

GOSPA-Driven Non-Myopic Multi-Sensor Management with Multi-Bernoulli Filtering

George Jones and Ángel F. García-Fernández

Abstract—In this paper, we propose a non-myopic sensor management algorithm for multi-target tracking, with multiple sensors operating in the same surveillance area. The algorithm is based on multi-Bernoulli filtering and selects the actions that solve a non-myopic minimisation problem, where the cost function is the mean square generalised optimal sub-pattern assignment (GOSPA) error, over a future time window. For tractability, the sensor management algorithm actually uses an upper bound of the GOSPA error and is implemented via Monte Carlo Tree Search (MCTS). The sensors have the ability to jointly optimise and select their actions with the considerations of all other sensors in the surveillance area. The benefits of the proposed algorithm are analysed via simulations.

Index Terms—Non-myopic, multi-sensor management, Monte Carlo tree search, multi-Bernoulli filtering.

I. INTRODUCTION

SENSOR management can be defined as the dynamic re-tasking of agile sensors to achieve an operational objective [1]. Sensors can be agile in a multitude of ways, from physically repositioning, changing direction or selecting a sensing mode.

Myopic sensor management, sometimes called greedy sensor management, optimises the sensor resources for the immediate benefit of the system, not considering the long term effects of the actions being selected now. Non-myopic sensor management operates on the policy of considering these long-term effects of the actions selected now. Whilst it has an increased computational demand, non-myopic planning often produces more desirable results [2], [3].

Sensor management is applicable to many domains, from space surveillance in [4], environmental field estimation in [5] and the general control of multiple sensors in [6]. This paper deals with multi-sensor management for multiple-target tracking (MTT). In this context the objective is to track the targets as accurately as possible whilst minimising the number of missed and false targets. One approach to tackle this sensor management problem is via reinforcement learning where a policy can be learnt off-line via simulation [4], [7], [8]. This approach has the benefit of not requiring a complete model of target dynamics and measurements at the expense of lower interpretability and a more complex training process.

In this work, we pose the MTT multi-sensor management problem using a partially observed Markov decision process (POMDP) [9], in which the hidden state is a set of targets. Solving the POMDP requires a Bayesian solution to the MTT problem, which is based on target birth/death models, target dynamic model and measurement model. In addition, a POMDP requires a cost function to minimise, or a reward function to maximise.

Sensor management for MTT can be approached in a variety of ways, including reward functions based on the use of information theory. These methods derive information theoretic approaches, utilising divergence measures such as the Rényi divergence [10]–[12], the Kullback-Leibler divergence (KLD) [13]–[15] and the Cauchy-Schwarz divergence [16], [17]. Their objective can broadly be understood to be to take the action that provides the largest gain in information about the variables of interest. Whilst these approaches often produce desirable behaviours, they are not explicitly transparent w.r.t. what the resulting policy is optimising for in terms of the real world.

Another popular cost function is the posterior Cramér-Rao lower bound (PCRLB) [18]–[22]. The PCRLB is a bound on the mean square error and is used instead of the mean square error because of its good performance and computational efficiency. However, to use the PCRLB as a cost function, external criteria must be added to be able to perform tracking of an time varying and unknown number of targets. In [23] [24], the cost function is the weighted sum of two terms: the tracking cost (the sum of the covariance matrix of each track) and the search cost (the expected number of targets that remain undetected).

In this paper, we utilise the generalised optimal sub pattern assignment (GOSPA) metric [25], or specifically an upper bound on the mean square GOSPA (MSGOSPA) error as a cost function to inform sensor actions and drive the sensor management [26], [2]. By using the GOSPA metric, we are selecting actions which specifically minimise the localisation, missed target and false target errors - the three quantities of interest in a multi-target estimation scenario. Therefore, the GOSPA cost function is a natural choice for planning sensor actions in search and track operations. A comparison between the GOSPA metric and other multi-target metrics for sensor management is given in [27]. An important benefit of GOSPA w.r.t. the optimal sub-pattern assignment (OSPA) metric [28] and the unnormalised OSPA metric, which is proportional to the Cardinalized Optimal Linear Assignment metric [29], is that the cost is additive for independent, far-away targets. This property is key to avoiding the spooky effect at a distance in

G. Jones is with Dept. of Autonomy, Frost Unmanned AB, Gothenburg 444 32, Sweden. A. F. García-Fernández is with the IPTC, ETSI de Telecomunicación, Universidad Politécnica de Madrid (emails: george.jones@frostunmanned.com, angel.garcia.fernandez@upm.es). This work was supported by the EPSRC Centre for Doctoral Training in Distributed Algorithms EP/S023445/1 at the University of Liverpool and Roke Manor Research Limited. This work was partially supported by the Spanish Ministry of Science, Innovation and Universities under the project PID2024-158149OB-C21.

optimal estimation and simplifies the planning stage [27].

The predecessors to this paper cover the utilisation of the GOSPA metric to manage a single sensor using a (Gaussian) Bernoulli filter for myopic and non-myopic sensor management [26] and then a multi-Bernoulli filter for myopic sensor management [30]. This paper has two main contributions:

- A non-myopic, multi-sensor management algorithm for MTT driven by the GOSPA metric based on Gaussian multi-Bernoulli filtering extending the work in [30] to handle the non-myopic case and multiple sensors.
- An efficient implementation of the algorithm using Monte Carlo Tree Search (MCTS) [31]. For the efficient MCTS implementation, we use hypothesis reduction by merging the detection and misdetection hypotheses for each target such that the edges of the tree only consider sensor actions, and not measurements, as done in [32].

Simulation results show the benefits of the proposed sensor management algorithm.

The remainder of this paper is organised as follows. In Section II, we provide the background and the objective of this paper. In Section III, we introduce the multi-sensor management algorithm for multi-Bernoulli filtering. In Section IV, we explain the MCTS implementation. In Section V, we analyse the simulation results of the sensor management algorithm. In Section VI, we provide the concluding remarks.

II. BACKGROUND & OBJECTIVE

This section presents the background and the objective of this paper. In Section II-A, we present the dynamic and measurement model. In Section II-B, we present the multi-Bernoulli filter, including its prediction and update step. In Section II-C, we present the GOSPA metric and, in Section II-D, we formulate the problem of non-myopic planning, using the time discounted predicted mean square GOSPA error. The main notation in the paper is provided in Table I.

A. Dynamic & Measurement Model

A single target state is denoted as $x \in \mathbb{R}^{n_x}$. The set of targets at time step k is X_k such that $X_k \in \mathcal{F}(\mathbb{R}^{n_x})$ where $\mathcal{F}(\mathbb{R}^{n_x})$ is the set of all finite subsets of $\mathbb{R}^{n_x}$ [33]. In this paper, we consider that there are S sensors. At time step k , there are a discrete number of actions $a_k = (a_k^1, \dots, a_k^S) \in \mathbb{A}_k$ where a_k^s represents the action associated with the s -th sensor. For sensor s , a target with state x in the surveillance area can either be detected with probability of detection $p_{a_k^s}^{D,s}(x)$ and one measurement is generated with density $l^s(z|x) = \mathcal{N}(z; H_{a_k^s}^s x + b_{a_k^s}^s, R_{a_k^s}^s)$, or missed (not detected) with probability $1 - p_{a_k^s}^{D,s}(x)$. At time step k , the set of measurements from sensor s , which is denoted by Z_k^s , can either be empty $Z_k^s = \emptyset$ or contain multiple measurements $|Z_k^s| \geq 1$, where measurements are the union of the target-generated detections and Poisson point process (PPP) clutter, defined by intensity $\lambda_{a_k^s}^C(\cdot)$. The sequence of sets of measurements from each sensor received at time step k is denoted as $Z_k = (Z_k^1, \dots, Z_k^S)$.

TABLE I
MAIN NOTATION

• Filter
– $f_{k k}^s(\cdot)$: MBM density of the set of targets at time step k after the s -th sensor measurement update at time step k .
– $f_{k k-1}^{m,s}(\cdot)$: predicted measurement density for the s -th sensor.
– $f_{k k}^i(\cdot; a_k)$: i -th Bernoulli density.
– $r_{k' k}^i, \bar{x}_{k' k}^i, P_{k' k}^i$: existence probability, mean and covariance at time step $k' \in \{k, k+1\}$ given the measurements up to time step k for the i -th Bernoulli.
• Non-myopic planning
– $a_{k:k'}$: sequence of actions from time step k to k' .
– $h_{k:k'}^i$: sequence of detections/misdictions for all sensors from time step k to k' for the i -th Bernoulli (Section III.F).
– $r_{k' k', a_{k:k'}, h_{k:k'}}^{i, h_{k:k'}^i}, \bar{x}_{k' k', a_{k:k'}, h_{k:k'}}^{i, h_{k:k'}^i}, P_{k' k', a_{k:k'}, h_{k:k'}}^{i, h_{k:k'}^i}$: existence probability, mean and covariance at time step k' given $h_{k:k'}^i$ and $a_{k:k'}$ for the i -th Bernoulli.
– $p(h_{k'}^i a_{k:k'-1}, h_{k:k'-1}^i)$: probability of the detection/misdictions $h_{k'}^i$ (for all sensors) at time step k' given previous $a_{k:k'-1}$ and $h_{k:k'-1}^i$.
– $C^i(a_{k:k'}, h_{k:k'}^i)$: cost for the i -th Bernoulli at time step k' given $a_{k:k'}$ and $h_{k:k'}^i$.
• Non-myopic planning with tree reduction
– $r_{k' k', a_{k:k'}, h_{k:k'}}^{i, h_{k:k'}^i}, \bar{x}_{k' k', a_{k:k'}, h_{k:k'}}^{i, h_{k:k'}^i}, P_{k' k', a_{k:k'}, h_{k:k'}}^{i, h_{k:k'}^i}$: merged existence probability, mean and covariance at time step k' given $a_{k:k'}$ for the i -th Bernoulli.
– $r_{k' k', a_{k:k'}, h_{k:k'}}^{i, h_{k:k'}^i}, \bar{x}_{k' k', a_{k:k'}, h_{k:k'}}^{i, h_{k:k'}^i}, P_{k' k', a_{k:k'}, h_{k:k'}}^{i, h_{k:k'}^i}$: pre-merged existence probability, mean and covariance at time step k' given $a_{k:k'}$ and $h_{k:k'}^i$ for the i -th Bernoulli.
– $p(h_{k'}^i a_{k:k'-1})$: probability of the detections/misdictions $h_{k'}^i$ at time step k' given previous actions $a_{k:k'-1}$.
– $C^i(a_{k:k'}, h_{k:k'}^i)$: cost for the i -th Bernoulli at time step k' given $a_{k:k'}$ and $h_{k:k'}^i$.
– $C_{k'}(a_{k:k'})$: cost at time step k' .

The dynamic model of the targets is as follows. The probability of a target with state x surviving and transitioning to the next time step is $p^S(x)$ with the single-target transition density being $g(\cdot|x)$, or the target can die with probability $1 - p^S(x)$. The multi-target state X_{k+1} is given by the union of the new and surviving targets.

We consider a multi-Bernoulli (MB) birth model which contains n_k^b Bernoulli components. The i -th Bernoulli component has an associated probability of birth $r_k^{b,i}$ and single-target density $p_k^{b,i}(\cdot)$. The density is given by [33], [34]

$$f_k^b(X_k) = \sum_{X^1 \uplus \dots \uplus X^{n_k^b} = X_k} \prod_{i=1}^{n_k^b} f_k^{b,i}(X^i) \quad (1)$$

where the i -th Bernoulli component density is

$$f_k^{b,i}(X_k) = \begin{cases} 1 - r_k^{b,i} & X_k = \emptyset \\ r_k^{b,i} p_k^{b,i}(x) & X_k = \{x\} \\ 0 & \text{otherwise.} \end{cases} \quad (2)$$

The summation in (1) is taken over all mutually disjoint and (potentially) empty sets $X^1, \dots, X^{n_k^b}$ whose union is X_k .

B. Multi-Bernoulli Filtering

For the previous dynamics and measurement models, the posterior density is a multi-Bernoulli mixture (MBM), which can be computed in closed-form via the MBM filtering recursion [34], [35] performing S sequential updates (one per sensor) at each update step.

In this paper for simplicity, we consider a multi-Bernoulli filter. In MB filtering, both the predicted and posterior densities at time step $k' \in \{k, k+1\}$, given the measurements up to time step k , are approximated as an MB of the form

$$f_{k'|k}(X_{k'}) = \sum_{X^1 \uplus \dots \uplus X^{n_{k'|k}} = X_{k'}} \prod_{i=1}^{n_{k'|k}} f_{k'|k}^i(X^i) \quad (3)$$

where $n_{k'|k}$ is the number of Bernoulli components.

For the i -th Bernoulli $f_{k'|k}^i(\cdot)$, the two quantities that must be propagated over time via the prediction and update steps are the spatial density $p_{k'|k}^i(\cdot)$ and the probability of existence $r_{k'|k}^i$.

1) *MB Filter Prediction Step:* The predicted density has $n_{k+1|k} = n_{k|k} + n_{k+1}^b$ Bernoulli components, including the previous and new Bernoulli components. The prediction equations for Bernoulli components relating to surviving targets are [36]

$$r_{k+1|k}^i = r_{k|k}^i \langle p^S, p_{k|k}^i \rangle \quad (4)$$

$$p_{k+1|k}^i(x) = \frac{\int g(x|y) p^S(y) p_{k|k}^i(y) dy}{\langle p_{k|k}^i, p^S \rangle} \quad (5)$$

where the notation $\langle h, g \rangle$ denotes the inner product between two functions $h(\cdot)$ and $g(\cdot)$ [33].

For Bernoulli components representing new born targets, $i \in \{n_{k|k} + 1, \dots, n_{k|k} + n_{k+1}^b\}$, the parameters are those of the birth model (1), which results in

$$r_{k+1|k}^i = r_{k+1}^{b, i-n_{k|k}} \quad (6)$$

$$p_{k+1|k}^i(x) = p_{k+1}^{b, i-n_{k|k}}(x). \quad (7)$$

2) *MB Filter Update Step:* To obtain a multi-sensor MB filter we can apply S MB sequential updates, one for each sensor. After each update, the density is of MBM form [35],

$$f_{k|k}^s(X_k) = \sum_{e \in \mathbb{E}_{k|k}} w_{k|k}^{s,e} \sum_{\uplus_{l=1}^{n_{k|k}} X^l = X_k} \prod_{i=1}^{n_{k|k}} f_{k|k}^{i,e^i}(X^i) \quad (8)$$

where $\mathbb{E}_{k|k}$ is the set that represents all global data association hypotheses and $e = (e^1, \dots, e^{n_{k|k}})$, e^i is the local hypothesis for Bernoulli component i . It should be mentioned that the standard notation for global hypotheses in Poisson multi-Bernoulli mixture filtering, see [37] [35] for details, is a and has not been used here as a is used to denote sensor actions. We proceed to explain the update of the local hypothesis.

Given the measurement set of the s -th sensor $Z_k^s = \{z_k^{s,1}, \dots, z_k^{s,m_k^s}\}$, the s -th update equations for a misdetection hypothesis for the i -th Bernoulli component are

$$w_{k|k}^{i,s,1} = 1 - r_{k|k}^{i,s-1} \langle p_{k|k}^{i,s-1}, p_{a_k}^{D,s} \rangle \quad (9)$$

$$r_{k|k}^{i,s,1} = \frac{r_{k|k}^{i,s-1} \langle p_{k|k}^{i,s-1}, 1 - p_{a_k}^{D,s} \rangle}{1 - r_{k|k}^{i,s-1} \langle p_{k|k}^{i,s-1}, p_{a_k}^{D,s} \rangle} \quad (10)$$

$$p_{k|k}^{i,s,1}(x) = \frac{(1 - p_{a_k}^{D,s}(x)) p_{k|k}^{i,s-1}(x)}{\langle p_{k|k}^{i,s-1}, 1 - p_{a_k}^{D,s} \rangle} \quad (11)$$

where the superindex (i, s, j) denotes the i -th Bernoulli component, the s -th sensor and the j -th local hypothesis, with $j = 1$ corresponding to a misdetection.

The equations for the update of the detection hypothesis for Bernoulli component i and measurement $z_k^{s,j}$ are given by

$$w_{k|k}^{i,s,1+j} = \frac{r_{k|k}^{i,s-1} \langle p_{k|k}^{i,s-1}, l^s(z_k^{s,j} | \cdot) p_{a_k}^{D,s} \rangle}{\lambda_{a_k}^C(z_k^{s,j})} \quad (12)$$

$$r_{k|k}^{i,s,1+j} = 1 \quad (13)$$

$$p_{k|k}^{i,s,1+j}(x) = \frac{l^s(z_k^{s,j} | x) p_{a_k}^{D,s}(x) p_{k|k}^{i,s-1}(x)}{\langle p_{k|k}^{i,s-1}, l^s(z_k^{s,j} | \cdot) p_{a_k}^{D,s} \rangle}. \quad (14)$$

These update equations give rise to the updated density after the s -th sensor update, which is of MBM form (8).

After each update, the MBM is projected to an MB, via Kullback-Leibler divergence (KLD) minimisation with auxiliary variables [36], [38]. After the s -th update, this MB density is characterised by

$$r_{k|k}^{i,s} = \sum_{e^i=1}^{h^i} w_{k|k}^{i,s,e^i} r_{k|k}^{i,s,e^i}, \quad (15)$$

$$p_{k|k}^{i,s}(x) = \frac{\sum_{e^i=1}^{h^i} w_{k|k}^{i,s,e^i} r_{k|k}^{i,s,e^i} p_{k|k}^{i,s,e^i}(x)}{\sum_{e^i=1}^{h^i} w_{k|k}^{i,s,e^i} r_{k|k}^{i,s,e^i}}, \quad (16)$$

$$w_{k|k}^{i,e^i} = \sum_{b \in \mathbb{E}_{k|k}: b^i = e^i} w_{k|k}^b \quad (17)$$

where $h^i = 1 + m_k^s$ is the number of local hypotheses of the i -th Bernoulli [35]. We also consider that $r_{k|k}^{i,0} = r_{k|k-1}^i$ and $p_{k|k}^{i,0}(\cdot) = p_{k|k-1}^i(\cdot)$.

There are many methods that can approximately perform this MB projection in a computationally efficient manner, for instance Murty's algorithm [39], loopy belief propagation [40], [41], [38] efficient hypothesis management [42] and sampling based approaches [43].

C. The GOSPA metric

The GOSPA metric [25] is a metric for sets of targets that penalises the localisation error, the missed detection error and the false detection error. Let $X = \{x_1, \dots, x_{|X|}\}$ be the ground truth set of targets and $Y = \{y_1, \dots, y_{|Y|}\}$ be the estimated set. Let γ be an assignment set from X to Y . The assignment set γ has these properties $\gamma \subseteq \{1, \dots, |X|\} \times \{1, \dots, |Y|\}$, $(i, j), (i, j') \in \gamma \implies j = j'$ and $(i, j), (i', j) \in \gamma \implies i = i'$. These properties imply that each target in X and Y can be assigned at most once. Then, given a single-target metric $d(\cdot, \cdot)$ parameter $p > 0$ and $\alpha = 2$, the GOSPA metric between set X and set Y is

$$d(X, Y) = \min_{\gamma \in \Gamma} \left(\sum_{(i,j) \in \gamma} d^p(x_i, y_j) + \frac{c^p}{2} (|X| + |Y| - 2|\gamma|) \right)^{1/p}. \quad (18)$$

In the remainder of the paper, we use $p = 2$.

D. Non-myopic Planning with Time-discounted MSGOSPA Error

In this paper, the objective is to plan the sensor actions by minimising the MSGOSPA error in a window of T time steps from time step k to time step $K = k + T - 1$. Sensor management consists of determining a policy $\mu_{k'}(\cdot)$ at time step k' that maps the available information $\mathcal{I}_{k'}$ at time step k' to an action at time step k' . The available information at time step k' is

$$\mathcal{I}_{k'} = \begin{cases} (f_{k|k-1}(\cdot)) & k' = k \\ (f_{k'|k'-1}(\cdot), a_{k:k'-1}, Z_{k:k'-1}) & k' \geq k \end{cases} \quad (19)$$

such that $a_{k'} = \mu_{k'}(\mathcal{I}_{k'})$.

The sequence of policies (which are the deterministic functions that map information to actions) up to the planning horizon is denoted by $\mu_{k:K}(\cdot) = (\mu_k(\cdot), \dots, \mu_K(\cdot))$. Including a decay factor (discount rate) of $\lambda \in (0, 1]$, the policy is chosen to minimise [9, Eq. (7.6)]

$$J_{\mu_{k:K}}(\cdot) = E_{\mu_{k:K}}(\cdot) \left[\sum_{k'=k}^K \lambda^{k'-k} d^2 \left(X_{k'}, \hat{X}_{k'}(a_{k:k'}, Z_{k:k'}) \right) \right] \quad (20)$$

where the expectation is taken with respect to the joint probability density of $(X_k, \dots, X_K, Z_k, \dots, Z_K)$ under the policy $\mu_{k:K}(\cdot)$, and $\hat{X}_{k'}(a_{k:k'}, Z_{k:k'})$ is the optimal MSGOSPA estimator at time step k' .

The optimisation over the policy in (20) can be written using a value function that is propagated backwards using a Bellman-type equation [9, Chapter 7], see (12)-(13) in [26]. The objective of this paper is to develop a computationally efficient approximation to solve (20) based on MB filtering.

III. MULTI-SENSOR MANAGEMENT

This section describes an algorithm for managing multiple sensors using multi-Bernoulli filtering, using the Gaussian distribution and the GOSPA metric.

As described in Section II-D, the objective is to perform planning by minimising a time-discounted MSGOSPA error, see (20). Since calculating the MSGOSPA error is intractable, we instead use an upper bound that we proceed to calculate. To do so, in Section III-A, we state the assumptions used in the sensor management algorithm. We compute the density of the measurements in Section III-B. We then derive an upper bound to the overall MSGOSPA error for a given MB posterior in Section III-C. This result is then used to derive an upper bound on the MSGOSPA error in the myopic case in Section III-D. In Section III-E, we explain the multi-sensor update that is required for sensor management and finally, in Section III-F, we provide the resulting non-myopic planning algorithm.

A. Assumptions of the Sensor Management Algorithm

This section presents the assumptions of the sensor management algorithm. In the MB filter, we consider a Gaussian implementation in which each single-target density is Gaussian with density

$$p_{k'|k}^i(x_{k'}) = \mathcal{N}(x_{k'}; \bar{x}_{k'|k}, P_{k'|k}) \quad (21)$$

where $\bar{x}_{k'|k}$ and $P_{k'|k}$ denote the mean and covariance matrix. The sensor management algorithm also makes the following assumptions:

- 1) We either detect zero measurements $Z_k^s = \emptyset$ or one measurement $Z_k^s = \{z_k\}$ at the predicted measurement for each Bernoulli component, without clutter measurements.
- 2) The probability of detection is approximated as a constant given by its predicted value for each Bernoulli component.
- 3) We use a computationally efficient upper bound for the resulting MSGOSPA error (see Section III-C).
- 4) The sensors are operating in a centralised manner, meaning they are operating with the same prediction information at each time step.
- 5) All Bernoulli components are far away from each other.
- 6) The measurements of the S sensors at a given time step are independent.

It should be noted that these assumptions are only made by the sensor management algorithm but not by the MB filter that is run once we have taken the actions and received the measurements. It is also relevant to notice that in [23], the sensor management algorithm also assumes that there is no clutter, but always considers a detection for each target. Considering misdetections as in Assumption 1) is more general and can become important for low probabilities of detection. In addition, while Assumptions 1)-5) can be strong approximations in some situations, they enable the development of a closed-form upper bound of the MSGOSPA error for sensor management with MB filters that is computationally efficient to implement.

B. Measurement Density

The predicted measurement density for an MB predicted density is Poisson multi-Bernoulli [44]. This is due to the fact that the clutter measurements follow a PPP and each potential target generates a potential measurement that follows a Bernoulli distribution. The expected probability of detection for the s -th sensor and i -th Bernoulli component with mean $\bar{x}_{k|k-1}^{i,s}$ and covariance $P_{k|k-1}^{i,s}$ is given by

$$\bar{p}_{a_k}^{D,s,i} = \int p_{a_k}^{D,s}(x) \mathcal{N}(x; \bar{x}_{k|k-1}^i, P_{k|k-1}^i) dx. \quad (22)$$

For an MB predicted density, the density of the predicted sequence of measurements at time step k under Assumption 6) is

$$f_{k|k-1}^m(Z_k) \approx \prod_{s=1}^S f_{k|k-1}^{m,s}(Z_k^s) \quad (23)$$

where

$$f_{k|k-1}^{m,s}(Z_k^s) = \sum_{\substack{\cup_{l=0}^{n_{k|k-1}-1} Z_{s,l} = Z_k^s}} f_{k|k-1}^{m,p,s}(Z^{s,0}) \prod_{i=1}^{n_{k|k-1}} f_{k|k-1}^{m,i,s}(Z^{s,i}) \quad (24)$$

is a Poisson multi-Bernoulli density as $f_{k|k-1}^{m,p,s}(Z^{s,0})$ is a PPP density with intensity $\lambda_{a_k}^C(\cdot)$ and under Assumption 2) and using (22), $f_{k|k-1}^{m,i,s}(Z^{s,i})$ is a Bernoulli density given by

$$f_{k|k-1}^{m,i,s}(Z_k^{s,i}; a_k) = \begin{cases} \bar{p}_{a_k}^{D,s,i} r_{k|k-1}^i \mathcal{N}(z_k^s, \hat{z}_{a_k}^{s,i}, S_{a_k}^{s,i}) & Z_k^{s,i} = \{z_k^s\} \\ 1 - r_{k|k-1}^i \bar{p}_{a_k}^{D,s,i} & Z_k^{s,i} = \emptyset \end{cases} \quad (25)$$

where $\hat{z}_{a_k}^{s,i}$ and $S_{a_k}^{s,i}$ are the Kalman filter predicted measurement and its covariance matrix (for the s -th sensor and the i -th Bernoulli) [45].

C. MSGOSPA Error for a Given MB Posterior

Applying Lemma 2 in [37], the MSGOSPA error at time step k for a given MB posterior $f_{k|k}(\cdot)$ is

$$\begin{aligned} & \int d^2(X_k, \hat{X}_k(a_k, Z_k)) f_{k|k}(X_k|Z_k; a_k) \delta X_k \\ &= \int d^2\left(\cup_{i=1}^{n_{k|k}} X^i, \cup_{i=1}^{n_{k|k}} \hat{X}^i(a_k, Z_k^i)\right) \\ & \quad \prod_{i=1}^{n_{k|k}} f_{k|k}^i(X^i; a_k) \delta X^{1:n_{k|k}} \end{aligned} \quad (26)$$

where $\hat{X}^i$ is the set estimate (which may be empty or contain a single state) of the i -th Bernoulli and $X^{1:n_{k|k}} = (X^1, \dots, X^{n_{k|k}})$. Under Assumption 5) and the property that the GOSPA metric is additive for far away targets [27], we obtain that the MSGOSPA error is

$$\begin{aligned} & \int d^2(X_k, \hat{X}_k(a_k, Z_k)) f_{k|k}(X_k|Z_k; a_k) \delta X_k \\ & \simeq \sum_{i=1}^{n_{k|k}} \int d^2(X^i, \hat{X}^i(a_k, Z_k^i)) f_{k|k}^i(X^i; a_k) \delta X^i. \end{aligned} \quad (27)$$

We consider that a target is detected if its probability of existence is higher than the detection threshold Γ_d . Then under Assumptions 1) and 5), we apply the upper bound in [26, Lem. 1] for Gaussian Bernoulli components such that

$$\begin{aligned} & \int d^2(X_k, \hat{X}_k(a_k, Z_k)) f_{k|k}(X_k|Z_k; a_k) \delta X_k \\ & \leq \sum_{i=1}^{n_{k|k}} C\left(\Gamma_d, r_{k|k,a_k}^{i,|Z_k^i|}, P_{k|k,a_k}^{i,|Z_k^i|}\right) \end{aligned} \quad (28)$$

and

$$C\left(\Gamma_d, r_{k|k,a_k}^{i,|Z_k^i|}, P_{k|k,a_k}^{i,|Z_k^i|}\right) = \begin{cases} \frac{c^2}{2} r_{k|k,a_k}^{i,|Z_k^i|} & r_{k|k,a_k}^{i,|Z_k^i|} \leq \Gamma_d \\ \frac{c^2}{2} (1 - r_{k|k,a_k}^{i,|Z_k^i|}) \\ + r_{k|k,a_k}^{i,|Z_k^i|} \min\left(\text{tr}(P_{k|k,a_k}^{i,|Z_k^i|}), c^2\right) & r_{k|k,a_k}^{i,|Z_k^i|} > \Gamma_d \end{cases} \quad (29)$$

where we use the notation $|Z_k^i| = (|Z_k^{1,i}|, \dots, |Z_k^{S,i}|)$ and $r_{k|k,a_k}^{i,|Z_k^i|}$ and $P_{k|k,a_k}^{i,|Z_k^i|}$ are the updated probability of existence and updated covariance of target i with the sequence of detections/misdictions in $|Z_k^i|$. Note that due to Assumption 1), $|Z_k^i| = 0$ or $|Z_k^i| = 1$.

We can also obtain the optimal detection threshold Γ_d^* for the bound (29) as [26].

$$\Gamma_d^* = \frac{1}{2 - \min\left(2 \frac{\text{tr}(P_{k|k,a_k}^{i,|Z_k^i|})}{c^2}, 1\right)}. \quad (30)$$

To simplify notation, we define

$$C\left(r_{k|k,a_k}^{i,|Z_k^i|}, P_{k|k,a_k}^{i,|Z_k^i|}\right) = C\left(\Gamma_d^*, r_{k|k,a_k}^{i,|Z_k^i|}, P_{k|k,a_k}^{i,|Z_k^i|}\right). \quad (31)$$

It should be mentioned that Assumption 5) has been used to obtain (27). Under Assumption 5), the optimal assignment problem of GOSPA, see (18), does not need to be explicitly solved since it is known what estimate corresponds to what target. If targets were in close proximity, this approximation would not be accurate, and one would need to solve the optimal assignment problem. Nevertheless, with targets in close proximity, (27) is an upper bound to the MSGOSPA error, which is in any case what is used later in (28). The reason why it is an upper bound to MSGOSPA is that the assignment is not necessarily the optimal one, so the obtained value is higher.

D. MSGOSPA Error Upper Bound in One Time Step

In this section, we derive an upper bound of the MSGOSPA error in a single time step, which is obtained by setting $K = k$ in (20) (myopic planning). The upper bound is provided via the following lemma.

Lemma 1. Let $r_{k|k,a_k}^{i,|Z_k^i|}$ and $P_{k|k,a_k}^{i,|Z_k^i|}$ be the updated probability of existence and covariance matrix of the i -th Bernoulli for the sequence of detections/misdictions $|Z_k^i| = (|Z_k^{1,i}|, \dots, |Z_k^{S,i}|)$ where $|Z_k^{s,i}| \in \{0, 1\}$ with $s \in \{1, \dots, S\}$. Under Assumptions 1)-2), 4)-6), an upper bound on the MSGOSPA error for a given sequence $Z_k = (Z_k^1, \dots, Z_k^S)$ of measurements at time step k is

$$\begin{aligned} & \int \left[\int d^2(X_k, \hat{X}_k(a_k, Z_k)) f_{k|k}(X_k|Z_k; a_k) \delta X_k \right] \\ & \quad \prod_{s=1}^S f_{k|k-1}^{m,s}(Z_k^s) \delta Z_k^{1:S} \\ & \leq \sum_{i=1}^{n_{k|k}} \sum_{|Z_k^{1,i}|=0}^1 \dots \sum_{|Z_k^{S,i}|=0}^1 \int C\left(r_{k|k,a_k}^{i,|Z_k^i|}, P_{k|k,a_k}^{i,|Z_k^i|}\right) \\ & \quad \times \prod_{s=1}^S \left[\left(1 - |Z_k^{s,i}|\right) \left(1 - r_{k|k-1}^i \bar{p}_{a_k}^{D,i}\right) + |Z_k^{s,i}| r_{k|k-1}^i \bar{p}_{a_k}^{D,i} \right]. \end{aligned}$$

Lemma 1 is proved in Appendix A.

This MSGOSPA upper bound is used by the sensor management algorithm due to its ease of computation.

E. Multi-sensor Update for sensor management

This section provides the updated mean and covariance for each Bernoulli in the multi-sensor update required by the sensor management algorithm to calculate the expression given in Lemma 1. The sequential multi-sensor update is computed under Assumptions 1)-2), 4)-6. We encode the sequence of detection/misdictions up to the processing of the s -th sensor as $h = (h_1, \dots, h_s)$ where $h_s = 0$ if there is a misdetection at sensor s and $h_s = 1$ if there is a detection at sensor s . More explicitly, when $h = (0, 1)$ this represents a dual-sensor setup, in which sensor 1 is in the misdetection hypothesis and sensor 2 is in the detection hypothesis.

We denote the probability of existence, mean and covariance with sequence of detections and misdetections in $h = (h_1, \dots, h_{s-1})$ taking action a_k for the i -th Bernoulli as $r_{k|k,a_k}^{i,(h)}$, $\bar{x}_{k|k,a_k}^{i,(h)}$, $P_{k|k,a_k}^{i,(h)}$. Then, the updates with a misdetection and a detection at the s -th sensor are the following.

For $Z_k^{i,s} = \emptyset$, we obtain [35]

$$\bar{x}_{k|k,a_k}^{i,(h,0)} = \bar{x}_{k|k,a_k}^{i,h} \quad (32)$$

$$P_{k|k,a_k}^{i,(h,0)} = P_{k|k,a_k}^{i,h} \quad (33)$$

$$r_{k|k,a_k}^{i,(h,0)} = \frac{(1 - \bar{p}_{a_k}^{D,s,i})r_{k|k,a_k}^{i,h}}{1 - r_{k|k,a_k}^{i,h} + (1 - \bar{p}_{a_k}^{D,s,i})r_{k|k,a_k}^{i,h}}. \quad (34)$$

For $Z_k^{i,s} = \{\hat{z}_{a_k}^{i,s}\}$, there is one measurement at the predicted value. Therefore, the updated mean coincides with the predicted mean. The updated covariance matrix is given by the Kalman filter update, and the updated probability of existence is one [35]

$$\bar{x}_{k|k,a_k}^{i,(h,1)} = \bar{x}_{k|k,a_k}^{i,h} \quad (35)$$

$$P_{k|k,a_k}^{i,(h,1)} = P_{k|k,a_k}^{i,h} - P_{k|k,a_k}^{i,h} (H_{a_k}^s)^T (S_{a_k}^{i,s,h})^{-1} H_{a_k}^s P_{k|k,a_k}^{i,h} \quad (36)$$

$$r_{k|k,a_k}^{i,(h,1)} = 1. \quad (37)$$

It is important to note that the predicted mean is not affected by the value of h_k^i or the sequence of actions. Thus, for the mean, we just need the predicted value $\bar{x}_{k'|k}^i$.

F. Non-myopic Planning

Following Section [26, Sec. III.E], we proceed to write the non-myopic planning problem (20) for the MSGOSPA error with the considered upper bound in Lemma 1 and the multi-sensor update.

Let $h_{k:k'}^i$ denote the sequence of detections/misdictions from time step k to k' for the i -th Bernoulli, see Section III-E. The probability of existence, mean and covariance matrix of the i -th Bernoulli for sequence of actions $a_{k:k'}$ and $h_{k:k'}^i$, computed via (32)-(37), are $r_{k'|k',a_{k:k'}}^{i,h_{k:k'}^i}$, $\bar{x}_{k'|k',a_{k:k'}}^{i,h_{k:k'}^i}$ and $P_{k'|k',a_{k:k'}}^{i,h_{k:k'}^i}$, respectively. Then, under the assumptions in Section III-A, the probability of $h_{k'}^i$ given previous actions and previous sequences of h_k^i is

$$p(h_{k'}^i | a_{k:k'}, h_{k:k'}^i) =$$

$$\prod_{s=1}^S \left[\left(1 - h_{k'}^{i,s}\right) \left(1 - r_{k'|k'-1,a_{k:k'-1}}^{i,h_{k:k'-1}^i} \bar{p}_{a_{k:k'}}^{D,s,i}\right) + h_{k'}^{i,s} r_{k'|k'-1,a_{k:k'-1}}^{i,h_{k:k'-1}^i} \bar{p}_{a_{k:k'}}^{D,s,i} \right] \quad (38)$$

where $r_{k'|k'-1,a_{k:k'-1}}^{i,h_{k:k'-1}^i}$ is the predicted probability of existence at time step k' for the i -th Bernoulli and sequence of detection/misdictions in $h_{k:k'-1}^i$ and actions $a_{k:k'-1}$. The probability of detection $\bar{p}_{a_{k:k'}}^{D,s,i}$ is given by (22) using $\bar{x}_{k'|k',a_{k:k'}}^{i,h_{k:k'}^i}$ and $P_{k'|k',a_{k:k'}}^{i,h_{k:k'}^i}$. Note that (38) is calculated for the Bernoulli components that are present at time step k and also for the Bernoulli components born between time step k and k' .

Let $r_{k'|k',a_{k:k'}}^{i,h_{k:k'}^i}$, $P_{k'|k',a_{k:k'}}^{i,h_{k:k'}^i}$ be the probability of existence and covariance matrix of the i -th Bernoulli at time step k' given the sequence of actions, $a_{k:k'}$ and observations $h_{k:k'}^i$. Then, using (31), the MSGOSPA bound at time step k' for actions $a_{k:k'}$ and sequence of detections/misdictions $h_{k:k'}^i$ for the i -th Bernoulli is

$$C^i(a_{k:k'}, h_{k:k'}^i) = \begin{cases} \frac{c^2}{2} r_{k'|k',a_{k:k'}}^{i,h_{k:k'}^i} & r_{k'|k',a_{k:k'}}^{i,h_{k:k'}^i} \leq \Gamma_d^* \\ \frac{c^2}{2} \left(1 - r_{k'|k',a_{k:k'}}^{i,h_{k:k'}^i}\right) + r_{k'|k',a_{k:k'}}^{i,h_{k:k'}^i} \min\left(\text{tr}\left(P_{k'|k',a_{k:k'}}^{i,h_{k:k'}^i}\right), c^2\right) & r_{k'|k',a_{k:k'}}^{i,h_{k:k'}^i} > \Gamma_d^* \end{cases} \quad (39)$$

Then, the optimal policy in (20) under the assumptions in Section III-A can be written as the nested optimisations

$$\begin{aligned} \hat{J}_{\mu_{k:K}^*}^*(\cdot) = & \min_{a_k} \sum_{i=1}^{n_k|k} \sum_{h_k^i \in \{0,1\}^S} \left[p(h_k^i | a_k) C^i(a_k, h_k^i) \right. \\ & + \lambda \min_{a_{k+1}} \sum_{i=1}^{n_k|k} \sum_{h_{k+1}^i \in \{0,1\}^S} \left[p(h_{k+1}^i | a_{k:k+1}, h_k^i) \right. \\ & \quad \times C^i(a_{k:k+1}, h_{k:k+1}^i) \\ & + \dots \\ & \left. + \lambda^{K-k+1} \min_{a_K} \sum_{i=1}^{n_k|k} \sum_{h_K^i \in \{0,1\}^S} p(h_K^i | a_{k:K-1}, h_{k:K-1}^i) \right. \\ & \quad \times C^i(a_{k:K}, h_{k:K}^i) \left. \right] \dots \end{aligned} \quad (40)$$

IV. MONTE CARLO TREE SEARCH IMPLEMENTATION

This section explains the MCTS implementation [31] of the non-myopic multi-sensor multi-Bernoulli sensor management algorithm presented in the previous section. The MCTS implementation for Bernoulli filtering is described in [26]. Section IV-A explains the hypothesis tree and the reduction we consider. Section IV-B indicates the information that is kept in each node of the tree. Section IV-C explains the MCTS implementation of the algorithm. Finally, Section IV-D discusses the applicability of the designed algorithm.

A. Tree Hypothesis Reduction

The MCTS algorithm aims to approximately solve (40) by selecting the action with the lowest cost. To do so, MCTS creates a tree of actions and detections/misdictions for each sensor. In this case, the number of child nodes per parent is $2^S \times |\mathbb{A}|^S$, where $|\mathbb{A}|$ is the number of actions per sensor. This represents the number of actions per sensor multiplied by the number of possible detection/mis-detection hypotheses per each sensor.

In this paper, we propose to simplify the tree structure by combining all detection and mis-detection hypotheses for the same Bernoulli into a single hypothesis. This results in the tree only representing actions, not observation states, as done in [32]. This results in a tree where the number of child nodes (at the following time step) per parent is $|\mathbb{A}|^S$, instead of $2^S \times |\mathbb{A}|^S$. This also implies that there is only one type of edge in the tree, representing actions, while the original tree has two types of edges, representing actions and detections/mis-detections. In addition, for computational efficiency and to be able to use the MSGOSPA bound in Lemma 1, we approximate each Bernoulli single-target density, which is a Gaussian mixture, as a Gaussian. We proceed to explain how to compute these Gaussian Bernoulli densities and the associated costs in the reduced tree.

Since the reduced tree only considers actions, the i -th Bernoulli is characterised by the parameters $r_{k'|k', a_{k:k'}}^i$, $\bar{x}_{k'|k', a_{k:k'}}^i$ and $P_{k'|k', a_{k:k'}}^i$. That is, these parameters do not depend on the sequence of observations $h_{k:k'}^i$ (as in the implementation with the entire tree). When we perform an update with a predicted Bernoulli of this form, following Section III-E, there is a Bernoulli density for each $h_{k'}^i$ (sequence of detection mis-detections) of the form

$$f_{k'|k', a_{k'}}^{i, h_{k'}^i}(X) = \begin{cases} r_{k'|k', a_{k:k'}}^i \mathcal{N}(x; \bar{x}_{k'|k', a_{k:k'}}^i, P_{k'|k', a_{k:k'}}^i) & X = \{x\} \\ 1 - r_{k'|k', a_{k:k'}}^i & X = \emptyset \\ 0 & \text{otherwise} \end{cases} \quad (41)$$

where $r_{k'|k', a_{k:k'}}^i$, $\bar{x}_{k'|k', a_{k:k'}}^i$ and $P_{k'|k', a_{k:k'}}^i$ are computed via (32)-(37).

Then, for each target, there is a mixture of these Bernoulli densities, weighted by the corresponding probability of $h_{k'}^i$ such that its density is

$$f_{k'|k', a_{k'}}^i(X) = \sum_{h_{k'}^i \in \{0,1\}^S} p(h_{k'}^i | a_{k:k'-1}) f_{k'|k', a_{k'}}^{i, h_{k'}^i}(X) \quad (42)$$

where $p(h_{k'}^i | a_{k:k'-1})$ is computed using (38) with probability of existence $r_{k'|k', a_{k:k'}}^i$.

A mixture of Bernoulli densities is another Bernoulli density. In this case, its single-target density is a Gaussian mixture. To speed up the algorithm and be able to use the MSGOSPA bound in Lemma 1, we approximate this Gaussian mixture as a single Gaussian via moment matching. This is a standard procedure in target tracking algorithms,

originating from the probabilistic data association filter [46] and the joint probabilistic data association filter [47]. The resulting Bernoulli density that is stored at each node of the tree is characterised by a mean $\bar{x}_{k'|k', a_{k:k'}}^i$, a covariance matrix $P_{k'|k', a_{k:k'}}^i$ and a probability of existence $r_{k'|k', a_{k:k'}}^i$. The formulas to calculate these parameters are provided in Appendix B for completeness.

The cost at time step k' for the i -th Bernoulli for each $h_{k'}^i$ is denoted by $C^i(a_{k:k'}, h_{k'}^i)$, which is given by substituting $r_{k'|k', a_{k:k'}}^i$ and $P_{k'|k', a_{k:k'}}^i$ into (39). Then, from (40), the cost at time step k' (without considering the time-discounting factor λ) for the node representing action $a_{k:k'}$ is the weighted sum of the costs for each $h_{k'}^i$ such that

$$C_{k'}(a_{k:k'}) = \sum_{i=1}^{n_{k'|k'}} \sum_{h_{k'}^i \in \{0,1\}^S} p(h_{k'}^i | a_{k:k'-1}, h_{k:k'-1}^i) C^i(a_{k:k'}, h_{k'}^i) \quad (43)$$

where $C(a_{k:k'}, h_{k:k'}^i)$ is given by (39). With this simplification, the number of children nodes per parent is $|\mathbb{A}|^S$.

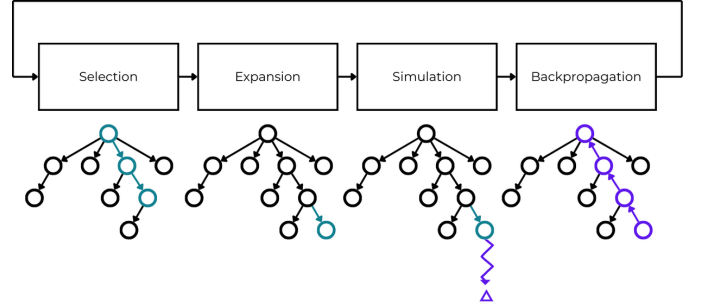


Fig. 1. The four stages of the MCTS algorithm. Where Δ is the cost returned from the simulation phase. Figure adapted from [23] [31].

B. Node Information

Each node of the tree, representing a sequence of actions $a_{k:k'}$ up to time step k' contains the following information:

- Visit count n .
- Expected initial cost of visiting node $C_{k'}$, given by (43) (not updated by simulation outcomes).
- Expected cost of visiting node over all time steps $\bar{C}_{k'}$, see (46) (updated based on simulation outcomes).
- Sensors current locations.
- Parent node.
- Set of child nodes J .
- Set of available actions $\mathbb{A}_{k'}$.
- Target means $\bar{x}_{k'|k', a_{k:k'}}^i$, covariance matrices $P_{k'|k', a_{k:k'}}^i$ and probabilities of existence $r_{k'|k', a_{k:k'}}^i$ for $i \in \{1, \dots, n_{k'|k'}\}$, see (55)-(54).
- Expected probabilities of detection for all targets and sensors $\bar{p}_{a_{k:k'}}^{D,s,i}$, see (22).
- Node depth in global tree k' .

It should be noted that at each time step k' , we add the Bernoulli components corresponding with the birth process.

C. MCTS Algorithm

The MCTS algorithm is a selective and iterative way of exploring a search space. In sensor management, each row of depth in the tree can be understood as a time step into the future, where the root node represents the current time step.

There are four phases to the MCTS algorithm which are outlined below and illustrated in Figure 1. One MCTS iteration consists of these four stages and the budget refers to how many iterations are completed when building the tree.

1) *Selection*: This phase always begins at the root node. The selection criteria (commonly the upper confidence bound for trees (UCT) (44)) [31] is used to continuously select children nodes, until a node is reached that does not have a full set of children. That is, we choose the node

$$\arg \max_{j \in J} \left\{ \bar{C}_{k',j} + \epsilon \sqrt{\frac{\ln n}{n_j}} \right\} \quad (44)$$

where ϵ is the trade-off parameter between exploration and exploitation, n_j represents the visit count of the child node and n the visit count of the current node, J is the set of children of the current node, and $\bar{C}_{k',j}$ is the expected cost of visiting the child node, as further explained in the backpropagation section.

Once a node has been selected that meets this criteria, we move to the expansion phase.

2) *Expansion*: Here, a new child node j is added to the node selected in the previous stage of the algorithm. To add this node, we randomly select a previously untried action, predict the set of target densities (and add in the densities from the birth model), generate the ideal measurement set, calculate the cost of selecting this action and then update the set of target densities based on the consideration of both the detection and misdetection hypotheses (32)-(37) for each target. We can then use the results of these calculations to initialise a new node within the tree structure. The budget of the tree is the maximum number of nodes that can be added to the tree. Each time a node is added to the tree via the expansion phase, the budget of the tree is decremented by one.

3) *Simulation*: Starting from the node that was added during the expansion phase, a rollout is begun, in which actions are selected in-line with the rollout policy, which is often random. The rollout phase continues to move down the depth of the tree until the terminating criteria is met. This is often a user-specified maximum search depth. Once the rollout phase has been completed, a total, time-discounted cost of selecting this path of actions (beginning at the root node) is obtained, denoted as Δ . During the simulation phase, no nodes are added to the tree and therefore the budget is not decremented by the simulation phase. We calculate Δ as

$$\Delta = \sum_{\kappa=k'+1}^K \lambda^{\kappa-k'} C_{\kappa}(a_{k:\kappa}) \quad (45)$$

where K is the length of the action path (which is also the total depth of the tree as it includes the simulation) and κ indicates the depth of the corresponding node (and cost) in that tree.

4) *Back-propagation*: The total, time discounted cost of the path of used nodes Δ is then incorporated into the cost of each of the nodes (already in the tree) that are on the path. The costs are incorporated as a mean average (46), using the visit count n of each node, yielding

$$\bar{C}_{k',new} = \frac{(\bar{C}_{k',old} \cdot n) + \Delta}{n + 1} \quad (46)$$

where $\bar{C}_{k',old}$ is the cost prior to the backpropagation phase, $\bar{C}_{k',new}$ refers to the updated cost associated with this action/node, n is the visit count of the node, and Δ is the cost calculated during the simulation phase. After the cost is updated, the visit count is incremented by one.

Finally, the pseudo code of the GOSPA driven, non-myopic, multi-sensor management algorithm for multi-Bernoulli filtering is provided in Algorithm 1.

Algorithm 1 Non-myopic, multi-agent, multi-Bernoulli GOSPA driven sensor management

- 1: Initialise tree with root node statistics
 - 2: **while** budget is remaining **do**
 - 3: Select child node using UCT (44) $\leftarrow$ SELECTION.
 - 4: Expand tree by adding node and calculating $\bar{x}_{k'|k',a_{k:k'}}^i, P_{k'|k',a_{k:k'}}^i$ and $r_{k'|k',a_{k:k'}}^i$, using (32)-(37), with the mixture merging (54)-(56), and also considering new born targets (6)-(7).
 - 5: Compute the cost $C_{k'}(a_{k:k'})$ using (43) and add it to the node $\leftarrow$ EXPANSION.
 - 6: Perform a simulation until terminating criteria is met, in accordance with rollout policy, obtaining the cost of the rollout Δ using (45) $\leftarrow$ SIMULATION.
 - 7: Update $\bar{C}_{k'}$ using (46) and update the number of visit counts n for all nodes on the action path $\leftarrow$ BACK-PROPAGATION.
 - 8: **end while**
 - 9: Select the action a_k^s which has the lowest cost $\bar{C}_{k'}$ (46).
-

D. Discussion on applicability

The developed multi-target multi-sensor management algorithm can be used for any problem that can be modelled with the multi-target dynamic and measurement models in Section II-A. That is, it can be used for any multi-sensor management problem with point targets. While the algorithm has been written for linear-Gaussian models, the extension to non-linear model is straightforward using linearisation techniques, which are widely used in Gaussian filtering [45]. For instance, the algorithm can be used to manage a set of sensors (e.g. drones) with a limited field of view that can move in the surveillance area. Another possible use of the algorithm is in cognitive radar [8], where different waveform parameters can lead to different probabilities of detection or single-measurement models. It is also possible to use this algorithm to control a telescope for space situation awareness [4].

V. SIMULATIONS

In this section, we provide simulation results with two sensors in a scenario in which obstacles block the movement of the sensors. All units in this section are in the international system but have been omitted for brevity.

In Section V-A, the target motion model and sensor model are described, in Section V-B the implementation details of the MCTS are provided and in Section V-C, we provide the simulation results.

A. Models

Each of the individual targets motion is governed by a nearly constant velocity model [48, Chap. 6]. The state vector is defined as $x = [p_x, v_x, p_y, v_y]^T$ in which p denotes the position, v denotes the velocity and subscript x and y denote the directional element of each of the vector quantities w.r.t. a two dimensional Cartesian grid. The transition and process noise covariance matrices are

$$F = I_2 \otimes \begin{pmatrix} 1 & \tau \\ 0 & 1 \end{pmatrix}, \quad Q = qI_2 \otimes \begin{pmatrix} \tau^3/3 & \tau^2/2 \\ \tau^2/2 & \tau \end{pmatrix} \quad (47)$$

where the sampling time $\tau = 1$ and $q = 0.8$. We consider a Bernoulli birth density that is parameterised by the mean $\bar{x}_B = [0, 0.1, 0, 0.1]^T$, covariance $P_B = \text{diag}([6, 6, 6, 6]^T)$ and probability of birth $r_k^b = 0.03$. That is, we know within reasonable accuracy the locations where targets may be born. The probability of survival is $p^S = 0.99$ and each simulation is 200 time steps in length.

We consider the case of $S = 2$ sensors. Each sensor has seven discrete actions to select from at each time step. The actions determine where the sensor will move at the next time step. Six of these actions are positioned around the circumference of a circle of constant radius 15, centered at the sensor's current position; the seventh action is for the sensor to remain in its current position. These actions can be understood as changes in direction with a constant velocity, with the added option of not moving.

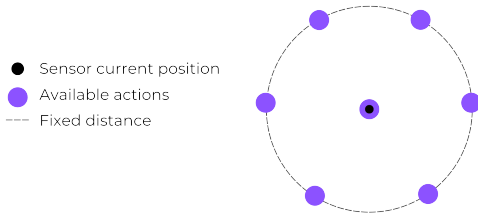


Fig. 2. Sensor movement model, showing seven possible actions for an individual sensor. Six actions cause the sensor to move and one action cause the sensor to hold its position.

The sensors collect imperfect measurements, including noise on detections of the ground truth and also clutter. The probability of detection for a target at a distance δ from sensor s is

$$p_{a_k}^{D,s} = p^D e^{-\frac{1}{2}(\frac{\delta}{r})^2} \quad (48)$$

where $p^D = 0.999$ and $r = 40$. That is, the probability of detection decays exponentially with distance. The expected

value of the probability of detection is approximated by its value at the predicted mean. The observation matrix H and observation noise matrix R are

$$H_{a_k}^s = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix}; R_{a_k}^s = \begin{bmatrix} 2 & 0 \\ 0 & 2 \end{bmatrix} \quad (49)$$

with a bias term $b_{a_k}^s = 0$. Clutter is generated by a PPP and uniformly distributed in the sensors FOV, which has a radius of $r = 40$. The clutter rate is $\bar{\lambda}_{a_k}^C = 0.1$.

The surveillance region is a square of size 500×500 . The movements of the sensor platforms are restricted by the obstacle as actions that breach their boundary are not available for selection.

In the simulations, the sensors are initialised such that the obstacle blocks their direct path to the birth location - meaning in all scenarios, the obstacle has to be overcome. This way, in every simulation that is ran, the sensor has to navigate around the obstacle in order to meet its operational objective of tracking the unknown and time varying number of targets.

An obstacle-free scenario has been omitted from the results but it can be noted that there is no major benefit for adopting computationally intense non-myopic approaches for these simple scenarios.

B. Tree Search Considerations

To speed up the algorithm, if the sensors are closer than a proximity threshold Ψ , then we build an MCTS for both sensors to optimise over both sensor actions jointly. If they are farther away, we build an MCTS for each sensor that is optimised independently for each sensor.

It can be mentioned here that if the proximity threshold Ψ is too small, in this scenario, the sensors continuously move between inside and outside of the threshold, which is not helpful for planning. The value of $\Psi = 3r$ has been selected in these simulations and prevents this from happening, whilst also allowing for the computational benefits of not jointly optimising at every time step.

For the scenario in which we jointly optimise the two sensors, we build a tree in which each parent node can have a maximum of $7^2 = 49$ children, meaning that each child node represents an action pair, one relating to each sensor. Conversely, in the scenario where the sensors are independently optimised, each parent node has a maximum of 7 children.

C. Results

We evaluate several variations of the MCTS implementation of the GOSPA driven (GD) sensor management algorithm¹. We also use the same MCTS tree structure for a set of information theoretic sensor management algorithms, driven by the KLD [10]. This algorithm aims to maximise the sum of the KLDs between the predicted densities and the posterior Bernoulli densities for each target [26, Appendix C].

For the MCTS implementations, we vary the computational budgets (number of nodes allowed to be added to the tree

¹The Python code of the proposed algorithm will be available at <https://github.com/sggjon5> after publication.

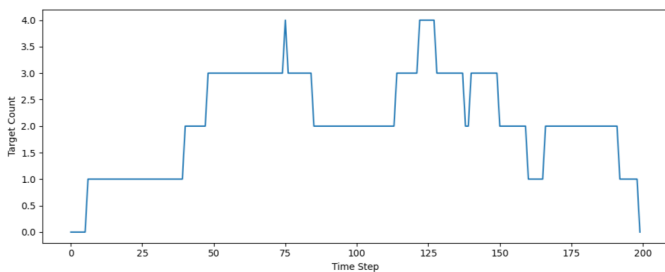


Fig. 3. Number of targets alive at each time step in the simulation. Starting with zero targets and a maximum number of four at time steps 75 and 125. The total number of targets in the simulation is 8.

via expansion) and lookahead for the MCTS (MCTS1-4). The parameters tested in the simulations are provided in Table II, including the budgets for joint and individual optimisation, see Section V-B. The reason for differing budgets for the two types of optimisation are due to the search spaces being larger in the joint optimisation case, and therefore a proportionally larger budget is provided. The decay factor ($\lambda = 0.9$) and distance threshold of joint optimisation ($\Psi = 3r$) where r is the radius of the sensors FOV, have been held constant for these simulations. We also include two myopic algorithms (Myopic - GD, Myopic - KLD) in which the parameters of the MCTS are set such that there is no rollout and the tree cannot grow beyond the first layer.

TABLE II
ALGORITHM PARAMETERS FOR GD AND KLD.

Algorithm	Budget - Joint	Budget - Individual	Lookahead
Myopic	49	7	1
MCTS1	49	7	5
MCTS2	49	7	10
MCTS3	200	40	5
MCTS4	200	40	10

In this implementation, we use loopy belief propagation [40] to compute the MB projection and we merge Bernoulli components after each update if their Mahalanobis distance is smaller than 1 [49, Eqs. (25-31)]. We merge similar Bernoulli components because of computational efficiency. This does however eventually cause the birth density to surpass the threshold of being declared as an existing target. Once this threshold of existence is surpassed, the errors (if within the maximum localisation error distance) become localisation errors.

The root mean square GOSPA (RMS-GOSPA) errors were calculated over a simulation of 200 time steps and 50 Monte Carlo runs. The GOSPA parameter $c = 2r$, where $r = 40$ is the radius of the sensor's field of view (FOV), and the parameter $p = 2$. The planning horizon is set to N , meaning the action space has a maximum of 7^N options to search through when optimized individually, and 7^{SN} options when jointly optimized, where N_s is the number of sensors being jointly optimized. The number of targets alive at each time step of the simulation can be seen in Figure 3, the most targets alive at any one time is four.

In this simulation, to ensure that the sensor management algorithms have to overcome the obstacle, when each run is initialised, the sensors direct path to observe the birth location (the only known Bernoulli component in the MB filter at conception), is blocked. This way, the algorithms must navigate around an obstacle to begin viewing the birth location. Due to this initialisation criteria, myopic algorithms quickly evidence their shortcomings, as they are unable to navigate around the obstacles and therefore get stuck. It should be noted here that all of the algorithms often miss the first target as they are initialised some distance from the birth location for the reasons described above.

Figure 4 shows the outputs from the myopic KLD planning algorithm, and the MCTS3 - GD algorithm at time step 152 in a Monte Carlo run. In the myopic (KLD and GD) algorithms, both sensors get stuck behind the obstacle. In the non-myopic case (MCTS1-4), the sensors do not get stuck behind the obstacle. In the MCTS3 - GD results, both sensors are near the birth area, maintaining track of the two targets currently alive. The purple outline on the sensors indicates that they are non-myopically planning and the purple line between the sensors indicates that the sensors are jointly optimised.

Figure 5 shows the RMS-GOSPA error at each time step. For clarity in the figure presentation, the errors for the non-myopic KLD algorithm are not shown here. As expected, non-myopic algorithms improve upon the myopic alternatives as they can navigate around the obstacle. In particular, myopic algorithms get stuck behind the obstacle and the errors are identical. MCTS3 - GD is the best performing algorithm in this scenario, outperforming the other non-myopic algorithms, even those with a longer lookahead. This is likely due to the longer lookahead allowing for increased uncertainty in the simulation phase and therefore the predictions that inform the sensor actions contain higher uncertainty; finding this balance of the appropriate lookahead we found to be dependent on the scenario. By this, we mean that the relationship between the size of the obstacles being navigated around and the distance covered by the lookahead length need to be commensurate. For example, if an obstacles is ten units square, and the sensor platform moves in steps of two, we would suggest a planning horizon of at least five to allow the algorithms to propose paths that navigate around the obstacle. In Figure 4, there are two targets currently alive, and a third has just died. This is reflected in Figure 5 by a sudden drop in the missed target error as at this point the target that died was not being tracked.

It should also be noted here that the large contribution of localisation and missed target errors in the myopic planning algorithm are caused by the merging of similar Bernoulli components at the birth location within the MB filter [49]. The localisation error for the myopic algorithms can be understood to increase as the targets move away from the birth area, with a sharp drop off when the targets either die, or they are re-categorised as missed targets once they reach the maximum localisation error c .

Table III shows the average RMS-GOSPA for each algorithm across all time steps for three differing clutter rates $\bar{\lambda}_{a_k}^C \in \{0.1, 1, 2\}$. From this we can see that the myopic algorithms have consistently higher error throughout

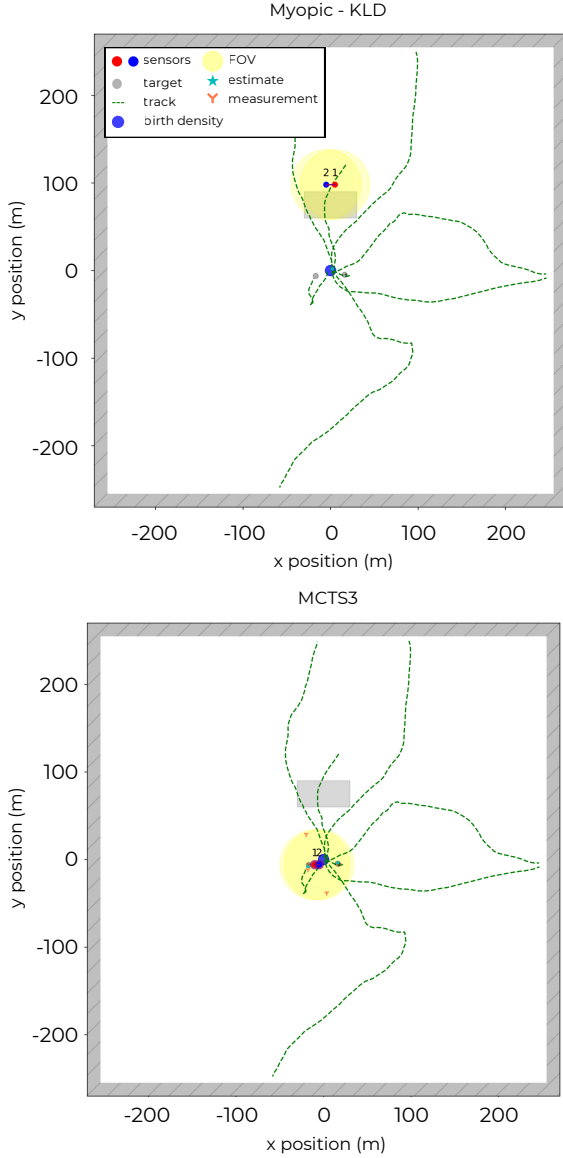


Fig. 4. Frame 152 in one Monte Carlo run, number of targets alive: 2. Top - myopic KLD scenario snapshot. Both sensors are stuck behind the obstacle as they do not have the ability to plan further ahead. Bottom - MCTS3 scenario snapshot. Sensors have navigated around the obstacle and are maintaining track of the two targets alive at this time step. Purple outline on each sensor indicates they are planning with a non-myopic policy, purple line between the two sensors indicates they are being jointly optimised at this time step. Birth density is centred at the origin.

the simulation as they are unable to navigate around the obstacle within the surveillance area and therefore unable to meet their operational objective. In addition, performance is quite consistent across these different clutter rates. It is also worth noting that the non-myopic GOSPA driven approach outperforms the KLD approach in all but one setting (MCTS2 with $\bar{\lambda}_{a_k}^C = 2$). We believe this may be due to the fact that it is not explicitly clear what maximising for information gain means in a multi-target tracking context, but minimising localisation, missed and false target errors (GOSPA error), is.

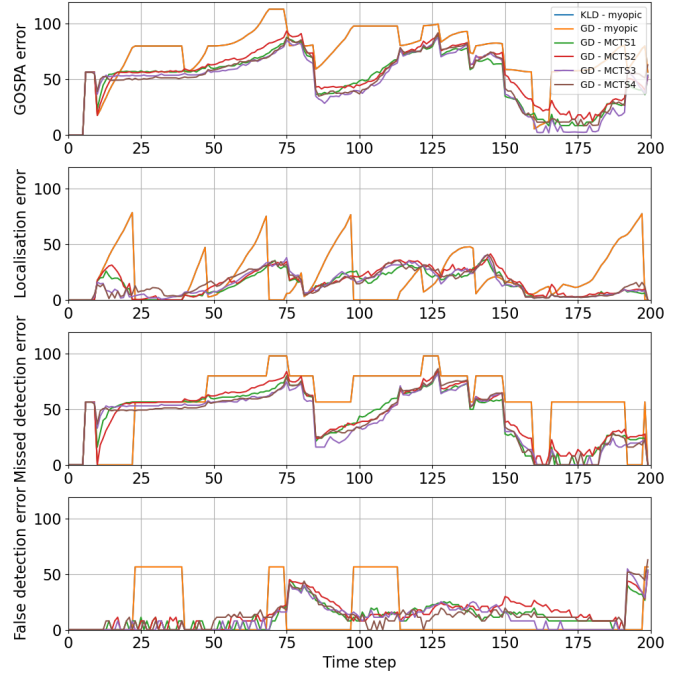


Fig. 5. GOSPA error breakdown for the obstacles scenario. The top plot shows the overall GOSPA error across each MC run for each time step. The second, third and fourth plots show the localisation, missed and false errors that contribute to the overall GOSPA error. Result set for $\bar{\lambda}_{a_k}^C = 1$.

TABLE III
RMS-GOSPA ERROR ACROSS ALL TIME STEPS FOR EACH ALGORITHM OVER 50 MC RUNS WITH VARYING CLUTTER RATES.

Algorithm	$\bar{\lambda}_{a_k}^C = 0.1$	$\bar{\lambda}_{a_k}^C = 1$	$\bar{\lambda}_{a_k}^C = 2$
Myopic - GD	72.80	72.80	72.80
MCTS1 - GD	49.05	50.46	51.78
MCTS2 - GD	54.59	54.00	56.18
MCTS3 - GD	47.17	47.88	50.53
MCTS4 - GD	48.24	49.96	51.75
Myopic - KLD	72.80	72.80	72.80
MCTS1 - KLD	51.94	53.16	53.16
MCTS2 - KLD	54.74	54.71	54.71
MCTS3 - KLD	51.38	51.51	51.51
MCTS4 - KLD	51.76	52.36	52.36

The simulations were ran on an Intel Core i7-14700KF 3.4GHz CPU and the wall clock run time of each algorithm ($\bar{\lambda}_{a_k}^C = 2$) is given in Table IV. This computational time remains basically unaltered for the three considered clutter rates. As we would expect, the algorithms with a larger lookahead and a larger computational budget (more nodes in the tree) take longer to process each time step, whilst the myopic approaches are the least computationally demanding.

VI. CONCLUSIONS

We have proposed a sensor management algorithm that is transparent, multi-sensor and multi-Bernoulli, driven by the GOSPA metric. Utilising GOSPA as the cost function in sensor management, provides a desirable level of interpretability and transparency within the decision making. This is because the

TABLE IV
WALL CLOCK RUN TIME OF EACH ALGORITHM.

Algorithm	Driven by	Runtime per time step [s]
Myopic	GOSPA	0.02
MCTS1	GOSPA	0.23
MCTS2	GOSPA	0.62
MCTS3	GOSPA	0.85
MCTS4	GOSPA	2.57
Myopic	KLD	0.02
MCTS1	KLD	0.23
MCTS2	KLD	0.61
MCTS3	KLD	0.85
MCTS4	KLD	2.56

GOSPA-derived cost function specifically optimises for the quantities of interest in a multi-target tracking scenario.

The proposed approach has general applicability to non-myopic multi-Bernoulli sensor management algorithms. We have demonstrated the benefits of non-myopic planning when operating in a surveillance area that contains obstacles.

A line of future research is to develop GOSPA-based non-myopic multi-target multi-sensor management algorithm with fewer approximations. These should have better performance, though with a higher computational complexity. Another line of future research is the development of reinforcement learning algorithms for multi-target multi-sensor management, for instance, using a reward based on the GOSPA metric.

REFERENCES

- [1] A. O. Hero III, D. A. Castañón, D. Cochran, K. Kastella, *Foundations and Applications of Sensor Management*, Springer, 2008.
- [2] M. Hernandez, A. F. García-Fernández, S. Maskell, Nonmyopic sensor control for target search and track using a sample-based GOSPA implementation, *IEEE Transactions on Aerospace and Electronic Systems* 60 (1) (2024) 387–404.
- [3] C. Kreucher, A. Hero III, K. Kastella, D. Chang, Efficient methods of non-myopic sensor management for multitarget tracking, in: 2004 43rd IEEE Conference on Decision and Control, Vol. 1, 2004, pp. 722–727 Vol.1.
- [4] B. Oakes, D. Richards, J. Barr, J. Ralph, Double deep Q networks for sensor management in space situational awareness, 2022, pp. 1–6.
- [5] V. Roy, A. Simonetto, G. Leus, Spatio-temporal sensor management for environmental field estimation, *Signal Processing* 128 (2016) 369–381.
- [6] X. Wang, R. Hoseinnezhad, A. K. Gostar, T. Rathnayake, B. Xu, A. Bab-Hadiashar, Multi-sensor control for multi-object Bayes filters, *Signal Processing* 142 (2018) 260–270.
- [7] C. Kreucher, D. Blatt, A. Hero, K. Kastella, Adaptive multi-modality sensor scheduling for detection and tracking of smart targets, *Digital Signal Processing* 16 (5) (2006) 546–567.
- [8] Y. Huang, R. Guo, Y. Zhang, Z. Chen, Deep-reinforcement-learning-based radar parameter adaptation for multiple-target tracking, *IEEE Transactions on Aerospace and Electronic Systems* 60 (5) (2024) 7125–7141.
- [9] V. Krishnamurthy, *Partially Observed Markov Decision Processes: From Filtering to Controlled Sensing*, Cambridge University Press, 2016.
- [10] C. M. Kreucher, A. O. Hero III, K. D. Kastella, M. R. Morelande, An information-based approach to sensor management in large dynamic networks, *Proceedings of the IEEE* 95 (5) (2007) 978–999.
- [11] B. Ristic, B. Vo, D. Clark, A note on the reward function for PHD filters with sensor control, *IEEE Transactions on Aerospace and Electronic Systems* 47 (2) (2011) 1521–1529.
- [12] C. Kreucher, K. Kastella, A. O. Hero III, Sensor management using an active sensing approach, *Signal Processing* 85 (3) (2005) 607–624.
- [13] K. A. LeGrand, P. Zhu, S. Ferrari, Cell multi-Bernoulli (cell-MB) sensor control for multi-object search-while-tracking (SWT), *IEEE Transactions on Pattern Analysis and Machine Intelligence* 45 (6) (2023) 7195–7207.
- [14] A. A. Saucan, S. Das, M. Z. Win, On multisensor activation policies for Bernoulli tracking, in: *IEEE Military Communications Conference (MILCOM)*, 2021, pp. 795–801.
- [15] E. H. Aoki, A. Bagchi, P. Mandal, Y. Boers, A theoretical look at information-driven sensor management criteria, in: *International Conference on Information Fusion*, 2011, pp. 1180–1187.
- [16] M. Beard, B. T. Vo, B. N. Vo, S. Arulampalam, Void probabilities and Cauchy-Schwarz divergence for generalized labeled multi-Bernoulli models, *IEEE Transactions on Signal Processing* 65 (19) (2017) 5047–5061.
- [17] A. K. Gostar, T. Rathnayake, R. Tennakoon, A. Bab-Hadiashar, G. Battistelli, L. Chisci, R. Hoseinnezhad, Cooperative sensor fusion in centralized sensor networks using Cauchy-Schwarz divergence, *Signal Processing* 167 (2020) 107278.
- [18] P. Tichavsky, C. Muravchik, A. Nehorai, Posterior Cramer-Rao bounds for discrete-time nonlinear filtering, *IEEE Transactions on Signal Processing* 46 (5) (1998) 1386–1396.
- [19] M. Hernandez, A. Farina, B. Ristic, PCRLB for tracking in cluttered environments: measurement sequence conditioning approach, *IEEE Transactions on Aerospace and Electronic Systems* 42 (2) (2006) 680–704.
- [20] R. Tharmarasa, T. Kirubarajan, M. Hernandez, A. Sinha, PCRLB-based multisensor array management for multitarget tracking, *IEEE Transactions on Aerospace and Electronic Systems* 43 (2) (2007) 539–555.
- [21] K. L. Bell, C. J. Baker, G. E. Smith, J. T. Johnson, M. Rangaswamy, Cognitive radar framework for target detection and tracking, *IEEE Journal of Selected Topics in Signal Processing* 9 (8) (2015) 1427–1439.
- [22] X. Tang, M. Li, R. Tharmarasa, T. Kirubarajan, Posterior Cramér-Rao lower bounds for extended target tracking with Gaussian process pmht, in: 2019 22th International Conference on Information Fusion (FUSION), 2019, pp. 1–8.
- [23] P. Boström-Rost, D. Axehill, G. Hendeby, Sensor management for search and track using the Poisson multi-Bernoulli mixture filter, *IEEE Transactions on Aerospace and Electronic Systems* 57 (5) (2021) 2771–2783.
- [24] P. Boström-Rost, D. Axehill, G. Hendeby, PMBM filter with partially grid-based birth model with applications in sensor management, *IEEE Transactions on Aerospace and Electronic Systems* 58 (1) (2022) 530–540.
- [25] A. S. Rahmathullah, A. F. García-Fernández, L. Svensson, Generalized optimal sub-pattern assignment metric, in: 2017 20th International Conference on Information Fusion (Fusion), 2017, pp. 1–8.
- [26] G. Jones, A. F. García-Fernández, C. Blackman, Non-myopic GOSPA-driven Gaussian Bernoulli sensor management, *IEEE Transactions on Aerospace and Electronic Systems* 60 (6) (2024) 7628–7642.
- [27] A. F. García-Fernández, M. Hernandez, S. Maskell, An analysis on metric-driven multi-target sensor management: GOSPA versus OSPA, in: 2021 IEEE 24th International Conference on Information Fusion (FUSION), 2021, pp. 1–8.
- [28] D. Schuhmacher, B. T. Vo, B. N. Vo, A consistent metric for performance evaluation of multi-object filters, *IEEE Transactions on Signal Processing* 56 (8) (2008) 3447–3457.
- [29] P. Barrios, G. Naqvi, M. Adams, K. Leung, F. Inostroza, The cardinalized optimal linear assignment (COLA) metric for multi-object error evaluation, in: 2015 18th International Conference on Information Fusion (Fusion), 2015, pp. 271–279.
- [30] G. Jones, A. F. García-Fernández, GOSPA-driven multi-Bernoulli Gaussian sensor management, in: 2024 IEEE International Conference on Multisensor Fusion and Integration for Intelligent Systems (MFI), 2024, pp. 1–8.
- [31] C. B. Browne, E. Powley, D. Whitehouse, S. M. Lucas, P. I. Cowling, P. Rohlfshagen, S. Tavener, D. Perez, S. Samothrakis, S. Colton, A survey of Monte Carlo tree search methods, *IEEE Transactions on Computational Intelligence and AI in Games* 4 (1) (2012) 1–43.
- [32] P. Salvagnini, F. Pernici, M. Cristani, G. Lisanti, A. D. Bimbo, V. Murino, Non-myopic information theoretic sensor management of a single pan-tilt-zoom camera for multiple object detection and tracking, *Computer Vision and Image Understanding* 134 (2015) 74–88.
- [33] R. P. S. Mahler, *Advances in Statistical Multisource-Multitarget Information Fusion*, Artech House, 2014.
- [34] A. F. García-Fernández, J. L. Williams, K. Granström, L. Svensson, Poisson multi-Bernoulli mixture filter: Direct derivation and implementation, *IEEE Transactions on Aerospace and Electronic Systems* 54 (4) (2018) 1883–1901.
- [35] A. F. García-Fernández, Y. Xia, K. Granström, L. Svensson, J. L. Williams, Gaussian implementation of the multi-Bernoulli mixture

- filter, in: 2019 22th International Conference on Information Fusion (FUSION), 2019, pp. 1–8.
- [36] A. F. García-Fernández, L. Svensson, J. L. Williams, Y. Xia, K. Granström, Trajectory Poisson multi-Bernoulli filters, *IEEE Transactions on Signal Processing* 68 (2020) 4933–4945.
 - [37] J. L. Williams, An efficient, variational approximation of the best fitting multi-Bernoulli filter, *IEEE Transactions on Signal Processing* 63 (1) (2015) 258–273.
 - [38] H. Kim, A. F. García-Fernández, Y. Ge, Y. Xia, L. Svensson, H. Wymeersch, Set-type belief propagation with applications to Poisson multi-Bernoulli SLAM, *IEEE Transactions on Signal Processing* 72 (2024) 1989–2005.
 - [39] K. G. Murty, Letter to the editor - an algorithm for ranking all the assignments in order of increasing cost, *Oper. Res.* 16 (1968) 682–687.
 - [40] J. Williams, R. Lau, Approximate evaluation of marginal association probabilities with belief propagation, *IEEE Transactions on Aerospace and Electronic Systems* 50 (4) (2014) 2942–2959.
 - [41] F. Meyer, T. Kropfreiter, J. L. Williams, R. Lau, F. Hlawatsch, P. Braca, M. Z. Win, Message passing algorithms for scalable multitarget tracking, *Proceedings of the IEEE* 106 (2) (2018) 221–259.
 - [42] P. Horridge, S. Maskell, Real-time tracking of hundreds of targets with efficient exact JPDAF implementation, in: 2006 9th International Conference on Information Fusion, 2006, pp. 1–8.
 - [43] M. R. Morelande, Joint data association using importance sampling, in: 2009 12th International Conference on Information Fusion, 2009, pp. 292–299.
 - [44] J. L. Williams, Marginal multi-Bernoulli filters: RFS derivation of MHT, JPDA, and association-based MeMBer, *IEEE Transactions on Aerospace and Electronic Systems* 51 (3) (2015) 1664–1687.
 - [45] S. Särkkä, L. Svensson, *Bayesian Filtering and Smoothing*, 2nd Edition, Cambridge University Press, 2023.
 - [46] Y. Bar-Shalom, F. Daum, J. Huang, The probabilistic data association filter, *IEEE Control Systems Magazine* 29 (6) (2009) 82–100.
 - [47] Y. Bar-Shalom, E. Tse, Tracking in a cluttered environment with probabilistic data association, *Automatica* 11 (5) (1975) 451–460.
 - [48] Y. Bar-Shalom, X. Li, T. Kirubarajan, *Estimation with Applications to Tracking and Navigation: Theory, Algorithms and Software*, 2004.
 - [49] M. Fontana, A. F. García-Fernández, S. Maskell, Data-driven clustering and Bernoulli merging for the Poisson multi-Bernoulli mixture filter, *IEEE Transactions on Aerospace and Electronic Systems* (2023) 1–14.

Supplementary material: GOSPA-Driven Non-Myopic Multi-Sensor Management with Multi-Bernoulli Filtering

APPENDIX A

In this appendix, we prove Lemma 1 by deriving the MSGOSPA error in a single time step, which is obtained by setting $K = k$ in (20) (myopic planning). Using (23) and (26) and integrating over all measurements, the MSGOSPA error is

$$E = \int \left[\int d^2 \left(X_k, \hat{X}_k(a_k, Z_k) \right) \times f_{k|k} \left(X_k | Z_k; a_k \right) \delta X_k \right] \prod_{s=1}^S f_{k|k-1}^{m,s} \left(Z_k^s \right) \delta Z_k^{1:S}. \quad (50)$$

Using (28), we obtain the upper bound B of E ($E \leq B$) with

$$B = \int \cdots \int \left[\sum_{i=1}^{n_{k|k}} C \left(r_{k|k,a_k}^{i,|Z_k^i|}, P_{k|k,a_k}^{i,|Z_k^i|} \right) \right] \times \prod_{s=1}^S \left[f_{k|k-1}^{m,p,s} \left(Z_k^{s,0} \right) \prod_{i=1}^{n_{k|k-1}} f_{k|k-1}^{m,i,s} \left(Z_k^{s,i} \right) \right] \left[\prod_{s=1}^S \prod_{i=0}^{n_{k|k-1}} \delta Z_k^{s,i} \right]. \quad (51)$$

Then, since the cost does not depend on $Z_k^{s,0}$ we can integrate the PPP densities out (each integral has value one). Then, the integral of the sum can be written as the sum of the integrals. This yields

$$B = \int \cdots \int \left[\sum_{i=1}^{n_{k|k}} C \left(r_{k|k,a_k}^{i,|Z_k^i|}, P_{k|k,a_k}^{i,|Z_k^i|} \right) \right] \times \prod_{s=1}^S \left[\prod_{i=1}^{n_{k|k-1}} f_{k|k-1}^{m,i,s} \left(Z_k^{s,i} \right) \right] \left[\prod_{s=1}^S \prod_{i=1}^{n_{k|k-1}} \delta Z_k^{s,i} \right] \\ = \sum_{i=1}^{n_{k|k}} \int \cdots \int C \left(r_{k|k,a_k}^{i,|Z_k^i|}, P_{k|k,a_k}^{i,|Z_k^i|} \right) \prod_{s=1}^S f_{k|k-1}^{m,i,s} \left(Z_k^{s,i} \right) \left[\prod_{s=1}^S \delta Z_k^{s,i} \right]. \quad (52)$$

Now we plug (25) into (52) and carry out the set integrals [33]. This results in the MSGOSPA error bound in Lemma 1 that is used in the proposed sensor management algorithm

$$B = \sum_{i=1}^{n_{k|k}} \sum_{|Z_k^{1,i}|=0}^1 \cdots \sum_{|Z_k^{S,i}|=0}^1 C \left(r_{k|k,a_k}^{i,|Z_k^i|}, P_{k|k,a_k}^{i,|Z_k^i|} \right) \times \prod_{s=1}^S \left[\left(1 - |Z_k^{s,i}| \right) \left(1 - r_{k|k-1}^i \bar{p}_{a_k}^{D,s,i} \right) + |Z_k^{s,i}| r_{k|k-1}^i \bar{p}_{a_k}^{D,s,i} \right]. \quad (53)$$

APPENDIX B

In this appendix, we provide the details of the Gaussian mixture merging used to approximate the single-target density in the Bernoulli density (42), which is a Gaussian mixture, as a Gaussian.

First of all, the probability of existence of the Bernoulli (42) is the weighted sum of the probability of existence of each Bernoulli

$$r_{k'|k',a_{k:k'}}^i = \sum_{h_{k'}^i \in \{0,1\}^S} p(h_{k'}^i | a_{k:k'-1}) r_{k'|k',a_{k:k'}}^{i,h_{k'}^i}. \quad (54)$$

Then, the merged mean and covariance matrix are given by the standard moment matching formula for Gaussian mixtures, where the weight of each Gaussian mixture component is proportional to $p(h_{k'}^i | a_{k:k'-1}) r_{k'|k',a_{k:k'}}^{i,h_{k'}^i}$, resulting in [48]

$$\bar{x}_{k'|k',a_{k:k'}}^i = \frac{1}{r_{k'|k',a_{k:k'}}^i} \sum_{h_{k'}^i \in \{0,1\}^S} p(h_{k'}^i | a_{k:k'-1}) r_{k'|k',a_{k:k'}}^{i,h_{k'}^i} \times \bar{x}_{k'|k',a_{k:k'}}^{i,h_{k'}^i} \quad (55)$$

$$P_{k'|k',a_{k:k'}}^i = \frac{1}{r_{k'|k',a_{k:k'}}^i} \sum_{h_{k'}^i \in \{0,1\}^S} p(h_{k'}^i | a_{k:k'-1}) r_{k'|k',a_{k:k'}}^{i,h_{k'}^i} \times \left[P_{k'|k',a_{k:k'}}^{i,h_{k'}^i} + \left(\bar{x}_{k'|k',a_{k:k'}}^{i,h_{k'}^i} - \bar{x}_{k'|k',a_{k:k'}}^i \right) \times \left(\bar{x}_{k'|k',a_{k:k'}}^{i,h_{k'}^i} - \bar{x}_{k'|k',a_{k:k'}}^i \right)^T \right]. \quad (56)$$