

See discussions, stats, and author profiles for this publication at: <https://www.researchgate.net/publication/330819039>

POSE.3C: Prediction-based Opportunistic Sensing using Distributed Classification, Clustering and Control in Heterogeneous Sensor Networks

Article in *IEEE Transactions on Control of Network Systems* · February 2019

DOI: 10.1109/TCNS.2019.2896949

CITATIONS

0

READS

41

3 authors:



James Zachary Hare

Army Research Laboratory

21 PUBLICATIONS 120 CITATIONS

[SEE PROFILE](#)



Shalabh Gupta

University of Connecticut

92 PUBLICATIONS 995 CITATIONS

[SEE PROFILE](#)



Thomas A. Wettergren

Naval Undersea Warfare Center

83 PUBLICATIONS 489 CITATIONS

[SEE PROFILE](#)

Some of the authors of this publication are also working on these related projects:



Distributed Optimal Control [View project](#)



Swarm intelligent control [View project](#)

POSE.3C: Prediction-based Opportunistic Sensing using Distributed Classification, Clustering and Control in Heterogeneous Sensor Networks

James Z. Hare[†] Shalabh Gupta^{†*} Thomas A. Wettergren[‡]

Abstract—This paper presents a distributed algorithm, called *Prediction-based Opportunistic Sensing using Distributed Classification, Clustering and Control* (POSE.3C), for self adaptation of sensor networks for energy management. The underlying 3C network autonomy concept enables utilization of the target classification information to form dynamic clusters around the predicted target position via selection of sensor nodes with the highest energies and maximum geometric diversity. Further, the nodes can probabilistically control their heterogeneous devices to track targets of interest and minimize energy consumption in a completely distributed manner. Theoretical properties of the POSE.3C network are established and derived in terms of the network lifetime and missed detection characteristics. The algorithm is validated through extensive simulations which demonstrate a significant increase in the network lifetime as compared to other network control approaches, while providing high tracking accuracy and low missed detection rates.

1. INTRODUCTION

Recent advancements in sensing, computing, and communication technologies have enabled *Distributed Sensor Networks* (DSN) to evolve into intelligent systems that are capable of adaptive sensing and target tracking [1]. However, a major limitation that affects the long-term reliability of DSN is the limited availability of energy resources [2]. Once a group of nodes are depleted of energy, they fail to sense, causing missed detections, and coverage gaps, which reduces network lifetime. Thus, it is important to maximize the network lifetime via minimizing the energy consumption per node, while maintaining high tracking accuracy and low missed detection rates.

To address this issue, *Opportunistic Sensing* [2], [3], [4] approaches have been proposed, where the objective is to preserve energy by activating nodes locally in the region around the target. The current approaches in this domain are mainly Cluster Head (CH) based [3] and only consider scheduling of binary operating states (On/Off). In contrast, current sensor nodes consist of multiple heterogeneous sensing devices consuming different amounts of power; thus, advanced heterogeneous network control approaches are needed to ensure energy-efficiency to maximize the network lifetime. In this regard, this paper presents a *distributed supervisory control* approach which probabilistically controls the heterogeneous devices on each node, such that the nodes around the target

activate their high power (e.g., active) sensing devices to track the target, while distant nodes either switch to low power (e.g., passive) sensing to stay aware, or sleep to preserve energy.

The above supervisory control requires predictive intelligence of target's whereabouts to enable the distributed nodes to pro-actively prepare for a target's arrival and to form a cluster with high power sensors activated around the target. However, there could be a large number of nodes located around the target, which can lead to redundant node activation and wasted energy. In this regard, the second issue addressed in this paper is *distributed clustering* via sensor node selection. Current clustering approaches typically select nodes based on their distance to the target or detection capability [5]. However, in applications where targets may frequently travel in the same section (e.g., a lane) of the network, the current approaches will select the same nodes again and again, thus depleting their energies and creating energy depleted lanes. Thus, it is also necessary to maintain uniform spatial distribution of remaining energy in nodes around regions of frequent target visits. In our approach, we pursue a multi-stage filtering process for node selection that selects a group of nodes with the highest remaining energies and maximum geometrical diversity. This process improves target tracking accuracy as well as facilitates uniform depletion of energy to minimize coverage gaps and further extend the network lifetime.

In the distributed clustering framework, energy wastage may still occur if clusters are formed around *Targets Not of Interest* (TNOIs), while their primary goal is to track *Targets of Interest* (TOIs) [6]. For example, in a border surveillance application, the TOIs could be humans and vehicles, while the TNOIs could be animals. Therefore, we incorporate *distributed classification* into the clustering strategy to adapt the cluster size based on the classification decisions. The objective is to activate $N_{sel} > 1$ nodes around a TOI to improve the estimation accuracy via distributed fusion, while activating only 1 node around a TNOI to maintain awareness and preserve energy. This approach drastically improves the network lifetime by opportunistic sensing only around TOIs.

To address the above issues of *Classification*→*Clustering*→*Control* (3C network autonomy), this paper presents a distributed algorithm, called *Prediction-based Opportunistic Sensing using Distributed Classification, Clustering and Control* (POSE.3C), which manages the heterogeneous devices on each node for energy-efficiency.

[†] Department of Electrical and Computer Engineering, University of Connecticut, Storrs, CT 06269, USA.

* Corresponding Author Email: shalabh.gupta@uconn.edu.

[‡] Naval Undersea Warfare Center, Newport, RI 02841, USA.

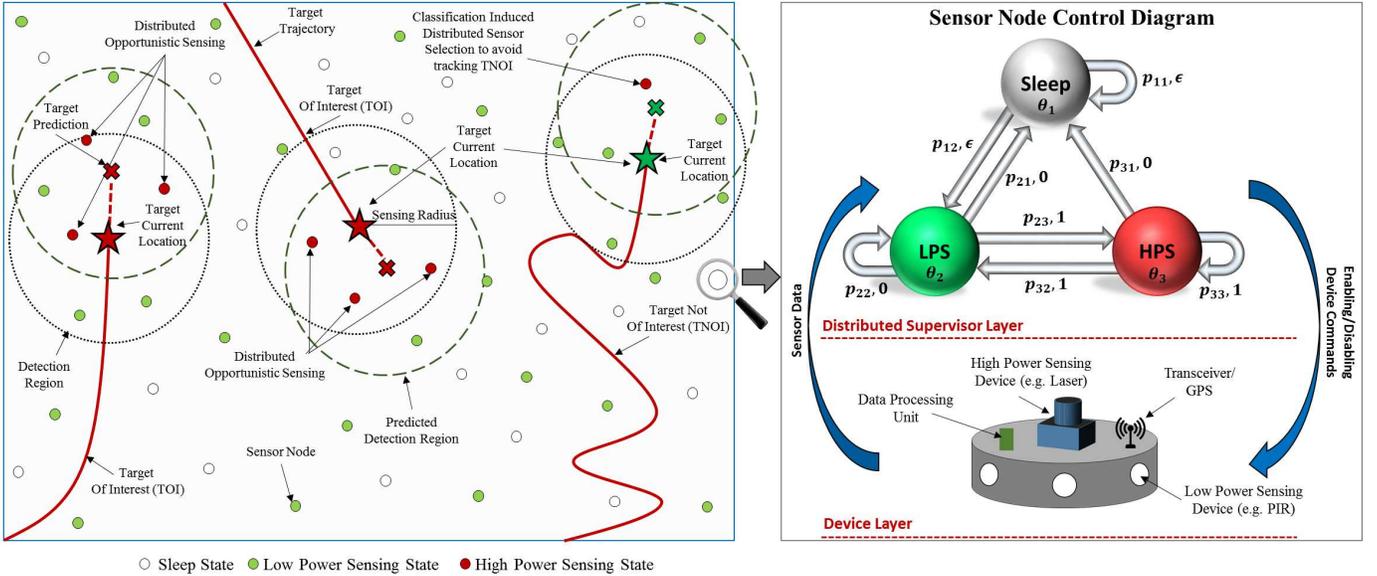


Figure 1: Illustration of the POSE.3C algorithm with a PFSA-based distributed supervisor acting on each node.

As shown in Fig. 1, a distributed supervisor designed as a *Probabilistic Finite State Automaton* (PFSA), is embedded on each node to enable/disable its sensing and communication devices. The PFSA states include: 1) *Sleep*, 2) *Low Power Sensing* (LPS), and 3) *High Power Sensing* (HPS). The *Sleep* state disables all devices to minimize energy consumption. The LPS state enables the LPS devices for target detection, while the HPS state enables the HPS devices for state estimation and target classification. The transceiver is turned on in both LPS and HPS states for communication.

The state transition probabilities of the PFSA are dynamically updated based on the target's predicted location. This information could be measured and computed locally by the nodes and/or received from their neighbors. The nodes perform distributed fusion to identify the target's class (TOI or a TNOI) and predict its state during the next time interval. This fused information is then used to form clusters of optimal nodes. The selected nodes probabilistically transition to the HPS state, while the nodes not selected cycle between the low power states (i.e., *Sleep* and LPS) to stay aware while preserving energy. Fig. 1 illustrates this approach with three targets traveling through the network. As illustrated, $N_{sel} = 3$ nodes are activated in the HPS state around the TOIs, while only 1 node is activated around the TNOI.

The main contributions of this paper are as follows:

- Development of a distributed algorithm that improves the lifetime of a heterogeneous sensor network, while providing high tracking accuracy with low missed detection rates, via enabling 3C network autonomy as follows:
 - Distributed classification: This governs the size of dynamic clusters for tracking the target based on target class inference to minimize energy wastage,
 - Distributed clustering: This is done to form dynamic clusters around the target's predicted state via select-

ing optimal nodes that maximize remaining energy and geometric diversity, and

- Distributed supervisory control: This PFSA-based supervisor probabilistically enables/disables the heterogeneous devices on each node.
- Theoretical derivation of the network's performance characteristics: (i) Expected energy consumption, (ii) Expected network lifetime, (iii) Probability of missed detection for a target birth and for a mature target.
- Comparative evaluation with other distributed methods which show significant improvement in network lifetime, high tracking accuracy, and low missed detection rates.

2. RELATED WORK

The following subsections describe the existing classification and scheduling methods employed in sensor networks.

A. Classification Methods in Sensor Networks

Diverse classification problems arise in sensor networks, include classification of vehicles, soldiers, and pedestrians [7], trespassers [8], [9], underwater mines [10], etc. Typically, the sensors detect acoustic signals, magnetic fields, or collect images, etc., that provide distinguishing features between target classes. Some classification approaches were developed based on kinematic features (e.g., range and azimuth of the target), which compute the posteriori probability of the target motion model given the observed track, known as Joint Tracking and Classification [11], [12]. These methods are typically used in radar/sonar applications (e.g., classifying commercial aircraft vs. fighter jets) and have a high computational complexity. Since centralized classification is impractical for large DSN, most networks perform CH-based classification by fusing local decisions or features from each sensor node [7], [9]. However, these approaches only deal with the classification problem and did not address the 3C network autonomy problem.

B. Network Control and Scheduling

Typical methods of network control deal with minimizing the number of active nodes around the target via sensor selection. This is achieved by Centralized or CH-based approaches [3], [11], [13], [14]. These methods select the optimal sensors to track the target by maximizing/minimizing one of the following cost functions: probability of detection [15], tracking accuracy [16], energy [17], Kullback-Liebler distance [5], or other information theoretic measures [11], [18]. However, the centralized approaches require frequent communication and heavy computational requirements of searching the entire network, thus making them impractical for large scale networks. CH-based approaches [3], [11], [13], [14] improve the communication cost but are not robust to sensor failure. If a CH were to fail, the entire cluster would stop receiving control decisions, rendering a large coverage gap in the network.

To alleviate these pitfalls, DSN were proposed that allow for robustness to node failure and energy-efficiency via event triggered communication. Kaplan [19] proposed a local node selection algorithm for DSN based on a cost function that minimizes the Mean Squared Error of the target state. This approach was extended by utilizing the innovation of the estimate [20] and using mutual information [21].

C. Literature Gaps

The following research gaps exist in the current literature, which are addressed in this paper:

- 1) Control of heterogeneous nodes has not been studied for energy-efficient networks. The literature typically deals with 2 operating conditions e.g., ON and OFF.
- 2) Control approaches are typically deterministic and not probabilistic; hence they lack robustness.
- 3) Classification is not used for clustering and control; which extends the network lifetime by focusing only on TOIs. The concept of 3C network autonomy is not proposed.
- 4) DSN sensor selection methods do not optimize for both energy and tracking accuracy; hence they lead to non-uniform depletion of energy.
- 5) Technical literature does not provide theoretical guarantees of network performance (i.e., network lifetime, probability of detection, and missed detection rates). This paper allows the network designer to predict network performance based on their specifications.

3. PROBLEM FORMULATION

Let $\Omega \subset \mathbb{R}^2$ be the ROI with area A_Ω . Let $\mathcal{S} = \{s_1, s_2, \dots, s_n\}$ be the set of n static sensor nodes randomly deployed throughout Ω , where node $s_i \in \mathcal{S}$ is positioned at $\mathbf{u}^{s_i} = (x^{s_i}, y^{s_i}) \in \Omega$. Let $\mathcal{T} = \{\tau_1, \tau_2, \dots, \tau_m\}$ be the set of m targets traveling through Ω . Let the position of a target $\tau_\ell \in \mathcal{T}$ at time step k be denoted as $\mathbf{u}^{\tau_\ell}(k) = (x^{\tau_\ell}, y^{\tau_\ell})(k) \in \Omega$.

A. Description of a Sensor Node

Each static sensor node $s_i \in \mathcal{S}$ contains a sensor suite of heterogeneous sensing devices, a data processing unit (DPU),

a communication device (transceiver), and a GPS device, as shown in the device layer of Fig. 1. The sensor suite includes several LPS devices (e.g. Passive Infrared sensors) for target detection, and HPS devices (e.g. Laser rangefinder) for further target interrogation (e.g. bearing and range measurements). For classification purposes additional sensors could be used to identify target classes (e.g. vibrations sensors to separate vehicles from humans). The transceiver allows the node to transmit and receive information within its neighborhood. Note: This work does not consider communication issues which will be studied in a future work.

Definition 3.1 (Neighborhood). *The neighborhood of a sensor node $s_i \in \mathcal{S}$ is defined as*

$$\mathcal{N}^{s_i} \triangleq \{s_j \in \{\mathcal{S} \setminus s_i\} : \|\mathbf{u}^{s_j} - \mathbf{u}^{s_i}\| \leq R_c\}, \quad (1)$$

where R_c is the communication radius of the node.

The energy consumed [22] by node s_i until time k is

$$E^{s_i}(k) = \sum_k \sum_j e_j^{s_i} \cdot \zeta_j^{s_i}(k) \Delta T, \quad (2)$$

where $e_j^{s_i}$ denotes the rate of energy consumption per unit time by a certain device $j \in \{DPU, LPS, HPS, \text{transmitter (TX), receiver (RX), clock}\}$; $\zeta_j^{s_i}(k) \in \{0, 1\}$ indicates whether the device is ON or OFF at time k ; and ΔT is the sample time interval. Thus, the total energy consumed by the network is $E_{net}(k) = \sum_{i=1}^n E^{s_i}(k)$.

B. Description of a Target

1) **Target Motion and Measurement Model:** The motion of a target, τ_ℓ , is modeled as a *Discrete White Noise Acceleration (DWNA)* model [23]

$$\mathbf{x}(k+1) = \mathbf{f}(\mathbf{x}(k), k) + \mathbf{v}(k), \quad (3)$$

where $\mathbf{x}(k) \triangleq [x(k), \dot{x}(k), y(k), \dot{y}(k), \psi(k)]'$ is the target state at time k , which includes the position and velocity in x and y and the turning rate $\psi(k)$; $\mathbf{f}(\mathbf{x}(k), k)$ is the state transition matrix; and $\mathbf{v}(k)$ is the zero-mean white Gaussian noise with covariance $E[\mathbf{v}(k)\mathbf{v}(k)'] = \mathbf{Q}$. The target is assumed to travel with a nearly coordinated turning model [23].

The HPS devices collect o range and azimuth observations at each time step k , $\mathbf{z}(k) = (\mathbf{z}_1(k), \dots, \mathbf{z}_o(k))$, where each observation is modeled as

$$\mathbf{z}_j(k) = \mathbf{h}(\mathbf{x}(k), k) + \mathbf{w}(k), \quad (4)$$

where $\mathbf{h}(\mathbf{x}(k), k)$ is the nonlinear measurement model and $\mathbf{w}(k)$ is the zero-mean white Gaussian measurement noise with covariance $E[\mathbf{w}(k)\mathbf{w}(k)'] = \mathbf{R}(k)$. The observations $\mathbf{z}(k)$ also include false measurements which are generated according to a Poisson distribution with mean μ_{cl} [23].

Note: Detailed derivations of the above models are provided in [23] and are beyond the scope of this paper.

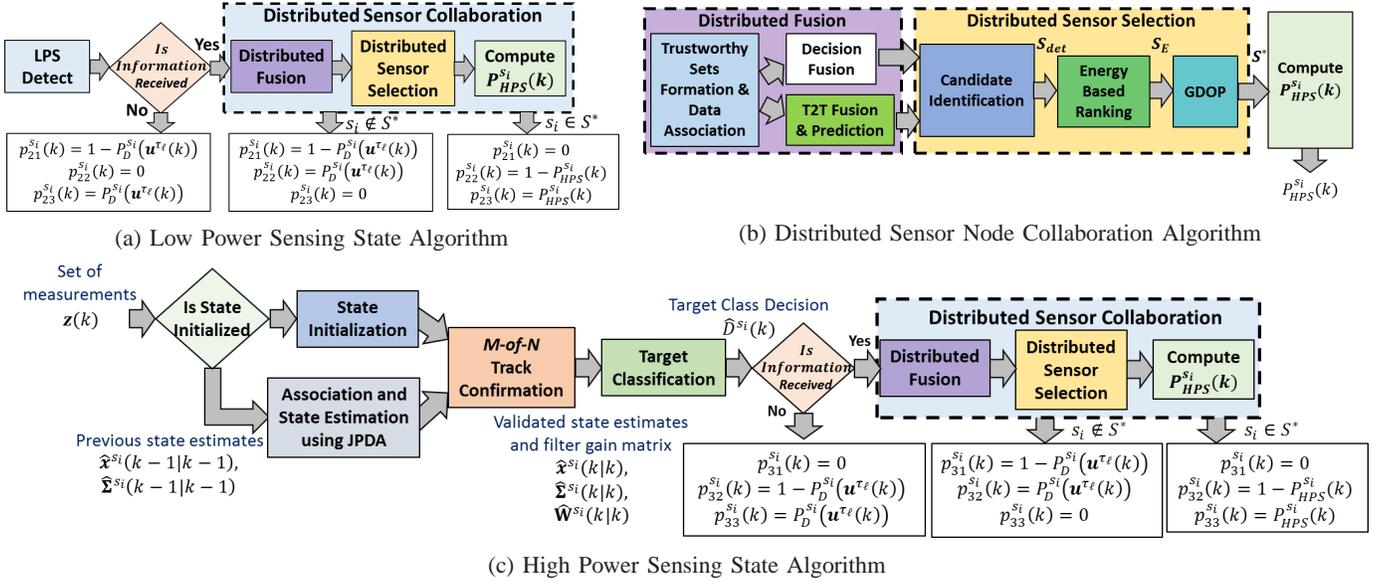


Figure 2: Flowcharts for the algorithms within each state of the PFSA

2) **Target Detection Model:** The detection model proposed in [24] is adopted in this paper as follows:

$$P_D^{s_i}(\mathbf{u}^{\tau_\ell}) = \begin{cases} \alpha & d(\tau_\ell, s_i) \leq R_{s,a} \\ \alpha e^{-\beta(d(\tau_\ell, s_i) - R_{s,a})} & R_{s,a} < d(\tau_\ell, s_i) \leq R_{s,b} \\ p_{fa} & \text{else} \end{cases}, \quad (5)$$

where $d(\tau_\ell, s_i) = \|\mathbf{u}^{\tau_\ell} - \mathbf{u}^{s_i}\|$, α and β are model parameters, $R_{s,a}$ is the reliable sensing radius, $R_{s,b}$ is the maximum sensing radius, and $p_{fa} = 1 - e^{-f\Delta T}$ is the false alarm probability [25] with a rate of f during a ΔT second scan. Note that $R_c \geq 2R_{s,b}$. For an application specific node design, a detailed sensor model can be substituted for Eq. (5).

C. Objective

The objective of the POSE.3C algorithm is to control the heterogeneous sensing states of each node in a distributed manner to achieve the following: (1) Improved network lifetime (see Defn. 5.1), (2) Improved uniformity of energy distribution in nodes around targets to prevent coverage holes, and (3) high tracking accuracy with low missed detection rates.

4. POSE.3C-ALGORITHM

As shown in Fig. 1, each sensor node is controlled by a distributed *PFSA*-based supervisor as defined below.

Definition 4.1 (PFSA). A *PFSA* is defined as a 3-tuple $\Xi = \langle \Theta, A, P \rangle$, where

- Θ is a finite set of states,
- A is a finite alphabet,
- $p: \Theta \times \Theta \rightarrow [0, 1]$ are the state transition probabilities which form a stochastic matrix $P \equiv [p_{ij}]$, where $p_{ij} \equiv p(\theta_i, \theta_j)$, $\forall \theta_i, \theta_j \in \Theta$, s.t. $\sum_{\theta' \in \Theta} p(\theta, \theta') = 1$, $\forall \theta \in \Theta$.

The alphabet is defined as $A = \{\epsilon, 0, 1\}$, where ϵ is the null symbol which is emitted when no information is available, 0 indicates no target detection, and 1 indicates target detection.

The state set Θ consists of three states: (1) Sleep (θ_1), (2) LPS (θ_2), and (3) HPS (θ_3), as shown in Fig. 1, whose operations and transitions are described below.

A. Description of PFSA States and Transition Probabilities

Consider a sensor node $s_i \in \mathcal{S}$. The operations within each state of its distributed *PFSA*-based supervisor and update of its state transition probabilities are discussed below.

1) **Sleep State:** The Sleep state θ_1 is designed to minimize the energy consumption of s_i when the target is away or if the node is not selected to track the target. It disables all sensing and communication devices on the node except the clock. After every time interval ΔT , the node can make a transition to the LPS state with a probability $p_{12}^{s_i}(k) = 1 - p_{sleep}$ or it can stay in the Sleep state with a probability $p_{11}^{s_i}(k) = p_{sleep}$, where p_{sleep} is a design parameter.

2) **LPS state:** The LPS state θ_2 is designed to conserve energy while enabling target detection. In this state, the LPS devices, the DPU, and the transceiver are powered on. Fig. 2a shows the flowchart for the algorithm within the LPS state.

• **Target Detection:** Here a target could be detected either by: (i) using the LPS devices or (ii) fusing the information received from neighbors. If a target is present, then the sensor node s_i detects it with a probability $P_D^{s_i}$ as per Eq. (5).

• **Distributed Collaboration:** Next, it checks if it has received any information from its neighbors. Let $\mathcal{N}_{HPS}^{s_i} \subseteq \mathcal{N}^{s_i}$ be the set of HPS sensors that have broadcasted the target state and class information to s_i . If information is received, i.e. $\mathcal{N}_{HPS}^{s_i} \neq \emptyset$, then the node s_i uses a collaboration algorithm (see Section 4-B for details), which fuses the received information to obtain the fused state prediction and classification

decision. The fused information is used to form a cluster of optimal nodes, \mathcal{S}^* , to track the target. If $s_i \in \mathcal{S}^*$, then it computes a probability $P_{HPS}^{s_i}(k)$ (see Eq. (16)) to transition to the HPS state. On the other hand, if no information is received, i.e. $\mathcal{N}_{HPS}^{s_i} = \emptyset$, then the node relies on its own detection probability $P_D^{s_i}(k)$ to transition to the HPS state.

• **Updating the State Transition Probabilities:** The node s_i updates its *PFSA* probabilities as follows:

- if ($\mathcal{N}_{HPS}^{s_i} \neq \emptyset$ and $s_i \in \mathcal{S}^*$), then:
 $p_{21}^{s_i}(k) = 0$; $p_{22}^{s_i}(k) = 1 - P_{HPS}^{s_i}(k)$; $p_{23}^{s_i}(k) = P_{HPS}^{s_i}(k)$,
- if ($\mathcal{N}_{HPS}^{s_i} \neq \emptyset$ and $s_i \notin \mathcal{S}^*$), then:
 $p_{21}^{s_i}(k) = 1 - P_D^{s_i}(k)$; $p_{22}^{s_i}(k) = P_D^{s_i}(k)$; $p_{23}^{s_i}(k) = 0$,
- if ($\mathcal{N}_{HPS}^{s_i} = \emptyset$), then:
 $p_{21}^{s_i}(k) = 1 - P_D^{s_i}(k)$; $p_{22}^{s_i}(k) = 0$; $p_{23}^{s_i}(k) = P_D^{s_i}(k)$.

3) **HPS state:** The HPS state θ_3 is designed to (i) estimate the target's state and class using measurements from its HPS devices and (ii) broadcast this information. Here, the HPS devices, DPU, and transceiver are all enabled. Fig. 2c shows the flowchart for the algorithm within the HPS state.

• **Data Association and State Estimation:** In the HPS state, s_i first receives a set of measurements, $\mathbf{z}(k)$, from its HPS devices. Subsequently, the previous state estimate $\hat{\mathbf{x}}^{s_i}(k-1|k-1)$, $\hat{\Sigma}^{s_i}(k-1|k-1)$ are updated using the *Joint Probabilistic Data Association* (JPDA) method [23] to generate $\hat{\mathbf{x}}^{s_i}(k|k)$, $\hat{\Sigma}^{s_i}(k|k)$. If the received measurements do not associate to the previous state estimate (e.g., when the target state is not initialized), then s_i must first perform state initialization [23].

The measurements received may contain false alarms, due to clutter, which can generate false tracks at each node. To ensure that false tracks do not propagate throughout the network, node s_i utilizes the *M-of-N Track Confirmation Logic* [26] to allow the network to be robust to false alarms. This approach ensures that M out of N consecutive measurements are associated to a target state estimate before the node confirms that it is not a false track. Furthermore, once the target track has been confirmed, the node can drop the track if M consecutive measurements do not associate to it.

• **Target Classification:** Next, s_i performs target classification to determine the target class. To keep our control algorithm general, it is assumed that the network designer has developed a classifier (similar to those reported in Section 2) for the particular application and its performance is represented by a *Confusion Matrix B*, as shown in Table I.

Table I: Confusion Matrix

		Estimated Class	
		TOI	TNOI
True Class	TOI	b_{11}	b_{12}
	TNOI	b_{21}	b_{22}

Then, node s_i classifies the target as a TOI with probability

$$\hat{P}_{TOI}^{s_i}(k) = \begin{cases} \frac{b_{11}}{b_{11}+b_{12}} & \text{Given TOI} \\ \frac{b_{21}}{b_{21}+b_{22}} & \text{Given TNOI} \end{cases}, \quad (6)$$

which leads to a class decision $\hat{D}^{s_i}(k) \in \{0, 1\}$, where 0 and 1 correspond to a TNOI and a TOI, respectively.

• **Distributed Collaboration:** Next, if a target has been detected then node s_i broadcasts its target state estimates $\hat{\mathbf{x}}^{s_i}(k|k)$ and $\hat{\Sigma}^{s_i}(k|k)$, the filter gain matrix $\hat{\mathbf{W}}^{s_i}(k)$, and the classification decision $\hat{D}^{s_i}(k)$. Since s_i is in the HPS state and has broadcasted information to its neighbors, we define $\mathcal{N}_{HPS}^{s_i} = \mathcal{N}_{HPS}^{s_i} \cup \{s_i\}$. However, if s_i has not confirmed a target track, then $\mathcal{N}_{HPS}^{s_i}$ does not include s_i . Then, if $\mathcal{N}_{HPS}^{s_i} \neq \emptyset$, it will run the collaboration algorithm (please see Section 4-B for details) in the same manner as in the LPS state. This generates a fused state and class decision, which are then used to form the cluster of optimal sensor nodes \mathcal{S}^* to track the target during the next time step. If $s_i \in \mathcal{S}^*$, then it computes $P_{HPS}^{s_i}(k)$ in Eq. (16) to stay in the HPS state.

• **Updating the State Transition Probabilities:** Finally, the *PFSA* probabilities are updated as follows:

- if ($\mathcal{N}_{HPS}^{s_i} \neq \emptyset$ and $s_i \in \mathcal{S}^*$), then:
 $p_{31}^{s_i}(k) = 0$; $p_{32}^{s_i}(k) = 1 - P_{HPS}^{s_i}(k)$; $p_{33}^{s_i}(k) = P_{HPS}^{s_i}(k)$,
- if ($\mathcal{N}_{HPS}^{s_i} \neq \emptyset$ and $s_i \notin \mathcal{S}^*$), then:
 $p_{31}^{s_i}(k) = 1 - P_D^{s_i}(k)$; $p_{32}^{s_i}(k) = P_D^{s_i}(k)$; $p_{33}^{s_i}(k) = 0$,
- if ($\mathcal{N}_{HPS}^{s_i} = \emptyset$), then:
 $p_{31}^{s_i}(k) = 0$; $p_{32}^{s_i}(k) = 1 - P_D^{s_i}(k)$; $p_{33}^{s_i}(k) = P_D^{s_i}(k)$.

Remark 4.1. The proposed approach is modular, i.e., the blocks in the flowcharts of LPS and HPS states could be furnished by appropriate methods as suited for the application.

B. Distributed Sensor Node Collaboration

The distributed sensor node collaboration consists of the distributed fusion, sensor node selection, and computation of the HPS transition probability, as described below:

1) **Distributed Fusion:** This algorithm, shown in Fig. 2b, fuses the received target state estimates and classification decisions to produce the fused state estimate and class decision.

Consider a node s_i which could be in the LPS or HPS state. The information ensemble it receives consists of

$$\hat{\mathbf{I}}^{s_i}(k) = \{(\hat{\mathbf{x}}^{s_j}, \hat{\Sigma}^{s_j}, \hat{\mathbf{W}}^{s_j}, \hat{D}^{s_j}), \forall s_j \in \mathcal{N}_{HPS}^{s_i}\}, \quad (7)$$

where $\hat{\mathbf{x}}^{s_j}(k|k)$, $\hat{\Sigma}^{s_j}(k|k)$, $\hat{\mathbf{W}}^{s_j}(k)$, and $\hat{D}^{s_j}(k)$ correspond to the state estimate, covariance, filter gain, and class decision, of the node s_j at time k .

However, due to noise and other factors, the information received must first be validated to ensure that it is accurate and reliable before processing. This is done by forming a set of trustworthy neighbors $\mathcal{N}_T^{s_i} \subseteq \mathcal{N}_{HPS}^{s_i}$ by evaluating the sum of the position error as follows

$$\mathcal{N}_T^{s_i} = \{s_j \in \mathcal{N}_{HPS}^{s_i} : \text{Trace}(\mathbf{H}(k)\hat{\Sigma}^{s_j}(k|k)\mathbf{H}(k)') \leq \xi\}, \quad (8)$$

where $\mathbf{H}(k)$ is the Jacobian of the measurement model defined in Section 3-B1 and ξ is the maximum tolerance of the estimate. In this paper, $\xi = \frac{R_{s,b}^2\sigma_\phi^2 + \sigma_R^2}{2}$, where σ_ϕ and σ_R are the standard deviations in the azimuth and range measurements

of the HPS sensor. Finally, node s_i receives the following trustworthy information:

$$\hat{\mathbf{I}}_T^{s_i}(k) = \{(\hat{\mathbf{x}}^{s_j}, \hat{\Sigma}^{s_j}, \hat{\mathbf{W}}^{s_j}, \hat{D}^{s_j}), \forall s_j \in \mathcal{N}_T^{s_i}\}, \quad (9)$$

Remark 4.2. The threshold ξ is the trace of the initialized covariance matrix, which is used to eliminate tracks associated to mismatched models or clutter.

Next, the trustworthy information is associated to ensure that it is related to the same target. In this work, the *Track-to-Track Association Method* (T2TA) [23] is used for this purpose. This method associates the trustworthy information into C distinct groups which correspond to the C different targets that could be present within the node s_i 's neighborhood; thus forming the information ensembles: $\hat{\mathbf{I}}_T^{s_i,c}(k) \subseteq \hat{\mathbf{I}}_T^{s_i}(k)$, where $c = 1, \dots, C$. Note that C may be different than the true number of targets present in the ROI. Subsequently, for each c , the state information in $\hat{\mathbf{I}}_T^{s_i,c}(k)$ is fused to form a single state ($\hat{\mathbf{x}}^{s_i,c}(k|k)$) and covariance ($\hat{\Sigma}^{s_i,c}(k|k)$) estimate, using the *Track-to-Track Fusion* (T2TF) algorithm [23].

Remark 4.3. T2TA associates targets based on their state, including position, velocity, and turning rate estimates. Thus, two targets are separated if their state differs from each other even if they are spatially co-located.

Once all the received state information is fused, node s_i computes a one-step prediction of the target's state using the Extended Kalman Filter [23]:

$$\begin{aligned} \hat{\mathbf{x}}^{s_i,c}(k+1|k) &= \mathbf{f}(\hat{\mathbf{x}}^{s_i,c}(k|k), k), \\ \hat{\Sigma}^{s_i,c}(k+1|k) &= \mathbf{F}(k)\hat{\Sigma}^{s_i,c}(k|k)\mathbf{F}(k)' + \mathbf{Q}, \end{aligned} \quad (10)$$

where $\mathbf{F}(k)$ is the Jacobian of the state transition matrix evaluated at $\hat{\mathbf{x}}^{s_i,c}(k|k)$. This predicted state is used in the distributed sensor node selection to identify the sensor nodes to track the target during next time step.

Furthermore, the associated target class decisions in $\hat{\mathbf{I}}_T^{s_i,c}(k)$ are also fused together using the majority vote rule as follows.

$$\hat{D}^{s_i,c}(k) = \begin{cases} 1 & \text{if } \frac{1}{|\hat{\mathbf{I}}_T^{s_i,c}(k)|} \sum \hat{D}^{s_j} \in \hat{\mathbf{I}}_T^{s_i,c}(k) \hat{D}^{s_j}(k) \geq 0.5 \\ 0 & \text{else} \end{cases}. \quad (11)$$

Therefore, if $\hat{D}^{s_i,c}(k) = 1$, $N_{sel} > 1$ nodes will be selected to track the target during the next time step, while if $\hat{D}^{s_i,c}(k) = 0$, $N_{sel} = 1$ nodes will be selected. The number of selected nodes around a TOI, i.e. $N_{sel} > 1$, is a design parameter.

2) **Distributed Sensor Node Selection:** The Distributed Sensor Node Selection (*DSS*) algorithm employs a multi-step filtering process performed at each node to identify the optimal nodes to track the target using their HPS devices at time $k+1$. The selected nodes are the ones with the highest remaining energy that also minimize the estimation error. The steps of the sensor node selection are described below.

Step 1: The node s_i first locates the nodes within its neighborhood that can detect a target c in the detection region

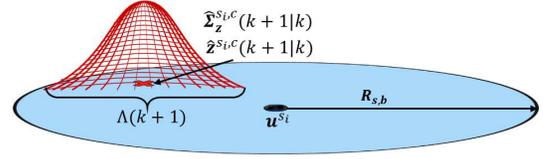


Figure 3: Computation of $P_{HPS}^{s_i}(k)$

around the target's predicted position during the next time step to form a candidate set \mathcal{S}_{det} , such that

$$\mathcal{S}_{det} = \{s_j \in (\mathcal{N}^{s_i} \cup s_i) : \|\mathbf{u}^{s_j} - \hat{\mathbf{z}}^{s_i,c}(k+1|k)\| \leq R_{s,b}\}, \quad (12)$$

where $\hat{\mathbf{z}}^{s_i,c}(k+1|k) = \mathbf{h}(\hat{\mathbf{x}}^{s_i,c}(k+1|k), k)$.

Step 2: Then, $N'_{sel} \geq N_{sel}$ nodes are filtered from \mathcal{S}_{det} that have the highest remaining energies. At this state, if node $s_i \in \mathcal{S}_{det}$, then it will broadcast its energy consumption $E^{s_i}(k)$ to its neighbors. Then, the node s_i computes the predicted remaining energy for each node in \mathcal{S}_{det} as follows:

$$E_R^{s_j} = 1 - \left(\frac{E^{s_j}(k) + E_{HPS}}{E_0} \right), \quad \forall s_j \in \mathcal{S}_{det}, \quad (13)$$

where $E^{s_j}(k)$ is given in Eq. (2); $E_{HPS} = (e_{HPS} + e_{RX} + e_{TX} + e_{DPU})\Delta T$ is the predicted energy cost of the HPS state during time $k+1$; ΔT is the time duration for which the node s_j will be in the HPS state if selected; and E_0 is the node's initial energy. Next, each node $s_j \in \mathcal{S}_{det}$ is ranked in descending order of remaining energy. Then the set $\mathcal{S}_E \subseteq \mathcal{S}_{det}$ is selected to consist of the top ranked N'_{sel} nodes.

Step 3: Finally, N_{sel} nodes are selected from the set \mathcal{S}_E , which are geometrically distributed around the target's predicted position to minimize the estimation error. This is done using the reciprocal of the *Geometric Dilution Of Precision* (GDOP) measure [19] defined as

$$\mu(\tilde{\mathcal{S}}) = \frac{\det(\mathbf{J}(\tilde{\mathcal{S}}))}{\text{trace}(\mathbf{J}(\tilde{\mathcal{S}}))}, \quad (14)$$

$$\mathbf{J}(\tilde{\mathcal{S}}) = \sum_{s_j \in \tilde{\mathcal{S}}} \frac{1}{\sigma_\phi^2 r_{s_j}^2} \begin{bmatrix} \sin^2 \phi_{s_j} & -\sin \phi_{s_j} \cos \phi_{s_j} \\ -\sin \phi_{s_j} \cos \phi_{s_j} & \cos^2 \phi_{s_j} \end{bmatrix},$$

where r_{s_j} is the range of sensor s_j to the target's predicted position; ϕ_{s_j} is the azimuth angle between s_j and the target's predicted position; and $\tilde{\mathcal{S}} \subseteq \mathcal{S}_E$, s.t. $|\tilde{\mathcal{S}}| = \min(|\mathcal{S}_E|, N_{sel})$. The optimal set $\mathcal{S}^* \subseteq \mathcal{S}_E \subseteq \mathcal{S}_{det}$ is generated as

$$\mathcal{S}^* = \underset{\tilde{\mathcal{S}} \subseteq \mathcal{S}_E}{\text{argmax}} (\mu(\tilde{\mathcal{S}})). \quad (15)$$

Note that \mathcal{S}^* is computed for each target track c .

3) **Computation of the HPS Transition Probability:** If $s_i \in \mathcal{S}^*$ for any target track, then it should transition to the HPS state to track the target during the next time step. As shown in Fig. 3, it first computes its expected probability of detecting the target based on target's predicted position. Let

$$\Lambda^{s_i,c}(k+1) = \iint_G P_D^{s_i}(x, y) N(\hat{\mathbf{z}}^{s_i,c}(k+1|k), \hat{\Sigma}^{s_i,c}(k+1|k)) dx dy,$$

where $G = \{(x, y) : \|(x, y) - \mathbf{u}^{s_i}\| \leq R_{s,b}\}$. Then the maximum probability of target detection over all tracks is

$$P_{HPS}^{s_i}(k) = \max_c \{\Lambda^{s_i,c}(k+1)\}, \quad (16)$$

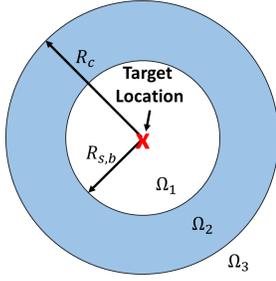


Figure 4: Illustration of regions around a single target.

Table II: Partition Regions and Related Parameters.

Ω_1	The detection region around each of the m TOIs.
Ω_2	The region in which a sensor cannot detect a target but receives state information from all broadcasting sensors within Ω_1 .
Ω_3	The region outside Ω_2 .
$A_{\Omega_1}, A_{\Omega_2}, A_{\Omega_3}$	Areas of regions Ω_1, Ω_2 , and Ω_3 , respectively.
S_1, S_2, S_3	Sets of nodes in Ω_1, Ω_2 , and Ω_3 , respectively.
S^*	Set of nodes selected in Ω_1
$S^{*'}$	Set of nodes not selected in Ω_1 , s.t. $S^{*'} = S_1 \setminus S^*$.
$\bar{E}^{S^*}, \bar{E}^{S^{*'}}, \bar{E}^{S_2}$, and \bar{E}^{S_3}	Expected energy consumption per node in the sets $S^*, S^{*'}, S_2$, and S_3 , respectively.

which it uses to transition to the HPS state as described in the computation of state transition probabilities in Section 4-A.

5. NETWORK CHARACTERISTICS

This section presents the characteristics of the POSE.3C network in terms of the expected energy consumption, network lifetime, and the missed detection probabilities.

Consider that m TOIs (or TNOIs) are present in Ω during a time interval ΔT . We partition Ω into three regions as follows:

$$\begin{aligned}
 \Omega_1 &= \bigcup_{\tau_\ell} \left\{ (x, y) : \|(x, y) - \mathbf{u}^{\tau_\ell}\| \leq R_{s,b} \right\}, \\
 \Omega_2 &= \bigcup_{\tau_\ell} \left\{ (x, y) : \|(x, y) - \mathbf{u}^{\tau_\ell}\| \leq R_c \right\} \setminus \Omega_1, \\
 \Omega_3 &= \Omega \setminus (\Omega_1 \cup \Omega_2),
 \end{aligned} \tag{17}$$

which are defined in Table II and shown in Fig. 4.

The energy consumed per node in a ΔT time interval is computed based on its state and region as follows:

$$\begin{aligned}
 E_{Sleep} &= e_{clock} \Delta T, \\
 E_{LPS}^{\Omega_1} &= (e_{LPS} + e_{TX} + e_{RX} + e_{DPV}) \Delta T, \\
 E_{HPS}^{\Omega_1} &= (e_{HPS} + e_{TX} + e_{RX} + e_{DPV}) \Delta T, \\
 E_{LPS}^{\Omega_2, \Omega_3} &= (e_{LPS} + e_{RX} + e_{DPV}) \Delta T, \text{ and} \\
 E_{HPS}^{\Omega_2, \Omega_3} &= (e_{HPS} + e_{RX} + e_{DPV}) \Delta T.
 \end{aligned}$$

These were derived from Eq. (2) where $\zeta_{TX}^{s_i}(k) = 0$ in regions Ω_2 and Ω_3 since the nodes do not detect the target and thus don't broadcast any information; however they can still receive the information from broadcasting neighbors.

Now, we present the POSE.3C network characteristics, i.e. the expected energy consumption per unit time, network lifetime, and the probability of missed detection. These are presented via Theorems 5.1, 5.2, and 5.3, respectively. The

information about these characteristics enables the design of a network with appropriate sensor density and p_{sleep} to achieve the desired lifetime and missed detection requirements.

A. Energy Consumption and Lifetime Characteristics

Theorem 5.1. *The expected energy consumption of the POSE.3C network during a ΔT time interval is given as*

$$\begin{aligned}
 \bar{E}_{\Delta T} &= N_{sel} m \bar{E}^{S^*} + (\rho A_{\Omega_1} - N_{sel} m) \bar{E}^{S^{*'}} + \\
 &\quad \rho A_{\Omega_2} \bar{E}^{S_2} + \rho A_{\Omega_3} \bar{E}^{S_3},
 \end{aligned}$$

where ρ is the sensor network density; \bar{E}^{S^*} , $\bar{E}^{S^{*'}}$, \bar{E}^{S_2} , and \bar{E}^{S_3} are defined in Table II and given as

$$\begin{aligned}
 \bar{E}^{S^*} &= E_{LPS}^{\Omega_1} p_2^{S^*} + E_{HPS}^{\Omega_1} p_3^{S^*}, \\
 \bar{E}^{S^{*'}} &= E_{Sleep} p_1^{S^{*'}} + E_{LPS}^{\Omega_1} p_2^{S^{*'}}, \\
 \bar{E}^{S_2} &= E_{Sleep} p_1^{S_2} + E_{LPS}^{\Omega_2, \Omega_3} p_2^{S_2}, \\
 \bar{E}^{S_3} &= E_{Sleep} p_1^{S_3} + E_{LPS}^{\Omega_2, \Omega_3} p_2^{S_3} + E_{HPS}^{\Omega_2, \Omega_3} p_3^{S_3},
 \end{aligned}$$

and $[p_1, p_2, p_3]$ are the steady state probabilities for the Sleep, LPS and HPS states, respectively, which for each set of nodes are given as follows:

$$\begin{aligned}
 \begin{bmatrix} p_1 \\ p_2 \\ p_3 \end{bmatrix}^{S^*} &= \begin{bmatrix} 0 \\ 1 - \alpha \\ \alpha \end{bmatrix}, \quad \begin{bmatrix} p_1 \\ p_2 \\ p_3 \end{bmatrix}^{S^{*'}} = \begin{bmatrix} \frac{1 - \alpha}{2 - p_{sleep} - \alpha} \\ \frac{1 - p_{sleep}}{2 - p_{sleep} - \alpha} \\ 0 \end{bmatrix}, \\
 \begin{bmatrix} p_1 \\ p_2 \\ p_3 \end{bmatrix}^{S_2} &= \begin{bmatrix} \frac{1 - p_{fa}}{2 - p_{sleep} - p_{fa}} \\ \frac{1 - p_{sleep}}{2 - p_{sleep} - p_{fa}} \\ 0 \end{bmatrix}, \quad \begin{bmatrix} p_1 \\ p_2 \\ p_3 \end{bmatrix}^{S_3} = \begin{bmatrix} \frac{(1 - 2p_{fa})}{2 - p_{sleep} - 2p_{fa}} \\ \frac{(1 - p_{fa})(1 - p_{sleep})}{2 - p_{sleep} - 2p_{fa}} \\ \frac{p_{fa}(1 - p_{sleep})}{2 - p_{sleep} - 2p_{fa}} \end{bmatrix}.
 \end{aligned}$$

Definition 5.1 (Network Lifetime). *Consider a path γ of length L in the region Ω that is taken by the maximum number of targets. Now consider a cylindrical tube $\Omega_\gamma \subset \Omega$ of radius $R_{s,b}$ around this path, which contains the set of nodes $S_\gamma \subset S$ that will die first in the network. The expected network lifetime, \bar{T}_{Life} , is defined as the time when the energy of the nodes within Ω_γ reduces to a certain fraction $\eta \in [0, 1)$, s.t.*

$$\frac{\sum_{s_j \in S_\gamma} (E_0^{s_j} - E^{s_j}(\bar{T}_{Life}))}{\sum_{s_j \in S_\gamma} E_0^{s_j}} = \eta.$$

Theorem 5.2. *The expected lifetime of a POSE.3C network is*

$$\bar{T}_{Life} = \frac{2\rho R_{s,b} L E_0 \Delta T (1 - \eta)}{\bar{E}_{\Delta T}}.$$

B. Missed Detection Characteristics

Definition 5.2 (Target Birth). *A target birth is the time instance when a target appears in the deployment region Ω .*

Definition 5.3 (Mature Target). *A mature target is a target that has travelled inside the region Ω for sufficient time such that node collaborations are taking place to track it.*

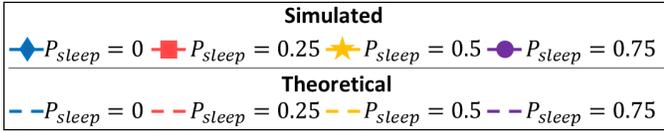
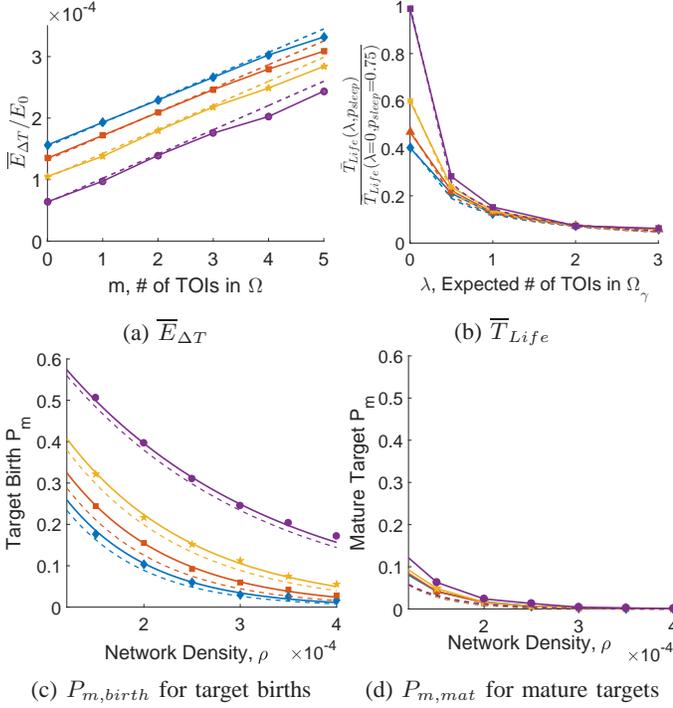


Figure 5: Validation of: (a) Theorem 5.1, (b) Theorem 5.2, (c) Theorem 5.3 Part a, and (d) Theorem 5.3 Part b.

Theorem 5.3. The missed detection probability characteristics of a POSE.3C network are given as follows:

a) For a target birth:

$$P_{m,bir} \geq \exp\left(-\frac{\pi R_{s,a}^2 \alpha \chi \rho (1 - p_{sleep})}{2 - p_{sleep} - 2p_{fa}}\right),$$

b) For a mature target:

$$P_{m,mat} \geq \exp\left(-\frac{\pi R_{s,a}^2 \alpha \chi \rho \left[(1 - p_{sleep}) + \frac{N_{sel}}{\rho \pi R_{s,b}^2} (1 - \alpha)\right]}{2 - p_{sleep} - \alpha}\right)$$

$$\text{where } \chi = 1 + \frac{2(1 + \beta R_{s,a})}{\beta^2 R_{s,a}^2} \left(1 - \frac{(1 + \beta R_{s,b}) e^{-\beta R_{s,b}}}{(1 + \beta R_{s,a}) e^{-\beta R_{s,a}}}\right).$$

C. Theorem Validations

In order to validate the theorems, the POSE.3C algorithm was simulated in a 1km x 1km deployment region. For a thorough analysis, 500 Monte-Carlo simulation runs were conducted where the distribution of sensor nodes was regenerated during each run according to a uniform distribution. Table III lists the different simulation parameters.

To validate Theorems 5.1 and 5.2, the network characteristics in terms of expected energy consumption and lifetime, were evaluated against the number of targets present and the

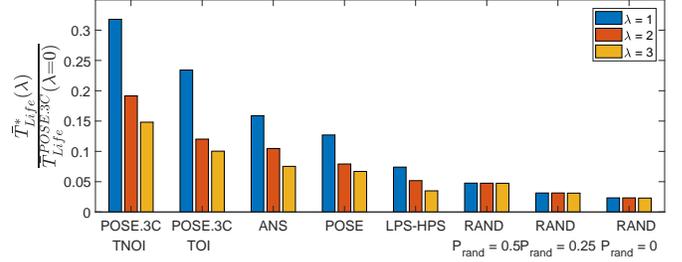


Figure 6: Network Lifetime Comparison

design parameter p_{sleep} . The network density was chosen to be $\rho = 4e^{-4} \frac{\text{nodes}}{m^2}$. For Theorem 5.1, m targets were deployed throughout Ω according to a uniform distribution. Subsequently, the average network energy consumption was acquired over the different simulation runs. Fig. 5a shows the comparison of the simulated vs theoretical results of the average energy consumption for different number of targets and for various p_{sleep} values. For Theorem 5.2, a tube $\Omega_\gamma \subset \Omega$ was considered of length $L = 600m$ and width $2R_{s,b} = 180m$. The targets were simulated with an arrival rate such that an average of λ targets are traveling through the tube during each time interval. The simulations were run until the network reached its lifetime according to Defn. 5.1. In this manner the network lifetime $\bar{T}_{Life}(\lambda, p_{sleep})$ was computed for different values of λ and p_{sleep} . The lifetime is normalized by the expected lifetime for $\lambda = 0$ and $p_{sleep} = 0.75$. Fig. 5b shows the comparison of the simulation and theoretical results for the expected lifetime. As seen in both Figs. 5a and 5b, the simulation results match the theoretical results. It is observed that as we increase p_{sleep} , the expected energy consumption decreases while the expected lifetime increases.

To validate Theorem 5.3 Part a, a random target birth was generated in the region Ω at each time instance, while to validate Part b, a moving target was generated travelling through the region according to the *DWNA* model. The probabilities of missed detection for target births and mature targets were computed over the Monte-carlo runs by counting the number of detections and misses. The simulations were repeated for various network densities and p_{sleep} values. The probabilities are plotted in Figs. 5c and 5d, which show that the simulation results match the theory. It is also seen that $P_{m,mat}$ is significantly lower than $P_{m,birth}$. This is because node collaborations allow the network to activate their LPS and HPS devices in advance to detect and track the target.

Table III: Simulation Parameters

$e_{clock} = 0.01W$	$R_{s,a} = 60m$	$b_{11} = b_{22} = 0.9$
$e_{LPS} = 2.5mW$	$R_{s,b} = 90m$	$b_{12} = b_{21} = 0.1$
$e_{HPS} = 22W$	$R_c = 180m$	$N'_{sel} = 5$
$e_{TX} = 1.26W$	$L = 600m$	$\sigma_\phi = 1^\circ$
$e_{RX} = 0.63W$	$\alpha = 0.95$	$\sigma_R = 1m$
$e_{DPU} = 1W$	$\beta = 0.00171$	$\sigma_{v,x} = \sigma_{v,y} = 1m$
$E_0 = 1.08MJ$	$\eta = 0.05$	$\sigma_{v,\psi} = 1^\circ$
$\Delta T = 0.5s$	$p_{fa} = 0.01$	$\mu_{cl} = 0.025$

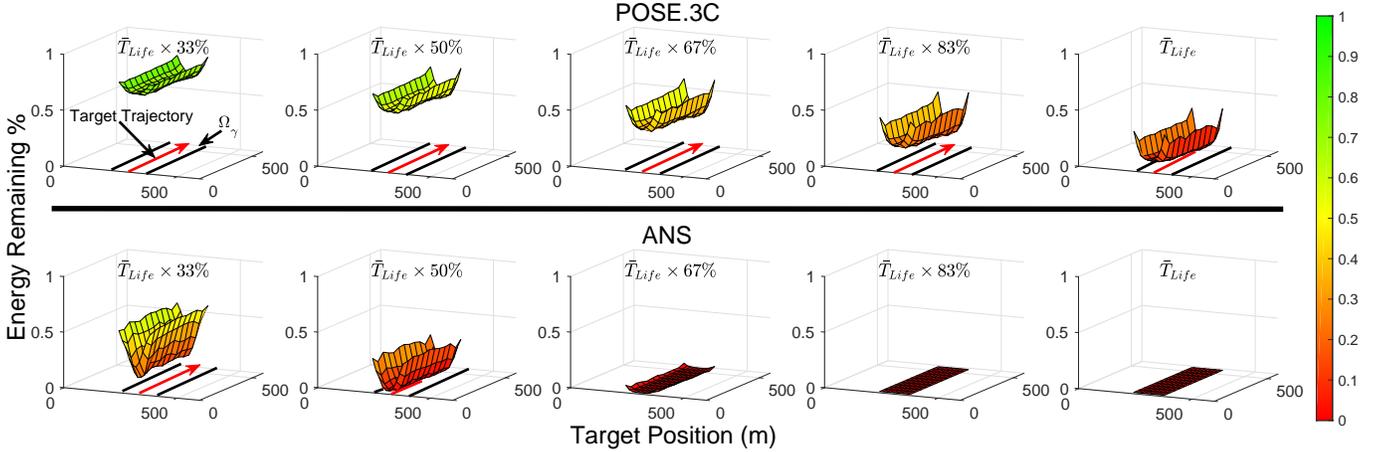


Figure 7: Distribution of Energy Remaining around a TOI

6. RESULTS AND DISCUSSION

To verify that the proposed network meets the performance requirements presented in Section 3-C, this section presents the results of the POSE.3C algorithm compared with existing methods. Specifically, POSE.3C is compared with the following distributed scheduling methods: (1) *Autonomous Node Selection* (ANS), (2) POSE (3) *LPS-HPS Scheduling*, and (4) *Random Scheduling*. The ANS algorithm [19] is a distributed sensor selection method that utilizes a cost function that minimizes the Mean Square Error (MSE) based on the GDOP. Here, the nodes collaborate in a distributed manner to make the scheduling decision; however the sensor states are only binary, i.e. passive (LPS) and active (HPS). The POSE algorithm [4] is a primitive version of the POSE.3C algorithm which does not include classification and sensor selection. The nodes in the POSE algorithm are selfish and transition to the HPS state if a target is predicted to travel within their coverage area. Due to no sensor selection, it leads to a large number of active nodes around a target resulting in significant energy wastage. Additionally, the PFSA in POSE algorithm consists of 4 states instead of 3 with an additional *Listening* state. The LPS-HPS Scheduling method is a distributed trigger-based activation method where the sensor nodes remain in the passive (LPS) state until a target is detected. Once a node detects a target, it switches to the active (HPS) state. The Random Scheduling method is a distributed method where the sensor nodes probabilistically cycle between actively sensing (HPS) and sleeping. Thus, during each time step the nodes sleep with a probability P_{rand} and sense the environment in the HPS state with a probability $1 - P_{rand}$. Thus, for $P_{rand} = 0$, the sensor nodes are always sensing and when $P_{rand} = 1$, they are always sleeping. Note that in both the LPS-HPS and Random Scheduling methods, the nodes do not collaborate.

The sensor nodes are assumed to have a hydrophone array [27] as the LPS device and an active sonar [28] as the HPS device. Table III lists the different simulation parameters including energy costs, sensing parameters, process noise parameters ($\sigma_{v,x}, \sigma_{v,y}, \sigma_{v,\psi}$), and measurement noise parameters (σ_{ϕ}, σ_R). The network density, $\rho = 4e^{-4}$, and

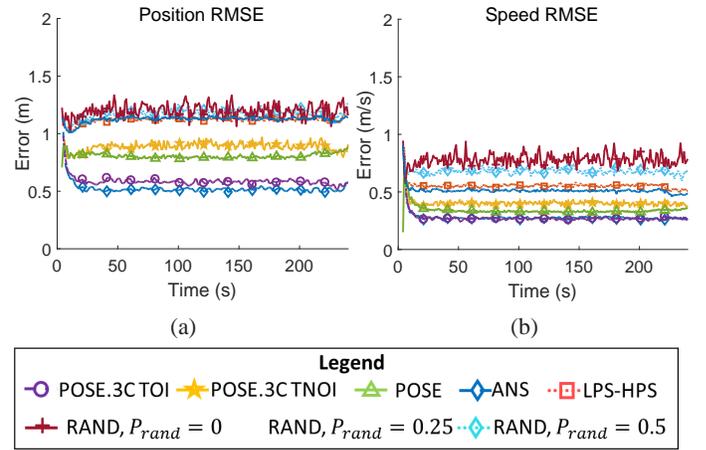


Figure 8: RMSE of (a) Position and (b) Speed Estimates

design parameter, $p_{sleep} = 0.5$, are chosen to ensure that the probability of missed detection for a mature target is less than 0.01. The number of sensor nodes selected to track a TOI is chosen to be $N_{sel} = 3$, while for a TNOI $N_{sel} = 1$.

First, the expected network lifetime of the POSE.3C and the other methods is computed by varying the expected number of targets, λ , that travel within a tube $\Omega_\gamma \in \Omega$. The results achieved are presented in Fig. 6 where each bar is normalized by the expected lifetime of the POSE.3C network for $\lambda = 0$. As seen, the expected lifetime of the POSE.3C network is higher than all of the other scheduling methods. Also, as expected, the lifetime achieved by the POSE.3C network while tracking TNOIs is higher than that of TOIs. Additionally, it can be concluded that if there is a mix of TOIs and TNOIs within the network, then the expected lifetime of the POSE.3C network will lie between the POSE.3C TNOI and TOI values.

Next, the distribution of energy remaining around the targets within a tube $\Omega_\gamma \in \Omega$ of the POSE.3C network is compared with that of the ANS algorithm. For this comparison, the expected number of targets within the tube is chosen as $\lambda = 1$. Fig. 7 shows several snapshots of the remaining energy distribution within Ω_γ until \bar{T}_{Life} of the POSE.3C is reached. As seen, the distribution of energy within the tube is much

Table IV: Computation Times of the POSE.3C Components

	JPDA	Distributed Fusion	Distributed Sensor Selection
TOI	1.4 ms	2.6 ms	0.37 ms
TNOI	1.3 ms	2.37 ms	0.36 ms

more uniform for the POSE.3C network as compared to the ANS network. This is because the ANS network depletes the energy rapidly near the track by always selecting the closest sensors to minimize the tracking error, while the POSE.3C network allows for energy based ranking while maintaining tracking accuracy. A slight increase in energy on the sides of the tube is seen due to boundary effects. Also, the ANS network dies much more rapidly as compared to POSE.3C.

Additionally, the tracking accuracy is evaluated for each network according to the Root Mean Squared Error (RMSE) and is presented in Fig. 8. As seen in the figure, the POSE.3C and ANS networks have very low tracking errors as compared to the Random and LPS-HPS methods due to distributed fusion and collaboration between neighbors. Additionally, this shows that merging the *Listening* state of the POSE algorithm with the LPS and HPS states allows the POSE.3C network to lower the RMSE error via node selection and fusion.

The complexity of the POSE.3C algorithm arises from the JPDA, Distributed Fusion, and Distributed Sensor Selection algorithms within the HPS and LPS states. The computational complexity was analyzed by measuring the average times taken for each of these processes and the results are shown in Table IV. These average times were generated in a Matlab environment on a i5 3.1 GHz CPU computer.

7. CONCLUSIONS

This paper developed the POSE.3C algorithm where a distributed PFSA-based supervisor is embedded on each node to enable/disable the heterogeneous devices on the node for energy-efficient target tracking. The algorithm relies on a 3C network autonomy approach where targets of interest are tracked by dynamic clusters of the most reliable and optimal nodes. Theoretical characteristics of the POSE.3C network have been established and validated in comparison with existing methods to show that the POSE.3C network significantly increases the network lifetime while providing high tracking accuracy and a low probability of missed detection.

REFERENCES

- [1] K. Mukherjee, S. Gupta, A. Ray, and T. A. Wettergren, "Statistical-mechanics-inspired optimization of sensor field configuration for detection of mobile targets," *IEEE Transactions on Systems, Man, and Cybernetics, Part B*, vol. 41, no. 3, pp. 783–791, 2011.
- [2] J. Hare, S. Gupta, and J. Wilson, "Decentralized smart sensor scheduling for multiple target tracking for border surveillance," in *IEEE International Conference on Robotics and Automation*, 2015, pp. 3265–3270.
- [3] J. Lin, W. Xiao, F. L. Lewis, and L. Xie, "Energy-efficient distributed adaptive multisensor scheduling for target tracking in wireless sensor networks," *IEEE Transactions on Instrumentation and Measurement*, vol. 58, no. 6, pp. 1886–1896, 2009.
- [4] J. Z. Hare, S. Gupta, and T. A. Wettergren, "Pose: Prediction-based opportunistic sensing for energy efficiency in sensor networks using distributed supervisors," *IEEE Transactions on Cybernetics*, vol. 48, no. 7, pp. 2114–2127, 2018.
- [5] D. Bajovic, B. Sinopoli, and J. Xavier, "Sensor selection for event detection in wireless sensor networks," *IEEE Transactions on Signal Processing*, vol. 59, no. 10, pp. 4938–4953, 2011.
- [6] J. Z. Hare, S. Gupta, J. Song, and T. A. Wettergren, "Classification induced distributed sensor scheduling for energy-efficiency in underwater target tracking sensor networks," in *OCEANS–Anchorage, 2017*. IEEE, 2017, pp. 1–7.
- [7] M. F. Duarte and Y. H. Hu, "Vehicle classification in distributed sensor networks," *Journal of Parallel and Distributed Computing*, vol. 64, no. 7, pp. 826–838, 2004.
- [8] X. Jin, S. Sarkar, A. Ray, S. Gupta, and T. Damarla, "Target detection and classification using seismic and pir sensors," *IEEE Sensors Journal*, vol. 12, no. 6, pp. 1709–1718, 2012.
- [9] N. Dziengel, M. Seiffert, M. Ziegert, S. Adler, S. Pfeiffer, and J. Schiller, "Deployment and evaluation of a fully applicable distributed event detection system in wireless sensor networks," *Ad Hoc Networks*, vol. 37, no. 2, pp. 160–182, 2016.
- [10] K. Mukherjee, S. Gupta, A. Ray, and S. Phoha, "Symbolic analysis of sonar data for underwater target detection," *IEEE Journal of Oceanic Engineering*, vol. 36, no. 2, pp. 219–230, 2011.
- [11] T. Vercauteren, D. Guo, and X. Wang, "Joint multiple target tracking and classification in collaborative sensor networks," *IEEE Journal on selected areas in communications*, vol. 23, no. 4, pp. 714–723, 2005.
- [12] D. Angelova and L. Mihaylova, "Joint target tracking and classification with particle filtering and mixture kalman filtering using kinematic radar information," *Digital Signal Processing*, vol. 16, no. 2, pp. 180–204, 2006.
- [13] S. Aeron, V. Saligrama, and D. A. Castanon, "Efficient sensor management policies for distributed target tracking in multihop sensor networks," *IEEE Transactions on Signal Processing*, vol. 56, no. 6, pp. 2562–2574, 2008.
- [14] O. Younis and S. Fahmy, "Heed: a hybrid, energy-efficient, distributed clustering approach for ad hoc sensor networks," *IEEE Transactions on Mobile Computing*, vol. 3, no. 4, pp. 366–379, 2004.
- [15] W. Li and W. Zhang, "Sensor selection for improving accuracy of target localisation in wireless visual sensor networks," *IET Wireless Sensor Systems*, vol. 2, no. 4, pp. 293–301, 2012.
- [16] L. M. Kaplan, "Global node selection for localization in a distributed sensor network," *IEEE Transactions on Aerospace and Electronic systems*, vol. 42, no. 1, pp. 113–135, 2006.
- [17] Y.-C. Chen and C.-Y. Wen, "Decentralized cooperative toa/aoa target tracking for hierarchical wireless sensor networks," *Sensors*, vol. 12, no. 11, pp. 15 308–15 337, 2012.
- [18] E. Ertin, J. W. Fisher, and L. C. Potter, "Maximum mutual information principle for dynamic sensor query problems," in *Information Processing in Sensor Networks*. Springer, 2003, pp. 405–416.
- [19] L. M. Kaplan, "Local node selection for localization in a distributed sensor network," *IEEE Transactions on Aerospace and Electronic Systems*, vol. 42, no. 1, pp. 136–146, 2006.
- [20] M. Zoghi and M. Kahaei, "Adaptive sensor selection in wireless sensor networks for target tracking," *IET Signal Processing*, vol. 4, no. 5, pp. 530–536, 2010.
- [21] F. Ghassemi and V. Krishnamurthy, "Decentralized node selection for localization in wireless unattended ground sensor networks," in *International Conference on Sensor Technologies and Applications*, 2008, pp. 294–299.
- [22] J. Chen, K. Cao, K. Li, and Y. Sun, "Distributed sensor activation algorithm for target tracking with binary sensor networks," *Cluster Computing*, vol. 14, no. 1, pp. 55–64, 2011.
- [23] Y. Bar-Shalom, P. K. Willett, and X. Tian, *Tracking and data fusion: A Handbook of Algorithms*. YBS Publishing, 2011.
- [24] Y. Zou and K. Chakrabarty, "Sensor deployment and target localization in distributed sensor networks," *ACM Transactions on Embedded Computing Systems*, vol. 3, no. 1, pp. 61–91, 2004.
- [25] T. A. Wettergren, "Performance of search via track-before-detect for distributed sensor networks," *IEEE Transactions on Aerospace and Electronic systems*, vol. 44, no. 1, pp. 314–325, 2008.
- [26] S. Coraluppi and C. Carthel, "Distributed tracking in multistatic sonar," *IEEE Transactions on Aerospace and Electronic Systems*, vol. 41, no. 3, pp. 1138–1147, 2005.
- [27] "Hti hydrophone <http://www.hightechincusa.com/products/hydrophones/>."
- [28] "M3 sonar <https://www.km.kongsberg.com/>."

APPENDIX: THEOREM PROOFS

Proof of Theorem 5.1

The state transition probability matrices for nodes in \mathcal{S}^* , $\mathcal{S}^{*'}$, \mathcal{S}_2 , and \mathcal{S}_3 are given as

$$\begin{aligned} P^{\mathcal{S}^*} &\triangleq \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1-\alpha & \alpha \\ 0 & 1-\alpha & \alpha \end{bmatrix}, \\ P^{\mathcal{S}^{*'}} &\triangleq \begin{bmatrix} p_{sleep} & 1-p_{sleep} & 0 \\ 1-\alpha & \alpha & 0 \\ 1-\alpha & \alpha & 0 \end{bmatrix}, \\ P^{\mathcal{S}_2} &\triangleq \begin{bmatrix} p_{sleep} & 1-p_{sleep} & 0 \\ 1-p_{fa} & p_{fa} & 0 \\ 1-p_{fa} & p_{fa} & 0 \end{bmatrix}, \\ P^{\mathcal{S}_3} &\triangleq \begin{bmatrix} p_{sleep} & 1-p_{sleep} & 0 \\ 1-p_{fa} & 0 & p_{fa} \\ 0 & 1-p_{fa} & p_{fa} \end{bmatrix}. \end{aligned} \quad (18)$$

Note that α is chosen as the best case conservative estimate of P_{HPS}^{si} and P_D^{si} in $P^{\mathcal{S}^*}$ and $P^{\mathcal{S}^{*'}}$ respectively. Similarly, $P_D^{si} = p_{fa}$ in $P^{\mathcal{S}_2}$, while $P^{\mathcal{S}_3}$ is constructed for $\mathcal{N}_{HPS}^{si} = \emptyset$.

Based on the above operators, the steady-state probabilities of each state within each region are computed as follows

$$\begin{aligned} [p_1, p_2, p_3] &= [p_1, p_2, p_3] P, \\ \text{s.t. } p_1 + p_2 + p_3 &= 1. \end{aligned} \quad (19)$$

For $P^{\mathcal{S}^*}$:

$$\begin{aligned} p_1^{\mathcal{S}^*} &= 0, \text{ since a sleeping node cant be selected,} \\ p_2^{\mathcal{S}^*} &= (1-\alpha)(p_2^{\mathcal{S}^*} + p_3^{\mathcal{S}^*}), \\ p_3^{\mathcal{S}^*} &= \alpha(p_2^{\mathcal{S}^*} + p_3^{\mathcal{S}^*}), \quad p_3^{\mathcal{S}^*} = 1 - p_2^{\mathcal{S}^*}, \\ \Rightarrow p_2^{\mathcal{S}^*} &= 1 - \alpha, \quad p_3^{\mathcal{S}^*} = \alpha. \end{aligned} \quad (20)$$

For $P^{\mathcal{S}^{*'}}$:

$$\begin{aligned} p_1^{\mathcal{S}^{*'}} &= p_{sleep}p_1^{\mathcal{S}^{*'}} + (1-\alpha)(p_2^{\mathcal{S}^{*'}} + p_3^{\mathcal{S}^{*'}}), \\ p_2^{\mathcal{S}^{*'}} &= (1-p_{sleep})p_1^{\mathcal{S}^{*'}} + \alpha(p_2^{\mathcal{S}^{*'}} + p_3^{\mathcal{S}^{*'}}), \\ p_3^{\mathcal{S}^{*'}} &= 0, \quad p_1^{\mathcal{S}^{*'}} = 1 - p_2^{\mathcal{S}^{*'}}, \\ \Rightarrow p_1^{\mathcal{S}^{*'}} &= \frac{1-\alpha}{2-p_{sleep}-\alpha}, \text{ and} \\ p_2^{\mathcal{S}^{*'}} &= \frac{1-p_{sleep}}{2-p_{sleep}-\alpha}. \end{aligned} \quad (21)$$

For $P^{\mathcal{S}_2}$: By replacing α with p_{fa} in Eq. (21), we find the steady-state probabilities as $p_1^{\mathcal{S}_2} = \frac{1-p_{fa}}{2-p_{sleep}-p_{fa}}$, $p_2^{\mathcal{S}_2} = \frac{1-p_{sleep}}{2-p_{sleep}-p_{fa}}$, and $p_3^{\mathcal{S}_2} = 0$.

For $P^{\mathcal{S}_3}$:

$$\begin{aligned} p_1^{\mathcal{S}_3} &= p_{sleep}p_1^{\mathcal{S}_3} + (1-p_{fa})p_2^{\mathcal{S}_3}, \\ p_2^{\mathcal{S}_3} &= (1-p_{sleep})p_1^{\mathcal{S}_3} + (1-p_{fa})p_3^{\mathcal{S}_3}, \\ p_3^{\mathcal{S}_3} &= p_{fa}(p_2^{\mathcal{S}_3} + p_3^{\mathcal{S}_3}), \quad p_1^{\mathcal{S}_3} = 1 - p_2^{\mathcal{S}_3} - \frac{p_{fa}p_2^{\mathcal{S}_3}}{1-p_{fa}}, \end{aligned}$$

Solving the above and assuming $p_{fa}^2 \ll 1$, we get

$$\begin{aligned} p_1^{\mathcal{S}_3} &= \frac{1-2p_{fa}}{2-p_{sleep}-2p_{fa}}, \\ p_2^{\mathcal{S}_3} &= \frac{(1-p_{fa})(1-p_{sleep})}{2-p_{sleep}-2p_{fa}}, \\ p_3^{\mathcal{S}_3} &= \frac{p_{fa}(1-p_{sleep})}{2-p_{sleep}-2p_{fa}}. \end{aligned} \quad (22)$$

Next, we compute the expected number of sensor nodes in each state per region. Since $|\mathcal{S}^*| = N_{sel}m$ nodes are selected around m targets in region Ω_1 we get:

$$\begin{aligned} \bar{N}_{Sleep}^{\Omega_1} &= (\rho A_{\Omega_1} - N_{sel}m)p_1^{\mathcal{S}^{*'}}; \quad \bar{N}_{Sleep}^{\Omega_2} = \rho A_{\Omega_2}p_1^{\mathcal{S}_2}; \\ \bar{N}_{Sleep}^{\Omega_3} &= \rho A_{\Omega_3}p_1^{\mathcal{S}_3}; \quad \bar{N}_{LPS}^{\Omega_1} = N_{sel}mp_2^{\mathcal{S}^*} + (\rho A_{\Omega_1} - N_{sel}m)p_2^{\mathcal{S}^{*'}}; \\ \bar{N}_{LPS}^{\Omega_2} &= \rho A_{\Omega_2}p_2^{\mathcal{S}_2}; \quad \bar{N}_{LPS}^{\Omega_3} = \rho A_{\Omega_3}p_2^{\mathcal{S}_3}; \end{aligned}$$

$$\bar{N}_{HPS}^{\Omega_1} = N_{sel}mp_3^{\mathcal{S}^*}; \quad \bar{N}_{HPS}^{\Omega_2} = 0; \quad \bar{N}_{HPS}^{\Omega_3} = \rho A_{\Omega_3}p_3^{\mathcal{S}_3}. \quad (23)$$

Now, the average energy consumed by the network is

$$\begin{aligned} \bar{E}_{\Delta T} &= E_{Sleep}\bar{N}_{Sleep}^{\Omega_1} + E_{LPS}^{\Omega_1}\bar{N}_{LPS}^{\Omega_1} + E_{HPS}^{\Omega_1}\bar{N}_{HPS}^{\Omega_1} + \\ &E_{Sleep}\bar{N}_{Sleep}^{\Omega_2} + E_{LPS}^{\Omega_2,\Omega_3}\bar{N}_{LPS}^{\Omega_2} + E_{HPS}^{\Omega_2,\Omega_3}\bar{N}_{HPS}^{\Omega_2} + \\ &E_{Sleep}\bar{N}_{Sleep}^{\Omega_3} + E_{LPS}^{\Omega_2,\Omega_3}\bar{N}_{LPS}^{\Omega_3} + E_{HPS}^{\Omega_2,\Omega_3}\bar{N}_{HPS}^{\Omega_3}. \end{aligned} \quad (24)$$

Thus, putting the number of nodes derived above into Eq. (24), we obtain the result of the theorem.

Proof of Theorem 5.2

Let the number of TOIs or (TNOIs) traveling through Ω_γ per ΔT time interval be represented by a Poisson process given by

$$P(N_\gamma = m_\gamma) = e^{-\lambda\Delta T} \frac{(\lambda\Delta T)^{m_\gamma}}{m_\gamma!}. \quad (25)$$

Then, the average number of targets in Ω_γ is $\bar{N}_\gamma = \lambda$. Let the average velocity of a target be \bar{V} . Also, let L be the tube length which the target travels. Then the expected number of time intervals that a target spends in Ω_γ is $\bar{T} = L/(\bar{V}\Delta T)$.

Let \bar{T}_{Life} be the expected life of the network. Now, consider multiple TOIs (or TNOIs) travelling in the tube Ω_γ . These targets lead to a partition of the tube into three regions, $\Omega_{1\gamma}$, $\Omega_{2\gamma}$, and $\Omega_{3\gamma}$, as shown in Fig. 9. Note that these regions are similar in characteristics to the regions Ω_1 , Ω_2 , and Ω_3 , respectively, except that these are defined within the tube Ω_γ . To be conservative, we assume that the targets have disjoint detection and communication regions. Therefore, the expected energy consumption of the sensor nodes in ΔT time interval is as given in Theorem 5.1, where the areas therein are replaced by the corresponding areas $A^{\Omega_{1\gamma}}$, $A^{\Omega_{2\gamma}}$, and $A^{\Omega_{3\gamma}}$. Here, $A^{\Omega_{1\gamma}} = \lambda\pi R_{s,b}^2$; $A^{\Omega_{2\gamma}} = \lambda \left(\pi(R_c^2 - R_{s,b}^2) - R_c^2(\delta - \sin\delta) \right)$, is the sector area as shown in Fig. 9, where $\delta = \left(\pi - 2\tan^{-1} \left(\frac{R_{s,b}}{\sqrt{R_c^2 - R_{s,b}^2}} \right) \right)$;

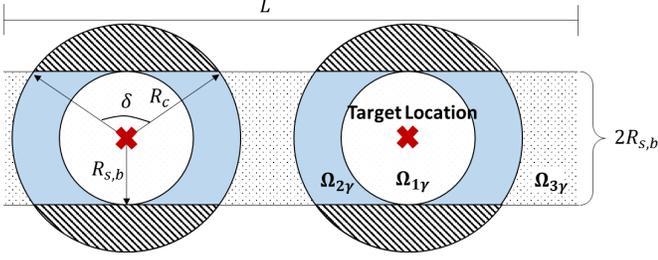


Figure 9: Illustration of the tube Ω_γ with 2 targets.

and $A^{\Omega_{3\gamma}} = 2LR_{s,b} - (A^{\Omega_{1\gamma}} + A^{\Omega_{2\gamma}})$. Now, for the overall lifetime, the total energy consumed is given as $\bar{E}_{\Delta T} \frac{\bar{T}_{Life}}{\Delta T}$. Using Defn. 5.1, we can then solve for \bar{T}_{Life} :

$$\begin{aligned} \frac{\sum_{s_j \in \mathcal{S}_\gamma} (E_0^{s_j} - E^{s_j}(\bar{T}_{Life}))}{\sum_{s_j \in \mathcal{S}_\gamma} E_0^{s_j}} &= \eta, \\ \Rightarrow 1 - \frac{\bar{E}_{\Delta T} \bar{T}_{Life}}{\Delta T \rho 2R_{s,b} L E_0} &= \eta, \\ \Rightarrow \bar{T}_{Life} &= \frac{2\rho R_{s,b} L E_0 \Delta T (1 - \eta)}{\bar{E}_{\Delta T}}. \end{aligned}$$

Proof of Theorem 5.3

The probability q that an individual sensor node s_i sampled from the deployment region Ω detects the target, is represented by a spatial Poisson process [25], such that

$$q = Pr \{ Det_i = 1 \} = 1 - e^{-\varphi(p_2 + p_3)}, \quad (26)$$

where $Det_i = 1$ denotes a detection, and φ is the coverage factor for the sensor which is computed as:

$$\varphi = \frac{1}{A_\Omega} \left(\int_0^{R_{s,a}} 2\pi r \alpha dr + \int_{R_{s,a}}^{R_{s,b}} 2\pi r \alpha e^{-\beta(r - R_{s,a})} dr \right). \quad (27)$$

Using integration by parts and simplifying we get:

$$\varphi = \frac{\pi R_{s,a}^2 \alpha}{A_\Omega} \left[1 + \frac{2(1 + \beta R_{s,a})}{\beta^2 R_{s,a}^2} \left(1 - \frac{(1 + \beta R_{s,b}) e^{-\beta R_{s,b}}}{(1 + \beta R_{s,a}) e^{-\beta R_{s,a}}} \right) \right]. \quad (28)$$

Since the term within the square brackets in Eq. (28) is equal to χ , $\varphi = \frac{\pi R_{s,a}^2 \alpha \chi}{A_\Omega}$. Here $p_2 + p_3$ is the probability that the node is in the LPS or HPS state, and thus capable of detecting the target. Since each sensor is statistically independent and identical, the probability of exactly κ sensor detections [25] is given by Bernoulli trials as follows

$$Pr \left\{ \sum Det_i = \kappa \right\} = \binom{n}{\kappa} q^\kappa (1 - q)^{n - \kappa}. \quad (29)$$

Then the probability of missed detections is given as

$$P_m = Pr \left\{ \sum Det_i = 0 \right\} = \binom{n}{0} q^0 (1 - q)^n = e^{-\varphi(p_2 + p_3)n}. \quad (30)$$

a) For a target birth: Since a target has just taken birth, the entire deployment region follows the state transition probabilities corresponding to the matrix P^{S_3} , as given in Eq. (18). Also, $n = \rho A_\Omega$. Substituting for φ , n , and $p_2^{S_3} + p_3^{S_3}$ in Eq. (30), the lower bound on P_m for a target birth is given as

$$P_{m,bir} \geq \exp \left(-\frac{\pi R_{s,a}^2 \alpha \chi \rho (1 - p_{sleep})}{2 - p_{sleep} - 2p_{fa}} \right).$$

b) For a mature target: Since the target is mature, node collaborations are taking place to select the optimal nodes for target tracking. Thus, the chosen node could be a selected node or a not selected node. Therefore, it follows the state transition probabilities corresponding to the matrix P^{S^*} or $P^{S^{*'}}$, respectively. Thus the probability $p_2 + p_3$ to find a node in the LPS or HPS state is given as

$$\begin{aligned} p_2 + p_3 &= \frac{N_{sel}}{\rho \pi R_{s,b}^2} (p_2^{S^*} + p_3^{S^*}) + \frac{\rho \pi R_{s,b}^2 - N_{sel}}{\rho \pi R_{s,b}^2} (p_2^{S^{*'}} + p_3^{S^{*'}}), \\ \Rightarrow p_2 + p_3 &= \frac{N_{sel}}{\rho \pi R_{s,b}^2} + \frac{\rho \pi R_{s,b}^2 - N_{sel}}{\rho \pi R_{s,b}^2} \left(\frac{1 - p_{sleep}}{2 - p_{sleep} - \alpha} \right), \\ \Rightarrow p_2 + p_3 &= \frac{N_{sel}(1 - \alpha) + \rho \pi R_{s,b}^2 (1 - p_{sleep})}{\rho \pi R_{s,b}^2 (2 - p_{sleep} - \alpha)}. \quad (31) \end{aligned}$$

Substituting for φ , n , and $p_2 + p_3$ in Eq. (30), the lower bound on P_m for a mature target is given as

$$P_{m,mat} \geq \exp \left(-\frac{\pi R_{s,a}^2 \alpha \chi \rho \left[(1 - p_{sleep}) + \frac{N_{sel}}{\rho \pi R_{s,b}^2} (1 - \alpha) \right]}{2 - p_{sleep} - \alpha} \right).$$

A. Comparison with POSE

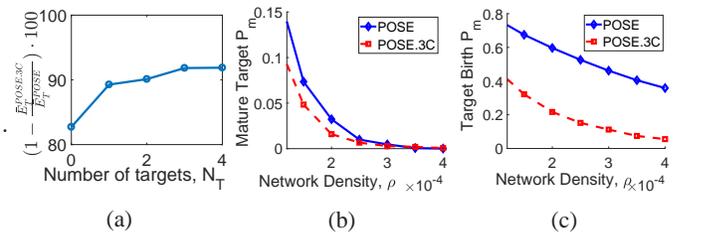


Figure 10: Comparison of POSE.3C vs. POSE, (a) Avg. % energy savings, (b) Mature P_m , and (c) Birth P_m .

While Figs. 6 and Fig. 8 show the comparison of lifetime and estimation errors with POSE, Figs. 10a, 10b, and 10c show a comparison of energy savings and missed detection characteristics. As seen in Fig. 10a, the average percent energy savings of the POSE.3C network over POSE is $\sim 90\%$. This is because the POSE.3C network is minimizing redundant HPS nodes via sensor selection. Additionally, the missed detection characteristics in Figures 10b and 10c show that merging the *Listening* state from the POSE algorithm into the LPS and HPS states improves the network's detection capabilities.



James Zachary Hare received the B.S. degree in Electrical Engineering in 2012, the M.S. degree in Electrical Engineering in 2016, and the Ph.D. degree in Electrical Engineering in 2018, all from the University of Connecticut Storrs, CT, USA. He is currently a Postdoctoral Fellow at the Army Research Laboratory, Adelphi, MD, USA. His research interests include distributed autonomy, intelligent network control systems, non-Bayesian social learning theory, modeling uncertainty, information fusion, and data analytics.



Shalabh Gupta received the B.E. degree in mechanical engineering from IIT Roorkee, India, in 2001. He received the M.S. degrees in mechanical and electrical engineering, and the Ph.D. degree in mechanical engineering from the Pennsylvania State University, University Park, PA, USA, in 2004, 2005, and 2006, respectively. He is currently an Associate Professor at the Department of Electrical and Computer Engineering, University of Connecticut. His current research interests include distributed autonomy, cyber-physical systems, robotics, motion planning, network intelligence, system resilience, data analytics, information fusion, and fault diagnostics in complex interconnected systems. Dr. Gupta is currently serving as the Chief Editor of *Frontiers in Robotics and AI* (Specialty Sensor Fusion and Machine Intelligence) and an Associate Editor of *Structural Health Monitoring: An International Journal*. Dr. Gupta has published around 100 peer-reviewed journal and conference papers. He is a member of the IEEE, IEEE Systems, Man and Cybernetics Society, and ASME.



Thomas A. Wettergren (SM06) received the B.S. degree in electrical engineering and the Ph.D. degree in applied mathematics from Rensselaer Polytechnic Institute, Troy, NY, USA. He works as a Senior Research Scientist at the Naval Undersea Warfare Center, Newport, RI, USA, where he currently serves as the US Navy Senior Technologist (ST) for Operational and Information Science. His personal research interests are in the planning and control of distributed systems, applied optimization, multiagent systems, and search theory. His professional awards include the IEEE-USA Harry Diamond Award, the NAVSEA Scientist of the Year, and the Dolores Etter Award for Top Navy Scientists and Engineers. Dr. Wettergren is a senior member of the IEEE, and a member of the Society for Industrial and Applied Mathematics and the Institute for Operations Research and the Management Sciences.