-2\sigma^2_{\tilde{\theta}_{i,k}}}\cos 2\bar{\theta}_{i,k} - e^{-\sigma^2_{\tilde{\theta}_{i,k}}}\sin^2\bar{\theta}_{i,k}, \tag{101}$$

$$\sigma^2_{\mathcal{C}_{i,k}\mathcal{S}_{i,k}} = \sigma^2_{\mathcal{S}_{i,k}\mathcal{C}_{i,k}} = \mathbb{E}[\{\cos(\tilde{\theta}_{i,k}) - \mathbb{E}(\cos(\tilde{\theta}_{i,k}))\}\times \{\sin(\tilde{\theta}_{i,k}) - \mathbb{E}(\sin(\tilde{\theta}_{i,k}))\}] = 0, \tag{102}$$

where $\mu_{\mathcal{C}_{i,k}}$ and $\mu_{\mathcal{S}_{i,k}}$ represent the mean, $\sigma^2_{\mathcal{C}_{i,k}}$ and $\sigma^2_{\mathcal{S}_{i,k}}$ represent the variances and $\sigma^2_{\mathcal{C}_{i,k}\mathcal{S}_{i,k}}$ and $\sigma^2_{\mathcal{S}_{i,k}\mathcal{C}_{i,k}}$ represent the covariances. Using above, the pdf can be approximated to a Gaussian $\tilde{\boldsymbol{\zeta}}_{i,k} \sim \mathcal{N}(\bar{\boldsymbol{\zeta}}_{i,k}, \boldsymbol{R}^{\tilde{\boldsymbol{\zeta}}}_{i,k})$ where:

$$\bar{\boldsymbol{\zeta}}_{i,k} = \begin{bmatrix} \mu_{\mathcal{C}_{i,k}} \\ \mu_{\mathcal{S}_{i,k}} \end{bmatrix}, \boldsymbol{R}^{\tilde{\boldsymbol{\zeta}}}_{i,k} = \begin{bmatrix} \sigma^2_{\mathcal{C}_{i,k}} & \sigma^2_{\mathcal{C}_{i,k}\mathcal{S}_{i,k}} \\ \sigma^2_{\mathcal{S}_{i,k}\mathcal{C}_{i,k}} & \sigma^2_{\mathcal{S}_{i,k}} \end{bmatrix}. \tag{103}$$

The approximation is valid in central and near central regions as explained in the Appendix A.

Acknowledgment

We acknowledge the support from the Pakistan Air Force (Govt. of Pakistan) and the Dept. of ACSE (University of Sheffield) for granting scholarships to the first author. We appreciate the support of the SETA project: An open, sustainable, ubiquitous data and service for efficient, effective, safe, resilient mobility in metropolitan areas funded from

the European Unions Horizon 2020 research and innovation programme under grant agreement no. 688082. We are thankful to the Associate Editor and anonymous reviewers helping us to improve this paper.

REFERENCES

[1] N. Wahlström and E. Özkan, "Extended target tracking using Gaussian processes," *IEEE Transactions on Signal Processing*, vol. 63, no. 16, pp. 4165–4178, 2015.

[2] A. Kräußling, *Intelligent Techniques and Tools for Novel System Architectures*. Springer Berlin Heidelberg, 2008, ch. Tracking Extended Moving Objects with a Mobile Robot, pp. 513–530.

[3] F. Faion, M. Baum, and U. D. Hanebeck, "Tracking 3D shapes in noisy point clouds with random hypersurface models," in *Proc. of the 15th International Conf. on Information Fusion*. IEEE, 2012, pp. 2230–2235.

[4] V. Edman, M. Andersson, K. Granstrom, and F. Gustafsson, "Pedestrian group tracking using the GM-PHD filter," in *Proc. of the 21st European Signal Processing Conf. (EUSIPCO)*. IEEE, 2013, pp. 1–5.

[5] F. Septier, A. Carmi, and S. Godsill, "Tracking of multiple contaminant clouds," in *Proc. of the 12th International Conf. on Information Fusion*. IEEE, 2009, pp. 1280–1287.

[6] 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 Transactions on Geoscience and Remote Sensing*, vol. 53, no. 12, pp. 6617–6631, 2015.

[7] F. Kunz, D. Nuss, J. Wiest, H. Deusch, S. Reuter, F. Gritschneder, A. Scheel, M. Stübler, M. Bach, P. Hatzelmann, and C. Wild, "Autonomous driving at Ulm University: A modular, robust, and sensor-independent fusion approach," in *Proc. of the IEEE Intelligent Vehicles Symposium (IV)*. IEEE, 2015, pp. 666–673.

[8] J. Han, D. Kim, M. Lee, and M. Sunwoo, "Enhanced road boundary and obstacle detection using a downward-looking LIDAR sensor," *IEEE Transactions on Vehicular Technology*, vol. 61, no. 3, pp. 971–985, 2012.

[9] K. Kodagoda, S. S. Ge, W. S. Wijesoma, and A. P. Balasuriya, "IMMPDAF approach for road-boundary tracking," *IEEE transactions on vehicular technology*, vol. 56, no. 2, pp. 478–486, 2007.

[10] M. Michaelis, P. Berthold, D. Meissner, and H.-J. Wuensche, "Heterogeneous multi-sensor fusion for extended objects in automotive scenarios using Gaussian processes and a GMPHD-filter," in *Sensor Data Fusion: Trends, Solutions, Applications (SDF), 2017*. IEEE, 2017, pp. 1–6.

[11] S. Blackman and R. Popoli, "Design and analysis of modern tracking systems (book)," *Norwood, MA: Artech House, 1999.*, 1999.

[12] Y. Bar-Shalom, X. R. Li, and T. Kirubarajan, *Estimation with applications to tracking and navigation: theory algorithms and software*. John Wiley & Sons, 2004.

[13] J. W. Koch, "Bayesian approach to extended object and cluster tracking using random matrices," *IEEE Transactions on Aerospace and Electronic Systems*, vol. 44, no. 3, pp. 1042–1059, July 2008.

[14] 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, April 2011.

[15] L. Mihaylova, A. Y. Carmi, F. Septier, A. Gning, S. K. Pang, and S. Godsill, "Overview of Bayesian sequential Monte Carlo methods for group and extended object tracking," *Digital Signal Processing*, vol. 25, pp. 1–16, 2014.

[16] K. Granstrom, M. Baum, and S. Reuter, "Extended object tracking: Introduction, overview and applications," *ISIF Journal of Advances in Information Fusion*, vol. 12, no. 2, pp. 139–174, 2017.

[17] X. R. Li and V. P. Jilkov, "Survey of maneuvering target tracking. Part I. Dynamic models," *IEEE Transactions on aerospace and electronic systems*, vol. 39, no. 4, pp. 1333–1364, 2003.

[18] ——, "Survey of maneuvering target tracking. Part V. Multiple-model methods," *IEEE Transactions on Aerospace and Electronic Systems*, vol. 41, no. 4, pp. 1255–1321, 2005.

[19] L.-j. Chi, X.-x. Feng, and L. Miao, "Generalized labeled multi-Bernoulli extended target tracking based on Gaussian process regression," in *MATEC Web of Confs.*, vol. 176. EDP Sciences, 2018, p. 01017.

[20] C. E. Rasmussen and C. K. Williams, *Gaussian processes for machine learning*. MIT press Cambridge, 2006, vol. 1.

[21] T. Hirscher, A. Scheel, S. Reuter, and K. Dietmayer, "Multiple extended object tracking using Gaussian processes," in *Proc. of the 19th International Conf. on Information Fusion (FUSION)*. IEEE, 2016, pp. 868–875.

[22] S. Sarkka, A. Solin, and J. Hartikainen, "Spatiotemporal learning via infinite-dimensional Bayesian filtering and smoothing: A look at Gaussian process regression through Kalman filtering," *IEEE Signal Processing Magazine*, vol. 30, no. 4, pp. 51–61, 2013.

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

[24] S. Särkkä and J. Sarmavuori, "Gaussian filtering and smoothing for continuous-discrete dynamic systems," *Signal Processing*, vol. 93, no. 2, pp. 500–510, 2013.

[25] M. Baum, F. Faion, and U. D. Hanebeck, "Modeling the target extent with multiplicative noise," in *Proc. of the 15th International Conf. on Information Fusion*. IEEE, 2012, pp. 2406–2412.

[26] N. Petrov, L. Mihaylova, A. Gning, and D. Angelova, "A novel sequential Monte Carlo approach for extended object tracking based on border parameterisation," in *Proc. of the 14th International Conf. on Information Fusion*. IEEE, 2011, pp. 1–8.

[27] M. Baum, V. Klumpp, and U. D. Hanebeck, "A novel Bayesian method for fitting a circle to noisy points," in *Proc. of the 13th International Conf. on Information Fusion*. IEEE, 2010, pp. 1–6.

[28] A. De Freitas, L. Mihaylova, A. Gning, D. Angelova, and V. Kadirkamanathan, "Autonomous crowds tracking with box particle filtering and convolution particle filtering," *Automatica*, vol. 69, pp. 380–394, 2016.

[29] D. Angelova, L. Mihaylova, N. Petrov, and A. Gning, "A convolution particle filtering approach for tracking elliptical extended objects," in *Proc. of the 16th International Conf. on Information Fusion*. IEEE, 2013, pp. 1542–1549.

[30] G. Vivone, P. Braca, K. Granström, A. Natale, and J. Chanussot, "Converted measurements random matrix approach to extended target tracking using X-band marine radar data," in *Proc. of the 18th International Conf. on Information Fusion*. IEEE, 2015, pp. 976–983.

[31] 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. of the 17th International Conf. on Information Fusion*. IEEE, 2014, pp. 1–8.

[32] J. Lan and X. R. Li, "Tracking of maneuvering non-ellipsoidal extended object or target group using random matrix." *IEEE Trans. Signal Processing*, vol. 62, no. 9, pp. 2450–2463, 2014.

[33] 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, 2014.

[34] M. Feldmann and D. Franken, "Tracking of extended objects and group targets using matrices random–a new approach," in *Proc. of the 11th International Conf. on Information Fusion*. IEEE, 2008, pp. 1–8.

[35] R. Murray-Smith and A. Girard, "Gaussian Process priors with ARMA noise models," in *Irish Signals and Systems Conference, Maynooth*, 2001, pp. 147–152.

[36] P. J. Diggle, J. Tawn, and R. Moyeed, "Model-based geostatistics," *Journal of the Royal Statistical Society: Series C (Applied Statistics)*, vol. 47, no. 3, pp. 299–350, 1998.

[37] R. M. Neal, "Monte Carlo implementation of Gaussian process models for Bayesian regression and classification," *arXiv preprint physics/9701026*, 1997.

[38] R. E. Kalman, "A new approach to linear filtering and prediction problems," *Journal of basic Engineering*, vol. 82, no. 1, pp. 35–45, 1960.

[39] F. Daum, "Nonlinear filters: beyond the Kalman filter," *IEEE Aerospace and Electronic Systems Magazine*, vol. 20, no. 8, pp. 57–69, 2005.

[40] S. Särkkä and J. Hartikainen, "Infinite-dimensional Kalman filtering approach to spatio-temporal Gaussian process regression," in *International Conf. on Artificial Intelligence and Statistics*, 2012, pp. 993–1001.

[41] W. Aftab, A. De Freitas, M. Arvaneh, and L. Mihaylova, "A Gaussian process approach for extended object tracking with random shapes and for dealing with intractable likelihoods," in *Proc. of the 22nd International Conf. on Digital Signal Processing*. IEEE, 2017, pp. 1–5.

[42] A. Girard, C. E. Rasmussen, J. Q. Candela, and R. Murray-Smith, "Gaussian process priors with uncertain inputs-application to multiple-step ahead time series forecasting," *Advances in neural information processing systems*, pp. 545–552, 2003.

[43] A. Girard and R. Murray-Smith, "Gaussian processes: Prediction at a noisy input and application to iterative multiple-step ahead forecasting of time-series," *Lecture notes in computer science*, vol. 3355, pp. 158–184, 2005.

[44] P. Bourke, "Calculating the area and centroid of a polygon," *Swinburne Univ. of Technology*, 1988.

[45] M. S. Handcock and M. L. Stein, "A Bayesian analysis of kriging," *Technometrics*, vol. 35, no. 4, pp. 403–410, 1993.

[46] P. Guttorp and T. Gneiting, "On the Whittle-Matérn correlation family," *National Research Center for Statistics and the Environment-Technical Report Series, Seattle, Washington*, 2005.

[47] G. E. Uhlenbeck and L. S. Ornstein, "On the theory of the brownian motion," *Physical review*, vol. 36, no. 5, p. 823, 1930.

[48] C. Wolf and J.-M. Jolion, "Object count/area graphs for the evaluation of object detection and segmentation algorithms," *International Journal of Document Analysis and Recognition*, vol. 8, no. 4, pp. 280–296, 2006.

[49] R. A. Singer, "Estimating optimal tracking filter performance for manned maneuvering targets," *IEEE Transactions on Aerospace and Electronic Systems*, no. 4, pp. 473–483, 1970.

[50] A. Akula, R. Ghosh, S. Kumar, and H. Sardana, "Moving target detection in thermal infrared imagery using spatiotemporal information," *JOSA A*, vol. 30, no. 8, pp. 1492–1501, 2013.

[51] B. H. Cantrell, "Description of an alpha-beta filter in Cartesian coordinates," Naval research lab Washington, D.C., Tech. Rep., 1973.

[52] S. J. Julier and J. K. Uhlmann, "Consistent debiased method for converting between polar and Cartesian coordinate systems," in *Acquisition, Tracking, and Pointing XI*, vol. 3086. International Society for Optics and Photonics, 1997, pp. 110–122.

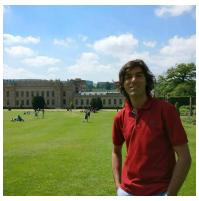
[53] T. Li, J. M. Corchado, J. Bajo, S. Sun, and J. F. De Paz, "Effectiveness of Bayesian filters: An information fusion perspective," *Information Sciences*, vol. 329, pp. 670–689, 2016.

[54] M. Longbin, S. Xiaoquan, Z. Yiyu, S. Z. Kang, and Y. Bar-Shalom, "Unbiased converted measurements for tracking," *IEEE Transactions on Aerospace and Electronic Systems*, vol. 34, no. 3, pp. 1023–1027, 1998.

[55] Z. Duan, C. Han, and X. R. Li, "Comments on " unbiased converted measurements for tracking"," *IEEE Transactions on Aerospace and Electronic Systems*, vol. 40, no. 4, p. 1374, 2004.

[56] S. V. Bordonaro, P. Willett, and Y. Bar-Shalom, "Unbiased tracking with converted measurements," in *Radar Conf. (RADAR), 2012 IEEE*. IEEE, 2012, pp. 0741–0745.

[57] S. Bordonaro, P. Willett, and Y. Bar-Shalom, "Decorrelated unbiased converted measurement Kalman filter," *IEEE Transactions on Aerospace and Electronic Systems*, vol. 50, no. 2, pp. 1431–1444, 2014.



Waqas Aftab received his B.Eng. (Avionics) degree from the College of Aeronautical Engineering (NUST), Pakistan and the MS (Control Systems Engineering) degree from Air University, Pakistan in 2005 and 2014, respectively. He is a Ph.D. research student at The University of Sheffield, United Kingdom. His main interests are multisensor multitarget tracking and Gaussian processes.

Dr. Roland Hostettler received the Dipl. Ing. degree in Electrical and Communication Engineering from Bern University of Applied Sciences, Switzerland in 2007, the M.Sc. in Electrical Engineering and Ph.D. in Automatic Control all from Lule University of Technology, Sweden in 2009 and 2014, respectively. He has held visiting positions at the Department of Electrical and Computer Engineering at Stony Brook University, NY, USA (2013, host: Prof. Petar Djuri) and the Department of Engineering at the University of Cambridge, UK (2017, host: Prof. Simon J. Godsill). Currently, he is a research fellow at the Department of Electrical Engineering and Automation at Aalto University, Finland. His research interests include statistical signal processing, probabilistic machine learning, and sensor systems.

Dr. Allan De Freitas received the B.Eng., B.Eng. (Hons) and M.Eng. degrees in Electronic Engineering from the University of Pretoria, South Africa in 2009, 2010 and 2013, respectively. In 2017 he obtained the Ph.D. degree in the Automatic Control and Systems Engineering department at the University of Sheffield, United Kingdom. His main scientific interests are in the areas of signal processing, object tracking, communications, image processing, and complex systems

Dr. Mahnaz Arvaneh received her B.Sc. degree in Control Engineering from K. N. Toosi University of Technology, Tehran, Iran, and the M.Sc. degree in Control Engineering from Ferdowsi University of Mashhad, Iran, in 2005 and 2007, respectively. She received a Ph.D. in Computer Science from Nanyang Technological University (NTU), Singapore in 2013. From 2009 to 2013, she was also an attached researcher at the Institute for Infocomm research, Agency for Science, Technology and Research, Singapore. She has worked at the University College Dublin as a lecturer in Biomedical Engineering and as a research fellow in Trinity College Institute of Neuroscience. She is currently a Lecturer in the Department of Automatic Control and Systems Engineering at the University of Sheffield, and an honorary research fellow in Trinity College Institute of Neuroscience. She is currently an associate editor in IEEE Transactions on Neural Systems and Rehabilitation Engineering.

Prof Lyudmila Mihaylova is a Professor of Signal Processing and Control at the Department of Automatic Control and Systems Engineering at the University of Sheffield, United Kingdom. Her research is in the areas of autonomous systems with various applications such as navigation, surveillance and sensor networks. She is an Associate Editor of the IEEE Transactions on Aerospace and Electronic Systems and of the Elsevier Signal Processing Journal. She is the President of the the International Society of Information Fusion (ISIF) in the period 2016-2018.