Distributed Ensemble Kalman Filtering

Arslan Shahid, Deniz Üstebay, and Mark Coates
Department of Electrical and Computer Engineering, McGill University
3480 University St., Montreal, QC, Canada H3A 2A7

Emails: {arslan.shahid,deniz.ustebay}@mail.mcgill.ca, mark.coates@mcgill.ca

***Abstract*—We address the problem of distributed filtering in a wireless sensor network and develop distributed approximations of three variants of the ensemble Kalman filter. We express the update equations in an alternative information form in order to formulate a distributed measurement update mechanism. The distributed filters use randomized gossip to reach consensus on the statistics needed to perform an update. Simulation results suggest that in the case of linear measurements and high-dimensional nonlinear measurements (with measurement model parameters known network-wide) with nonlinear state dynamics the proposed schemes achieve accuracy comparable to state-of-the-art distributed filters while significantly reducing the communication overhead.**

I. Introduction

Distributed filters are attractive in sensor networks because they do not have a single point of failure. Local processing of measurements can reduce the communication overhead when the data dimension is high; it also eliminates the need for centralized knowledge of sensor measurement models and allows any sensor node to be queried to retrieve a state estimate. Distributed Kalman filtering approaches [1] perform poorly when the state dynamics and/or measurement model are nonlinear. The distributed unscented Kalman filter [2] and distributed particle filters [3]–[7] are more suitable for nonlinear and non-Gaussian problems, but require much more computation and communication.

In this paper, we focus on filters that lie between the Kalman filter and the particle filter. We develop distributed approximations of three forms of the ensemble Kalman filter [8], [9]. The ensemble Kalman filter (EnKF) uses a set of samples to approximate the first two moments of the state probability density function. To derive the distributed EnKFs, we express the update equations in an alternative information form. We use randomized gossip [10] to calculate in a distributed fashion the statistics required to perform the update. Simulation results show that in the case of linear measurements or high-dimensional nonlinear measurements (with measurement models known network-wide) the proposed algorithms achieve accuracy comparable to state-of-the-art methods while significantly reducing the communication overhead.

The rest of the paper is organized as follows. Section III presents the problem statement and Section II provides background information. Section IV explains the distributed filtering approach. We present the simulation results in Section V and conclude in Section VI.

II. Background: Ensemble Kalman Filtering

We now present a short review of ensemble Kalman filtering. The ensemble Kalman filter (EnKF) was introduced in [8]. Similar to the particle filter, the EnKF uses a set of state realizations (samples), but they are not weighted and they are used to approximate only the mean and covariance of the state probability density function. In order to obtain an estimate at time k, the samples from time $k-1$ are first forecast (propagated) using the system model and then their weights and locations are updated based on the measurements.

Compared to the (extended) Kalman fiter, the advantage of using the EnKF is that the samples are propagated using the nonlinear system function $f(\cdot)$ rather than a linear approximation. However, the algorithm continues to assume a linear measurement function $h(\cdot)$ (in the case of a nonlinear function a linearization has to be performed). All flavours of the EnKF use the same forecast step and differ only in the update step.

In the original EnKF [8] the i-th state sample is updated as follows:

$$\mathbf{K} = \hat{\mathbf{P}}^f \mathbf{H}^T (\mathbf{H}\hat{\mathbf{P}}^f \mathbf{H}^T + \mathbf{R})^{-1}, \tag{1}$$

$$\mathbf{x}^u_{(i)} = \mathbf{x}^f_{(i)} + \mathbf{K}(\mathbf{y}_{(i)} - \mathbf{H}\mathbf{x}^f_{(i)}) \ . \tag{2}$$

Here $\mathbf{x}^u_{(i)}$ is the updated sample, $\mathbf{x}^f_{(i)}$ is the forecast sample, $\hat{\mathbf{P}}^f$ is the forecast sample covariance, $\mathbf{R}$ is the measurement noise covariance and the matrix $\mathbf{H}$ defines the linear measurement function. $\mathbf{K}$ represents the Kalman gain and $\mathbf{y}_{(i)} = \mathbf{y} + \epsilon_{(i)}$ is a perturbed measurement, generated by adding a realization of the modelled measurement noise to the original measurement.

The ensemble square root filter (ESRF) [9] is a variant of the EnKF that uses a linear transformation to map the forecast samples to the updated samples. The update is given by

$$\hat{\mathbf{x}}^u = \hat{\mathbf{x}}^f + \mathbf{K}(\mathbf{y} - \mathbf{H}\hat{\mathbf{x}}^f), \tag{3}$$

$$\mathbf{T} = (\mathbf{I} - \mathbf{K}\mathbf{H})^{1/2}, \tag{4}$$

$$\mathbf{x}^u_{(i)} = \hat{\mathbf{x}}^u + \mathbf{T}(\mathbf{x}^f_{(i)} - \hat{\mathbf{x}}^f) \ . \tag{5}$$

Here $\hat{\mathbf{x}}^f$ is the mean of the forecast samples, $\mathbf{y}$ is the measurement, $\mathbf{I}$ is the identity matrix, and $(\cdot)^{1/2}$ denotes the unique positive definite square root of a positive definite matrix. The deterministic ensemble Kalman filter (DEnKF) [9] is similar to the ESRF with the advantage that the computation of the transformation matrix does not involve a matrix square root. In this approach the transformation matrix is given by $\mathbf{T} = \mathbf{I} - \frac{1}{2}\mathbf{K}\mathbf{H}$.

978-1-4799-1481-4/14/$31.00 ©2014 IEEE

III. Problem statement

We consider a set of V wireless sensor nodes denoted by $\mathcal{V}$. At each time step, the sensor nodes acquire local measurements related to the system state. We assume one of the following scenarios: (i) the measurements are linear but the sensor nodes are unaware of the measurement parameters (or modalities) of the other nodes; or (ii) the measurements are non-linear and high-dimensional, but there is global knowledge of the local measurement functions. Either assumption renders filtering based on sharing of raw measurements impractical. The goal is to compute an estimate of the system state based on all the measurements. The relation between the measurement and the system state is defined by the measurement model

$$\mathbf{y}_v = h_v(\mathbf{x}) + \epsilon_v, \qquad v \in \mathcal{V}, \tag{6}$$

where $\mathbf{x} \in \mathcal{R}^M$ is the system state, and $h_v(\cdot)$ is the sensor-dependent measurement function (possibly nonlinear) and and ϵ_v is the measurement noise. We assume that it is possible to identify a linearization $\mathbf{H}_{v(i)}$ of the measurement function h_v at a specific location in the state-space $\mathbf{x}_{(i)}$. The sensor nodes have uncorrelated measurement noise and the noise for each sensor follows a Gaussian distribution with mean zero and covariance $\mathbf{R}_v$. The system state evolves according to a discrete-time Markov process with state dynamics defined by

$$\mathbf{x}_k = f(\mathbf{x}_{k-1}) + \mathbf{n}_{k-1}, \tag{7}$$

where $f(\cdot)$ is a nonlinear system function and $\mathbf{n}_{k-1}$ is a Gaussian noise with zero mean and covariance $\mathbf{Q}$.

IV. Distributed EnKF Algorithms

In the distributed approximations of the EnKFs we develop here, every sensor node runs a local copy of the filter and the nodes communicate via randomized gossip to compute a global estimate. In order to do this we require two conditions to hold. First, the random number generators of all the nodes must be synchronized (initialized with the same seed). Second, the nodes must be aware of the network size. We can use a decentralized routine [4] to accomplish both tasks before the start of the distributed estimation activity.

Every node can calculate its local forecast by propagating the updated samples from the previous time step using the system model (7). The forecast is given by

$$\mathbf{x}^f_{(i)k} = f(\mathbf{x}^u_{(i)k-1}) + \mathbf{n}_{(i)k-1}, \tag{8}$$

$$\hat{\mathbf{P}}^f_k = \frac{1}{N-1}\sum_{i=1}^{N}(\mathbf{x}^f_{(i)k} - \hat{\mathbf{x}}^f_k)(\mathbf{x}^f_{(i)k} - \hat{\mathbf{x}}^f_k)^T, \tag{9}$$

where $\mathbf{n}_{(i)k-1}$ is a system noise realization and N is the number of samples. Since the random number generators are synchronized the sensors will produce identical noise samples and consequently identical forecast samples if they have the same updated samples at the previous time step.

The main challenge in formulating a distributed update step lies in the fact that the update equations (1)-(5) require the knowledge of the complete set of measurements, measurement functions and measurement noise statistics. Sensor nodes only have access to their local measurements and may only be aware of their own measurement model parameters. We address this challenge by expressing the update equation in an alternative form.

In the derivation that follows, we exclude the common time index subscript k from the notation. In the information filter form, we write the Kalman gain as

$$\hat{\mathbf{P}}^u = \left[(\hat{\mathbf{P}}^f)^{-1} + \mathbf{H}^T\mathbf{R}^{-1}\mathbf{H}\right]^{-1}, \tag{10}$$

$$\mathbf{K} = \hat{\mathbf{P}}^u\mathbf{H}\mathbf{R}^{-1}. \tag{11}$$

We can substitute the above expressions (10) and (11) into the update equation of the EnKF (2) and then express it in terms of the measurements and measurement model parameters of the individual sensor nodes,

$$\hat{\mathbf{P}}^u = [(\hat{\mathbf{P}}^f)^{-1} + \sum_{v\in\mathcal{V}}(\mathbf{H}^T_{v(i)}\mathbf{R}_v^{-1}\mathbf{H}_{v(i)})]^{-1}, \tag{12}$$

$$\mathbf{x}^u_{(i)} = \mathbf{x}^f_{(i)} + \hat{\mathbf{P}}^u\sum_{v\in\mathcal{V}}[\mathbf{H}^T_{v(i)}\mathbf{R}_v^{-1}\mathbf{y}_{v(i)} - \mathbf{H}^T_{v(i)}\mathbf{R}_v^{-1}\mathbf{H}_{v(i)}\mathbf{x}^f_{(i)}]. \tag{13}$$

Here the subscript v denotes sensor-specific versions of the correlation and measurement matrices; and the subscript (i) denotes sample-specific quantities.

A. Linear Measurements

In the linear measurement scenario, $\mathbf{H}_{v(i)} = \mathbf{H}_v$ does not depend on the state value. We can express several of the computations involved in the update (13) as network averages over local sensor variables. In the local update step of the EnKF, each sensor node $v \in \mathcal{V}$ initializes local variables:

$$\{\mathbf{Y}_{v(i)}\}_{i=1}^N = \{|\mathcal{V}|(\mathbf{H}^T_{v(i)}\mathbf{R}_v^{-1}\mathbf{y}_{v(i)})\}_{i=1}^N, \tag{14}$$

$$\mathbf{S}_v = |\mathcal{V}|(\mathbf{H}_v^T\mathbf{R}_v^{-1}\mathbf{H}_v). \tag{15}$$

Two randomized gossip procedures [10] are applied to these local variables. First, the sensor nodes perform an averaging gossip routine to compute approximately the element-wise average of these variables over the sensor network. Then the sensor nodes run a max-gossip routine to try to ensure that all the nodes have exactly the same values. In each iteration of the max-gossip routine, the local update operation is an element-wise maximum. Each sensor node can then compute a local update using the equation

$$\mathbf{x}^u_{(i)} = \mathbf{x}^f_{(i)} + [(\hat{\mathbf{P}}^f)^{-1} + \widetilde{\mathbf{S}}]^{-1}(\widetilde{\mathbf{Y}}_{(i)} - \widetilde{\mathbf{S}}\mathbf{x}^f_{(i)}), \tag{16}$$

where $\widetilde{\mathbf{Y}}_{(i)}$ and $\widetilde{\mathbf{S}}$ are the values obtained after running average and max gossip routines on $\{\mathbf{Y}_{v(i)}\}_{v\in V}$ and $\{\mathbf{S}_v\}_{v\in V}$, respectively.

A similar approach can be used to develop a distributed update procedure for the ESRF and DEnKF algorithms. In these algorithms sensor nodes need to perform gossip on the following local variables

$$\mathbf{Y}_v = |\mathcal{V}|(\mathbf{H}_v^T\mathbf{R}_v^{-1}\mathbf{y}_v), \tag{17}$$

$$\mathbf{S}_v = |\mathcal{V}|(\mathbf{H}_v^T\mathbf{R}_v^{-1}\mathbf{H}_v). \tag{18}$$

After running the gossip routines on these local variables, each sensor node can compute the mean of the updated samples:

$$\hat{\mathbf{x}}^u = \hat{\mathbf{x}}^f + [(\hat{\mathbf{P}}^f)^{-1} + \widetilde{\mathbf{S}}]^{-1}(\widetilde{\mathbf{Y}} - \widetilde{\mathbf{S}}\hat{\mathbf{x}}^f). \tag{19}$$

If the sensor nodes possess an estimate of the transformation matrix they can locally update the samples using the equation (5). In the ESRF approach the transformation matrix can be locally computed as

$$\mathbf{T} = (\mathbf{I} - [(\hat{\mathbf{P}}^f)^{-1} + \widetilde{\mathbf{S}}]^{-1}\widetilde{\mathbf{S}})^{1/2}. \tag{20}$$

Similarly, in the DEnKF a local computation of the transformation matrix can be performed as follows

$$\mathbf{T} = \mathbf{I} - \frac{1}{2}[(\hat{\mathbf{P}}^f)^{-1} + \widetilde{\mathbf{S}}]^{-1}\widetilde{\mathbf{S}}. \tag{21}$$

B. Non-linear measurements with globally-known models

The other scenario we address involves non-linear measurements, but we assume that there is network-wide knowledge of the local measurement models and parameters. In this case, there is no longer a need to gossip to calculate $\widetilde{\mathbf{S}}$, since each node can compute it locally. Note, however, that we use a sample-dependent $\widetilde{\mathbf{S}}_{(i)}$ in the distributed EnKF update (16), because the linearization is performed at the sample state value. Similarly, in the distributed DEnKF and the ESRF, a different operator $\mathbf{T}_{(i)}$ in (20) and (21) must be calculated for each sample.

C. Communication Overhead

In the case of a linear measurement model, the distributed ensemble Kalman filter requires the communication of $NM + M^2$ scalars by each node per gossip iteration (recall that M is the state dimension and N is the number of samples in the ensemble). For the deterministic ensemble Kalman filter and the ensemble square root filter, this communication requirement is reduced to $M + M^2$. For these latter filters, the removal of the dependence on the number of samples enables the filters to perform accurate tracking even when exchanging relatively few values between nodes.

When the measurement models are non-linear, but known network-wide, the communication overhead reduces to NM for the distributed ensemble Kalman filer and M for the distributed ESRF and DEnKF.

V. Simulations

In order to examine the efficacy of the proposed approach we conduct a set of numerical simulations using Matlab. In this study we consider two measurement functions: (i) a linear function defined by $\mathbf{H} = [1\ 0\ 0\ 0; 0\ 1\ 0\ 0]$, such that $\mathbf{y}_v = \mathbf{H}\mathbf{x} - \mathbf{x}_v + \epsilon_v$, where $\mathbf{x}_v$ is the location of node v; and (ii) a nonlinear *radio-frequency (RF) tomography* measurement function to demonstrate the case of high dimensional measurements (see [11] for a complete description of the RF tomography model). The standard deviation of the system noise is 0.25 and the standard deviation of the measurement noise is 0.25 for the linear model and 0.50 for the RF tomography model. The parameters of the RF tomography model, σ_λ is set to 0.05 and ϕ is set to 5.

In our simulation setup we deploy sensor nodes in a square region of length $50m$. In the case of linear/range models 25 sensor nodes are uniformly spaced on a grid. For the RF tomography model 24 sensor nodes are deployed at the boundary of the square region. Hence, in the RF tomography setup each sensor node acquires a measurement consisting of 23 scalars. A pair of nodes can communicate if their separation is less than the transmission range ($15m$). The sensor nodes track a single target for 50 seconds. The target makes multiple clockwise turns, and its state is given by $\mathbf{x} = [x_1, x_2, \dot{x}_1, \dot{x}_2]^T$, where x_1 and x_2 represent the position and $\dot{x}_1$ and $\dot{x}_2$ represent the velocity along the x and y axes, respectively. The target motion is modeled by the nearly coordinated turn model, which assumes unknown cartesian velocity but known turn rate. The sampling period is 1 second. A sensor node only acquires a measurement if the target is within its sensing range of $10m$. All sensor nodes use the same measurement function and noise statistics.

We compare our proposed algorithms to the following distributed tracking techniques: Gaussian approximation (GA) [4], SIR version of likelihood consensus (LC) [5]; set membership (SM) filter [6], top-m selective gossip (Top-m) [7], and the distributed unscented Kalman filter (UKF) [2]. In the set membership approach we use the same oversampling parameter value (10) as in [6]. For the top-m approach we set $m = N/4$, where N is the number of samples in the particle filter. In the distributed UKF we use just one mixture component in order to minimize communication cost. For the particle filter approaches, we tune parameter choices (number of particles, gossip iterations) to optimize performance. For the algorithms proposed in this paper, we do not tune, but adopt default choices. Specifically, we choose a moderate number of samples, since practical experience suggests that even for very high dimensional problems the EnKF requires around 50 samples to reach its best performance [12].

We compare performance using the root-mean-squared (RMS) position error. We average results over 100 Monte Carlo trials; each trial has a different measurement realization and a different particle initialization. For all particle filters we generate initial samples by adding zero mean Gaussian noise of standard deviation 0.25 and 1 to the starting position and velocity of the target, respectively. If in any trial, the RMS error exceeds $2m$ we consider it as a lost track and exclude it from the error analysis.

In Table I we present the average RMS error ± standard deviation and percentage track loss as a function of the number of scalars transmitted for all the distributed filters for the linear measurement model. We include the results for a centralized bootstrap particle filter with 2000 samples as a performance benchmark. In some cases we are unable to find any operating point of a filter for a particular communication cost. We mark these cases with a dash. For a high communication cost all filters achieve tracking accuracy close to that of the centralized filter. As the number of scalars is decreased all

TABLE I
PERFORMANCE (LINEAR MEASUREMENT MODEL).

	Ave. RMSE ± Std. Dev. (% track loss)		
Scalars	1000	500	200
GA [4]	0.26 ± 0.02	0.26 ± 0.03	0.28 ± 0.02(1)
LC [5]	0.24 ± 0.02	0.24 ± 0.02	0.83 ± 0.14
SM [6]	0.36 ± 0.14	0.45 ± 0.16	0.49 ± 0.17(1)
Top-m [7]	0.26 ± 0.04	0.32 ± 0.08	0.69 ± 0.33(14)
UKF [2]	0.24 ± 0.02	0.26 ± 0.03	0.96 ± 0.52(6)
EnKF	0.41 ± 0.28(12)	0.41 ± 0.27(13)	0.47 ± 0.29(15)
ESRF	0.24 ± 0.02	0.24 ± 0.02	0.27 ± 0.02
DEnKF	0.24 ± 0.02	0.24 ± 0.02	0.28 ± 0.02
Centr. PF	0.24 ± 0.02	0.24 ± 0.02	0.24 ± 0.02

TABLE II
PERFORMANCE (RF TOMOGRAPHY MEASUREMENT MODEL).

	Ave. RMSE ± Std. Dev. (% track loss)		
Scalars	1000	500	200
GA [4]	0.17 ± 0.01	0.20 ± 0.02(2)	0.35 ± 0.18(22)
LC [5]	0.33 ± 0.22	-	-
SM [6]	0.26 ± 0.18(3)	0.33 ± 0.19(7)	0.36 ± 0.25(16)
Top-m [7]	0.27 ± 0.22(2)	0.34 ± 0.20(11)	0.70 ± 0.41(50)
UKF [2]	0.21 ± 0.09(18)	0.61 ± 0.49(74)	0.40 ± 0.26(95)
EnKF	-	-	-
ESRF	0.19 ± 0.05	0.18 ± 0.05	0.22 ± 0.10(6)
DEnKF	0.23 ± 0.09	0.21 ± 0.08	0.22 ± 0.09(1)
Centr. PF	0.10 ± 0.01	0.10 ± 0.01	0.10 ± 0.01

the filters except for the ESRF and the DEnKF experience significant performance degradation. The SM and the Top-m schemes gossip on particle weights so they have a much higher communication overhead. The LC and the UKF gossip on low dimensional variables but require more gossip iterations to ensure that there is an accurate consensus for these values. The GA method performs well but is prone to occasional track loss and is sensitive to the choice of its algorithmic parameters. The EnKF gossips on twice as many scalar values as the ESRF and the DEnKF, so its RMS error increases more sharply. The ESRF and the DEnKF require fewer gossip iterations and exchange low dimensional variables, so they perform well at lower communication costs.

In Table II we present the results for the RF tomography measurement model. In this case the sensor nodes are aware of the model parameters of the other nodes in the network. Hence, the EnKF, the ESRF, the DEnKF and the UKF schemes only need to gossip on the local variables that involve the measurements. The distributed ensemble Kalman filters can compute the linearized approximations locally and update the corresponding statistics. However, the distributed particle filtering methods experience no change in the dimensionality of the data on which they have to gossip. The distributed ESRF and DEnKF show a similar performance trend as in the case of the linear measurement model and outperform the existing distributed filters at lower communication costs. Figure 1 shows the target trajectory and the median estimate of the distributed DEnKF for 200 scalars.

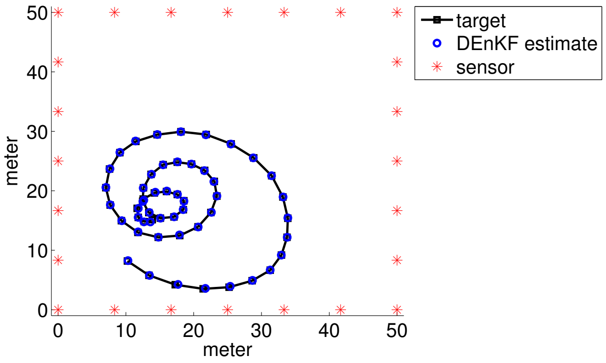

Fig. 1. RF tomography sensor network, target trajectory and the DEnKF track corresponding to the median error performance (200 scalars communicated).

VI. CONCLUSION

We developed distributed approximations of three ensemble Kalman filters for the scenario where measurements are distributed over a sensor network. The proposed distributed schemes express the measurement update equation in an alternative, information form, and employ randomized gossip between sensor nodes to perform distributed computation of the statistics required to perform an update. Simulation results suggest that for linear measurements and high-dimensional nonlinear measurements (model parameters known locally) two of the proposed algorithms can achieve performance similar to the state-of-the-art distributed filtering methods while significantly reducing the communication cost.

REFERENCES

[1] R. Olfati-Saber, "Distributed Kalman filtering for sensor networks," in *Proc. IEEE Eur. Control Conf.*, Seville, Spain, Dec. 2007.

[2] W. Li and Y. Jia, "Consensus-based distributed multiple model UKF for jump Markov nonlinear systems," *IEEE Trans. Autom. Control*, vol. 57, no. 1, pp. 227–233, Jan. 2012.

[3] M. Coates, "Distributed particle filters for sensor networks," in *Proc. Int. Symp. Inf. Proc. Sens. Net. (IPSN)*, Berkeley, CA, USA, Apr. 2004.

[4] B.N. Oreshkin and M.J. Coates, "Asynchronous distributed particle filter via decentralized evaluation of Gaussian products," in *Proc. ISIF Int. Conf. Information Fusion*, Edinburgh, UK, July 2010.

[5] O. Hlinka, O. Sluciak, F. Hlawatsch, P.M. Djuric, and M. Rupp, "Likelihood consensus: Principles and application to distributed particle filtering," in *Proc. Asilomar Conf. Sig., Sys. Comp.*, Nov. 2010.

[6] S. Farahmand, S.I. Roumeliotis, and G.B. Giannakis, "Set-membership constrained particle filter: Distributed adaptation for sensor networks," *IEEE Trans. Signal Proc.*, vol. 59, no. 9, pp. 4122–4138, Sept. 2011.

[7] D. Üstebay, M. Coates, and M. Rabbat, "Distributed auxiliary particle filters using selective gossip," in *Proc. IEEE Int. Conf. Acoustics Speech and Sig. Proc. (ICASSP)*, Prague, Czech Republic, May 2011.

[8] G. Evensen, "Sequential data assimilation with a nonlinear quasi-geostrophic model using Monte Carlo methods to forecast error statistics," *J. Geophy. Res.-All Ser.*, vol. 99, pp. 10–10, May 1994.

[9] P. Sakov and P.R. Oke, "A deterministic formulation of the ensemble Kalman filter: an alternative to ensemble square root filters," *Tellus A*, vol. 60, no. 2, pp. 361–371, Jan. 2008.

[10] A.G. Dimakis, S. Kar, J.M.F. Moura, M.G. Rabbat, and A. Scaglione, "Gossip algorithms for distributed signal processing," *Proc. IEEE*, vol. 98, no. 11, pp. 1847–1864, Nov. 2010.

[11] F. Thouin, S. Nannuru, and M. J. Coates, "Multi-target tracking for measurement models with additive contributions," in *Int. Conf. Inf. Fusion*, Chicago, IL, USA, Jul. 2011, pp. 1–8.

[12] G. Evensen, "The ensemble Kalman filter for combined state and parameter estimation," *IEEE Control Syst. Mag.*, vol. 29, no. 3, pp. 83–104, Jun. 2009.