

A Cooperative Approach to Sensor Localisation in Distributed Fusion Networks

Murat Üney, *Member, IEEE*, Bernard Mulgrew, *Fellow, IEEE*, Daniel E. Clark, *Member, IEEE*

Abstract—We consider self-localisation of networked sensor platforms which are located disparately and collect cluttered and noisy measurements from an unknown number of objects (or, targets). These nodes perform local filtering of their measurements and exchange posterior densities of object states over the network to improve upon their myopic performance. Sensor locations need to be known, however, in order to register the incoming information in a common coordinate frame for fusion. In this work, we are interested in scenarios in which these locations need to be estimated solely based on the multi-object scene. We propose a cooperative scheme which features nodes using only the information they already receive for distributed fusion: we first introduce node-wise separable parameter likelihoods for sensor pairs, which are recursively updated using the incoming multi-object information and the local measurements. Second, we establish a network coordinate system through a pairwise Markov random field model which has the introduced likelihoods as its edge potentials. The resulting algorithm consists of consecutive edge potential updates and Belief Propagation message passing operations. These potentials are capable of incorporating multi-object information without the need to find explicit object-measurement associations and updated in linear complexity with the number of measurements. We demonstrate the efficacy of our algorithm through simulations with multiple objects and complex measurement models.

Index Terms—cooperative localisation, multi-target tracking, simultaneous localisation and tracking, sensor networks, graphical models, Monte Carlo algorithms, dynamical Markov random fields.

I. INTRODUCTION

FUSION networks comprised of geographically dispersed and networked sensor platforms are one of the key enablers of wide area surveillance applications. These networks have the potential to enhance situational awareness in a number of aspects including coverage, accuracy and ease of deployment by providing ad-hoc deployability through a reconfigurable and scalable system structure. The sensor platforms have moderate sensing, computation and communication capabilities and energy resources (as opposed to the rather stringent resource constraints of the commonly considered wireless sensor networks). In order to monitor an unknown number of objects (or, targets), they use sensor signals with an inherently imperfect detection process and obtain noisy measurements from a subset of the objects in

their coverage and false detections from the surroundings, all in their sensor centric coordinate system (SCCS). Given the limited bandwidth of the links and the high energy cost of communications, it is often not feasible to forward network wide collected measurements to a designated centre which in turn would have a high computational load [1].

Decentralised paradigms have more desirable properties in fusion networks such as better resource utilisation and flexibility [2]. Typically, the nodes locally filter their measurements to estimate the object trajectories. Then, they exchange the filtered distributions with other nodes over the network to improve upon the accuracy they achieve myopically based on only their local measurements (e.g., [3]). These information messages, however, can be combined only after they are registered in a common coordinate system [4], e.g., the local coordinate frame. Respective sensor locations constitute a fundamental component of sensor registration parameters that specify these coordinate transforms. Geographical routing algorithms underpinning the communication network also rely on a reasonably accurate knowledge of these locations [5].

We are interested in locating the sensors based solely on measurements from the multi-object scene. Such a constraint arise in a range of applications: For example, underwater fusion networks cannot exploit global navigation space systems (GNSS) due to signal propagation constraints of their environment [6]. In terrestrial settings, GNSS might fail to perform reliably considering their vulnerabilities to deliberate interferences such as jamming [7]. The use of reference (or, cooperative) vehicles [8] does not match well the flexibility requirements. Another alternative which has been investigated intensely is to use communication front-end and/or network statistics such as received signal strength (RSS) and time of arrival (TOA) [9]. Localisation based on RSS and TOA type noisy distance measurements, however, is often not sufficiently accurate for networks with the degree of connectivity typical in fusion applications [10, Chp.6].

Our problem can be treated as a particular type of sensor registration (or, calibration [11]) using targets of opportunity. The latter topic has been investigated in the context of multi-target tracking, however, mostly for mitigating biases of model parameters [12]. The work which implicitly or explicitly address locating sensors include solutions in centralised settings based on conventional multi-target trackers [13], [14] and random finite sets (RFS) based multi-target filtering [15], [16]. These work involve selection of either a maximum likelihood (ML) or a Bayesian estimation paradigm and specifying the parameter likelihood of the problem (see, e.g., [17, Sec.IV]) in accordance with the multi-object and measurement models.

This work was supported by the Engineering and Physical Sciences Research Council (EPSRC) grants EP/J015180/1 and EP/K014277/1, and the MOD University Defence Research Collaboration in Signal Processing.

Murat Üney and Bernard Mulgrew are with the Institute for Digital Communications, School of Engineering, the University of Edinburgh, Edinburgh (e-mail: m.üney@ed.ac.uk, b.mulgrew@ed.ac.uk).

Daniel E. Clark is with the School of Engineering and Physical Sciences, Heriot-Watt University, Edinburgh (e-mail: d.e.clark@hw.ac.uk).

This likelihood, however, requires all the target measurements collected across the network to be filtered together, and, in turn, centralised processing. Distributed alternatives often resort to joint filtering which embodies all the drawbacks of centralised fusion, both in the case of ML [18] and Bayesian [19] paradigms.

In this article, we propose a distributed online self-localisation scheme that avoids any form of centralisation and operates alongside distribution fusion. The computational structure is composed of iterative local message passings yielding full decentralisation. The algorithm is developed in two steps: First, we consider a pair of fusion sensors and develop a calibration likelihood which can be computed based solely on the multi-object distributions exchanged and local target measurements. Specifically, we approximate the parameter likelihood with a node-wise separable form which removes the need for local centralisations.

Second, we merge these likelihoods in a pairwise Markov random field (MRF) model which captures the communication structure of the network. Such models are equipped with estimation algorithms which operate in a message passing fashion and have proved useful for distributed estimation [20] and target tracking [21] applications in wireless sensor network as well as self-localisation based on RSS measurements [22]. In our model, we use the proposed node-wise separable likelihoods as edge potentials [23]. These potentials have a time-recursive structure as the respective parameters are indirectly observed through measurements from a hidden process.

In the resulting scheme, nodes simultaneously perform distributed fusion and update node-wise separable likelihoods—equivalently, edge potentials—with their neighbours using the incoming multi-object posteriors and local target measurements. At the end of a fixed length time window, the updated potentials are used in Belief Propagation (BP) [24] message passing iterations. We accommodate and benefit from the rich information from multiple targets of opportunity using Poisson multi-object models [25] in the update recursions. These models are propagated by local multi-object filters from which our scheme inherits the capability of operating with complex measurements involving false alarms and noisy measurements of the objects with imperfect detection rates. The potential updates feature linear complexity with the number of measurements without any need to explicitly find target-measurement associations.

The article is structured as follows: We provide the problem statement in Section II. Then, we introduce a pairwise MRF model for calibration and overview distributed estimation of calibration marginals in Section III. In Section IV, we develop node-wise separable likelihoods for calibration problems. We combine these likelihoods with the MRF model introduced and describe our collaborative scheme in Section V. We detail a Monte Carlo (MC) algorithm for self-localisation in Section VI and demonstrate its efficacy in Section VII. Finally, we conclude in Section VIII.

II. PROBLEM STATEMENT

We consider networked sensor platforms listed as $\mathcal{V} = \{1, \dots, N\}$. The communication links

available between pairs of sensors (i, j) specify the edge set $\mathcal{E} \subset \mathcal{V} \times \mathcal{V}$ with respect to the relation $\mathcal{E} = \{(i, j) | i \text{ and } j \text{ share a communication link}\}$. We assume bidirectional communication links which is captured by using an undirected graph \mathcal{G} for which if $(i, j) \in \mathcal{E}$, then it holds that $(i, j) \in \mathcal{E} \Leftrightarrow (j, i) \in \mathcal{E}$. The neighbours of node i in \mathcal{G} constitute the set $ne(i) \triangleq \{j | (i, j) \in \mathcal{E}\}$. The network topology in \mathcal{G} is connected and might contain cycles.

We employ widely used models for capturing complex and uncertain interactions between sensors and an unknown number of manoeuvring objects.

An object in the network surveillance region is described by a state vector x in the state space \mathcal{X} . Typically, x contains the Cartesian coordinates of the object in an Euclidean plane x^l and its velocity x^v , i.e., $x = [x^l, x^v]$.

The evolution of a moving object's state and the measurement it induces are described by a hidden Markov model (HMM). The state at time k is distributed according to the density $\pi(x_k | x_{k-1})$ where x_{k-1} is the realisation of the previous state with initial density $\pi_0(x_0)$ ¹. An observation $z_{k,j}$ is generated at sensor j through a likelihood of the form $l(z_{k,j} | x_k)$ independently from other sensors.

Sensor likelihoods are specified by models that characterise the detectors processing sensor signals. In multi-sensor settings, it is useful to explicitly condition the likelihoods further on calibration parameters which relate a given target state in a desired reference frame to sensor readings. These parameters might involve respective quantities such as the location and the orientation of the sensors with respect to the reference frame, as well as scale factors.

For example, consider a commonly used model involving a nonlinear mapping of the target state with additive uncertainties:

$$z_{k,j} = h_j([x_k]_j) + n_j \quad (1)$$

where n_j is a measurement noise realisation with probability density $g_j(n_j)$, and, h_j is a known mapping. Here, $[x_k]_j$ denotes x_k in the SCCS of the j th sensor. For a sensor localisation problem,

$$[x_k^l]_j = x_k^l - \theta_j \quad (2)$$

where θ_j is the sensor location and x_k^l is the position component of the state vector x_k , all represented in the reference frame. The likelihood density for this model is then found as

$$l_j(z_{k,j} | x_k; \theta_j) = g_j(z_{k,j} - h_j([x_k]_j)).$$

For more general calibration problems, it is useful to introduce a transform from the reference frame to the SCCS of sensor j parameterised by θ_j , i.e.,

$$[x_k]_j = \tau_j(x_k; \theta_j). \quad (3)$$

For example, if the respective orientation angle α_j in the plane is unknown in addition to the location of sensor j , then it is

¹The state transition could be time-dependant which is omitted here for brevity.

useful to consider the transform given by

$$\tau_j(x_k; \theta_j) = [\alpha_j, \theta_j^l] = \begin{bmatrix} \mathbf{R}(\alpha_j)(x_k^l - \theta_j^l) \\ \mathbf{R}(\alpha_j)x_k^v \end{bmatrix}, \quad (4)$$

where $\mathbf{R}(\alpha_j)$ is the rotation matrix for α_j and θ_j^l is the position component of θ_j in the reference frame. The likelihood density for this model is then

$$l_j(z_{k,j}|x_k; \theta_j) = g_j(z_{k,j} - h_j(\tau_j(x_k, \theta_j))). \quad (5)$$

In practice, it is reasonable to assume that θ_j takes values from a bounded set $\mathcal{B} \subset \mathbb{R}^d$ of dimensionality d . For example, $d = 2$ when localisation in a plane is considered. Our goal is to find these parameters

$$\theta \triangleq (\theta_1, \dots, \theta_N)$$

using local message passings on \mathcal{G} and based on measurements from the multi-object scene (or, targets of opportunity), models for which are discussed next².

Consider the set of realisations of M_k hidden processes $X_k \triangleq \{x_k^1, \dots, x_k^{M_k}\}$ at time k . A point $x \in X_k$ induces a measurement at sensor j with probability $P_{D,j}(x)$, independently. Let us denote the set of measurements from the objects at sensor j by \tilde{Z}_k^j . In addition to \tilde{Z}_k^j , sensor j collects false detections from the surroundings, or clutter. We use a Poisson process to model the clutter points, i.e., C^j is a Poisson realisation denoted by $C^j \sim \text{Pois}(\cdot; \lambda_{C,j}, s_{C,j}(z))$ where $\lambda_{C,j}$ is the average number of (Poisson distributed) clutter points and $s_{C,j}(z)$ their spatial density³. Therefore, the set of measurements sensor j receives at time k is given by

$$Z_k^j = \tilde{Z}_k^j \cup C_{k,j}.$$

Sensor calibration, hence, involves finding the joint parameter likelihood relating the measurement histories $\{Z_{1:k}^j\}_{j \in \mathcal{V}}$ and θ , i.e., $l(Z_{1:k}^1, \dots, Z_{1:k}^N | \theta)$. We consider a random θ and use this likelihood to update a prior distribution. Because θ is bounded, a uniform prior $p_0(\theta)$ over \mathcal{B}^N can be used, and, the posterior density is given by

$$p(\theta | Z_{1:k}^1, \dots, Z_{1:k}^N) \propto p_0(\theta) l(Z_{1:k}^1, \dots, Z_{1:k}^N | \theta). \quad (6)$$

We are interested in finding the minimum mean squared error (MMSE) estimate of θ based on this posterior⁴.

A. Centralised sensor calibration

In the centralised estimation of θ , a designated centre has access to all measurement histories $\{Z_{1:k}^j\}_{j \in \mathcal{V}}$ to evaluate the parameter likelihood

$$l(Z_{1:k}^1, \dots, Z_{1:k}^N | \theta) = \prod_{t=0}^{k-1} p(Z_{t+1}^1, \dots, Z_{t+1}^N | Z_{1:t}^1, \dots, Z_{1:t}^N, \theta), \quad (7)$$

²The neighbourhood structure of \mathcal{G} does not necessarily replicate the distance relations among θ_j s.

³Note that, it is possible to use a non-stationary Poisson process model for the clutter. Here, we omit dependency to time for brevity.

⁴The MMSE estimate is often easier to compute compared to, for example, the maximum a posteriori (MAP) estimation rule. We note that the MMSE solution approximates the MAP one reasonably well in the case of convex, unimodal distributions. The latter is also equivalent to using this likelihood in a ML paradigm (e.g., [18]), in the case of uniform priors.

where the factorisation follows from the chain rule of probabilities [17, Sec.IV]. The factors in the right hand side (RHS) are independent contributions of the measurement sets collected at each time step. The Markov property of the hidden object processes admit that the current measurements are conditionally independent of the measurement histories, given the current object states and the sensor locations. Using this relation together with the (conditional) mutual independence of measurements across the sensors, the instantaneous contribution of measurements collected at $t + 1$ to the likelihood at k is found as [17]

$$\begin{aligned} & p(Z_{t+1}^1, \dots, Z_{t+1}^N | Z_{1:t}^1, \dots, Z_{1:t}^N, \theta) \\ &= \int p(Z_{t+1}^1, \dots, Z_{t+1}^N | X_{t+1}, \theta) p(X_{t+1} | Z_{1:t}^1, \dots, Z_{1:t}^N, \theta) dX_{t+1}, \\ &= \int \left(\prod_{j=1}^N p(Z_{t+1}^j | X_{t+1}, \theta_j) \right) p(X_{t+1} | Z_{1:t}^1, \dots, Z_{1:t}^N, \theta) dX_{t+1}, \end{aligned} \quad (8)$$

where the product term inside the integral is the multi-sensor likelihood. The second term is a prediction density for the multi-object scene at time $t + 1$ based on the network history until t , and, can be found by the Bayesian filtering recursions, i.e., it is output at the prediction step of a ‘‘centralised’’ filter.

In addition to the difficulties in collecting $Z_{1:k}^j$ s at a designated fusion centre under resource constraints, the estimator obtained by substituting from (8) to (7) and (6) inherits the computational issues related to both multi-sensor filtering and parameter estimation in state space models: The centralised filter needed for the realisation of (8) faces the same computational complexity issues which are encountered in multi-object filtering and which are exacerbated in the case of multiple sensors⁵. The usual non-Gaussian/non-linear dynamics involved in such problem settings suggest Sequential Monte Carlo (SMC) methods to be used as a computational paradigm, and, in the case of parameter estimation problems including calibration, iterative sampling strategies with relatively high computational and memory complexities should be used to achieve reasonably accurate estimates [28]. Thus, the described centralised solution suffers from poor scalability with the number of sensors and objects.

We circumvent these problems through a number of modelling approximations introduced in the rest of this article, and provide a distributed scheme with efficient local computations.

B. Distributed fusion architecture and information exchange

Each sensor platform locally filters its measurement history and maintains a local representation of its environment including the multi-object state \mathbf{X}_k . There is no restriction on the multi-object models and inference algorithms used by these platforms. The neighbouring nodes in \mathcal{G} exchange their representations to improve upon their myopic accuracy. As far as our self-localisation (calibration) scheme is concerned, the only requirement is that a Poisson RFS approximation of \mathbf{X}_k is available from the information received. In other words, the neighbours provide –directly or indirectly– a model

⁵See, for example, the related discussions in [26] and [27] in the cases of hypothesis based and RFS models, respectively.

in which X_k is a realisation of a Poisson RFS conditioned on the sensor history, i.e., $X_k \sim \text{Pois}(\cdot; \lambda_{k|k}, s_{k|k}(x))$, which first draws M_k from a Poisson distribution with parameter $\lambda_{k|k}$ and then generates M_k points from $s_{k|k}(x)$. Hence, X_k evaluates its density as

$$f(X_k) = e^{-\lambda_{k|k}} \lambda_{k|k}^{M_k} \prod_{x \in X_k} s_{k|k}(x) \quad (9)$$

where the product selects each element only once.

Poisson multi-object models are provided directly by the Probability Hypothesis Density (PHD) filter [29]. In the case that another RFS model is provided (from filtering algorithms such as the Cardinalised PHD [30], labelled [31] and variational [32] multi-Bernoulli filters), a best Poisson approximation can be found using the first order statistical moment of this distribution [25]. More conventional algorithms such as multiple hypothesis tracking (MHT) consider hypotheses on measurement object correspondances, on the other hand, and find the likelihood of these association hypotheses together with single object posteriors. This form of information is convertible to RFS distributions [33] from which Poisson multi-object models can readily be obtained.

Besides that they can be obtained from a wide range of trackers, there are additional appealing features of Poisson multi-object models: because they are RFS models, the corresponding sensor likelihoods evaluate without the need to explicitly find measurement-object associations [25, Eq.(12.41)]. Moreover, marginalisations over the state variable as in the RHS of (8) have closed form expressions and can be computed using Monte Carlo methods with linear complexity in the number of measurements for a single sensor. These features of multi-object scalability are later combined with the MRF model we introduce in Sections III and IV for multi-sensor scalability.

III. A DYNAMIC PAIRWISE MARKOV RANDOM FIELD MODEL FOR CALIBRATION

In this section, we introduce an approximation to the centralised estimator (Eq.s(8),(7) and (6)) which enables the estimation of θ in a distributed fashion, through local message passing operations.

We consider a parsimonious representation for the calibration posterior in (6). Specifically, we assume that θ is Markov with respect to the communication topology \mathcal{G} , i.e., if the sets of nodes A and B are separated by C on \mathcal{G} , then the random variables associated with A , i.e., $\theta_A = \{\theta_i | i \in A\}$, and θ_B are conditionally independent given θ_C . This conditional independence relation is often denoted by $\theta_A \perp\!\!\!\perp \theta_B | \theta_C$ [23]. All such relations admitted by \mathcal{G} factorise (6) to positive functions (or, potential functions) over the cliques of \mathcal{G} (connected subsets of \mathcal{V}) [23].

The distributed fusion architecture discussed in Section II-B involves nearest neighbour multi-object posterior exchanges. Consequently, the local information available at the nodes are related to the respective locations of pairs of (neighbouring) sensors. Therefore, we take into account only the singleton and two-node cliques and use a pairwise MRF model

$$\begin{aligned} \tilde{p}(\theta | Z_{1:k}^1, \dots, Z_{1:k}^N) &\propto \prod_{i \in \mathcal{V}} \psi_i(\theta_i) \prod_{(i,j) \in \mathcal{E}} \psi_{ij}^k(\theta_i, \theta_j), \quad (10) \\ \psi_i(\theta_i) &= p_{0,i}(\theta_i), \\ \psi_{ij}^k(\theta_i, \theta_j) &= l(Z_{1:k}^i, Z_{1:k}^j | \theta_i, \theta_j), \end{aligned}$$

where the node potential functions ψ_i are arbitrary priors for θ_i (e.g., uniform distributions over \mathcal{B}) and the edge potentials ψ_{ij}^k s are predictive parameter likelihoods for the pairs (i, j) s based on sensor histories up to time k . These edge potentials have the time-recursive structure in (7), i.e.,

$$\begin{aligned} \psi_{ij}^k(\theta_i, \theta_j) &= \prod_{t=0}^{k-1} p(Z_{t+1}^i, Z_{t+1}^j | Z_{1:t}^i, Z_{1:t}^j, \theta_i, \theta_j) \\ &= \psi_{ij}^{k-1}(\theta_i, \theta_j) p(Z_k^i, Z_k^j | Z_{1:k-1}^i, Z_{1:k-1}^j, \theta_i, \theta_j) \quad (11) \end{aligned}$$

and render a dynamical MRF.

One challenge in calibration using this model in distributed fusion networks is the computation of ψ_{ij}^k s in a distributed fashion, without communicating Z_k^i s. Another issue is the computational load even if the measurements could be exchanged among the neighbouring nodes. We introduce node-wise separable likelihoods to tackle these challenges later in Section IV. Next, we give a brief outline of decentralised estimation in sensor networks based on pairwise MRFs.

A. Decentralised Estimation Using Belief Propagation

Consider the MMSE parameter estimator based on the joint posterior in (10). This can be decomposed as a concatenation of the MMSE estimates of θ_i s based on the marginal distributions. The pairwise MRF model in (10) allows the computation of the marginal densities through iterative local message passings such as Belief Propagation (BP) [24], when \mathcal{G} contains no cycles. Specifically, the nodes maintain distributions over their local variables and update them based on messages from their neighbours which summarise the information neighbours have gained on these variables. This is described by the set of equations

$$m_{ji}(\theta_i) = \int \psi_{ij}^k(\theta_i, \theta_j) \psi_j(\theta_j) \prod_{i' \in \text{ne}(j) \setminus i} m_{i'j}(\theta_j) d\theta_j, \quad (12)$$

$$\tilde{p}_i(\theta_i) = k_i \psi_i(\theta_i) \prod_{j \in \text{ne}(i)} m_{ji}(\theta_i), \quad (13)$$

for all $i \in \mathcal{V}$, where k_i s are scale factors ensuring that $\tilde{p}_i(\theta_i)$ s integrate to unity.

The fixed points of the set of equations above are the marginals of (10) in the case of a cycle-free \mathcal{G} . In BP iterations, nodes simultaneously send messages to their neighbours using (12) (often starting with constants as the previously received messages) and update their local “belief” using (13). If \mathcal{G} contains no cycles, \tilde{p}_i s are guaranteed to converge to the fixed points, i.e., the marginals of (10), in a finite number of steps [24].

For the case in which \mathcal{G} contains cycles, message and update equations are still well defined. BP iterations on loopy graphs, however, are not guaranteed to convergence to an equilibrium, in general [23, Sec.4]. The fixed points of Eq.s (13) and (12), if they exist, are often not equal to the marginals of the

$$\begin{aligned}
 & D(p(Z_k^i, Z_k^j | Z_{1:k-1}^i, Z_{1:k-1}^j, \theta_{i,j}) | | q(Z_k^i, Z_k^j | Z_{1:k-1}^i, Z_{1:k-1}^j, \theta_{i,j})) \\
 & = I(Z_k^j; Z_{1:k-1}^j | Z_{1:k-1}^i, \theta_{i,j}) + I(Z_k^i; Z_{1:k-1}^i | Z_{1:k-1}^j, \theta_{i,j}) + I(Z_k^i; Z_k^j | Z_{1:k-1}^i, Z_{1:k-1}^j, \theta_{i,j}), \quad (18) \\
 & \leq H(\mathbf{X}_k | Z_{1:k-1}^i, \theta_{i,j}) + H(\mathbf{X}_k | Z_{1:k-1}^j, \theta_{i,j}) - H(\mathbf{X}_k | Z_{1:k-1}^i, Z_{1:k-1}^j, \theta_{i,j}) \\
 & \quad - \max\{H(\mathbf{X}_k | Z_k^i, Z_{1:k-1}^i, Z_{1:k-1}^j, \theta_{i,j}), H(\mathbf{X}_k | Z_k^j, Z_{1:k-1}^i, Z_{1:k-1}^j, \theta_{i,j})\}. \quad (19)
 \end{aligned}$$

global distribution, but reside in vicinities of them. Additional details on the convergence of BP on graphs with cycles can be found in [23] and the references therein. Loopy versions of BP has been very successful in finding approximate marginal distributions in many applications including fusion problems in sensor network (see, for example [20], and the references therein).

As a result, we distribute the calibration task over the network by using BP iterations to compute the marginals of (10) (exactly or approximately, in the cases of tree or loopy structured \mathcal{G} s, respectively). Since the neighbours in \mathcal{G} share a communication link, the messages of BP map directly onto the communication network and the i th sensor localises itself using, e.g., the MMSE rule, with its local marginal.

The next step involves the challenge of designing efficient computational procedures for approximating the edge potentials of the pairwise MRF model, which is discussed next.

IV. NODE-WISE SEPARABLE EDGE POTENTIALS

The MRF model in (10) and (11) does not readily allow for distributed calibration because the potential functions are the centralised likelihoods for sensor pairs (i.e., Eq.s (7) and (8) for sensors i and j). We overcome this problem by introducing an approximation for the instantaneous likelihood, or, update, term in the RHS of (11) which can be computed in a distributed fashion. This approximation has a node-wise separable form as a product of terms based on locally available information such as multi-object models received from the neighbours (for example, posterior multi-object Poisson distributions) and locally collected measurements of the multi-object scene. Specifically, we approximate the edge potential with a product of the form

$$\tilde{\psi}_{ij}^k(\theta_i, \theta_j) = l_{ij}^k(\theta_i, \theta_j) l_{ji}^k(\theta_i, \theta_j) \quad (14)$$

where l_{ij}^k is computed at node i in a recursive fashion and based on only $Z_{1:k}^i$ and the incoming posteriors from sensor j and vice versa.

Let us consider the update term in the RHS of (11) and denote the pair (θ_i, θ_j) by $\theta_{i,j}$. This term can be further factorised in alternative ways as follows:

$$\begin{aligned}
 & p(Z_k^i, Z_k^j | Z_{1:k-1}^i, Z_{1:k-1}^j, \theta_{i,j}) \\
 & = p(Z_k^i | Z_k^j, Z_{1:k-1}^i, Z_{1:k-1}^j, \theta_{i,j}) p(Z_k^j | Z_{1:k-1}^i, Z_{1:k-1}^j, \theta_{i,j}) \\
 & = p(Z_k^j | Z_k^i, Z_{1:k-1}^i, Z_{1:k-1}^j, \theta_{i,j}) p(Z_k^i | Z_{1:k-1}^i, Z_{1:k-1}^j, \theta_{i,j}) \\
 & = \left(p(Z_k^i | Z_k^j, Z_{1:k-1}^i, Z_{1:k-1}^j, \theta_{i,j}) \times \right. \\
 & \quad \left. p(Z_k^j | Z_{1:k-1}^i, Z_{1:k-1}^j, \theta_{i,j}) \right)^{1/2} \times \\
 & \quad \left(p(Z_k^j | Z_k^i, Z_{1:k-1}^i, Z_{1:k-1}^j, \theta_{i,j}) \times \right. \\
 & \quad \left. p(Z_k^i | Z_{1:k-1}^i, Z_{1:k-1}^j, \theta_{i,j}) \right)^{1/2} \quad (15)
 \end{aligned}$$

In the first and second lines above, the chain rule is used and the third equality follows from taking the geometric mean of the first two expressions. The conditioning of these factors to the measurement histories of both sensors prevents decentralisation. An alternative is to approximate the four factors in Eq.(15) by leaving out the history of sensor i (sensor j) in conditioning of the first and last (second and third) terms of (15) together with any current measurements, i.e.,

$$\begin{aligned}
 & q(Z_k^i, Z_k^j | Z_{1:k-1}^i, Z_{1:k-1}^j, \theta_{i,j}) \\
 & \triangleq \frac{1}{\kappa} \left(p(Z_k^i | Z_{1:k-1}^j, \theta_{i,j}) p(Z_k^j | Z_{1:k-1}^i, \theta_{i,j}) \right)^{1/2} \\
 & \quad \times \left(p(Z_k^j | Z_{1:k-1}^i, \theta_{i,j}) p(Z_k^i | Z_{1:k-1}^j, \theta_{i,j}) \right)^{1/2} \\
 & = p(Z_k^i | Z_{1:k-1}^j, \theta_{i,j}) p(Z_k^j | Z_{1:k-1}^i, \theta_{i,j}) \quad (16)
 \end{aligned}$$

$$\begin{aligned}
 & p(Z_k^i, Z_k^j | Z_{1:k-1}^i, Z_{1:k-1}^j, \theta_{i,j}) \approx \\
 & \quad q(Z_k^i, Z_k^j | Z_{1:k-1}^i, Z_{1:k-1}^j, \theta_{i,j}) \quad (17)
 \end{aligned}$$

where κ above is the normalisation constant and it can easily be shown that $\kappa = 1$.

The appeal of this approximation is that its factors depend on single sensor histories allowing us to avoid centralisation. The computation of these factors in terms of local information and incoming multi-object posteriors from the neighbours and the utilisation of the update in (17) in a message passing scheme are detailed in Section V. Now, we consider the approximation quality in terms of the Kullback-Leibler (KL) divergence [34] between the centralised update term in the left hand side (LHS) of (17) with respect to its approximation on the RHS:

Proposition 4.1: The KL divergence between the centralised update in (8) and the node-wise separable approximation in (16) equals to a sum of Mutual Information (MI) [34] terms given in (18).

Proof. See Appendix A. ■

The first two MI terms in Proposition 4.1 measure the departure of the current measurement and the sensor history from conditional independence when they are conditioned on the other sensor history instead of the current state \mathbf{X}_k for which $Z_k^i \perp\!\!\!\perp Z_{1:k-1}^i | \mathbf{X}_k, \theta_{i,j}$ and $Z_k^j \perp\!\!\!\perp Z_{1:k-1}^j | \mathbf{X}_k, \theta_{i,j}$ hold leading to

$$I(Z_k^i; Z_{1:k-1}^i | \mathbf{X}_k, \theta_{i,j}) = I(Z_k^j; Z_{1:k-1}^j | \mathbf{X}_k, \theta_{i,j}) = 0.$$

The third term is a measure of departure from conditional independence for the current measurements given the measurement histories. Therefore, the RHS of (18) is zero if $Z_k^i \perp\!\!\!\perp Z_{1:k-1}^i | Z_{1:k-1}^j, \theta_{i,j}$ and $Z_k^j \perp\!\!\!\perp Z_{1:k-1}^j | Z_{1:k-1}^i, \theta_{i,j}$ hold together with $Z_k^i \perp\!\!\!\perp Z_k^j | Z_{1:k-1}^i, Z_{1:k-1}^j, \theta_{i,j}$ simultaneously. This condition is satisfied, for example, in the case that either of the measurement histories $Z_{1:k-1}^i$ and $Z_{1:k-1}^j$ are

sufficient statistics for X_k and the true state can be predicted by both sensors with probability one. One should not expect this level of prediction accuracy in realistic tracking scenarios, therefore, it is instructive to relate the KL divergence in (18) further to the uncertainty on X_k given the sensor histories.

Corollary 4.2: The KL divergence term in (18) is upper bounded by the difference between the total local state prediction entropies and the entropies of the joint prediction and its most uncertain single sensor update given in (19) with H denoting the Shannon Entropy [34].

Proof. See Appendix B. \blacksquare

Corollary 4.2 relates the approximation quality of the node-wise separable updates to the uncertainties in the object state prediction and estimation. The first two terms in the RHS of (19) measure the uncertainties in the locally predicted object states. Subtracted from their sum are the uncertainty in the joint prediction based on the histories of both of the sensors and the entropy of the single sensor update upon the joint prediction that has the highest value. Therefore, a better quality of approximation should be expected as the local prediction densities become more concentrated around a single point in the state space.

Consider, for example, range-bearing sensors providing noisy measurements of object positions in polar coordinates. Tracking filters can provide fairly accurate predictions and estimates of object locations and velocities using typical measurements and with increasing k . This in turn results with smaller values for the bound in (19). An alternative in which these conditions cannot be satisfied with guarantees is a bearing-only sensor scenario consisting of a single object and two identical sensors. Suppose that the object moves along the line-of-sight (LOS) of one of the sensors. The local target prediction as well as the posterior distribution at time k will typically have a probability mass spread around the line-of-sight. The amount of this spread, which can be measured by the entropy H , will drop significantly when state distributions are conditioned also on the other sensor's history [35] yielding a large value for (19).

The use of the node-wise separable term in (16) to update the dynamic MRF edge potentials given by (11) leads to the following recursive formulae:

$$\begin{aligned} \tilde{\psi}_{ij}^k(\theta_i, \theta_j) &= \tilde{\psi}_{ij}^{k-1}(\theta_i, \theta_j) q(Z_k^i, Z_k^j | Z_{1:k-1}^i, Z_{1:k-1}^j, \theta_{i,j}), \\ &= \prod_{t=0}^{k-1} p(Z_t^i | Z_{1:t-1}^i, \theta_{i,j}) p(Z_t^j | Z_{1:t-1}^j, \theta_{i,j}), \\ &= l_{ij}^k(\theta_i, \theta_j) l_{ji}^k(\theta_i, \theta_j), \end{aligned} \quad (20)$$

where the node-wise terms in (14) are the products of individual node-wise separable update factors over time defined in a recursive fashion:

$$l_{ij}^k(\theta_i, \theta_j) \triangleq l_{ij}^{k-1}(\theta_i, \theta_j) p(Z_k^i | Z_{1:k-1}^i, \theta_{i,j}), \quad (21)$$

$$l_{ji}^k(\theta_i, \theta_j) \triangleq l_{ji}^{k-1}(\theta_i, \theta_j) p(Z_k^j | Z_{1:k-1}^j, \theta_{i,j}). \quad (22)$$

Here, subscript ij indicates that the associated term is computed at sensor i and will be transmitted to sensor j and vice versa.

V. THE COOPERATIVE CALIBRATION SCHEME

In this section, we describe the cooperative calibration scheme based on the node-wise separable edge potentials. First, in order to facilitate online processing within the Bayesian paradigm, the measurement histories are partitioned into time windows of length T , and, θ is evolved with artificial dynamics between two consecutive windows leading to prediction-update cycles [28]. Let us denote the n th window of sensor j by $Z_n^j \triangleq Z_{(n-1)T+1:nT}^j$. A recursive rule is, then, given by

$$p_n(\theta_n | Z_{0:n}^1, \dots, Z_{0:n}^N) \propto l(Z_n^1, \dots, Z_n^N | \theta_n) p_{n|n-1}(\theta_n | Z_{0:n-1}^1, \dots, Z_{0:n-1}^N), \quad (23)$$

$$p_{n|n-1,i}(\theta_{n,i} | Z_{0:n-1}^1, \dots, Z_{0:n-1}^N) = \int \beta_n(\theta_{n,i} | \theta_{n-1,i}) p_{n-1,i}(\theta_{n-1,i} | Z_{0:n-1}^1, \dots, Z_{0:n-1}^N) d\theta_{n-1,i}, \quad (24)$$

for $i = 1, \dots, N$ where subscript i denotes the i th marginal. In the second line we apply dynamics to the single sensor parameters $\theta_{n-1,i}$ independently which is specified by the conditional density $\beta_n(\theta_{n,i} | \theta_{n-1,i})$. β_n models (small) Brownian motion steps, i.e.,

$$\beta_n(\theta_{n,i} | \theta_{n-1,i}) = \mathcal{N}(\theta_{n,i} - \theta_{n-1,i}; \mathbf{0}, \mathbf{D}_n), \quad (25)$$

where \mathcal{N} is a multi-dimensional Gaussian with an all zero mean vector. $\mathbf{D}_n = \text{diag}(\sigma_{n,1}^2, \dots, \sigma_{n,d}^2)$ is a $d \times d$ diagonal covariance matrix and $\sigma_{n,1}, \dots, \sigma_{n,d}$ are the Brownian motion parameters specifying the step sizes in each direction. Consequently, the prior distribution in (23) becomes the product of these marginals, i.e.,

$$p_{n|n-1}(\theta_n | Z_{0:n-1}^1, \dots, Z_{0:n-1}^N) = \prod_{i \in \mathcal{V}} p_{n|n-1,i}(\theta_{n,i} | Z_{0:n-1}^1, \dots, Z_{0:n-1}^N).$$

The prediction step in (24) can already be performed locally. The discussion in Sections III and IV relates to the update step in (23). In other words, we replace (23) with

$$\tilde{p}_n(\theta_n | Z_{0:n}^1, \dots, Z_{0:n}^N) \propto \prod_{(i,j) \in \mathcal{E}} \tilde{\psi}_{i,j}^k(\theta_{n,i}, \theta_{n,j}) \prod_{i \in \mathcal{V}} p_{n|n-1,i}(\theta_{n,i}). \quad (26)$$

We describe the cooperative update of the marginal distributions for a time-window of length T . Then, these steps are repeated in consecutive time windows.

The conceptual steps followed at platform i is given as pseudo-code in Algorithm 1. Initially, platform i specifies an a priori distribution over its calibration parameters and the multi-object scene. Here, k becomes a time index for traversing the current window n . In an infinite loop, platform i proceeds with distributed fusion by first collecting measurements from the multi-object scene and filtering them to find the local multi-object density $f_i(X_k | Z_{1:k}^i)$. This density is exchanged with the neighbouring platforms and then fused with the incoming state densities from the neighbours.

For cooperative calibration, the local node-wise update term $p(Z_k^i | Z_{1:k-1}^i, \theta_{i,j})$ is found for all neighbours $j \in ne(j)$. The

Algorithm 1 Pseudo-code for fusion node i in the cooperative calibration scheme.

```

1: Specify  $p_{1|0,i}(\theta_i)$ ,  $f_i(X_0)$ ,  $T$ ,  $S$ 
2:  $n \leftarrow 1$ ,  $k \leftarrow 1$ 
3: Exchange  $\hat{\theta}_i = E_{p_{1|0,i}}\{\theta_i\}$  s with  $j \in ne(i)$ 
4: while 0 do
5:   Collect measurements and store in  $Z_k^i$ 
6:   Filter  $Z_k^i$  and find the Poisson multi-object density  $f_i(X_k|Z_{1:k}^i)$  ▷ Local filtering
7:   Exchange  $f_i(X_k|Z_{1:k}^i)$  s with  $j \in ne(i)$  ▷ Distributed fusion 1
8:   Fuse  $f_i(X_k|Z_{1:k}^i)$  and  $f_j(X_k|Z_{1:k}^j)$ s using  $\hat{\theta}_i$  and  $\hat{\theta}_j$ s ▷ Distributed fusion 2
9:   Find  $p(Z_k^i|Z_{1:k-1}^j, \theta_{i,j})$  using  $Z_k^i$  and  $f_j(X_{k-1}|Z_{1:k-1}^j)$  for all  $j \in ne(i)$  ▷ Eq.s(27) and (29)
10:  Update  $l_{ij}^k$  with  $p(Z_k^i|Z_{1:k-1}^j, \theta_{i,j})$  for all  $j \in ne(i)$  ▷ Eq.(21)
11:  if  $k = T$  then
12:    Exchange  $l_{ij}^k$  s with  $j \in ne(i)$ 
13:    Find  $\tilde{\psi}_{ij}^k$  for  $j \in ne(i)$  ▷ Eq.(14)
14:    Perform loopy BP for  $S$  steps to find  $\tilde{p}_{n,i}(\theta_i)$  ▷ Eq.s(12) and (13) to estimate the  $i$ th marginal of (26).
15:    Exchange  $\hat{\theta}_i = E_{\tilde{p}_{n,i}}\{\theta_i\}$  s with  $j \in ne(i)$ 
16:     $p_{n+1|i,n,i} \leftarrow \tilde{p}_{n,i} * \mathcal{N}(\cdot; \mathbf{0}, \mathbf{C}_n)$  ▷ Brownian motion (Eqs.(24) and (25))
17:     $f_i(X_0) \leftarrow f_i(X_k|Z_{1:k}^i)$ 
18:     $k \leftarrow 0$ ,  $n \leftarrow n + 1$ 
19:  end if
20:   $k \leftarrow k + 1$ 
21: end while

```

computation is carried out in two steps: First, based on the Poisson density received at $k - 1$, i.e.,

$$f_j([X_{k-1}]_j|Z_{1:k-1}^j) = Pois([X_{k-1}]_j; \lambda_{k-1,j}, s_{k-1,j}([x]_j)),$$

the corresponding prediction density is found. Here, we explicitly indicate that the argument of the incoming density is in the SCCS of sensor j . Poisson predictions often incorporate a probability for an object with state x to continue to exist at the next time step denoted by $P_S(x)$. The predicted Poisson model is characterised by [25]

$$\lambda_{k|k-1,j}(\theta_{i,j}) = \lambda_{k-1,j} \int P_S(\tau_i \circ \tau_j^{-1}(x; \theta_{i,j})) \times s_{k-1,j}(x) dx,$$

$$s_{k|k-1,j}([x]_i; \theta_{i,j}) \propto \int \pi([x]_i | \tau_i \circ \tau_j^{-1}(x'; \theta_{i,j})) \times P_S(\tau_i \circ \tau_j^{-1}(x'; \theta_{i,j})) s_{k-1,j}(x') dx', \quad (27)$$

where π is the Markov transition modelling object motion (Section II). Here, the integral variables are in the j th SCCS and the composite function $\tau_i \circ \tau_j^{-1}$ transforms its argument in the j th SCCS to its counterpart in the i th SCCS given the calibration parameters $\theta_{i,j}$.

The predictive Poisson model $f_j(X_k|Z_{1:k-1}^j; \theta_{i,j})$ specified by (27) has its X_k argument in the i th SCCS. As a second step, the node-wise update term is computed based on this predictive density and the current measurements. Using the conditional independence relation $\mathbf{Z}_k^i \perp\!\!\!\perp \mathbf{Z}_{1:k-1}^j | \mathbf{X}_k$, it can easily be shown that

$$p(Z_k^i|Z_{1:k-1}^j, \theta_{i,j}) = \int p(Z_k^i|X_k) f_j(X_k|Z_{1:k-1}^j; \theta_{i,j}) \delta X_k, \quad (28)$$

where the integral variable is set valued and (28) is a set integral (see Appendix C for an overview of such integrals).

In addition, the RHS of (28) takes a simple form for a Poisson f_j and the sensor measurement model described in Section II. In particular,

$$p(Z_k^i|Z_{1:k-1}^j, \theta_{i,j}) = \exp\left(-\lambda_{C,i} - \lambda_{k|k-1,j}(\theta_{i,j}) \int P_{D,i}(x) s_{k|k-1,j}(x; \theta_{i,j}) dx\right) \times \prod_{z \in Z_k^i} \left(\lambda_{C,i} s_{C,i}(z) + \lambda_{k|k-1,j}(\theta_{i,j}) \times \int P_{D,i}(x) l_i(z|x) s_{k|k-1,j}(x; \theta_{i,j}) dx \right), \quad (29)$$

where subscript i denotes that the quantity belongs to the measurement model of node i (see, e.g., [36, Proposition 11.3], [29, Eq.(116)], or [16, Eq.(8)]). This term is then used to update the likelihood factor l_{ij}^k in (21).

At the end of the time window of length T , the node-wise terms are exchanged with the neighbouring nodes and the edge potentials are constructed using (14). Then, S steps of loopy BP is performed using (12) and (13) to estimate the i th local marginal density of the MRF calibration model in (26). In the next step, the expected value of this marginal is found which becomes the current estimate of the parameters, and is exchanged with the neighbours. Before proceeding with the next time window, the local parameter distribution is convolved with a zero mean Gaussian realising the Brownian motion step given by (24) and (25).

It is worth noting that (28) is valid for arbitrary sensor coverages: If different sensor pairs observe different targets in common due to partially overlapping coverages, it suffices to select the probability of detection $P_{D,i}(x)$ embedded in the likelihood term as zero outside the coverage. The explicit

evaluation given by (29), however, is valid when there are no targets observed by sensor i but not by sensor j . For arbitrary coverages, (29) would have the same form with the spurious measurement model (i.e., the terms related to clutter) adapting to include observations from those targets in the coverage of sensor i but not j . Further elaboration on this case remains as future work. For the rest of this article, we restrict our attention to the setting in which all sensors observe the same targets.

VI. COOPERATIVE SENSOR SELF-LOCALISATION USING MONTE CARLO METHODS

This section describes a realisation of the cooperative calibration scheme introduced in the previous section for sensor self-localisation using Monte Carlo methods. Central to the realisation of the conceptual procedure in Algorithm 1 are particle representations of the continuous densities involved and approximate computations which in turn require slight modifications to the original steps.

For sensor self-localisation, we consider a co-planar network and the respective calibration parameters as the sensor locations in a network coordinate system. An arbitrary node is selected as the centre of the network coordinate frame. For example, the SCCS of node 1 can be selected as the reference by setting $\theta_1 = [0, 0]^T$, without loss of generality. Thus, the transform in (2) maps the points in the 1st SCCS to their j th SCCS counterparts.

Consider the pair $(i, j) \in \mathcal{E}$. The composite function $\tau_i \circ \tau_j^{-1}$ appearing in (27) is found as

$$\begin{aligned} [x]_i &= \tau_i \circ \tau_j^{-1}([x]_j; \theta_i, \theta_j) \\ &= [x]_j - \theta_i + \theta_j. \end{aligned} \quad (30)$$

Local filtering can be performed using, for example, Sequential Monte Carlo methods [37] within the filtering approaches discussed in Section II-B such as Sequential Monte Carlo realisations of RFS filters [38]. Using its local filter, node j will have transmitted an approximation to $Pois(\cdot; \lambda_{k-1,j}, s_{k-1,j}(x))$ specified by an estimate $\hat{\lambda}_{k-1,j}$ for the expected number of objects and a weighted set of independent, identically distributed (i.i.d) samples (or, particles) $\{x_{k-1,j}^{(m)}, \zeta_{k-1,j}^{(m)}\}_{m=1}^M$ that encode an empirical distribution given by

$$\hat{S}_{k-1,j}(dx) = \sum_{m=1}^M \zeta_{k-1,j}^{(m)} \delta_{x_{k-1,j}^{(m)}}(dx), \quad (31)$$

where δ_x is the Dirac measure concentrated at x .

The predictive Poisson model in (27) is then approximated by substituting the empirical distribution in (31) together with $\hat{\lambda}_{k-1,j}$ into (27), and using the Monte Carlo integration principle [39, Chp.3]. This is akin to the prediction stage of the Sequential Monte Carlo PHD filter [40] and the resulting approximation (also considering (30)) is given by

$$\begin{aligned} \hat{\lambda}_{k|k-1,j}(\theta_{i,j}) &= \lambda_{k-1,j} \sum_m \zeta_{k-1,j}^{(m)} P_S(x_{k-1,j}^{(m)} - \theta_j + \theta_i), \\ \hat{S}_{k|k-1,j}(dx; \theta_{i,j}) &= \sum_m \zeta_{k|k-1,j}^{(m)} \delta_{x_{k|k-1,j}^{(m)}}(dx), \\ x_{k|k-1,j}^{(m)} &\sim \pi(x|x_{k-1,j}^{(m)} - \theta_j + \theta_i), \end{aligned} \quad (32)$$

$$\zeta_{k|k-1,j}^{(m)} = \frac{\zeta_{k-1,j}^{(m)} P_S(x_{k-1,j}^{(m)} - \theta_j + \theta_i)}{\sum_{m'} \zeta_{k-1,j}^{(m')} P_S(x_{k-1,j}^{(m')} - \theta_j + \theta_i)}.$$

The update term in (29) is computed using these quantities which yields

$$\begin{aligned} \hat{p}(Z_k^i | Z_{1:k-1}^j, \theta_{i,j}) &= \\ \exp\left(-\lambda_{C,i} - \hat{\lambda}_{k|k-1,j}(\theta_{i,j}) \sum_m \zeta_{k|k-1,j}^{(m)} P_{D,i}(x_{k|k-1,j}^{(m)})\right) &\times \\ \prod_{z \in Z_k^i} \left(\lambda_{C,i} s_{C,i}(z) + \hat{\lambda}_{k|k-1,j}(\theta_{i,j}) \times \right. & \\ \left. \sum_m \zeta_{k|k-1,j}^{(m)} P_{D,i}(x_{k|k-1,j}^{(m)}) l_i(z|x_{k|k-1,j}^{(m)})\right). & \quad (33) \end{aligned}$$

The likelihood update $p(Z_k^i | Z_{1:k-1}^j, \theta_{i,j})$ local to node i can only be estimated for a finite set of $\theta_{i,j}$ values which in turn will be used to estimate the node-wise likelihood term l_{ij}^k . It is beneficial to use the same set of parameter values at node j in order to estimate l_{ij}^k as the edge potential $\tilde{\psi}_{ij}^k$, then, can be found by simply taking the product of these estimates. For this reason, we generate L equally weighted samples from $p_{n|n-1,i}(\theta_i)$ and $p_{n|n-1,j}(\theta_j)$, and, obtain $\{\theta_{n|n-1,i}^{(l)}\}_{l=1}^L$ and $\{\theta_{n|n-1,j}^{(l)}\}_{l=1}^L$, respectively⁶. We use the concatenation of these values, i.e.,

$$\theta_{i,j}^{(l)} \triangleq \left[\left(\theta_{n|n-1,i}^{(l)} \right)^T, \left(\theta_{n|n-1,j}^{(l)} \right)^T \right]^T, \quad (34)$$

in computing (33) at both node i and j .

In order to construct (34), the nodes exchange their local particle sets representing $p_{n|n-1,i}(\theta_i)$ s as an additional communication step to Algorithm 1, before starting to process a new time window. Thus, at each time step k within the window, the update term (33) is computed for all $\{\theta_{i,j}^{(l)}\}_{l=1}^L$ and for all $j \in ne(i)$ in order to update $\{\hat{l}_{i,j}^k(\theta_i^l, \theta_j^l)\}_{l=1}^L$. At the last step of the time window $k = T$, the estimated node wise terms are exchanged and the edge potentials $\{\tilde{\psi}_{i,j}^k(\theta_i^l, \theta_j^l)\}_{l=1}^L$ are found by simply taking the element-wise product of the node-wise separable terms (Eq.(14)).

The loopy BP realisation with these edge potentials follows the non-parametric BP (NBP) approach [41] and represents both the messages in (12) and the local marginals in (13) by particle sets.

First, we describe the message computations: Consider (12) for the MRF model in (26) and suppose that i.i.d samples from the (scaled) product of the j th local prior and the incoming messages from all neighbours except i are given, i.e.,

$$\bar{\theta}_j^{(l)} \sim p_{n|n-1}(\theta_j) \prod_{i' \in ne(j)/i} m_{i'j}(\theta_j) \text{ for } l = 1, \dots, L. \quad (35)$$

Based on these samples, the message from node j to i (scaled to one) is approximated by a smoothed density estimate in NBP [41]. We use Gaussian kernels leading to the

⁶The particle weights are equal to $1/L$ for the case and omitted in the notation for the simplicity of exposition.

approximation given by

$$\begin{aligned} \hat{m}_{ji}(\theta_i) &= \sum_{l=1}^L \omega_{ji}^{(l)} \mathcal{N}(\theta_i; \theta_{ji}^{(l)}, \Lambda_{ji}), \\ \theta_{ji}^{(l)} &= \tau_i^{-1} \circ \tau_j(\bar{\theta}_j^{(l)}; \theta_i^{(l)}, \theta_j^{(l)}) \\ &= \bar{\theta}_j^{(l)} + \theta_i^{(l)} - \theta_j^{(l)}, \\ \omega_{ji}^{(l)} &= \frac{\hat{\psi}_{i,j}^k(\theta_i^{(l)}, \theta_j^{(l)})}{\sum_{l'=1}^L \hat{\psi}_{i,j}^k(\theta_i^{(l')}, \theta_j^{(l')})}, \end{aligned} \quad (36)$$

where the kernel weights are the normalised edge potentials. Λ_{ji} is related to a *bandwidth* parameter that can be found using Kernel Density Estimation (KDE) techniques. In particular, we use the rule-of-thumb method in [42] and find

$$\begin{aligned} \Lambda_{ji} &= (5/4L)^{-1/3} \hat{C}_{ji}, \\ \hat{C}_{ji} &= \sum_{l'} \sum_l \omega_{ji}^{(l')} \omega_{ji}^{(l)} (\theta_{ji}^{(l')} - \hat{m}_{ji})(\theta_{ji}^{(l)} - \hat{m}_{ji})^T, \\ \hat{m}_{ji} &= \sum_{l=1}^L \omega_{ji}^{(l)} \theta_{ji}^{(l)} \end{aligned}$$

where \hat{m}_{ji} and \hat{C}_{ji} are the empirical mean and covariance of the samples, respectively.

Second, we consider sampling from the updated marginal in (13) for the MRF in (26). We use the weighted bootstrap, or, sampling/importance resampling, approach [43] with samples generated from the (scaled) product of Gaussian densities with mean and covariance equal to the empirical mean and covariance of the particle sets, respectively. In other words, given \hat{m}_{ji} and \hat{C}_{ji} as above, we generate

$$\begin{aligned} \theta_{i,\beta}^{(l)} &\sim \beta(\theta_i), \quad l = 1, \dots, L, \\ \beta(\theta_i) &\propto \mathcal{N}(\theta_i; \hat{m}_{n|n-1,i}, \hat{C}_{n|n-1,i}) \prod_{j \in ne(i)} \mathcal{N}(\theta_i; \hat{m}_{ji}, \hat{C}_{ji}). \end{aligned}$$

The particle weights for these samples to represent the updated marginal is given by

$$\begin{aligned} \omega_{i,\beta}^{(l)} &= \hat{\omega}_{i,\beta}^{(l)} / \sum_{l'=1}^L \hat{\omega}_{i,\beta}^{(l')} \\ \hat{\omega}_{i,\beta}^{(l)} &= \left(\hat{p}_{n|n-1,i}(\theta_{i,\beta}^{(l)}) \prod_{j \in ne(i)} \hat{m}_{ji}(\theta_{i,\beta}^{(l)}) \right) / \beta(\theta_{i,\beta}^{(l)}) \end{aligned}$$

where $\hat{p}_{n|n-1,i}$ is a KDE found using $\{\theta_{n|n-1,i}^{(l)}\}_{l=1}^L$ with the rule-of-thumb method described above⁷. Thus, the local calibration marginal is estimated by

$$\hat{P}_{n,i}(d\theta_i) = \sum_{l=1}^L \omega_{i,\beta}^{(l)} \delta_{\theta_{i,\beta}^{(l)}}(d\theta_i). \quad (37)$$

As the final step of the bootstrap, $\{\theta_{i,\beta}^{(l)}, \hat{\omega}_{i,\beta}^{(l)}\}_{l=1}^M$ is resampled (with replacement) leading to equally weighted particles from $p_{n,i}(\theta_i)$, i.e., $\{\theta_{n,i}^{(l)}\}_{l=1}^L$. We follow similar bootstrap steps in order to generate the samples in (35).

After nodes iterate the BP computations described above for S times, each node estimates its location by finding the

empirical mean of $\{\theta_{n,i}^{(l)}\}_{l=1}^L$.

Before proceeding with the next time window, Brownian motion is applied to $\{\theta_{n,i}^{(l)}\}_{l=1}^L$ yielding i.i.d. samples from $p_{n+1|n,i}(\theta_i)$, i.e.,

$$\theta_{n+1|n,i}^{(l)} = \theta_{n,i}^{(l)} + \epsilon_n^{(l)}, \quad \epsilon_n^{(l)} \sim \mathcal{N}(\cdot; \mathbf{0}, \sigma_n^2 \mathbf{I}), \quad (38)$$

where \mathbf{I} is the 2×2 identity matrix.

The communication messages in this implementation strategy for the conceptual procedure in Algorithm 1 consist of particle sets which are equivalently (unordered) numerical arrays. For distributed fusion, node i broadcasts an array of average length $\mathcal{O}(M_k M)$ to its neighbours in line 7 of Algorithm 1, where M_k is the number of targets and M is the number of particles used per target.

The exchanges for cooperative localisation have a period of T . First, nodes broadcast L particles, i.e., an $\mathcal{O}(L)$ array, to construct (34). Because these particles are generated from the *a priori* localisation distributions in lines 3 and 16, it is convenient to have these broadcasts take place in these lines. Hence, the expectation in line 3 can be performed at the neighbours using the empirical mean of the messages. After the node-wise separable terms are computed, they are exchanged in line 12 which corresponds to the transmission of an $\mathcal{O}(LD_i)$ array where D_i is the degree (or, the number of neighbours) of node i . The loopy BP messages in line 14 are encoded by the L particles given in (35). The mixture representation in (36) is recovered from these particles by using the weights already known both at the transmitting and the receiving nodes, and, the rule-of-thumb KDE method. As a result, S iterations of loopy BP corresponds to the transmission of an $\mathcal{O}(SLD_i)$ array. Note that, the exchange of location estimates from the current posterior marginals in line 15 leads to a comparably negligible, constant communication load. As a result, the communication complexity of the cooperative scheme for node i is $\mathcal{O}(L((S+1)D_i + 1))$.

VII. EXAMPLE

We demonstrate the cooperative self-calibration algorithm introduced in Section V for self-localisation using the Monte Carlo realisation described in Section VI. The example scenario is depicted in Figure 1 and consists of four objects moving in a surveillance region being observed by nine sensors for 200 steps⁸. The nodes networked through communication links which are the edges of the communication graph \mathcal{G} (blue lines).

The object states are described by their planar position and velocity, i.e., $x_n = [(x_n^l)^T, (x_n^v)^T]^T$. The trajectories are obtained using a linear constant velocity motion model with (slight) additive process noise:

$$\begin{aligned} \pi(x_n | x_{n-1}) &= \mathcal{N}(x_n; Fx_{n-1}, Q) \\ F &= \begin{bmatrix} \mathbf{I} & \mathbf{I} \\ \mathbf{0} & \mathbf{I} \end{bmatrix}, \quad Q = 0.075 \begin{bmatrix} \frac{1}{3}\mathbf{I} & \frac{1}{2}\mathbf{I} \\ \frac{1}{2}\mathbf{I} & \mathbf{I} \end{bmatrix} \end{aligned}$$

⁸Results using this scheme on a problem with a smaller scale can be found in [44].

⁷If $p_{n|n-1,i}$ can be evaluated exactly, it is replaced with $\hat{p}_{n|n-1,i}$.

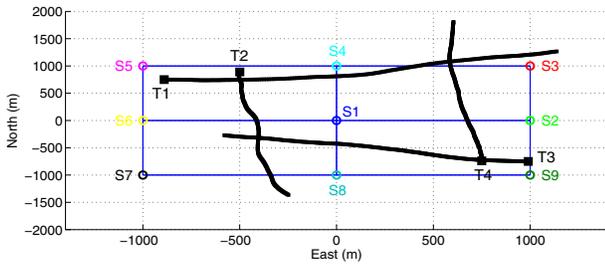


Fig. 1. Nine sensor nodes networked through communication links (blue edges) observing four objects. Black curves are the object trajectories boxes indicating the initial positions.

where \mathbf{I} and $\mathbf{0}$ are the 2×2 identity and zero matrices, respectively.

The sensors measure the bearing angle and the range of the objects in their SCCS with standard deviations $\sigma_\phi = 1^\circ$ and $\sigma_R = 5m$, respectively. The likelihood model in (5) is then found as

$$l_i(z|x; \theta_i) = \mathcal{N}(\angle(x^l - \theta_i); 0, \sigma_\phi^2) \mathcal{N}(\|x^l - \theta_i\|; 0, \sigma_R^2).$$

Each object is detected with $P_D = 0.98$. The number of false alarms at time k is Poisson distributed with mean $\lambda_{C,i} = 2$ for $i \in \mathcal{V}$ and the associated spatial distribution is uniform in the sensor field-of-views. A typical realisation of measurements at sensor 7 over time can be seen in Figure 2.

The nodes filter their local measurements using the SMC PHD algorithm in [45]. The probability that an object with state x remains to exist in the next step is select $P_S(x) = 0.9$. The resulting Poisson multi-object models are exchanged with the neighbours in \mathcal{G} (for distributed fusion).

Node 1 is selected as the origin of the network coordinate system. For the other nodes, the localisation prior for the first time window, i.e., $p_{1|0,i}(\theta_i)$, is selected to be a uniform distribution over the sensing region. A window length of $T = 10$ is used for computing the node-wise separable edge potentials as detailed in Section VI. The first time window starts at $k = 40$ to avoid using the posteriors from the early (transient) stages of filtering. Because uniform localisation priors are used, the initial set of location values in (34) – at which the node-wise terms are evaluated – are selected

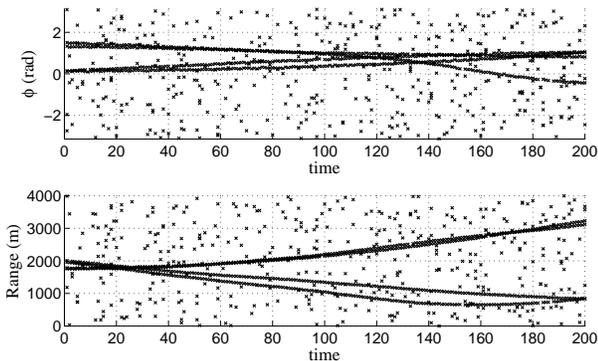


Fig. 2. Typical range and bearing measurements collected at sensor node 7. The field-of-view is the circular region around the sensor with a radius of 4000m.

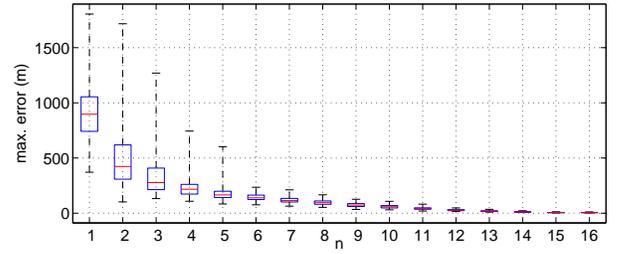


Fig. 3. Maximum localisation error in the network versus the iteration number of the proposed algorithm for 200 Monte Carlo runs. The boxes are centred at the median (red) with edges (blue) at the 25th and 75th percentiles.

as random permutations of an $L = 900$ node uniform grid over the sensing region. Thus, at each time step k within the window, the local likelihood updates for these values are computed using (32)–(34) and the node-wise terms (21),(22) are updated. At the end of the time window, these terms are exchanged to find the edge potentials (20). This stage is followed by $S = 8$ steps of nonparametric BP (Eq.s (35)–(37)).

Following the last NBP step, each node estimates its location as the empirical mean of the local marginal represented by $L = 900$ particles. A total of 16 iterations of the proposed algorithm is performed starting from $k = 50$ to 200 in every $T = 10$ steps. The Brownian motion parameter σ_n in (38) varies from $\sigma_1 = 600$ to $\sigma_{16} = 1$ in accordance with a square law.

We consider the maximum localisation error in the network for 200 Monte Carlo runs. In Fig. 3 we present the box-plot of these errors for all runs with respect to the time window index n . The average error at the final iteration $n = 16$ is 5.95m which is comparably close to the uncertainties in sensor measurements. The highest error, or, the error margin is 12.09m. Next, we consider the error margins for all steps and normalise them with the minimum distance between the nodes in this deployment, i.e., 1000m. The semi-log plot of the normalised error margins can be seen in Fig. 4. Note that the decay speed is close to a log-linear regime and reaches a minimum corresponding to approximately 1.2% of the normalising distance at $n = 16$. These results demonstrate that the proposed scheme is capable of providing self-localisation with good accuracy and small error margins.

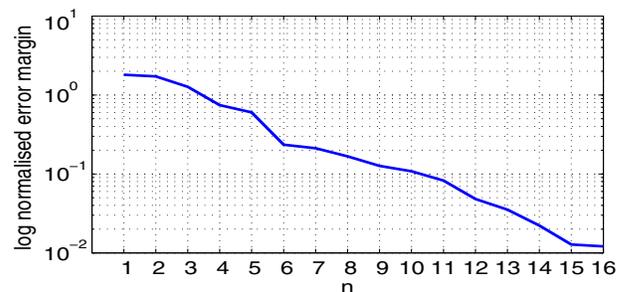


Fig. 4. Log-normalised error margin versus the iteration number n .

VIII. CONCLUSION

In this article, we propose a cooperative self-calibration scheme for sensor localisation in distribution fusion networks. The nodes in such networks collect measurements from a multi-object scene which involve uncertainties due to noise, probability of detection being less than one, and, the presence of false detections from the surroundings. Multi-sensor scalability is achieved by local filtering of these measurements, which in turn provides multi-object posteriors as the information entities to communicate and be fused for multi-sensor exploitation, as opposed to transmitting raw detections across the network.

We introduce a probabilistic model to estimate sensor calibration parameters without violating the locality structure of computations and by using nearest neighbour message exchanges. In particular, we develop node-wise separable parameter likelihoods which can be computed based on the locally available information for distributed fusion. The interactions between pairs of sensors sharing a communication link are used to build up a pairwise MRF model with these separable likelihoods as its edge potentials. Distributed estimation is, then, carried out using (loopy) BP over the communication graph.

Our approach inherits the capability of handling highly uncertain measurements from the local multi-object filters without posing any significant constraint on the system configuration. It can also be used in solving general calibration problems and at a fusion centre to avoid the computational burden of centralised multi-sensor filtering in parameter estimation. In the case of mobile sensor platforms, our framework can accommodate dynamical parameter models, as well.

There is a variety of possible extensions of this work. Different node wise separable parameter likelihoods can be found by using different strategies to approximate (15). More sophisticated sampling schemes can be adopted for the realisation of the algorithm. It is also possible to use more accurate multi-object models, in this framework, with the cost of increasing communication and/or computational demand.

APPENDIX

A. Proof of Proposition 4.1

We begin the proof by substituting the distributions of concern in the definition of (conditional) KL divergence. For the sake of a shorter notation, we denote the sensor histories from time 1 to $k-1$, i.e., $Z_{1:k-1}^i$ and $Z_{1:k-1}^j$ by \mathcal{H}^i and \mathcal{H}^j , respectively. We also drop the time indices in the current measurements Z_k^i and Z_k^j . The KL divergence in (18) is given by

$$\begin{aligned} D(p(Z^i, Z^j | \mathcal{H}^i, \mathcal{H}^j, \theta_{i,j}) || q(Z^i, Z^j | \mathcal{H}^i, \mathcal{H}^j, \theta_{i,j})) \\ = \int \delta Z^i \delta Z^j \delta \mathcal{H}^i \delta \mathcal{H}^j d\theta_{i,j} p(Z^i, Z^j, \mathcal{H}^i, \mathcal{H}^j, \theta_{i,j}) \times \\ \log \frac{p(Z^i, Z^j | \mathcal{H}^i, \mathcal{H}^j, \theta_{i,j})}{q(Z^i, Z^j | \mathcal{H}^i, \mathcal{H}^j, \theta_{i,j})} \\ = \int \delta Z^i \delta Z^j \delta \mathcal{H}^i \delta \mathcal{H}^j d\theta_{i,j} p(Z^i, Z^j, \mathcal{H}^i, \mathcal{H}^j, \theta_{i,j}) \times \\ \log \left(\frac{p(Z^i, Z^j | \mathcal{H}^i, \mathcal{H}^j, \theta_{i,j})}{p(Z^i | \mathcal{H}^j, \theta_{i,j}) p(Z^j | \mathcal{H}^i, \theta_{i,j})} \right). \quad (39) \end{aligned}$$

In the integration above, the variables regarding sensor measurements and histories are sets (as opposed to vectors), and, their integration is defined in Appendix C. As far as the proof is concerned, it suffices to have well-defined integration rules for the variables involved. The results hold true for any such variables including vector valued sensor measurements.

Next, we consider (39) and multiply the numerator and the denominator inside the log term by the product of $p(Z^i, Z^j | \mathcal{H}^i, \mathcal{H}^j, \theta_{i,j})$, $p(Z^i | \mathcal{H}^i, \mathcal{H}^j, \theta_{i,j})$, $p(Z^j | \mathcal{H}^i, \mathcal{H}^j, \theta_{i,j})$, $p(\mathcal{H}^i | \mathcal{H}^j, \theta_{i,j})$, and, $p(\mathcal{H}^j | \mathcal{H}^i, \theta_{i,j})$. Thus,

$$\begin{aligned} D(p(Z^i, Z^j | \mathcal{H}^i, \mathcal{H}^j, \theta_{i,j}) || q(Z^i, Z^j | \mathcal{H}^i, \mathcal{H}^j, \theta_{i,j})) \\ = \int \delta Z^i \delta Z^j \delta \mathcal{H}^i \delta \mathcal{H}^j d\theta_{i,j} p(Z^i, Z^j, \mathcal{H}^i, \mathcal{H}^j, \theta_{i,j}) \times \\ \left(\log \frac{p(Z^i, Z^j, \mathcal{H}^i | \mathcal{H}^i, \theta_{i,j})}{p(Z^j | \mathcal{H}^i, \theta_{i,j}) p(Z^i, \mathcal{H}^j | \mathcal{H}^i, \theta_{i,j})} \right. \\ \left. + \log \frac{p(Z^i, Z^j, \mathcal{H}^i | \mathcal{H}^j, \theta_{i,j})}{p(Z^i | \mathcal{H}^j, \theta_{i,j}) p(Z^j, \mathcal{H}^i | \mathcal{H}^j, \theta_{i,j})} \right. \\ \left. + \log \frac{p(Z^i | \mathcal{H}^i, \mathcal{H}^j, \theta_{i,j}) p(Z^j | \mathcal{H}^i, \mathcal{H}^j, \theta_{i,j})}{p(Z^i, Z^j | \mathcal{H}^j, \mathcal{H}^i, \theta_{i,j})} \right) \\ = I(Z^j; Z^i, \mathcal{H}^j | \mathcal{H}^i, \theta_{i,j}) + I(Z^i; Z^j, \mathcal{H}^i | \mathcal{H}^j, \theta_{i,j}) \\ - I(Z^j; Z^i | \mathcal{H}^j, \mathcal{H}^i, \theta_{i,j}) \\ = I(Z^j; \mathcal{H}^j | \mathcal{H}^i, \theta_{i,j}) + I(Z^j; Z^i | \mathcal{H}^j, \mathcal{H}^i, \theta_{i,j}) \\ + I(Z^i; \mathcal{H}^i | \mathcal{H}^j, \theta_{i,j}) + I(Z^j; Z^i | \mathcal{H}^j, \mathcal{H}^i, \theta_{i,j}) \\ - I(Z^j; Z^i | \mathcal{H}^j, \mathcal{H}^i, \theta_{i,j}) \\ = I(Z^j; \mathcal{H}^j | \mathcal{H}^i, \theta_{i,j}) + I(Z^i; \mathcal{H}^i | \mathcal{H}^j, \theta_{i,j}) \\ + I(Z^j; Z^i | \mathcal{H}^j, \mathcal{H}^i, \theta_{i,j}). \quad (40) \end{aligned}$$

The second equality above follows from the definition of mutual information (MI) [34], and, in the penultimate line, the chain rule for information [34] is used with the positive terms. ■

B. Proof of Corollary 4.2

We use the simplified notation introduced in Appendix A for the proof of Proposition 4.1. The data processing inequality (DPI) [34] applied simultaneously to the first two terms in the RHS of (18) (or, the RHS of (40) in the simplified notation) in accordance with the (conditional) chains $Z^j \leftrightarrow \mathbf{X}_k \leftrightarrow \mathcal{H}^j | \mathcal{H}^i, \theta_{i,j}$ and $Z^i \leftrightarrow \mathbf{X}_k \leftrightarrow \mathcal{H}^i | \mathcal{H}^j, \theta_{i,j}$, respectively, yields

$$\begin{aligned} I(Z^j; \mathcal{H}^j | \mathcal{H}^i, \theta_{i,j}) + I(Z^i; \mathcal{H}^i | \mathcal{H}^j, \theta_{i,j}) \\ \leq I(\mathbf{X}_k; \mathcal{H}^j | \mathcal{H}^i, \theta_{i,j}) + I(\mathbf{X}_k; \mathcal{H}^i | \mathcal{H}^j, \theta_{i,j}) \\ = H(\mathbf{X}_k | \mathcal{H}^i, \theta_{i,j}) - H(\mathbf{X}_k | \mathcal{H}^j, \mathcal{H}^i, \theta_{i,j}) \\ + H(\mathbf{X}_k | \mathcal{H}^j, \theta_{i,j}) - H(\mathbf{X}_k | \mathcal{H}^i, \mathcal{H}^j, \theta_{i,j}). \quad (41) \end{aligned}$$

For the third MI term in (18) (equivalently in (40)), there are two alternative ways to use the DPI in accordance with $Z_k^j \leftrightarrow \mathbf{X}_k \leftrightarrow Z_k^i | \mathcal{H}^i, \mathcal{H}^j, \theta_{i,j}$ which hold true simultaneously:

$$\begin{aligned} I(Z^j; Z^i | \mathcal{H}^j, \mathcal{H}^i, \theta_{i,j}) \\ \leq I(\mathbf{X}_k; Z^i | \mathcal{H}^j, \mathcal{H}^i, \theta_{i,j}) \\ = H(\mathbf{X}_k | \mathcal{H}^j, \mathcal{H}^i, \theta_{i,j}) - H(\mathbf{X}_k | Z^i, \mathcal{H}^j, \mathcal{H}^i, \theta_{i,j}), \quad (42) \end{aligned}$$

and,

$$\begin{aligned} I(\mathbf{Z}^j; \mathbf{Z}^i | \mathcal{H}^j, \mathcal{H}^i, \boldsymbol{\theta}_{i,j}) \\ \leq I(\mathbf{X}_k; \mathbf{Z}^j | \mathcal{H}^j, \mathcal{H}^i, \boldsymbol{\theta}_{i,j}) \\ = H(\mathbf{X}_k | \mathcal{H}^j, \mathcal{H}^i, \boldsymbol{\theta}_{i,j}) - H(\mathbf{X}_k | \mathbf{Z}^j, \mathcal{H}^j, \mathcal{H}^i, \boldsymbol{\theta}_{i,j}). \end{aligned} \quad (43)$$

Therefore, combining (41),(42) and (43) leads to the upper bound for (18) given by (19). ■

C. Marginalisation of RFS variables

Throughout the article, RFS variables are often marginalised by integrating densities over all possible values. Let X be a RFS with elements in \mathcal{X} . X takes values in the space of all finite sets denoted by $\mathcal{F}(\mathcal{X})$. Consider a RFS probability density $f : \mathcal{F}(\mathcal{X}) \rightarrow [0, \infty)$. The probability that X is contained in a closed subset \mathcal{S} of \mathcal{X} is given by the set integral [25, Chp.11]

$$Pr\{X \subseteq \mathcal{S}\} = \int_{\mathcal{S}} f(X) \delta X,$$

which is defined as

$$\int_{\mathcal{S}} f(X) \delta X = \sum_{m=0}^{\infty} \frac{1}{m!} \int_{\mathcal{S}^m} f(\{x_1, \dots, x_m\}) dx_1 \cdots dx_m. \quad (44)$$

Marginalisation of X involves integrating a probability density over $\mathcal{F}(\mathcal{X})$ which can equivalently be carried out through the set integral defined above:

$$\int_{\mathcal{F}(\mathcal{X})} f(X) \mu(dX) = \int_{\mathcal{X}} f(X) \delta X \quad (45)$$

where μ is an appropriate measure, for example, the unnormalised distribution of a Poisson point process with a uniform rate (further details can be found in Section II.B in [40], and the references therein).

This identity holds for any real valued measurable function $f : \mathcal{F}(\mathcal{X}) \rightarrow \mathbb{R}$ (Appendix B in [40]), thus, along with marginalisation of joint densities, computations regarding information measures and divergences of RFS distributions can be carried out using set integrals as well [46].

REFERENCES

- [1] M. Liggins II, C.-Y. Chong, I. Kadar, M. Alford, V. Vannicola, and S. Thomopoulos, "Distributed fusion architectures and algorithms for target tracking," *Proceedings of the IEEE*, vol. 85, no. 1, pp. 95–107, 1997.
- [2] D. Hall, C.-Y. Chong, J. Llinas, and M. L. II, Eds., *Distributed Data Fusion for Network-Centric Operations*. CRC Press, 2013.
- [3] M. Uney, D. Clark, and S. Julier, "Distributed fusion of PHD filters via exponential mixture densities," *IEEE Journal of Selected Topics in Signal Processing*, vol. 7, no. 3, pp. 521–531, June 2013.
- [4] M. P. Dana, *Registration: A prerequisite for multiple sensor tracking*. Artech House, 1990, ch. 5, pp. 155–185.
- [5] J. Gao and L. Guibas, "Geometric algorithms for sensor networks," *Philosophical Transactions of the Royal Society A*, vol. 370, no. 1958, pp. 27–51, January 2012.
- [6] I. F. Akyildiz, D. Pompili, and T. Melodia, "Underwater acoustic sensor networks: research challenges," *Ad Hoc Networks*, vol. 3, no. 3, pp. 257 – 279, 2005. [Online]. Available: <http://www.sciencedirect.com/science/article/pii/S1570870505000168>
- [7] The Royal Academy of Engineering, "Global navigation space systems: Reliance and vulnerabilities," 2011. [Online]. Available: www.raeng.org.uk/gnss
- [8] M. Erol-Kantarci, H. Mouftah, and S. Oktug, "A survey of architectures and localization techniques for underwater acoustic sensor networks," *IEEE Communications Surveys Tutorials*, vol. 13, no. 3, pp. 487–502, Third Quarter 2011.
- [9] N. Patwari, J. Ash, S. Kyperountas, A. Hero, R. Moses, and N. Correal, "Locating the nodes: cooperative localization in wireless sensor networks," *IEEE Signal Processing Magazine*, vol. 22, no. 4, pp. 54–69, 2005.
- [10] A. T. Ihler, "Inference in sensor networks: Graphical models and particle methods," Ph.D. dissertation, Massachusetts Institute of Technology, 2005.
- [11] R. L. Moses and R. Patterson, "Self-calibration of sensor networks," in *SPIE vol. 4743: Unattended ground sensor tech.s and app.s IV*, 2002, pp. 108–119.
- [12] X. Lin, Y. Bar-Shalom, and T. Kirubarajan, "Multisensor multitarget bias estimation for general asynchronous sensors," *IEEE Transactions on Aerospace and Electronic Systems*, vol. 41, no. 3, pp. 899–921, July 2005.
- [13] J. Vermaak, S. Maskell, and M. Briers, "Online sensor registration," in *IEEE Aerospace Conference 2005*, 2005, pp. 2117–2125.
- [14] J. Li, W. Ng, and S. Godsill, "Online multiple target tracking and sensor registration using sequential monte carlo methods," in *IEEE Aerospace Conference 2007*, 2007, pp. 1–9.
- [15] N. W. Singh, Sumeetpal S. and S. J. Godsill, "Approximate likelihood estimation of static parameters in multi-target models," in *Bayesian Time Series Models*, 1st ed., S. C. David Barber, Ali Taylan Cemgil, Ed. Cambridge University Press, 2011. [Online]. Available: <http://dx.doi.org/10.1017/CBO9780511984679.012>
- [16] B. Ristic, D. Clark, and N. Gordon, "Calibration of multi-target tracking algorithms using non-cooperative targets," *IEEE Journal of Selected Topics in Signal Processing*, vol. 7, no. 3, pp. 390–398, 2013.
- [17] O. Cappé, S. J. Godsill, and E. Moulines, "An overview of existing methods and recent advances in sequential Monte Carlo," *Proceedings of the IEEE*, vol. 95, pp. 899–924, 2007.
- [18] N. Kantas, S. Singh, and A. Doucet, "Distributed maximum likelihood for simultaneous self-localization and tracking in sensor networks," *IEEE Transactions on Signal Processing*, vol. 60, no. 10, pp. 5038–5047, 2012.
- [19] S. Funiak, C. Guestrin, M. Paskin, and R. Sukthankar, "Distributed localization of networked cameras," in *Proceedings of The Fifth International Conference on Information Processing in Sensor Networks (IPSN)*, 2006, pp. 34–42.
- [20] M. Cetin, L. Chen, J. Fisher, A. Ihler, R. Moses, M. Wainwright, and A. Willksy, "Distributed fusion in sensor networks: A graphical models perspective," *IEEE Signal Processing Magazine*, vol. 23, no. 4, pp. 42–55, 2006.
- [21] M. Uney and M. Cetin, "Graphical model-based approaches to target tracking in sensor networks: An overview of some recent work and challenges," in *Proc. of The Fifth International Symposium on Image and Signal Processing and Analysis (ISPA)*, 2007, pp. 492–497.
- [22] A. Ihler, J. Fisher, R. Moses, and A. Willksy, "Nonparametric belief propagation for self-localization of sensor networks," *IEEE Journal on Selected Areas in Communications*, vol. 23, no. 4, pp. 809–819, 2005.
- [23] M. J. Wainwright and M. I. Jordan, "Graphical models, exponential families, and variational inference," *Found. Trends Mach. Learn.*, vol. 1, no. 1-2, pp. 1–305, Jan. 2008. [Online]. Available: <http://dx.doi.org/10.1561/2200000001>
- [24] J. S. Yedidia, W. T. Freeman, and Y. Weiss, "Understanding belief propagation and its generalizations," Mitsubishi Electric Research Laboratories, San Francisco, CA, USA, Tech. Rep. TR-2001-22, Jan. 2002.
- [25] R. P. S. Mahler, *Statistical Multisource Multitarget Information Fusion*. Springer, 2007.
- [26] L. Chen, M. J. Wainwright, M. Cetin, and A. S. Willksy, "Data association based on optimization in graphical models with application to sensor networks," *Mathematical and Computer Modelling*, vol. 43, no. 9–10, pp. 1114 – 1135, 2006, optimization and Control for Military Applications Optimization and Control for Military Applications. [Online]. Available: <http://www.sciencedirect.com/science/article/pii/S0895717705005406>
- [27] R. Mahler, "The multisensor PHD filter: I. general solution via multitarget calculus," in *Proceedings of the SPIE*, vol. 7336, 2009, pp. 73 360E–73 360E–12. [Online]. Available: <http://dx.doi.org/10.1117/12.818024>
- [28] N. Kantas, A. Doucet, S. S. Singh, and J. Maciejowski, "An overview of sequential Monte Carlo methods for parameter estimation on general state space models," in *Proceedings of the 15th IFAC Symposium on System Identification*, 2009, pp. 774–785.

- [29] R. P. S. Mahler, "Multitarget Bayes Filtering via First-Order Multitarget Moments," *IEEE Transactions on Aerospace and Electronic Systems*, vol. 39, no. 4, pp. 1152–1178, Oct. 2003.
- [30] R. Mahler, "PHD filters of higher order in target number," *IEEE Transactions on Aerospace and Electronic Systems*, vol. 43, no. 4, pp. 1523–1543, October 2007.
- [31] B.-T. Vo and B.-N. Vo, "Labeled random finite sets and multi-object conjugate priors," *IEEE Transactions on Signal Processing*, vol. 61, no. 13, pp. 3460–3475, July 2013.
- [32] J. Williams, "An efficient, variational approximation of the best fitting multi-Bernoulli filter," *IEEE Transactions on Signal Processing*, vol. 63, no. 1, pp. 258–273, Jan 2015.
- [33] J. L. Williams, "Marginal multi-Bernoulli filters: RFS derivation of MHT, JIPDA and association-based MeMBer," *IEEE Transactions on Aerospace and Electronic Systems*, accepted for publication. [Online]. Available: <http://arxiv.org/abs/1203.2995v2>
- [34] T. M. Cover and J. A. Thomas, *Elements of Information Theory*. John Wiley & Sons, 1991.
- [35] M. Uney, B. Mulgrew, and D. Clark, "Target aided online sensor localisation in bearing only clusters," in *Sensor Signal Processing for Defence (SSPD) 2014*, Sept 2014, pp. 1–5.
- [36] S. S. Singh, N. Whiteley, and S. J. Godsill, "Approximate likelihood estimation of static parameters in multi-target models," in *Bayesian Time Series Models*, D. Barber, A. T. Cemgil, and S. Chiappa, Eds. Cambridge University Press, 2011, pp. 225–244, Cambridge Books Online. [Online]. Available: <http://dx.doi.org/10.1017/CBO9780511984679.012>
- [37] A. Doucet, N. de Freitas, and N. Gordon, Eds., *Sequential Monte Carlo Methods in Practice*. Springer, 2001.
- [38] B. Ristic, *Particle Filters for Random Set Models*. Springer, 2013.
- [39] C. P. Robert and G. Casella, *Monte Carlo Statistical Methods*, 2nd ed. Springer, 2004.
- [40] B. Vo, S. Singh, and A. Doucet, "Sequential Monte Carlo methods for Multi-target Filtering with Random Finite Sets," *IEEE Transactions on Aerospace and Electronic Systems*, vol. 41, No.4, pp. 1224–1245, 2005.
- [41] E. B. Sudderth, A. T. Ihler, M. Isard, W. T. Freeman, and A. S. Willsky, "Nonparametric belief propagation," *Communications of the ACM*, vol. 53, no. 10, pp. 95–103, Oct. 2010. [Online]. Available: <http://doi.acm.org/10.1145/1831407.1831431>
- [42] B. Silverman, *Density Estimation for Statistics and Data Analysis*. Chapman and Hall, 1986.
- [43] A. F. M. Smith and A. E. Gelfand, "Bayes statistics without tears: A sampling-resampling perspective," *The American Statistician*, vol. 46, no. 2, pp. 84–88, May 1992.
- [44] M. Uney, B. Mulgrew, and D. Clark, "Cooperative sensor localisation in distributed fusion networks by exploiting non-cooperative targets," in *IEEE Workshop on Statistical Signal Processing (SSP) 2014*, June 2014, pp. 516–519.
- [45] B. Ristic, D. Clark, B.-N. Vo, and B.-T. Vo, "Adaptive target birth intensity for PHD and CPHD filters," *IEEE Transactions on Aerospace and Electronic Systems*, vol. 48, no. 2, pp. 1656–1668, 2012.
- [46] M. Uney, D. Clark, and S. Julier, "Information measures in distributed multitarget tracking," in *Proceedings of the 14th International Conference on Information Fusion (FUSION)*, July 2011, pp. 1–8.



Murat Üney (S'97-M'10) is a Research Fellow in the School of Engineering, the University of Edinburgh, Edinburgh. Between 2010 and 2013 he was with Heriot-Watt University, Edinburgh, and prior to that he was a member of the Signal Processing and Information Systems (SPIS) Laboratory, Sabancı University, İstanbul. He received his Ph.D. degree in signal processing and control from the Middle East Technical University (ODTÜ), Ankara, Turkey, in 2009. He has industrial research and development experience in aerospace/defence and telecommunication sectors. His research interests are in the broad scope of statistical signal and information processing with a particular emphasis on distributed, multi-modal and resource constrained problem settings, and sensor fusion applications.



Professor Bernard (Bernie) Mulgrew (FIEEE, FREng, FRSE, FIET) received his B.Sc. degree in 1979 from Queen's University Belfast. After graduation, he worked for 4 years as a Development Engineer in the Radar Systems Department at Ferranti, Edinburgh. From 1983–1986 he was a research associate in the Department of Electrical Engineering at the University of Edinburgh. He was appointed to lectureship in 1986, received his Ph.D. in 1987, promoted to senior lecturer in 1994 and became a reader in 1996. The University of Edinburgh appointed him to a Personal Chair in October 1999 (Professor of Signals and Systems). His research interests are in adaptive signal processing and estimation theory and in their application to radar and sensor systems. Prof. Mulgrew is a co-author of three books on signal processing.



Daniel E. Clark is an Associate Professor in the School of Engineering and Physical Sciences at Heriot-Watt University. His research interests are in the development of the theory and applications of multi-object estimation algorithms for sensor fusion problems. He has led a range of projects spanning theoretical algorithm development to practical deployment. He was awarded his PhD in 2006 from Heriot-Watt University.