Resource Management for Wireless Networks of Bearings-Only Sensors

A Thesis
Presented to
The Academic Faculty

by

Qiang Le

In Partial Fulfillment
of the Requirements for the Degree
Doctor of Philosophy

School of Electrical and Computer Engineering
Georgia Institute of Technology
May 2006

Resource Management for Wireless Networks of Bearings-Only Sensors

Approved by:

Dr. James H. McClellan, Advisor
School of Electrical and Computer Engineering
Georgia Institute of Technology

Dr. Lance M. Kaplan
School of Electrical and Computer Engineering
Georgia Institute of Technology

Dr. Chin-Hui Lee
School of Electrical and Computer Engineering
Georgia Institute of Technology

Dr. Hayriye Ayhan
School of Industrial and Systems Engineering
Georgia Institute of Technology

Dr. Jennifer E. Michaels
School of Electrical and Computer Engineering
Georgia Institute of Technology

Date Approved: March 24, 2006

To my parents,

to my husband,

to those,

who are always there to support me

ACKNOWLEDGEMENTS

Many People have touched my life not only from an academic point of view, but both friendship and spiritual support have been a source of encouragement and strength to complete my thesis work.

I would like to express my sincere gratitude and appreciation to Dr. James H. McClellan, my teacher and dissertation advisor, for your encouragement and support throughout the research period. Your patience and kindness with your profound knowledge about this research topic has guided me along the way. I still remember the day when I made my first presentation in our group meeting. Although I was shy and shaky, seeing your encouragement in the face made my fear go away. Occasionally when I was asked who my advisor was in GaTech, I always spoke loudly and proudly announced that how lucky I am for being one of your students. I would also like to thank Dr. Lance M. Kaplan, my co-advisor, for your invaluable guidance and insightful comments on this research work. Your generous and consistent support not just in finance but also in spirit, encouragement, patience, and helpful suggestion through all these years have left a deep impression on my life. Although our weekly talk did push me to work like bees, I love listening to your advice and enjoyed every talk we had. I am really honored and grateful to have had the opportunity to work for you. I am so lucky to have you as my mentors.

I would also like to thank other committee members: Dr. Chin-Hui Lee, whom I always remember as a knowledgable and diligent professor, Dr. Jennifer E Michaels for your advice in my proposal and Dr. Hayriye Ayhan, whom I first met in ISYE department building and who generously considered a possibility of being one of committee members. Thank you all.

I am grateful to all CSIP faculty and staff: Christy Ellis, Keith May, Sam Smith, etc., all CTSPS faculty and staff from Clark Atlanta University: Katrina Barnum, and Dr. Carlos Handy for giving me advice on communication skills, etc., and Dr. Marvin N. Cohen from GTRI. Thank you for your administrative work.

Without other colleagues' help, I could not go this far in my thesis work. Volkan Cevher. I could always count on you for giving me advice on my research. Mubashir Alam and Rajbabu Velmurugan. I used to waste your guys time in chatting with you about our daily life, career, and future. Sam Lee, Ali Cafer Gurbuz, Milind Borkar, Sevgi Zbeyde ERTAN , Ryan Hersey, Yeongseon Lee and Faisal Shah. Thank you for your friendship and good luck on your research.

Finally, I would like to take a chance to thank my parents. Being the only child in my family, I wish I did not disappoint you when I was not there for you. Yi Liu, my husband and my best friend. Thank you for always being there for me.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

SUMMARY

The thesis focuses on resource management or sensor allocation when we use bearings-only measurements to track targets in an unattended ground sensor (UGS) network. Intelligent resource management is necessary because each UGS sensor node has limited power and it is desirable that estimation performance not degrade very much when only a few nodes are active to maximize the effective tracking lifetime. For scheduling to prolong the tracking lifetime, a new energy-based (EB) metric is proposed to model the number of snapshots remaining for a hypothesized node set, i.e., the remaining battery energy divided by the energy to sense and share information amongst the node set. Unlike other methods that use the total energy consumed for the given snapshot as the energy-based metric, the new EB metric can achieve load balancing of the nodes without resorting to computationally demanding non-myopic optimization. The metrics to choose nodes at a given snapshot could be geometry-based (GB) to minimize the estimation error, EB, or multiobjective. In determining the active set, each node only knows the existence of itself, the active set of nodes from the previous snapshot and the node's neighbors, i.e., the set of nodes within a distance of r_{nei}. When measuring the tracking lifetime of the system, we propose an adaptive transmission range control, known as the knowledge pool (KP) where the transmission range is determined by the knowledge of the network and the currently remaining battery level. The KP saves more energy usage than another adaptive transmission range control bounded with the GB metric when the global location information is available. We also provide practical search algorithms to optimize a constraint metric (multiobjective function) using one metric as the

optimization metric under the constraint of the other. We also demonstrate the resource management schemes for multitarget tracking with the field data.

CHAPTER I

INTRODUCTION

1.1 *Objective and contributions of this work*

Unattended ground sensor (UGS) networks that consist of a dense set of sensor nodes randomly distributed in the battlefield have drawn interest from the military. Such networks promise to provide a low power and high performance solution for surveillance. A UGS node consists of one or more sensors, e.g., microphones, to collect measurements, a battery to supply power, a computer to process the sensor data, and a radio to share information with other nodes (see Fig. 1.1). In theory, the best localization performance is achieved when all nodes share their raw sensor data. In practice, each node is constrained to operate using the finite amount of energy supplied by its battery since recharging the battery in hostile places is impractical. To maximize the effective lifetime of the network, it is desirable for only a small subset of nodes to actively track and classify targets. This will limit the energy burned by the nodes for sharing and processing. Thus, intelligent node resource management is necessary so that target localization performance does not degrade very much when only a few nodes are active.

The node selection algorithm embedded in the resource manager to determine which set of nodes should be active can be either distributed and performed at each node, or global and performed only at a central unit in a centralized system. In a distributed architecture, each node is also capable of implementing a tracking filter to extract useful information out of the locally obtained measurements, broadcasting the state, and at the same time integrating its results into a global state. A distributed system differs from a centralized system in that a node has some degree of freedom to process its measurements and make an active/inactive decision [69]. In a centralized system, decisions are made at a central unit that is assumed to be the most powerful, i.e., power is not an issue for a central unit. Subordinate nodes constrained by limited power follow the central unit's commands and take actions. In the distributed UGS system, each node is equally and relatively powerful. Thus it is impractical to elect one node as a central node in the UGS network. Another major reason for using the distributed architecture in the UGS network is that it is more robust and survivable when one node is vulnerable.

Resource management could be viewed as a scheduling problem whose solution can be provided by dynamic programming [39, 36, 35, 37, 3, 4, 8, 65, 66]. Kreucher, Kastella and Hero's work focuses on an application of sensor management to electronic scanned arrays (ESA) [39, 36, 35, 37]. Their work is to task a sensor action (e.g., sensor pointing or sensor mode) to obtain the maximum amount of information gain. Recursively solving Bellman's equation or approximating a value-to-go function to account for the long-time effect of current actions only works for the simplest problems. When sensor management is used for multiple sensors and we have a multiobjective function to optimize, non-myopic scheduling seems impractical or computationally prohibitive.

Generally, the goal of the resource manager is to choose suitable subset of nodes to be active for a given snapshot under some cost function. The chosen nodes could minimize the estimation error, maximize the probability of target detection, or maximize the lifetime of tracking. For the purpose of minimizing the estimation error, a geometry-based (GB) objective function was proposed to select nodes [34, 28]. Kaplan used the Kalman-based root mean squared (RMS) position error with/without prior covariance information as the objective function where it is assumed that both the dynamical and the measurement models are Gaussian [34, 28]. If the system is non-Gaussian, node selection algorithms can be implemented using information theory [69, 71, 47, 17, 38, 50]. From the information-theoretic point of view, nodes are selected to increase the information or reduce the uncertainty about the current environment. The differential information or the information gain between before and after applying the node combination can be computed based upon Shannon's entropy or Kullback-Leibler's cross entropy. Such entropy-based objective functions as functions of the node combination are prohibitive to evaluate. At this time, no comparisons between Kalman-based and entropy-based approaches are available.

Another important goal of the resource manager for the UGS network is to maximize the lifetime of tracking. The lifetime must be considered as the operational time because the tracking system could be non-operational long before the last node dies. Therefore, the lifetime of tracking should be defined as the time when α percent of data packages are lost in transmission, or when the first failed transmission occurs [5, 24]. Purely maximizing the lifetime unaware of the estimation performance involves the definition of an energy consumption model. Most communication papers that address the energy consumption involve finding an optimal route from a sender (a currently active node) to a receiver (a next active node) to minimize the total energy consumption on the path [60, 62]. Using the lifetime as the metric is more desirable in routing than using the total energy usage. The reason is that some nodes will be overused and run out of energy very quickly because the total energy usage metric does not consider the currently remaining battery level of all the nodes. In addition, multiple hops may exist on the route. A large number of hops along the route makes the delay longer and the tracking less real-time. Hence for real-time tracking, as well as for simplicity it is the desirable that no relay nodes lie between the sender and the receiver, and that the next active nodes receive the information via the direct broadcast of currently active nodes. Routing is not an issue for our node selection procedure. And we prefer an energy-based metric accounting for the battery level to maximize the tracking lifetime.

Reaching a certain level of estimation performance at a cost of a small amount of energy consumption has also drawn great interest from researchers[21, 73, 72, 16, 13, 14, 15]. Their work did not consider how much energy remains in the battery of the sensor node during tracking, but did assume that the global topology of the network is available for evaluating the objective functions. Instead of optimizing a multiobjective function, some work considered a hierarchical network structure to reduce the energy consumption where some nodes are elected as cluster heads to perform tracking and decision making algorithms, and subordinate nodes only report whether there is a target [21, 73, 72]. Chhetri introduced a composite optimization where a sensor action minimizes the total energy from the current time step k up to a future time step $k + M$ while maintaining the tracking error below a predefined limit in the future M time steps [15]. Chhetri's work also required a hierarchical network structure. Williams also used secondary objectives as constraints to address the trade-off between estimation performance and communication cost, and a planning over the future M time steps with the first step in this planning executed, is obtained based on adaptive Lagrangian relaxation in approximating dynamic programming [66, 65]. The fact that Williams' longer planning horizon saves accrued communication cost is given under a careful setting of the bounds for the dual variable in Lagrangian relaxation. The bounds are set experimentally to avoid undesirable behaviors that may result since utilizing all sensors may not meet the constraints. In addition, Williams' scheduling or planning also assumed knowledge of the global topology.

In this thesis, we propose a new energy-based metric which represents the number of snapshots remaining for a node set. Unlike other methods that use the total energy consumed for the given snapshot as the metric, the new metric can achieve load balancing of the nodes without resorting to computationally demanding non-myopic optimization. The non-myopic optimal lifetime metric must consider the utility K time steps in the future and is hard to optimize when the lifetime is defined as the earliest time point when transmission fails. The proposed EB metric models how long a hypothesized node set could run without further information handoff to another different node set. Optimization of the EB metric could be implemented via Greedy Search in practice.

We also investigate the effects of different metrics: GB minimizing the RMS position error, EB, or constrained metrics using one metric as the optimization metric under the constraint of the other, on the estimation performance and the tracking lifetime. We propose practical search algorithms to optimize the constrained metrics. Simulations results show the tradeoff between the estimation performance and the lifetime using different metrics.

These metrics to select nodes at a given snapshot could be embedded into a distributed system where sensor nodes are equally powerful and make independent active/inactive decisions by fusing available data. The distributed system, whose estimation performance or lifetime could be measured, must be responsible for answering the following questions:

1. what are the responsibilities of the active nodes,

2. what do inactive nodes do when they are within the broadcast range of active nodes,

3. how does the information get across in the network or how is the broadcast range of an active node determined,

4. if a node has partial knowledge of the network (a node only knows the physical location of its neighbors), how does the search space for evaluating the metric change?

To this end, we propose an adaptive transmission range control, called the knowledge pool (KP), in a distributed system where the transmission range depends on the knowledge of the network and the remaining battery level. Each active node transmits just far enough to reach all the active nodes. In other words, the energy consumption in one hop is proportional to the physical distance between the active set of nodes and is divided into two stages, one for active nodes to share the currently obtained information (the local target state estimates and error covariances only based upon the locally obtained measurements), the other for the currently active nodes to hand off the useful information (the predicted global target state estimates and error covariances after fusing the local information into the global) to the next active set. The proposed transmission range control is different from deterministic (common or fixed) broadcast range control [70, 12, 68, 22]. The common broadcast range is the reason for the existence of multiple hops or routing problems because the broadcast range could be so small that the packets must be multihopped to their destinations through intermediate or relay nodes. The advantage of the common broadcast range is that knowing the network topology becomes insignificant. Since the routing is not our focus and available neighbor information could be helpful for determining the broadcast range, the variable-range transmission range control becomes reasonable and practical. The proposed transmission range control is also different from another variable-range range control, known as the critical range (CR), which was originally introduced to measure the energy usage in a situation where there is no available neighbor information [31, 29]. Simulations show that more energy is saved using KP as opposed to using CR when enough neighbor information is available.

The goal of this thesis is to investigate node selection algorithms for different objectives, and to embed them into existing tracking filters. The tracking filters must address a complex environment where a target may be maneuvering, missed detections may happen, and false measurements may exist. Here the tracking filters performed at each acoustic node extract target location information out of bearings-only measurements.

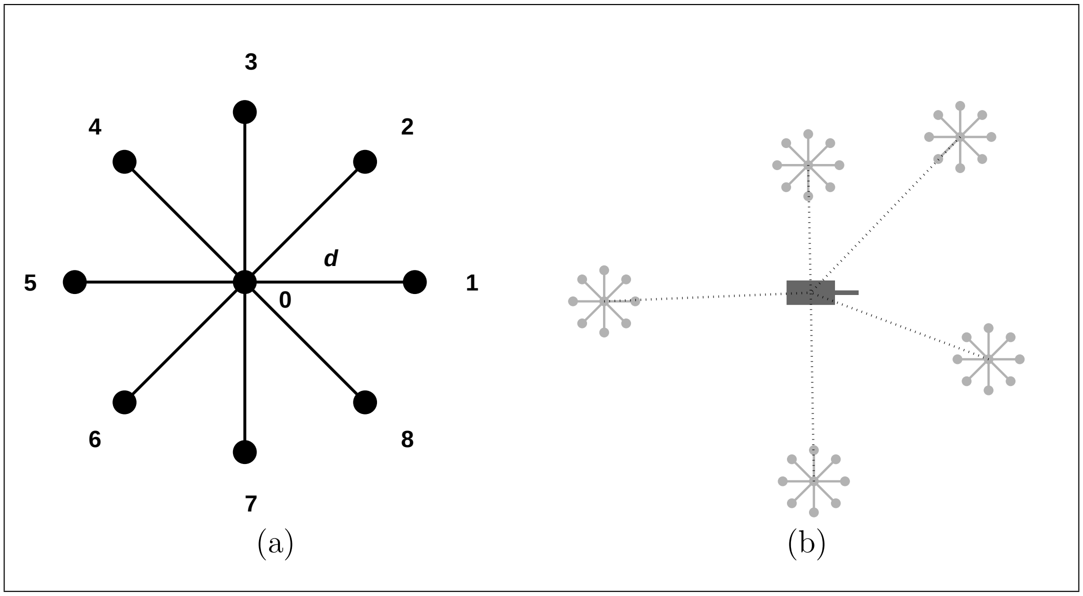


Figure 1: (a) UGS node and (b) triangulation using multiple bearing measurements.

Bearings-only measurement tracking is a classic tracking topic with techniques that locate targets via bearing measurements found in the bearings-only target motion analysis (TMA) literature [20, 44, 53, 55]. In the bearings-only TMA problem, a single sensor located on a moving platform collects bearing measurements over a number of snapshots. Different from a single sensor tracking using batch processing, we intend to use data from all the active nodes to better track a target. At a given snapshot, an acoustic node is capable of measuring target bearing angles. Target position estimates can be obtained using triangulation of multiple bearing lines at any given snapshot as shown in Fig. 1.1(b). Target velocity estimates are observable using two-point differences over multiple snapshots [1].

By incorporating a Markov model and a history of measurements, the Kalman filtering (KF) was developed for tracking and is used in many applications today. The Kalman-Bucy filter gives an unbiased estimate and is optimal in a mean-squared-error sense when observation and dynamic equations are linear [26, 27, 56]. For bearings-only tracking, the measurements are a nonlinear function of the target states. Hence, the extended Kalman filter (EKF) with linearization of measurements was developed to deal with the nonlinearity of the system [1]. The unscented Kalman Filter (UKF) provides a novel way to implement the Kalman filter [25]. It avoids the linearization of nonlinear dynamic and measurement equations by propagating sigma points representing the distribution of the target state through the nonlinearities. The UKF is expected to outperform the EKF due to the fact that the UKF approximates the relationship between the bearing measurements and the target state over a number of sigma points instead of linearizing the relationship about the predicted target state. In fact, Julier *et. al* shows that the UKF at least reaches second-order accuracy, and outperforms the EKF [25].

For better tracking of a maneuvering target, the KF requires exact a priori knowledge of the process and measurement noise statistics. For example in a constant velocity dynamic model, the target acceleration is modeled as process noise. In reality, such parameters are inexactly known. Wrong use of the process noise can lead to large estimation errors, or even to divergence. To solve this problem, one way is to modify or adjust mode parameters such as process noise or measurement noise by estimating them through filter gain or target turn rates [18, 19, 51, 52]. Another way is to introduce multiple dynamic models to describe possible target motions, known as a multiple-mode (MM) method. The MM uses a bank of filters running in parallel, each based upon a set of assumed mode parameters, to obtain mode-conditioned estimates. Then the global state estimate is nothing but a weighted sum [49, 1, 2, 7, 45, 46, 64].

In the presence of multiple targets, a node may receive multiple measurements from existing targets, new targets, or noise, but the node has no idea about which measurement comes from which target. In addition, the node may miss the target detection in noisy environments. Figure 2 shows typical bearing measurements collected by a node at different time intervals. Clearly two measurement tracks are obvious from Time 40 to Time 130. However, a large number of false measurements are also obtained. On the other hand, three measurement tracks exist from Time 130 to Time 200. The Probabilistic Data Association (PDA) filter was first proposed to handle this sort of data association problem [2, 7]. The PDA assumes one of the measurements originated from an existing target and the rest are from random clutter, but does not take into account that one of the remaining measurements may come from another target. The Joint Probabilistic Data Association (JPDA) filter only differs from the PDA in the calculation of the measurement-to-target association. The JPDA filter accounts for the existence of multiple targets. The measurement-to-target association in JDPA is assumed to be marginal so that all possible joint events involving that measurement-to-target pair are enumerated, where the joint event is defined as a set of measurement-to-target assignments. Both the PDA and the JPDA are target-oriented association filters where each measurement makes more or less of a contribution to each previously established target. On the contrary, the original multiple hypotheses tracking (MHT) is a measurement-oriented association filter where the origins of the measurements are taken into account [6, 59]. Recently, sequential Monte Carlo methods, also known as particle filters, have been used to track multiple targets where a particle represents a possible target state and its weight [54, 9]. Because the number of particles determines the tracking accuracy and the tracking speed, we use classic PDA or JPDA as multitarget trackers.

1.2 Organization of the thesis

The thesis is organized as follows. In Chapter 2, we introduce basic tracking filters such as EKF, UKF, MM or interacting MM (IMM) filters. EKF and UKF deal with the cases when the dynamical equations or observation equations are nonlinear. MM or IMM addresses the maneuvering target problem.

In Chapter 3, we introduce the existing geometry-based (GB) node selection algorithms in detail, and a variable-range transmission control that is called the critical range (CR) and bounded with the GB metric. We apply GB to a more realistic

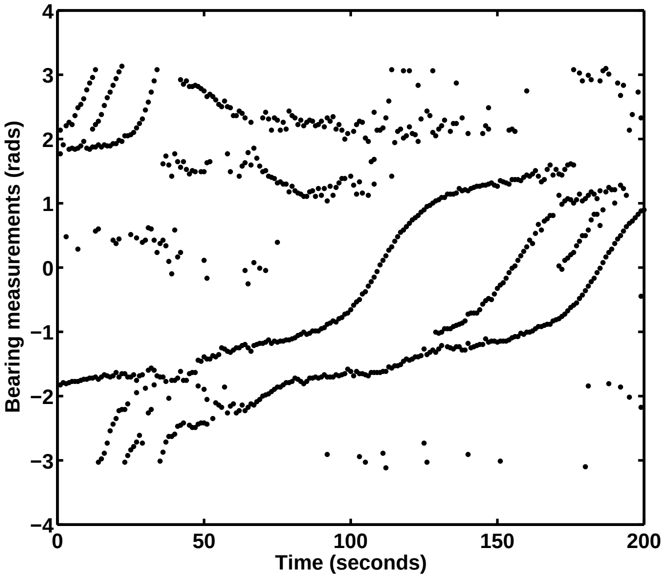


Figure 2: Bearing measurements processed from real data at a UGS node.

environment where a node only knows the existence of itself, the active set of nodes from the previous snapshot and the node's neighbors, i.e., the set of nodes within a distance of r_{nei}. The partial node selection (PNS) algorithm, where r_{nei} affects the search space, is believed to be a bridge between a localized node selection (LNS) algorithm where a node has a minimum knowledge of the network (i.e., itself and active nodes) and a global node selection (GNS) algorithm. Simulation results show that the RMS errors using PNS or GNS are comparable while GNS is more energy-efficient when the energy is measured using CR, which is consistent with Kaplan's results about LNS and GNS in [31]. Since CR is bounded with GB and was originally proposed in a scenario where a node has a minimum knowledge, we consider another variable-range transmission control called the knowledge pool (KP) which could be applied to other metrics. Simulation results show that using KP saves more energy than using CR when global location information is available.

In Chapter 4, we investigate the performance of a multiple target tracker that exploits bearings-only measurements from an acoustic UGS network using real data collected by the U.S Army Research Laboratory (ARL) [43]. To conserve energy while interrogating multiple maneuvering targets, the tracker integrates node resource management with the multiple-mode probabilistic data association (PDA) or the joint probabilistic data association (JPDA) filter. Experiments show that for sufficiently separated targets, the GB metric leads to better geolocation performance than the 'closest' selection approach when the number of active nodes is set to two per snapshot and global network knowledge is available.

In Chapter 5, a new energy-based (EB) metric is first proposed which represents the number of snapshots remaining for a hypothesized set, i.e., the remaining battery energy divided by the energy to sense and share information amongst the node set. Simulation results show EB provides longer lifetime than GB. Then we develop a more complicated resource management strategy that determines which nodes actively sense and communicate for each snapshot in order to achieve a tolerable level of some measure (e.g., geolocalization accuracy) while attempting to optimize the other (e.g., the effective lifetime of the network). Fast search algorithms are provided to optimize the constraint metric. Transmission range is determined by the knowledge of the network and the current battery level. Simulation results show the tradeoff between tracking lifetime and RMS errors for the system which employs the constraint metrics. The effective lifetime of the sensor network is measured by two definitions: the earliest time when one or more active nodes has too little energy to reach the other active nodes, and Type 2) the earliest time when the use of some nodes is unable to meet the constraint. We also investigate the effect of r_{nei} on the decision maker which utilizes the constraint metric to make the activation decision.

Finally, Chapter 6 concludes this thesis with the contributions and the possible future work.

CHAPTER II

BASIC BACKGROUND

2.1 Bearing measurement model

The bearing measurement obtained at the i-th node for a given snapshot is the true retarded bearing angle embedded in additive white Gaussian noise [32].,

$$\hat{\theta}_i = \theta_i + \eta_i, \tag{1}$$

where θ_i is the true bearing angle [32] given by

$$\theta_i = \theta_{i,0} + \arcsin\left(\frac{v}{c}\sin(\theta_{i,0} - \phi)\right), \tag{2}$$

where $\theta_{i,0} = \frac{P_{y,0}-S_{y,i}}{P_{x,0}-S_{x,i}}$ and the second term accounts for the propagation delay. The target position and velocity are labeled as $P_0 = [P_{x,0}, P_{y,0}]^T$ and $V = [V_x, V_y]^T$, respectively. The target state $[P_0^T, V^T]$ consists of the target position and velocity. The target speed $v = |V|$, the heading is $\phi = \arctan(V_y/V_x)$, and c is the speed of the sound, 347m/s. The i-th sensor node location is $S_i = [S_{x,i}, S_{y,i}]^T$. The measurement error η_i is zero-mean white Gaussian noise with a bearing measurement variance denoted as σ_i^2, i.e., $\eta_i \sim N(0, \sigma_i^2)$, and $R = diag(\sigma_1^2, \sigma_2^2, \cdots, \sigma_{N_s}^2)$. In this thesis, we assume the standard deviation $\sigma_i = 5^\circ$ for $i = 1, 2, \ldots, N_s$, where N_s is the number of nodes. The retarded bearing angle model given by (2) is used to generate measurements in the simulations.

2.2 Initial estimates

The maximum likelihood (ML) localization methods under different assumptions about the bearing measurements were developed for the single snapshot case in [33].

The ML methods, true target states or estimates via two-point differencing could be used for the tracking initialization [1].

2.3 Extended Kalman filter

The original Kalman filter is optimal in a mean-squared-error sense if the dynamic and measurement equations are linear, and the process and measurement noise are Gaussian. The extended Kalman filter (EKF) was developed for the case where the equations are nonlinear [40, 32].

The target can be assumed to follow a constant velocity dynamical evolution given by

$$x(k+1) = Fx(k) + Av(k+1), \tag{3}$$

where

$$F = \begin{bmatrix} 1 & 0 & T & 0 \\ 0 & 1 & 0 & T \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}, \text{ and } A = \begin{bmatrix} 0.5T^2 & 0 \\ 0 & 0.5T^2 \\ T & 0 \\ 0 & T \end{bmatrix}.$$

The vector $v(k+1)$ represents unknown accelerations as zero mean Gaussian noise with covariance $Q_v = \sigma_v^2 I$, and $x(k)$ is the target state S for the k-th snapshot. Finally, T is the time interval between successive snapshots, which means that every T seconds we use the Kalman filter to update the state estimates. And the Kalman filter assumes that the mean measurement is the non-retarded bearing $\theta_{i,0}$.

The predicted target state is computed from the previous Kalman filtered state $x(k|k)$ by

$$x(k+1|k) = Fx(k|k), \tag{4}$$

and the covariance of the predicted state is computed from the previous filtered covariance $P(k|k)$ by

$$P(k+1|k) = FP(k|k)F^T + AQ_vA^T. \tag{5}$$

The updated state and covariance expressions are written as

$$x(k+1|k+1) = x(k+1|k) + W\left(z(k+1) - h(x(k+1|k))\right),$$

$$P(k+1|k+1) = P(k+1|k) - W(HP(k+1|k)H^T + R)W^T, \tag{6}$$

where the Kalman gain matrix is

$$W = P(k+1|k)H^T(HP(k+1|k)H^T + R)^{-1}, \tag{7}$$

and H is the Jacobian of the nonlinear observation function given by

$$H = [\nabla\theta_1(x)|_{x(k+1|k)}, \cdots, \nabla\theta_{N_s}(x)|_{x(k+1|k)}]^T, \tag{8}$$

where $\nabla\theta_j = [\frac{\partial}{\partial P_{0,x}}, \frac{\partial}{\partial P_{0,y}}, \frac{\partial}{\partial V_x}, \frac{\partial}{\partial V_y}]^T\theta_j$, and $h(x) = [h_1(x), \ldots, h_{N_s}(x)]$. Using (6)-(7) and the matrix inversion lemma, the updated covariance can be rewritten as:

$$P^{-1}(k+1|k+1) = P^{-1}(k+1|k) + H^TR^{-1}H. \tag{9}$$

2.4 Fusion of multiple nodes

When the Kalman filter is implemented in a centralized way, a powerful central unit is required to collect all the bearing measurements from multiple nodes, and update the target state. The subordinate nodes do nothing but send their locally obtained measurements to the central unit. In a distributed system, every node acts as a central unit and maintains a global state about the current environment. Although redundant computation is done at each node, the system is no longer vulnerable when one node is destroyed. In other words, a node is capable of not only extracting useful information based solely on the locally obtained measurement, but also integrating it into a global state [58]. Let $x_j(k|k)$ and $P_j(k|k)$ be the filtered state and covariance output at the j-th node computed only on the local measurement using (6). A global state is assimilated as:

$$x(k+1|k+1) = P(k+1|k+1)P^{-1}(k+1|k)x(k+1|k) + P(k+1|k+1)\sum_{j=1}^{N_s}\mathbf{i}_j,$$

$$P^{-1}(k+1|k+1) = P^{-1}(k+1|k) + \sum_{j=1}^{N_s} \{P_j^{-1}(k+1|k+1) - P^{-1}(k+1|k)\}, \quad (10)$$

where $\mathbf{i}_j = P_j^{-1}(k+1|k+1)x_j(k+1|k+1) - P^{-1}(k+1|k)x(k+1|k)$.

2.5 Unscented Kalman filter

The UKF provides a novel way to implement the Kalman filter. It avoids the linearization of nonlinear dynamical and measurement equations by propagating $2n+1$ sigma points representing the distribution of the target state through the nonlinearities where the dimension of the target state n is 4. Specifically, it samples the points on the axes of the error hyperellipse defined by the n-dimensional vector $x(k|k)$ and matrix $P(k|k)$ as given by

$$\beta_0(k) = x(k|k), \qquad \beta_i(k) = x(k|k) + \sqrt{n+1}U_i,$$

$$\beta_{i+n}(k) = x(k|k) - \sqrt{n+1}U_i, \qquad \text{for } i = 1, 2, \cdots, n,$$

where $P(k|k) = UU^T$ and U_i is the i-th row or column of the matrix U. These points are weighted by

$$w_0 = \frac{1}{n+1}, w_i = w_{i+n} = \frac{1}{2(n+1)} \text{ for } i = 1, 2, \cdots, n,$$

in order to capture the mean and covariance of $x(k)$. In other words,

$$x(k|k) = \sum_{i=0}^{2n} w_i \beta_i(k),$$

$$P(k|k) = \sum_{i=0}^{2n} w_i \beta_i(k) \beta_i^T(k) - x(k|k)x^T(k|k).$$

In our application, the predicted state and covariance happen to form equivalent expressions as in the EKF because of the linearity of the dynamical equation (see (3)). The filtered equations are still given by (6) except that the $2n+1$ points determine the Kalman Gain matrix by

$$W = P_{xz}P_{zz}^{-1},$$

where

$$P_{xz} = \sum_{i=0}^{2n} w_i (x_i - x(k+1|k))(z_i - \tilde{z})^T,$$

$$P_{zz} = \sum_{i=0}^{2n} w_i (z_i - \tilde{z})(z_i - \tilde{z})^T,$$

$$\tilde{z} = \sum_{i=0}^{2n} w_i z_i, \quad x_i = F\beta_i(k),$$

and

$$z_i = h(F\beta_i(k)).$$

In our simulation to compare EKF and UKF, we considered a target moving at a constant velocity 10m/s through a 2000m×1000m field with ten nodes. The process noise to model target acceleration σ_v is assumed to be zero since the target is moving at a constant velocity. For each value of σ, we ran 100 Monte Carlo simulations. Figure 3 shows the performances of the EKF and the UKF when two bad runs are eliminated out of 100 runs. Julier shows that the UKF is better than the EKF because the predicted state and measurement is more precisely computed in the UKF [25]. We note that our dynamic equation is linear. Therefore, the UKF is better than the EKF only because the predicted measurement is more precise. When the bearing error is lower than five degrees, there is no obvious advantage to more precise predicted measurement in the UKF. The UKF outperforms the EKF only when the bearing error is larger than eight degrees.

2.6 Multiple-mode (MM) and interacting MM

The MM tracker employs multiple mode-matched filters. The mode-matched filter could be the standard EKF, Probabilistic Data Association filter (PDA) or JPDA [42]. The initial state and covariance for each mode-matched filter is the same and given by the previous global state and covariance. The global state update is the

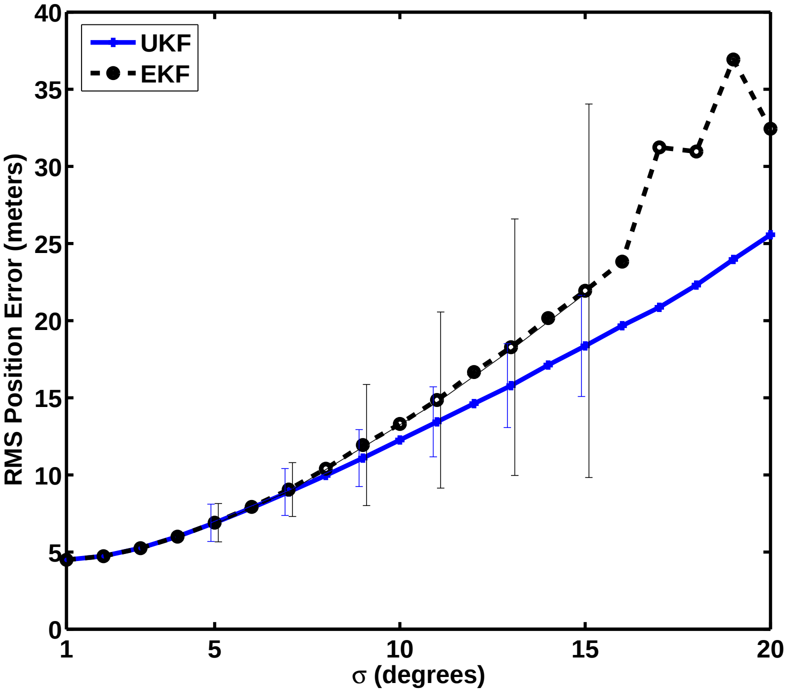


Figure 3: RMS position errors versus σ through simulations when the target velocity=10m/s. Error bars are shown at $\sigma = 5, 7, 9, 11, 13, 15$ degrees.

weighted sum of the state outputs of each mode-matched filter expressed by

$$x(k|k) = \sum_{i=1}^{N} x_i(k|k)\mu_i(k), \tag{11}$$

where $\mu_i(k)$ is the mode probability, $x_i(k|k)$ is the state output of the ith mode-matched filter and N is the number of the mode-matched filters. The weights are derived from the likelihood of each mode representing the true dynamics of the target. We consider modes that represent different parameterizations of the coordinated turn (CT) dynamic model given by

$$x(k+1) = F(\omega)x(k) + Av(k+1), \tag{12}$$

where

$$F(\omega) = \begin{bmatrix} 1 & 0 & \frac{\sin\omega T}{\omega} & \frac{\cos\omega T - 1}{\omega} \\ 0 & 1 & \frac{1-\cos\omega T}{\omega} & \frac{\sin\omega T}{\omega} \\ 0 & 0 & \cos\omega T & -\sin\omega T \\ 0 & 0 & \sin\omega T & \cos\omega T \end{bmatrix}, \text{ and } A = \begin{bmatrix} 0.5T^2 & 0 \\ 0 & 0.5T^2 \\ T & 0 \\ 0 & T \end{bmatrix}.$$

The vector $v(k+1)$ represents unknown accelerations as zero mean Gaussian noise with covariance $Q_\nu = \sigma_\nu^2 I$. T is the time interval between successive snapshots. This model is parameterized by the target turn rate ω in units of rad/s. The mode i filter has its own mode parameter, turn rate, ω_i. Chen in [11] claimed that although multiplicative fusion is Bayesian, when it comes to sensor reliability and weighting a sensor with higher reliability more, spatial additive fusion may outperform multiplicative fusion. Therefore, we use the spatial additive fusion strategy to compute the likelihood of the current measurement set for the i-th mode, i.e.,

$$\wedge_i(k) = \sum_{j=1}^{N_s}\sum_{l=1}^{m_j} \wedge_{ij}^l, \tag{13}$$

where the likelihood of the lth measurement $z_j^l(k)$ at node j is assumed to be a Gaussian centered at $z_{i,j}(k|k-1)$ with covariance $\tilde{\sigma}_j^2$:

$$\wedge_{ij}^l = \mathcal{N}(z_j^l(k); z_{i,j}(k|k-1), \tilde{\sigma}_j^2). \tag{14}$$

The number of measurements at node j is m_j. The predicted measurement related to mode i at node j is $z_{i,j}(k|k-1) = h_j(x_i(k|k-1))$. Let $\tilde{\sigma}_j^2 = H_j P_i(k|k-1) H_j^T + \sigma_j^2$. The $x_i(k|k-1)$ and $P_i(k|k-1)$ are mode-related predictions, and H_j is the jth row in (8). The mode likelihood (14) measures the difference between the assumed model denoted by the given mode parameter and the true target dynamic model. In other words, if the model difference is larger, the likelihood that the current measurement follows the assumed model gets smaller. Then the mode probability is normalized by:

$$\mu_i(k) = \frac{\wedge_i(k)\mu_i(k-1)}{\sum_{j=1}^{N} \wedge_j(k)\mu_j(k-1)}.$$

The global covariance update is written by

$$P(k|k) = \sum_{i=1}^{N} \{(x(k|k) - x_i(k|k))(x(k|k) - x_i(k|k))^T + P_i(k|k)\}\mu_i(k). \qquad (15)$$

where $P_i(k|k)$ is the covariance output of the ith mode-matched filter and $x(k|k)$ is computed by (11).

The IMM tracker considers two consecutive snapshots and introduces a Markov model for the mode transition. Each element of the mode transition matrix T is defined as: $T_{ji} = Pr(\text{the current mode is } i|\text{the previous mode is } j)$. Usually, the transition matrix might represent the following three types of modes:

- Mode 1: coordinated turn mode with negative fixed turn rate and fixed process noise.
- Mode 2: constant velocity mode with fixed process noise, i.e. σ_ν.
- Mode 3: coordinated turn mode with positive fixed turn rate and fixed process noise.

The mode probability in IMM as computed after the measurement data is received is based on not only the likelihood of the current measurement (13), but also on the mode probability at the previous snapshot and transition matrix T, i.e.,

$$\mu_i(k) = \frac{\wedge_i(k)C_i(k-1)}{\sum_{j=1}^{N} \wedge_j(k)C_j(k-1)},$$

where the predicted mode probability

$$C_i(k-1) = \sum_{j=1}^{N} \mu_j(k-1)T_{ji}. \tag{16}$$

The interacting/mixing step in IMM is to provide the initial guess for each mode-matched filter. The initial state for the ith mode-matched filter, $i = 1, 2, \ldots, N$, is computed by:

$$x_i^o(k-1|k-1) = \sum_{j=1}^{N} \mu_{ji}(k-1)x_j(k-1|k-1),$$

where $\mu_{ji}(k-1)$ is the probability that the target made the mode transition from j to i given that the target is currently in state i at time k. Thus,

$$\mu_{ji}(k-1) = \frac{\mu_j(k-1)T_{ji}}{C_i(k-1)}.$$

The initial covariance is written as

$$P_i^o(k-1|k-1) = \sum_{j=1}^{N} \{\tilde{x}_j(k-1)\tilde{x}_j(k-1)^T + P_j(k-1|k-1)\}\mu_{ji}(k-1), \tag{17}$$

where $\tilde{x}_j(k-1) = x_j(k-1|k-1) - x_i^o(k-1|k-1), i = 1, 2, \ldots, N$. The global state and covariance updates follow (11) and (15).

We evaluate the utility of MM-EKF using real acoustic measurements collected by the U.S. Army Research Laboratory (ARL). Six nodes consisting of arrays of microphones sampled the acoustic energy as different ground vehicles traveled around an oval track whose length and width are about 660 and 125 meters, respectively. Figure 4 illustrates the node/track geometry. ARL processed the raw data using an incoherent wideband minimum variance distortionless response (MVDR) beamformer [67] to obtain bearing measurements. We received the time stamped MVDR output and ground truth data collected from a GPS on the vehicles.

Using the real data, we evaluate the performance of different parameterizations of single mode standard EKF (SM-EKF) and MM-EKF trackers. For the SM-EKF, we used a constant velocity target model, i.e., $\omega = 0$. We also evaluated a clairvoyant SM tracker that uses the ground truth data to determine the true turn rate parameter for each snapshot. In MM-EKF or IMM-EKF, we used N CT models where the turn rates are uniformly sampled over [-30° 30°], i.e. $\omega_i = -30 + \frac{i-1}{N-1}60$, for $i = 1, 2, \ldots, N$. The initial mode probability $\mu_i(0)$ is set uniformly so that $\mu_i(0) = 1/N$.

For the 3-mode IMM with turn rates -30°, 0 and 30°, we could assume the target stays at the same mode with large probability 0.6 and makes the mode transition with probability 0.2 in one unit of time so that the mode transition matrix is

$$T1 = \begin{bmatrix} 0.6 & 0.2 & 0.2 \\ 0.2 & 0.6 & 0.2 \\ 0.2 & 0.2 & 0.6 \end{bmatrix}.$$

If some prior information tells us that the target stays at constant velocity mode for the longest time and at mode with right turn for the shortest time, the mode transition matrix could be predefined as:

$$T2 = \begin{bmatrix} 0.4 & 0.3 & 0.3 \\ 0.1 & 0.6 & 0.3 \\ 0.1 & 0.3 & 0.6 \end{bmatrix}.$$

Figure 4 and Table 1 investigate the estimation performances for MMs and IMMs at appropriate process noise levels. The appropriate process noise searched over a limited process noise range minimizes the RMS position error. As expected, the realizable SM has the largest critical process noise and the largest corresponding RMS error. The MM with larger number of modes has smaller critical process noise than the MM with smaller number of modes. From the point of view of the RMS error, it is better to use 3-mode MM among the MMs when the process noise is set appropriately. The MM is quicker to adapt to a maneuvering target than the realizable SM tracker.

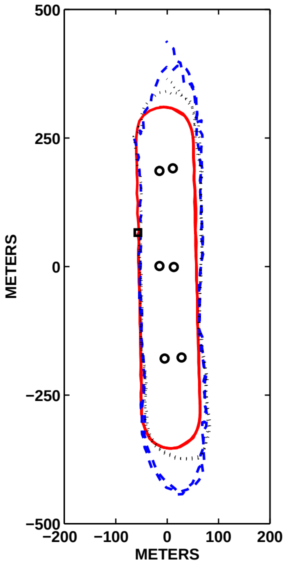


Figure 4: Node locations and estimated tracks with the correct process noise when all 6 nodes are active. The solid line is the true track. The square represents the initial target location. The target travels counter clockwise. The circles denote the six node positions. The dotted and dashed lines denote the 3-mode IMM with T1 and the 3-mode MM, respectively.

The IMM outperforms the MM when the transition probability is set reasonably. When the process noise is chosen correctly, it is better to use an IMM tracker with $N = 3$ modes.

Table 1: RMS position errors at appropriate process noise levels for MMs or IMMs.

	Appropriate process noise σ_ν(m/s^2)	RMS position error (meters)
Realizable SM	30	23.66
Clairvoyant SM	1.5	20.66
3-mode MM	7	22.25
5-mode MM	1.4	22.37
7-mode MM	1.3	22.52
13-mode MM	1.3	22.73
21-mode MM	1.3	22.83
3-mode IMM with T1	0	14.94
3-mode IMM with T2	0	14.44

CHAPTER III

GEOMETRY-BASED NODE SELECTION ALGORITHMS

The node selection algorithm embedded in the resource manager to determine the set of active nodes can be performed at each node in a distributed system. Each node must also be capable of implementing one of the tracking filters (EKF, UKF, PDA, etc.) to extract useful information out of the locally obtained measurements, broadcast the information and at the same time integrate it into a global state. The chosen nodes could either minimize the estimation error, maximize the lifetime of tracking, or optimize a multiobjective function. In this Chapter, we focus on minimizing the estimation error to choose the desired number of active nodes per snapshot. We do not consider how much energy is left in the battery (the capacity of the battery is not an issue), only how much energy is consumed.

For the purpose of minimizing the estimation error, a geometry-based (GB) objective function was proposed to select nodes [34, 28, 29]. Kaplan used the Kalman-based root mean squared (RMS) position error with/without prior covariance information as the objective function. He proposed a global node selection algorithm (GNS) in [28] and an autonomous node selection algorithm (ANS) in [34, 31]. We add the prefix GB in front of GNS/ANS in order to discriminate them from energy-based methods. The GB-GNS is a node selection algorithm to minimize the RMS position error when the location information of all the nodes in the network is available at each node. In the GB-GNS method, an inactive node could become active if it is within the next set of active nodes computed based upon the global location information. In the

GB-ANS method, an inactive node within the broadcast range of all the active nodes could become active if its contribution for a given active node set exceeds a threshold. The GB-ANS method is a local node selection algorithm (LNS) where a node only knows the physical location of itself and the active nodes.

3.1 GB metric and critical range (CR)

Since we assume the measurement is only dependent on the target positions, the updated covariance derived from the Kalman filter, using the information form, is explicitly expressed as:

$$\mathbf{J}_f(\mathcal{N}_a) = \mathbf{J}_p + \begin{pmatrix} \mathbf{J}_m(\mathcal{N}_a) & \mathbf{0_2} \\ \mathbf{0_2} & \mathbf{0_2} \end{pmatrix},$$

where

$$\mathbf{J}_m(\mathcal{N}_a) = \sum_{i \in \mathcal{N}_a} \frac{1}{\sigma_i^2} \frac{1}{r_i^2} \begin{pmatrix} \sin^2 \phi_i & -\sin\phi_i \cos\phi_i \\ -\sin\phi_i \cos\phi_i & \cos^2 \phi_i \end{pmatrix},$$

and where r_i and ϕ_i are the 2D polar coordinates of the i-th node to the predicted target location. The information matrices are $\mathbf{J}_f = P^{-1}(k|k)$ and $\mathbf{J}_p = P^{-1}(k+1|k)$. Then a utility function can be defined as:

$$\mu(\mathcal{N}_a) = \frac{1}{\rho^2(\mathcal{N}_a)}, \tag{18}$$

where $\rho(\mathcal{N}_a)$ could be either the posterior RMS position error defined by

$$\rho(\mathcal{N}_a) = \sqrt{\text{trace}([\mathbf{J}_f^{-1}]_{1:2,1:2})}, \tag{19}$$

or the current measurement-related RMS position error defined by

$$\rho(\mathcal{N}_a) = \sqrt{\text{trace}\left\{\mathbf{J}_m^{-1}\right\}} \tag{20}$$

without incorporating the prior information. $[\mathbf{A}]_{i:j,k:l}$ represents the $(j-i+1) \times (l-k+1)$ subblock of $\mathbf{A}$. Obviously $\rho(\mathcal{N}_a)$ is a function of the node set $\mathcal{N}_a$ and the predicted target state. Let $\mathcal{N}_g$ denote the set of all nodes in the network, and N_d the desired number of active nodes per snapshot. For a subset of N_a nodes that comprise the active set $\mathcal{N}_a$, we define the differential utility for the i-th node as:

$$d\mu(i|\mathcal{N}_a) = \begin{cases} \mu(\mathcal{N}_a) - \mu(\mathcal{N}_a \setminus \{i\}) & \text{if } i \in \mathcal{N}_a \\ \max_{a\in\mathcal{N}_a} \mu((\mathcal{N}_a \setminus \{a\}) \cup \{i\}) - \mu(\mathcal{N}_a \setminus \{a\}) & \text{otherwise} \end{cases} \tag{21}$$

Given a node whose *virtual range* [28] to the target is $r_i\sigma_i$, its differential utility using any active set is bounded by:

$$0 \leq d\mu(i|\mathcal{N}_a) < \frac{(1+\gamma)^2}{4\sigma_i^2 r_i^2}, \tag{22}$$

where Kaplan proved that $0 \leq \gamma \leq 1$ and gave the calculation of γ in [30, 31]. Let a threshold $\tau = \min_{i\in\mathcal{N}_a} d\mu(i|\mathcal{N}_a)$ be the minimum contribution that one node in the set $\mathcal{N}_a$ could make. Any potentially active node that is currently not in $\mathcal{N}_a$ is expected to make a greater contribution than τ, i.e., $d\mu(j|\mathcal{N}_a) > \tau$ for $j \ni \mathcal{N}_a$. Inspired by (22), a node far away from the predicted target or one having a large bearing measurement error has limited differential utility. In other words, for $j \ni \mathcal{N}_a$,

$$r_j < \frac{1}{\sigma_j\sqrt{d\mu(j|\mathcal{N}_a)}} < \frac{1}{\sigma_j\sqrt{\tau}}. \tag{23}$$

Inequality (23) shows that a potentially active node that is currently inactive and could make a greater contribution or break the threshold must be within a *critical range* $r_c \equiv \frac{1}{\sigma\sqrt{\tau}}$, assuming $\sigma_i = \sigma$ for each node. The critical range r_c is a function of the chosen node set $\mathcal{N}_a$ and the predicted target state. A currently active node's broadcast range must be

$$b_i = r_i + r_c \tag{24}$$

to cover $\mathcal{N}_a$ and also the potentially active nodes that may exceed the threshold.

When we measure the energy usage for GB-GNS or GB-ANS, we could use the critical range to determine the broadcast range (24). So we add the suffix CR to the node selection algorithm type to explicitly indicate that such a node selection algorithm is evaluated using the critical range. Note that neither the GB-GNS-CR nor the GB-ANS-CR has considered the capacity of the battery at this point.

Let

$$\mathcal{N}_r = \bigcap_{j \in \mathcal{N}_a} \{i | d_{i,j} < b_j\} \tag{25}$$

be the reachable set of the active set $\mathcal{N}_a$, where $d_{i,j}$ is the distance between the i-th node and the j-th node, and b_j is the broadcast range for the j-th active node. The steps of the GB-GNS-CR and GB-ANS-CR to determine the next active node set are listed in Figs. 5 and 6, respectively. Let $\mathcal{N}_a$ be the currently active node set. When measuring the energy usage, the information handoff is determined based upon the critical range within which nodes providing good geometric view must lie.

1. Determine the next set of active nodes by computing

$$\mathcal{N}_a^* = \arg \min_{\mathcal{N} \subseteq \mathcal{N}_r} \rho(\mathcal{N}),$$

where $|\mathcal{N}| = N_d$ and $\mathcal{N}_r$ is the reachable set of the active nodes (25).

2. Determine the broadcast range of the active nodes by (24).

Figure 5: One cycle of the GB-GNS-CR by Kaplan. Step 2 can be eliminated when energy usage is not measured.

3.2 Transmission range control: knowledge pool (KP)

In this section, we introduce another adaptive range control to determine the broadcast range using available location knowledge of the network and call it the knowledge pool (KP). Let $\mathcal{N}_a(k)$ be the active set at time k. The energy consumed for transmitting d meters is ϵd^4 in a multipath environment where $\epsilon = l\epsilon_{amp}$, l is the package size in bits, and ϵ_{amp} is a constant to run the power amplifiers in the transmitters. Such a consumption model is based upon the fact that the energy usage in computation

For any node $i \in \mathcal{N}_a$,

1. Determine whether to remain active by computing

$$\mathcal{N}_a^* = \arg \min_{\mathcal{N} \subseteq \mathcal{N}_a} \rho(\mathcal{N}),$$

   where $|\mathcal{N}| = N_d$. If $i \in \mathcal{N}_a^*$, i remains active; otherwise not.

2. If active, set a threshold τ to be the k_d-th largest differential utility by (21) where k_d is the user-defined parameter; determine its broadcast range by (24).

$\mathcal{N}_a = \mathcal{N}_a^*$.
For any node $i \in \mathcal{N}_r \setminus \mathcal{N}_a$ where $\mathcal{N}_r$ is computed by (25),

1. Determine whether to be active: compute $d\mu(i|\mathcal{N}_a)$ by (21). If $d\mu(i|\mathcal{N}_a) > \tau$, i becomes active. Otherwise not.

2. If active, determine its broadcast range by (24).

Figure 6: One cycle of the GB-ANS-CR by Kaplan.

is modest compared to that needed for radio transmissions [57]. The model derived from [23] only accounts for the radio transmission.

In GB-GNS-CR, the active nodes communicate their data and target state information in one shot. The broadcast range of a currently active node must be large enough to cover not only every other currently active node, but also the potentially active nodes for the next snapshot. Therefore, the i-th node in $\mathcal{N}_a(k)$ consumes ϵb_i^4 for the information exchange and handoff, where $b_i = r_c + r_i$ and r_c is the critical range within which the potentially active nodes providing a better geometric view must lie. The problem with GB-GNS-CR is that the broadcast range is usually larger than necessary.

In GB-GNS-CR illustrated in Fig. 5, we note that the currently active nodes, or the inactive nodes within the broadcast range of all active nodes after the currently active nodes share the locally obtained information, will have an idea of where the next active nodes are, or which nodes will be active, because the location information is global. On the other hand, the transmission range control using the critical range is bounded for the GB metric, which means that the critical range is to determine where the better nodes in terms of the GB metric will lie. If the evaluation metric is not GB, such transmission control may be loose. Hence, for a more general consideration, we propose a transmission range control using available network information, i.e., a knowledge pool. The activation decision is performed in a decentralized manner over the active set of nodes after the currently obtained information is shared among the active set of nodes. To this end, each active node determines the next active set by evaluating a metric, and decides whether it remains active or whether it should wake up and hand off information to nonactive nodes that are members of the next active set. Therefore, the broadcast range could be just long enough to reach all the active nodes for information sharing or the next active set for information handoff. The energy is consumed by information sharing over the currently active set and information handoff to the next active set. We add the suffix KP to the node selection algorithm type to explicitly indicate that such an algorithm uses the knowledge pool to determine the broadcast range.

3.3 GB-GNS-KP

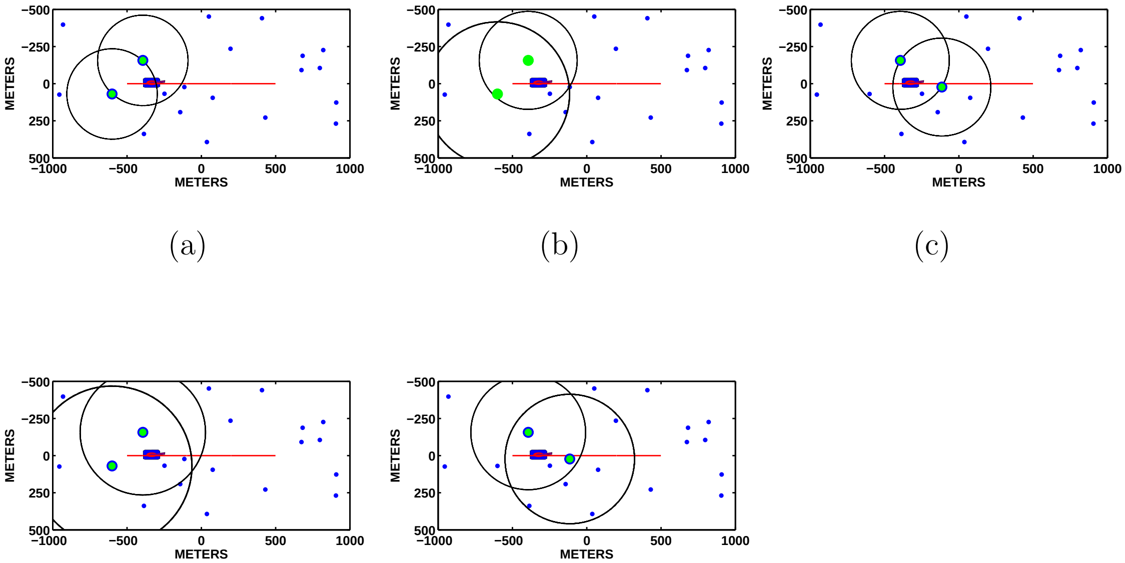
In GB-GNS-KP, the available global location information is used to determine the broadcast range of the active nodes. The energy consumption is divided into two classes: one for the active nodes to exchange the current information, i.e., the locally updated target state and covariance, the other for information handoff from the currently active nodes to the next set of active nodes. After the currently active nodes share and process the current information, they know which set of nodes will be active for the next snapshot. Thus, they could adjust their broadcast ranges so that the potentially active nodes that are currently inactive will get the state information about the target. The information handoff is dependent on the global location knowledge instead of the critical range in the GB-GNS-CR. The problem with the knowledge pool is that the currently active nodes must ping an inactive node to wake up. As a result, the i-th node in $\mathcal{N}_a(k)$ consumes not only $\epsilon d^4_{i,\mathcal{N}_a(k)}$ for the information sharing, but also $\epsilon d^4_{i,\mathcal{N}_a(k+1)}$ for the information handoff to the next set of active nodes, $\mathcal{N}_a(k+1)$, where $d_{i,\mathcal{N}} = \max_{j\in\mathcal{N}} d_{i,j}$.

Figure 7 shows one cycle of GB-GNS-CR and GB-GNS-KP where the chosen nodes are the origins of the broadcast ranges denoted by the black circles. For GB-GNS-CR in Fig. 7(d-e), the broadcast range determined by the critical range could be loose. For GB-GNS-KP in Fig. 7(b), the $\mathcal{N}_a(k)$ pings the inactive nodes to wake them up by adjusting the broadcast range, so the energy is burned for the pinging operation.

In the simulations, we ran ten different node configurations, each with twenty randomly placed nodes. A target is moving at a constant velocity of 10 m/s. The number of active nodes, N_d, is user-defined and varies from two to eight. The power scaling parameter is $\epsilon = l\epsilon_{amp}$, where $l = 384$ bits and $\epsilon_{amp} = 0.00013$ pJ. For each value of N_d, we ran fifty Monte Carlo simulations. Figures 8 and 9 show the comparison between GB-GNS-CR, where we use the posterior RMS error as the evaluation metric and a tight bound to compute r_c (22), and GB-GNS-KP, where we use measurement RMS error without incorporating the prior information as the evaluation metric. It is clear that GB-GNS-KP achieves similar estimation performance at a cost of a little bit more than two thirds of the energy usage in GB-GNS-CR.

3.4 GB partial node selection with CR (GB-PNS-CR)

GB-GNS assumes that each node in the network knows where every other node is located. As a result, every node subset with N_d nodes can be evaluated to minimize the RMS error. However, when there is a power constraint, each node may know only where its neighbors are located at the initial setup stage. Let $\mathcal{N}_{nei}(i) = \{j | d_{i,j} \leq r_{nei}, i \neq j\}$ be the neighbor node set of the i-th node where r_{nei} is the range that



(a)

(b)

(c)

(d)

(e)

Figure 7: Illustration of the information handoff using GB-GNS-CR and GB-GNS-KP. In (a)-(c): GB-GNS-KP, (a) $\mathcal{N}_a(k)$ (b) $\mathcal{N}_a(k)$ adjusts the broadcast range to transmit the information to the next active nodes and (c) $\mathcal{N}_a(k+1)$ becomes active. In (d,e): GB-GNS-CR, (d) $\mathcal{N}_a(k)$ and (e) $\mathcal{N}_a(k+1)$.

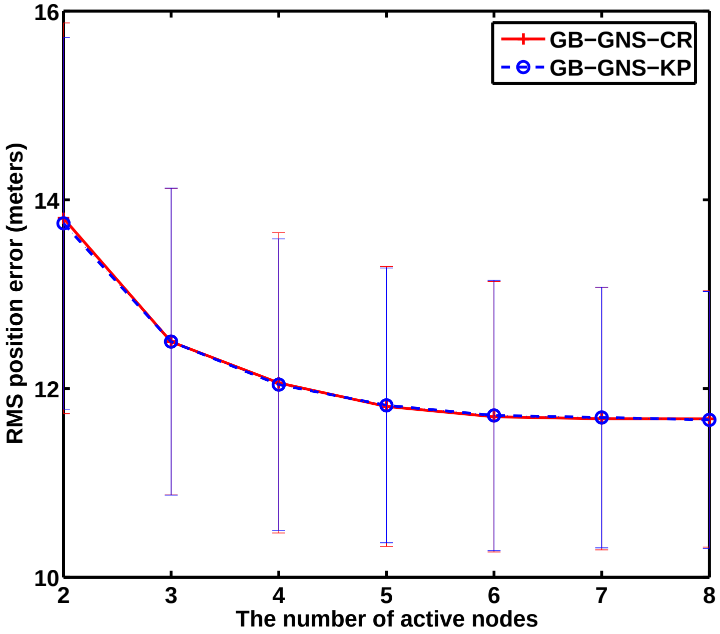


Figure 8: Comparison between the GB-GNS-CR and the GB-GNS-KP by RMS position error. Error bars computed from 50 runs are shown at each data point.

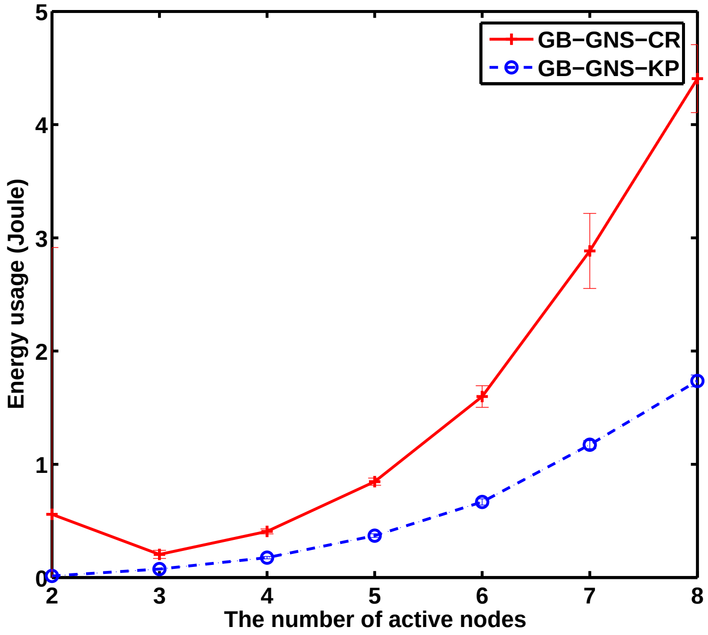


Figure 9: Comparison between the GB-GNS-CR and the GB-GNS-KP by energy usage. Error bars computed from 50 runs are shown at each data point.

defines neighbors by virtue of a broadcast power constraint. Any node within a circle of r_{nei} meters is known at the i-th node.

The range r_{nei} affects the size of the combinatorial search space in evaluating the GB metric. The combinatorial search space varies at different nodes, having different numbers of neighbors.

Inspired by Kaplan's GB-GNS-CR, the decision making steps in GB-PNS-CR investigating neighbor information are shown in Fig. 10. The meaning of the critical range to determine an active node's broadcast range is kept. The critical range is a function of a set of active nodes and the predicted target state. Step 1 is to maintain a minimum core with N_d active nodes so that $N_a = |\mathcal{N}_a| \geq N_d$. Otherwise, if a minimum core is not maintained, $N_a < N_d$ and, what is worse, the node selection algorithms will suffer from the case where only one node is active especially when $r_{nei} = 0$. Note that the neighbor information does not affect an active node's decision making because an active node's neighbors are not considered during the evaluation of Step 1. As a result a common critical range is determined at each active node (Step 2). Since the critical range is one of the parameters transmitted, a node within the broadcast range of multiple active nodes may receive different critical ranges if critical ranges are not the same at the active nodes. As a consequence, an inactive node may not know which critical range should be chosen to determine its broadcast range if it decides to be active. The GB-PNS-CR method with $r_{nei} = \infty$ differs slightly from GB-GNS-CR (Fig. 5). The GB-GNS-CR method ensures that $N_a = N_d$ while in GB-PNS-CR, even with $r_{nei} = \infty$, it is possible that $N_a > N_d$, because a minimum core with length N_d is kept (Step 1 of Fig. 10). In addition, GB-PNS-CR with $r_{nei} = 0$ is also different from GB-ANS-CR in that the inactive nodes make active/inactive decisions. The differential utility criteria in GB-ANS-CR allow all nodes that meet the criteria of GB-PNS-CR with $r_{nei} = 0$m to the join the active set, but it does allow for other nodes to join the set as well.

In Fig. 10, the currently active nodes $\mathcal{N}_a(k)$ are responsible for determining a minimum core with a desired number of active nodes and a common critical range, and then broadcasting the useful information including the predicted target position $x(k+1|k)$, the predicted error covariance $P(k+1|k)$, their own locations, and the updated critical range. In other words, the active nodes must implement Step 1 and Step 2 in Fig. 10. The inactive nodes within earshot of the active nodes are responsible for making their own active/inactive decisions by implementing Step 4 based on the received information. The inactive nodes beyond earshot of the active nodes remain inactive because they receive no new information. Each updated active node $i \in \mathcal{N}_a(k+1)$ obtains a local bearing measurement $z_i(k+1)$, computes a local target state update $x_i(k+1|k+1)$ and a covariance update $P_i(k+1|k+1)$ by using the EKF, PDA, JPDA, etc. As a result, each node consumes an amount of energy for transmitting $x_i(k+1|k+1)$ and $P_i(k+1|k+1)$ to other active nodes, but finally it builds up a global state $x(k+1|k+1)$ and covariance $P(k+1|k+1)$ (see Section 2.4). The flowchart of the functions combining GB-PNS-CR with Kalman filtering in a distributed way is shown in Fig. 11 where the dotted box contains the functions for active nodes and the dashed box the functions for inactive nodes at time k.

3.5 GB-PNS-KP

Inspired by GB-GNS-KP, the steps of decision making in GB-PNS-KP are shown in Fig. 12. We let the available location information guide the broadcast range of an active node instead of using the critical range. Step 1 is to determine the next set of the active nodes, $\mathcal{N}_a^*$, derived from the currently active nodes $\mathcal{N}_a$. The set $\mathcal{N}_{a,i}^*$ is the locally determined set of the active nodes based on node i's neighbors (see Steps 2 and 4). In other words, the i-th node thinks that $\mathcal{N}_{a,i}^*$ should be active for the next snapshot. Therefore, the broadcast range for the information exchange at the next snapshot is large enough to cover the sets $\mathcal{N}_a^*$ and $\mathcal{N}_{a,i}^*$, i.e., $R_i = \max\{d_{i,\mathcal{N}_{a,i}^*}, d_{i,\mathcal{N}_a^*}\}$.

1. Determine the next set of the active nodes with length N_d, derived from the currently active nodes $\mathcal{N}_a$

   $$\mathcal{N}_a^* = \arg \min_{\mathcal{N} \subseteq \mathcal{N}_a} \rho(\mathcal{N}),$$

   where $|\mathcal{N}| = N_d$.

2. For any node $i \in \mathcal{N}_a^*$,

   - set a threshold $\tau = \min_{i \in \mathcal{N}_a^*} d\mu(i|\mathcal{N}_a^*)$.
   - set a common critical range $r_c = \frac{1}{\sigma\sqrt{\tau}}$.
   - determine its broadcast range $b_i = r_i + r_c$.

3. Let $\mathcal{N}_{ia} = \mathcal{N}_r \setminus \mathcal{N}_a^*$ where $\mathcal{N}_r$ is a set of nodes that can hear every node in $\mathcal{N}_a^*$, that is, $\mathcal{N}_r = \bigcap_{j \in \mathcal{N}_a^*} \{i | d_{i,j} < b_j\}$.

4. For any node $i \in \mathcal{N}_{ia}$,

   - Define a known node set for node i:

     $$\mathcal{N}_{known}(i) = \mathcal{N}_a^* \cup \{i\} \cup \{\mathcal{N}_{nei}(i) \cap \mathcal{N}_r\}.$$

   - Compute

     $$\mathcal{N}_{a,i}^* = \arg \min_{\mathcal{N} \subseteq \mathcal{N}_{known}(i)} \rho(\mathcal{N}),$$

     where $|\mathcal{N}| = N_d$.
   - If $i \in \mathcal{N}_{a,i}^*$, i becomes active; $\mathcal{N}_a^* = [\mathcal{N}_a^* \; i]$; determine its broadcast range b_i by (24).

5. For $i \in \mathcal{N}_a^*$, ϵb_i^4 is consumed.

Figure 10: One cycle of the GB-PNS-CR method.

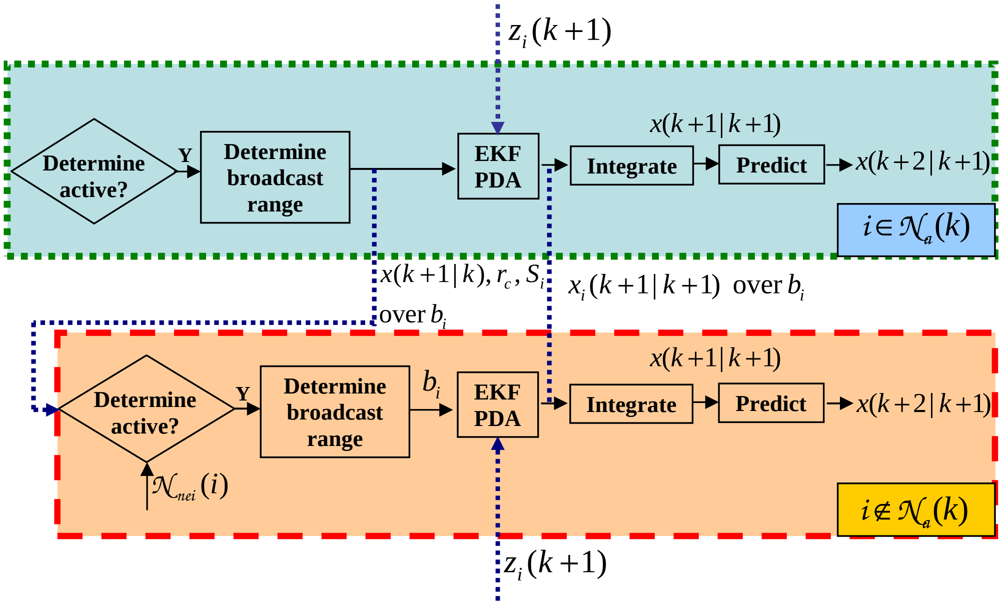


Figure 11: Flowchart of the functions for active/inactive nodes combining GB-PNS-CR with tracking filters.

On the other hand, since the potentially active nodes that are currently inactive may not be within the current broadcast range of the active nodes, the node in $\mathcal{N}_a^*$ needs to ping those potentially active nodes by adjusting its broadcast range for the information handoff. As a result, a node in $\mathcal{N}_a^*$ also consumes ϵD_i^4 for the information handoff where $D_i = \max\{d_{i,\mathcal{N}_{a,i}^*}, d_{i,\mathcal{N}_a^*}\}$.

Clearly, the broadcast range of an active node for the information handoff is not dependent on a common critical range, but on the knowledge of the active nodes' neighbors (see Step 2 in Fig. 12). The responsibilities of the active nodes are not only to maintain a minimum active node set, but also to insightfully determine their broadcast ranges to the potentially active nodes based upon the knowledge of their neighbors. The inactive nodes also investigate their neighbor information to make their own active/inactive decisions (see Step 4 in Fig. 12). A tracking filter such as EKF, PDA and JDPA is implemented using the new bearing measurement after a node becomes active. The flowchart of the functions combining GB-PNS-KP with the tracking filters in a distributed way is shown in Fig. 13, where the dotted box contains the functions for active nodes and the dashed box contains the functions for inactive nodes at time k. In Fig. 13, the information is exchanged and shared over the broadcast range that is probed by the active nodes based on the known location information and the r_{nei} affects the decision making of a currently active node on whether or not it should remain active. On the contrary, in Fig. 11 the information is shared over the broadcast range determined by the critical range and r_{nei} only affects the decision making of a currently inactive node on whether or not to become active.

Figure 14 shows one cycle of GB-PNS-KP with $r_{nei} = 1000$ m where the chosen nodes indicated by shaded circles are the origins of the broadcast ranges denoted by the black circles. The GB-PNS-KP requires the extra pinging operation to cover the promising active nodes, see Fig. 14(b). In the simulations, we ran ten node configurations in a field of size 2000m×1000m, each with twenty nodes randomly generated. For each value of r_{nei} and a given configuration, fifty Monte Carlo trials with $\sigma = 5^\circ$ are run. The desired number of active nodes is $N_d = 3$. A target is moving along a straight line as shown in Fig. 14.

Figures 15-18 show comparisons between GB-PNS-CR and GB-PNS-KP with posterior RMS errors as the GB metric. For GB-PNS-KP, which has more knowledge of its neighbors, there is smaller RMS error at a cost of more energy usage and node usage. The reason is that when r_{nei} is larger, an active node would know the existence of a nearby node that brings a better geometric view and then it would ping this potentially active node to wake up at a cost of more energy usage. As shown in Figs. 15-16, the advantage of using a knowledge pool over the critical range to determine the broadcast range is obvious when r_{nei} is greater than 700m. When r_{nei} is smaller than 700m, using the critical range could cover nodes that possibly provide better geometric views at a cost of more energy usage because of longer reach. For smaller r_{nei}, using knowledge pool may not help cover better nodes although energy would be saved.

1. Determine the next set of the active nodes with length N_d, derived from the currently active nodes $\mathcal{N}_a$

   $$\mathcal{N}_a^* = \arg\min_{\mathcal{N}\subseteq\mathcal{N}_a} \rho(\mathcal{N}),$$

   where $|\mathcal{N}| = N_d$.

2. For any node $i \in \mathcal{N}_a^*$,

   - define a known node set for node i: $\mathcal{N}_{known}(i) = \mathcal{N}_a \bigcup \mathcal{N}_{nei}(i)$.
   - determine the broadcast range for the information exchange at the next snapshot by computing

     $$\mathcal{N}_{a,i}^* = \arg\min_{\mathcal{N}\subseteq\mathcal{N}_{known}(i)} \rho(\mathcal{N}),$$

     where $|\mathcal{N}| = N_d$; $R_i = \max\{d_{i,\mathcal{N}_{a,i}^*}, d_{i,\mathcal{N}_a^*}\}$.
   - adjust its broadcast range for the information handoff to D_i where $D_i = R_i$; consume ϵD_i^4 for the information handoff.

3. Let $\mathcal{N}_{ia} = \mathcal{N}_r \setminus \mathcal{N}_a^*$ where $\mathcal{N}_r$ is a set of nodes that can hear every node in $\mathcal{N}_a^*$, that is, $\mathcal{N}_r = \bigcap_{j\in\mathcal{N}_a^*}\{i|d_{i,j} \leq D_j\}$.

4. For any node $i \in \mathcal{N}_{ia}$,

   - define a known node set for node i :

     $$\mathcal{N}_{known}(i) = \mathcal{N}_a^* \cup \{i\} \cup \{\mathcal{N}_{nei}(i) \cap \mathcal{N}_r\}.$$

   - compute

     $$\mathcal{N}_{a,i}^* = \arg\min_{\mathcal{N}\subseteq\mathcal{N}_{known}(i)} \rho(\mathcal{N}),$$

     where $|\mathcal{N}| = N_d$.
   - if $i \in \mathcal{N}_{a,i}^*$, i becomes active; $\mathcal{N}_a^* = [\mathcal{N}_a^*\ i]$; determine the broadcast range for the information exchange at the next snapshot: $R_i = \max\{d_{i,\mathcal{N}_{a,i}^*}, d_{i,\mathcal{N}_a^*}\}$.

5. For $i \in \mathcal{N}_a^*$, ϵR_i^4 is consumed for the information exchange at the next snapshot.

Figure 12: One cycle of the GB-PNS-KP method.

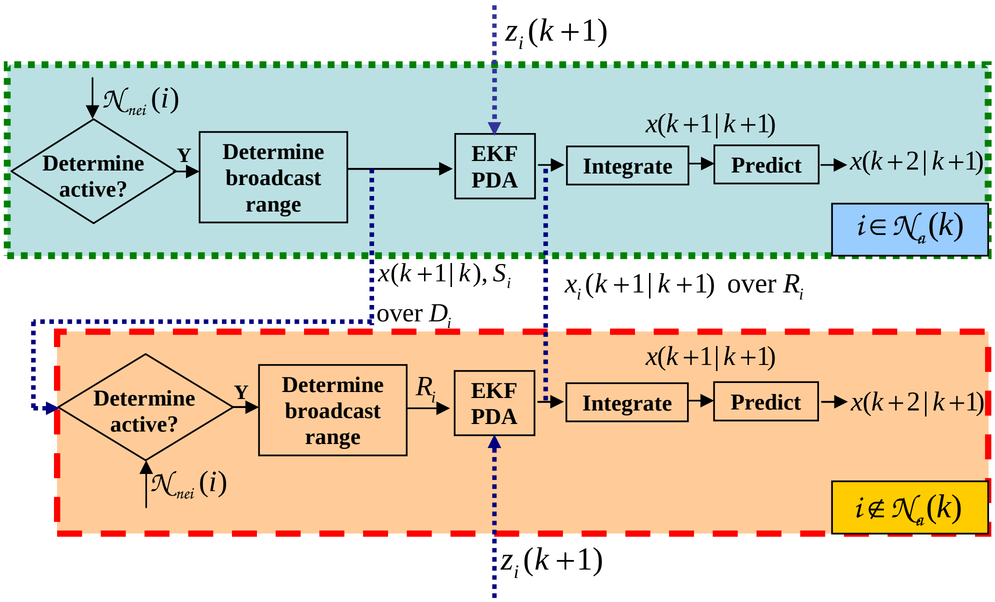
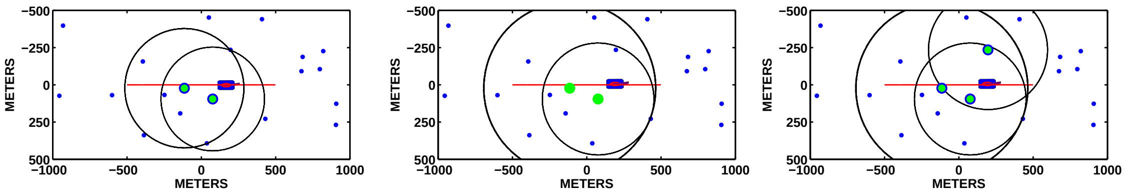


Figure 13: Flowchart of the functions for active/inactive nodes combining GB-PNS-KP with tracking filters where D_i is the broadcast range for information handoff and R_i is the broadcast range for information exchange (see Section 3.5).

For GB-PNS-CR in Fig. 15, the variable r_{nei} does not significantly affect the RMS error. The reason is that r_{nei} only affects the decision making of inactive nodes. The localization performance of GB-PNS-CR is consistent with that of GB-GNS-CR knowing the global information in [31]. Kaplan demonstrated that GB-GNS-CR and GB-ANS-CR are comparable while GB-GNS-CR is more energy efficient [31]. In addition, GB-PNS-CR is more energy efficient as r_{nei} becomes larger (see GB-PNS-CR in Fig. 16). The reason is that a larger r_{nei} slightly reduces node usage (see GB-PNS-CR in Fig. 17) because when r_{nei} is small, a node may decide to become active because it does not know about the existence of another nearby node that provides a better geometry. As r_{nei} becomes larger, the node would know of the existence of its neighbor and remain quiet. On the other hand, GB-PNS-CR/GB-PNS-KP with $r_{nei} = 1000$m does not exactly match the corresponding GNS because the PNS maintains a minimum core using the previous active nodes (see Step 2 in Figs. 10 and 12).



(a) (b) (c)

Figure 14: Illustration of GB-PNS-KP when $N_d = 2$ and $r_{nei} = 1000$m: (a) $\mathcal{N}_a(k)$, (b) a minimum core of $\mathcal{N}_a(k)$ adjusts the broadcast range to transmit the information to the potentially active nodes, and (c) $\mathcal{N}_a(k+1)$ becomes active.

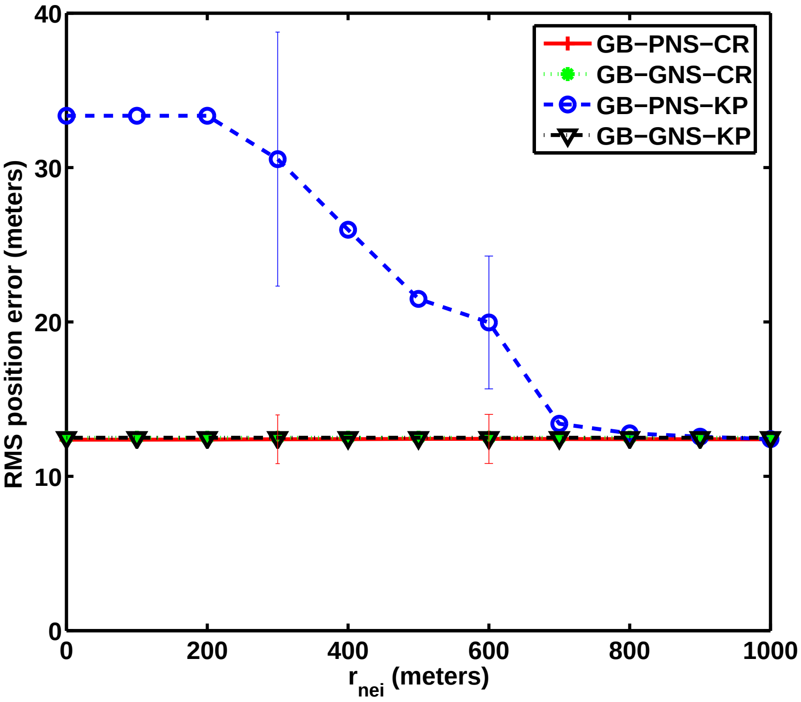


Figure 15: Comparison between GB-PNS-CR and GB-PNS-KP by RMS position error. Error bars computed from 10 node configurations each with 50 runs are shown for GB-PNS-CR and GB-PNS-KP when r_{nei}=300 and 600m.

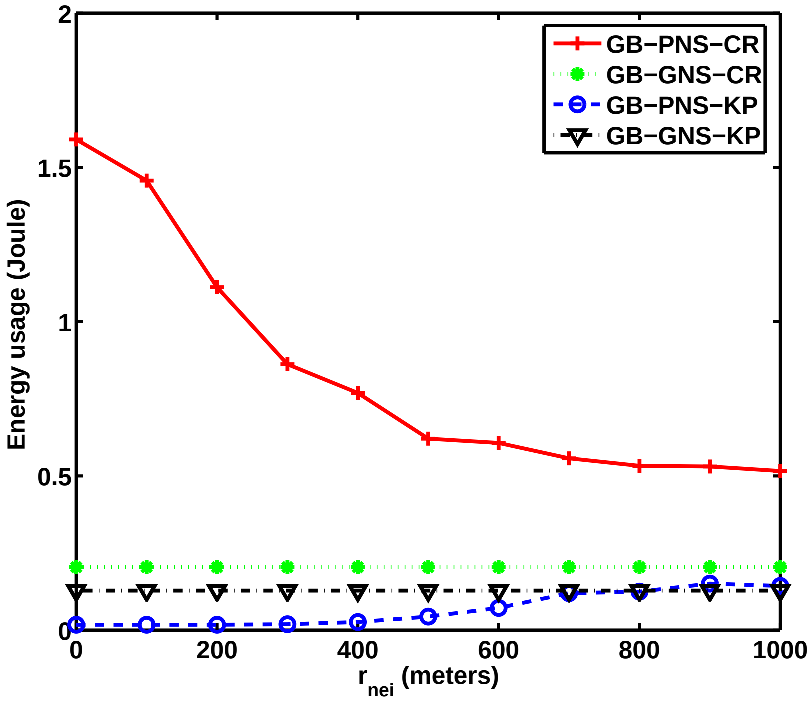


Figure 16: Comparison between GB-PNS-CR and GB-PNS-KP by energy usage; and std. error for GB-PNS-CR: 3.1J (r_{nei}=300m) and 1.8J (r_{nei}=600m); GB-PNS-KP: 0.0003J (r_{nei}=300m) and 0.0051J (r_{nei}=600m).

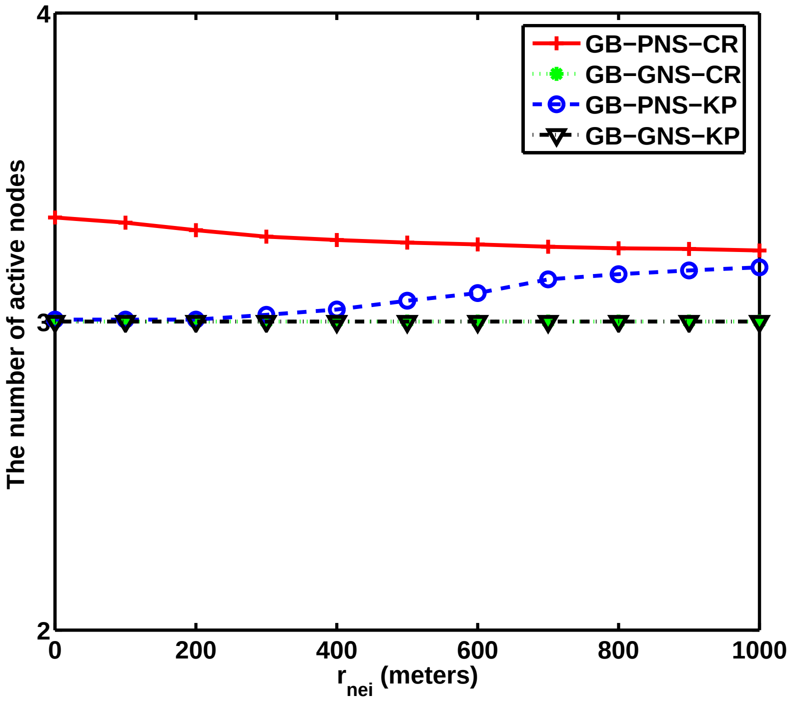


Figure 17: Comparison between GB-PNS-CR and GB-PNS-KP by node usage.

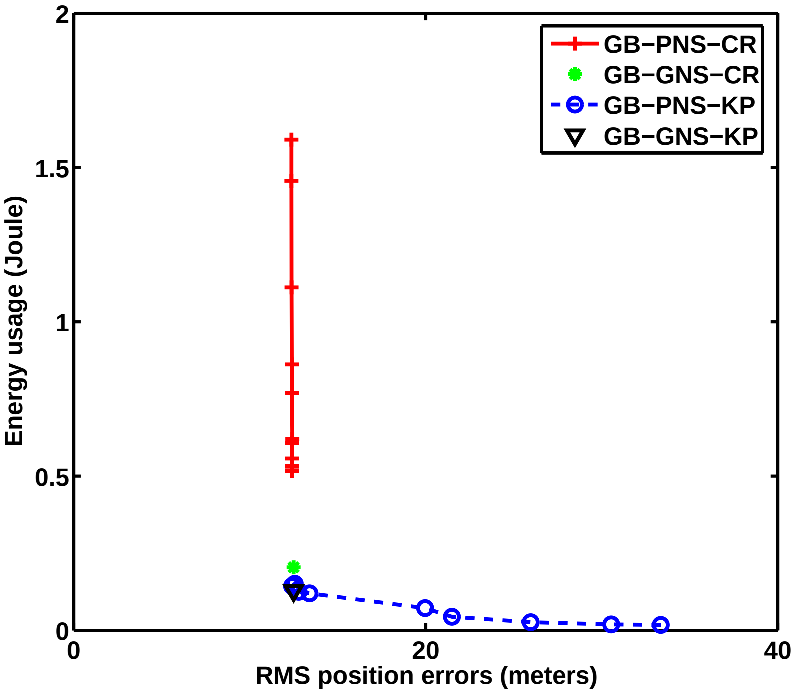


Figure 18: Comparison between GB-PNS-CR and GB-PNS-KP in the RMS vs. lifetime plane.

CHAPTER IV

NODE SELECTION FOR UNATTENDED GROUND SENSOR NETWORKS WHILE INTERROGATING MULTIPLE TARGETS

This chapter investigates the performance of a multiple target tracker that exploits bearings-only measurements from a network of unattended ground sensors (UGS) using real data collected by the U.S Army Research Laboratory (ARL) [43]. To conserve energy while interrogating multiple maneuvering target, the tracker integrates node resource management with either the multiple-mode probabilistic data association (PDA) or the joint probabilistic data association(JPDA) filter. Experiments show that for sufficiently separated targets, global node selection leads to better geolocation performance than the "closest" selection approach when the number of active nodes is set to two per snapshot. A track purity metric is also introduced to quantify the quality of the measurement-to-track association performance of the tracking filter.

4.1 Measurement and dynamic Model

In this section, the position and velocity for the t-th target are labeled as $P^t = [P^t_x, P^t_y]^T$ and $V^t = [V^t_x, V^t_y]^T$, respectively. The state for the t-th target $x^t(k)$ at time k is concatenated by the target position and velocity, i.e. $x^t(k) = [{P^t}^T, {V^t}^T]^T$. The bearing angles are used as the measurements. The UGS network consists of N_s nodes where the j-th node reports m_j measurements at a given snapshot. The l-th measurement reported by the j-th node at snapshot time k is related to the target state via the nonlinear equation

$$z_j^l(k) = H_j(x^{f(l)}(k)) + \eta_j^l(k)$$

where

$$H_j(x^t) = \arctan\left(\frac{P_y^t - S_{j,y}}{P_x^t - S_{j,x}}\right) \tag{26}$$

is the bearing angle and $S_j = [S_{j,x}S_{j,y}]$ is the position of the j-th node. The state index $f(l)$ represents the measurement-to-track association. The measurement error $\eta_j^l(k)$ is modeled as zero mean Gaussian noise with variance σ^2. This error is uncorrelated between the different measurements and nodes, i.e., $E\{\eta_i^p(k)\eta_j^l(k)\} = \sigma^2\delta_{i,j}\delta_{l,p}$.

The target motion can be represented by the coordinated turn (CT) dynamic model where the mode parameter is denoted by the turn rate ω:

$$x(k+1) = F(\omega)x(k) + Av(k+1),$$

where

$$F(\omega) = \begin{bmatrix} 1 & 0 & \frac{\sin\omega T}{\omega} & \frac{\cos\omega T - 1}{\omega} \\ 0 & 1 & \frac{1-\cos\omega T}{\omega} & \frac{\sin\omega T}{\omega} \\ 0 & 0 & \cos\omega T & -\sin\omega T \\ 0 & 0 & \sin\omega T & \cos\omega T \end{bmatrix}, \text{ and } A = \begin{bmatrix} 0.5T^2 & 0 \\ 0 & 0.5T^2 \\ T & 0 \\ 0 & T \end{bmatrix}.$$

The vector $v(k+1)$ is the process noise assumed to be Gaussian with covariance $\sigma_\mu^2 I$. We also need a stationary dynamical model to follow a stationary or low-velocity target, so we use the following state transition matrix

$$F = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}.$$

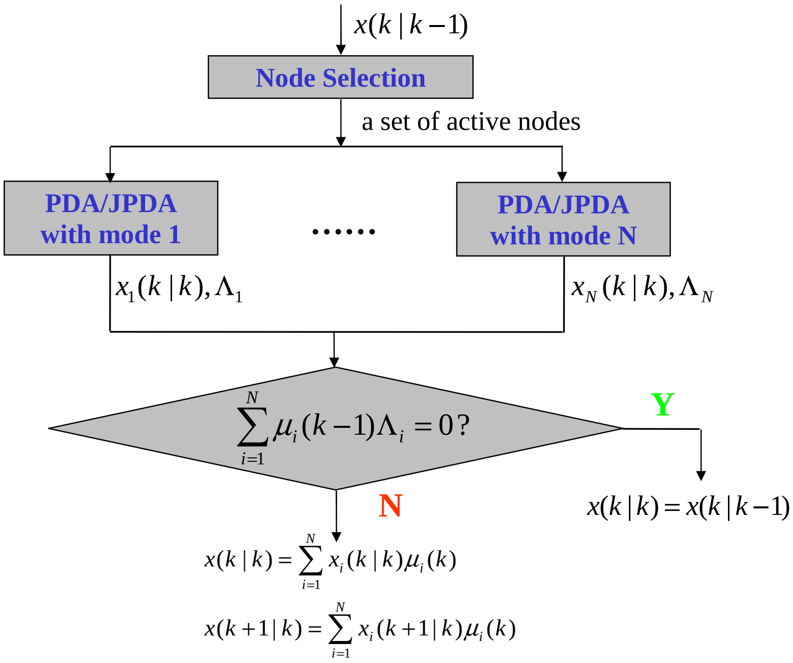


Figure 19: Structure of node selection with MM-PDA or MM-JPDA.

4.2 The Tracker

The measurement equation and the set of dynamical models leads to a bank of extended Kalman filters (EKF). This section discusses how the tracker integrates the Kalman filtering and node selection. Based upon empirical evidence, we assume that the bearing error $\sigma = 5^\circ$ for this work. The subsequent sections provide experimental results for the process noise parameter σ_μ^2 that leads to the smallest root mean squared (RMS) position error.

The integration of the node selection and filtering in the tracker is illustrated in Fig. 19. The initialization methods are described in the following subsection. The subsequent subsections describe the modules in Fig. 19. At this point, the track manager simply maintains the current tracks over the entire data collection interval.

4.2.1 Initialization

Multiple tracks are initialized (see [33]) by minimizing

$$C(P) = \sum_{j=1}^{N_s} \min_{l} |z_j^l(k) - \angle(P - S_j)|^2. \tag{27}$$

The number of bearing measurements m_j varies at different nodes because of false alarms and missed detections. Some of the local minima should correspond to true targets. However, other local minima could appear due to ghosting and noise. For this work, the location of the local minima closest to the ground truth target positions are used to initialize the Kalman filters even though these are, strictly speaking, unknown. Furthermore, the initial velocity is set to zero. We also considered another initialization using the true target positions.

4.2.2 Probabilistic Data Association (PDA) and Joint PDA (JPDA)

In PDA/JPDA, the number of estimated tracks is assumed or maintained by the track manager. In the filtering stage, each track is updated using a weighted sum of measurement residuals via

$$x(k|k) = x(k|k-1) + \sum_{j=1}^{N_s} W_j \sum_{l=1}^{\tilde{m}_j} \beta_j^l (z_j^l(k) - z_j(k|k-1)), \tag{28}$$

$$P^{-1}(k|k) = P^{-1}(k|k-1) + P_d P_g \sum_{j=1}^{N_s} \frac{1}{\sigma_j^2} H_j H_j^T, \tag{29}$$

where P_d represents the probability of detecting a measurement and P_g the probability the measurement passes the gating threshold. The integer m_j is the number of measurements and $\tilde{m}_j$ is the validated measurements defined by P_g. Furthermore, W_j is the Kalman gain (7) and H_j is the Jacobian of the nonlinear measurement equation given by (8), i.e., explicitly,

$$H_j = \frac{1}{r_j} \begin{bmatrix} \sin\phi_j & -\cos\phi_j & 0 & 0 \end{bmatrix}^T,$$

where r_j and ϕ_j are the 2-D polar coordinates for the position of the i-th node relative to the target when no propagation delay is assumed in the measurement equation. Finally, β_j^l is the association probability, i.e., the likelihood that the l-th measurement should be associated to a given track.

Both PDA and JPDA use a gate to eliminate measurement-to-track associations that are clearly poor. Simply given an existing target track t, $\epsilon = \frac{(z_j(k)-z_j^t(k|k-1))^2}{\tilde{\sigma}_{j,t}^2}$ follows a chi-squared distribution with 1-degree of freedom. Note that $\tilde{\sigma}_{j,t}^2 = H_j P(k|k-1){H_j}^T + \sigma_j^2$, where $z_j^t(k|k-1) = h_j(x(k|k-1))$ is the predicted measurement related to target t. The gate probability P_g =Prob($\epsilon < g^2$) is a user defined parameter. Once it is set, g can be computed. Given m_j measurements at the j-th node, the validated measurements for track t are $\{z_j^l(k)|\frac{|z_j^l(k)-z_j^t(k|k-1)|}{\tilde{\sigma}_{j,t}} < g, l = 1, 2, \ldots, m_j\}$. Clearly, the size of the validated measurement set is $\tilde{m}_j$ for a given track.

In PDA, one of the validated measurements is assumed to originate from an existing target, and the remainder are from random clutter, so PDA does not take into account that one of the remaining measurements might come from another target. We assume the detection probability is the same for each node and each target. Assuming uniform clutter distribution, the probability that none of the measurements originate from a target is

$$\beta_j^0 = \frac{b}{b + \sum_{l=1}^{l=\tilde{m}_j} \wedge_j^l},$$

while the probability that measurement l originates from a target is

$$\beta_j^l = \frac{\wedge_j^l}{b + \sum_{l=1}^{l=\tilde{m}_j} \wedge_j^l},$$

where $\wedge_j^l$ is the likelihood that $z_j^l(k)$, the l-th validated measurement from node j, is associated to the track t so that

$$\wedge_j^l = \mathcal{N}(z_j^l(k); z_j^t(k|k-1), \tilde{\sigma}_{j,t}^2). \tag{30}$$

In other words, if the measurement is actually associated to the track, then the measurement residual is zero mean Gaussian with variance $\tilde{\sigma}_{j,t}^2$. Also, $b = \frac{\tilde{m}_j}{V}(1-P_d P_g)/P_d$

where V is the volume of the gate. Explicitly, $V = 2g|\tilde{\sigma}_{j,t}|^{0.5}$. The JPDA tracker is similar to the PDA tracker with the exception of the calculation of the measurement-to-track association probability [2, 61]. Since JPDA takes into account the existence of multiple targets, the measurement-to-track association probability must be marginal, i.e., the probability of track t being associated with measurement l for node j is computed by enumerating all possible joint events that contain association (t, l). Therefore, $\beta_j^l = \sum P\{\theta\}/c$ where θ is a joint event that contains an association (t, l) and c is a normalization constant. A joint event or a hypothesis is a set of measurement-to-track associations which have measurements assigned to either clutter or tracks and each track assigned to only one measurement or declared missed. The probability of any joint event or hypothesis in a non-parameter version is computed by

$$P\{\theta\} = \frac{\phi!}{V^{\phi}} \prod_{l=1}^{m_j} \{\mathcal{N}(z_j^l; z_j^t(k|k-1), \tilde{\sigma}_{j,t}^2)\}^{\tau_l} \times \prod_{t=1}^{N_t} (P_d)^{\delta_t}(1 - P_d)^{1-\delta_t}, \qquad (31)$$

where ϕ is the number of clutter hits, τ_l is a binary number indicating whether measurement l is assigned to track t, δ_t is a binary number indicating whether or not track t is assigned to a measurement, and N_t is the number of tracks. Given a case with a total of four measurements, where Measurements 1 and 4 both fall inside the gates of the two established tracks, but Measurement 2 can only be associated with Track 1 and Measurement 3 can only be associated with Track 2, then there are a total of fourteen hypotheses and four of them contain the association of Track 1 with Measurement 2 (1,2):

- $\theta_1 = (1, 2), (2, 1),$ (clutter, 3&4),
- $\theta_2 = (1, 2), (2, 3),$ (clutter, 1&4),
- $\theta_3 = (1, 2), (2, 4),$ (clutter, 1&3),
- $\theta_4 = (1, 2),$ (2, missed),(clutter,1&3&4).

The JPDA method is better than the PDA because of the more precise calculation of the association probabilities.

4.2.3 Multiple-mode(MM) tracking

The MM tracker [2] employs a bank of mode-matched filters that can include a standard EKF, a PDA filter or a JPDA filter (see Figure 19). The initial state and covariance for each mode-matched filter is identical and given by the previous global state and covariance in the MM tracker. The global state update is the weighted sum of the state outputs of each mode-matched filter

$$x(k|k) = \sum_{i=1}^{N} x_i(k|k)\mu_i(k), \tag{32}$$

where $\mu_i(k)$ is the mode probability, $x_i(k|k)$ is the state output of the ith mode-matched filter and N is the number of mode-matched filters. Likewise, the covariance update is given by

$$P(k|k) = \sum_{i=1}^{N}\{(x(k|k) - x_i(k|k))(x(k|k) - x_i(k|k))^T + P_i(k|k)\}\mu_i(k), \tag{33}$$

where $P_i(k|k)$ is the covariance output of the i-th mode-matched filter.

The weights $\mu_i(k)$ are derived from the likelihood of each mode representing the true dynamics of the target. The measurement-to-track likelihood is computed via (30) for each mode i and is explicitly labeled as $\wedge_{ij}^{l}$. Using the additive fusion strategy suggested in [11], the likelihood that the target dynamics follow mode i given the current measurement set is

$$\wedge_i(k) = \sum_{j=1}^{N_s} \sum_{l=1}^{\tilde{m}_j} \wedge_{ij}^{l}. \tag{34}$$

The mode likelihood measures the difference between the assumed model expressed by the predicted measurement and the true model denoted by the received measurement. When this difference gets larger, the likelihood that the current measurement follows the assumed model gets smaller. Finally, the mode probability $\mu_i(k)$ is updated via a Bayesian rule where $\mu_i(k-1)$ is the prior probability, i.e.,

$$\mu_i(k) = \frac{\wedge_i(k)\mu_i(k-1)}{\sum_{j=1}^{N} \wedge_j(k)\mu_j(k-1)}. \tag{35}$$

4.2.4 Node selection

A node selection algorithm which is embedded in a resource manager determines which subset of nodes will be active for a given snapshot of data collection. In this chapter, we use the global node selection (GNS) approach [28]. The selection is global in the sense that each node knows the exact locations of all other nodes in the network.

In this chapter, GNS is a nearly optimal approach to determine which active set of nodes $\mathcal{N}_a$ provides the best geometry to localize a target, i.e., GNS is the same as GB-GNS in Chapter 3. Since the GB metric is the one without accounting for the prior measurements (20), at this moment we are not worrying about the transmission range control for multiple targets (a subject of future work). The GNS is a Greedy simplex approach to find the best N_a nodes. It starts by determining the best two nodes via exhaustive search. Then, it adds one node at a time to the active set. Finally, single node replacements that reduce (20) are performed until that strategy is exhausted. The GNS approach reduces the computational complexity from $O(N^{N_a})$ with exhaustive search to $O(N^2)$. Effectively, the GNS method selects nodes that surround the target and are within close proximity of the target. As a baseline to compare with the GNS method, we also consider the 'closest' node selection approach that selects the N_a nodes which lie closest to the predicted target location.

Figure 19 shows how node selection is integrated into the tracking filter. Prediction is critical in the combined node selection and MM-PDA/MM-JPDA tracker for the following reasons: 1) Node selection algorithms use the prediction to determine which subset of nodes will be active, and 2) when none of the current measurements lie in the gate for a given mode i, the mode likelihood is zero, i.e., $\wedge_i(k) = 0$. When $\sum_{i=1}^{N} \wedge_i(k)\mu_i(k-1) = 0$, we can infer that none of the assumed modes is correct. In this case, it is better to use the predicted target state and covariance instead of using the filtered ones. The predicted state and covariance in the MM-PDA/MM-JPDA with the node selection are given by (see [7]):

$$x(k|k-1) = \sum_{i=1}^{N} x_i(k|k-1)u_i(k-1),$$

$$P(k|k-1) = \sum_{i=1}^{N} P_i(k|k-1)u_i(k-1),$$

where $x_i(k|k-1)$ and $P_i(k|k-1)$ are the predictions for i-th mode.

4.3 Track metrics

To compare trackers, we score the resulting tracks via RMS position error and track purity. These metrics are clearly defined in the following subsections.

4.3.1 RMS error

The RMS position error is simply the sum of the position errors between the tracks and the corresponding target. In other words, at snapshot k, the error is

$$RMS(k) = \frac{1}{N_t} \sum_{t=1}^{N_t} \left(\sum_{s=1}^{2} ([x^t(k|k)]_s - [P^t(k)]_s)^2 \right)^{\frac{1}{2}}, \tag{36}$$

or

$$RMS(k) = \frac{1}{N_t} \sum_{t=1}^{N_t} \min_{1 \leq f \leq M_t} \left(\sum_{s=1}^{2} ([x^t(k|k)]_s - [P^f(k)]_s)^2 \right)^{\frac{1}{2}}, \tag{37}$$

where $[x]_s$ extracts the s-th element of vector x, N_t is the assumed number of targets usually computed in the track formation stage, and M_t is the exact number of targets. The RMS error computed by (37) must give smaller values than the RMS error in (36). In either case, the average RMS error over all snapshots is

$$e = \frac{1}{N_k} \sum_{k=1}^{N_k} RMS(k).$$

4.3.2 Track purity

In the experiments in Section 4.4, we report a simple purity metric that could indicate a track switch or merge. The intent of the metric is to quantify the accuracy of a measurement-to-track association algorithm in a multi-node multi-mode multi-target tracker. The correct measurement-to-track association is derived using the ground truth provided by the GPS units located in the targets. Let $C_{t,j}$ be the correct association for track t at node j. If some measurements from node j pass the gate for track t, then

$$C_{t,j} = \arg\min_{l \in \{l : |z_j^l - \theta_t| < \tau\}} |z_j^l - \theta_t|. \tag{38}$$

Otherwise, $C_{t,j} = 0$, which indicates that target t is not detected. For mode i and a chosen node $j \in \mathcal{N}_a$, the association of the target t with the measurement l is $\beta_{t,i,j}^l$ and $\sum_{l=1}^{m_j} \beta_{t,i,j}^l \leq 1$. The purity of the measurement-to-track association is defined as:

$$Q_{t,i,j} = \begin{cases} \beta_{t,i,j}^{C_{t,j}} & \text{if } C_{t,j} \neq 0 \\ 1 - \sum_{l=1}^{m_j} \beta_{t,i,j}^l & \text{otherwise} \end{cases} \tag{39}$$

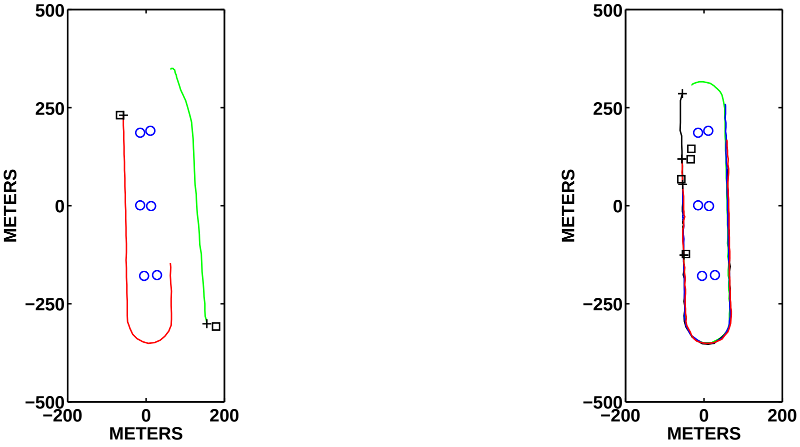
Considering all active nodes, the average purity is

$$Q_{t,i} = \frac{1}{N_a} \sum_{j \in \mathcal{N}_a} Q_{t,i,j}.$$

Considering all dynamic modes, the average purity of the measurement-to-track association is

$$Q_t = \sum_{i=1}^{N} \mu_i(k) Q_{t,i}.$$

Finally, the purity of the measurement-to-target associations Q_m is the average value of Q_t over the number of the targets.



(a) (b)

Figure 20: Node locations and tracks: (a) Scenario 1 with two vehicles on separate paths, and (b) Scenario 2 with four vehicles traveling around the oval track. The circles represent the node locations, the plus symbols represent the initial target positions, the square symbols represent the estimated target positions, and the lines represent the target trajectories.

4.4 Experiments

4.4.1 Real data test

The real data, collected by the U.S. Army Research Laboratory (ARL) at Aberdeen Proving Grounds, contains multiple targets traveling along an oval track or an adjacent road. Six acoustic nodes were situated in the middle of the oval track. The targets were fitted with GPS to obtain ground truth information. Figure 20 shows the tracks of targets for two different test scenarios and the initial position estimates as computed by the method in Section 4.2.1.

ARL processed the raw data using an incoherent wideband minimum variance distortionless response (MVDR) beamformer [67] to obtain bearing measurements. Figures 21 and 22 show the bearing measurements obtained by two different nodes for the two different test scenarios. In Scenario 1, one target is traveling along the oval track, while the other target is traveling down the road parallel to the oval track. In Scenario 2, a convoy of four vehicles is traveling around the track. The

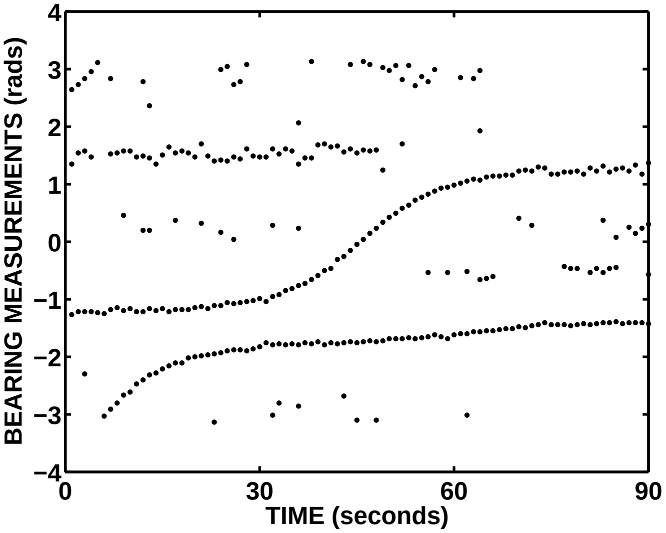


Figure 21: Bearing measurement output of MVDR for the top right node in Scenario 1.

measurements are assumed to have 5° errors for each snapshot. In addition, at each node, the bearing measurements might be missing or false measurements could be detected. In Fig. 22, two measurement tracks are obvious for k between 40 to 130. However, a large number of false measurement are also obtained. On the other hand, three measurement tracks exist for k between 130 to 200, but in these snapshots, the targets are not always detected.

For both scenarios, we initialize two tracks. In Scenario 1 the number of targets is correctly modeled, but for Scenario 2, it is underestimated. The purpose for underestimating the number of targets in Scenario 2 is to avoid the occurrence of track swaps or merges, which leads to poor geolocation performance. For Scenario 2, we either track the front and back targets, or the two middle targets. The middle targets pose a greater challenge to the measurement-to-track association portion of the track filter.

In these experiments, P_d and P_g are both fixed to be 0.9999 for Scenario 1, and 0.98 for Scenario 2. Here, we do not consider adaptively changing P_d or P_g. A bank of four mode-matched filters are used where three modes represent the CT model for

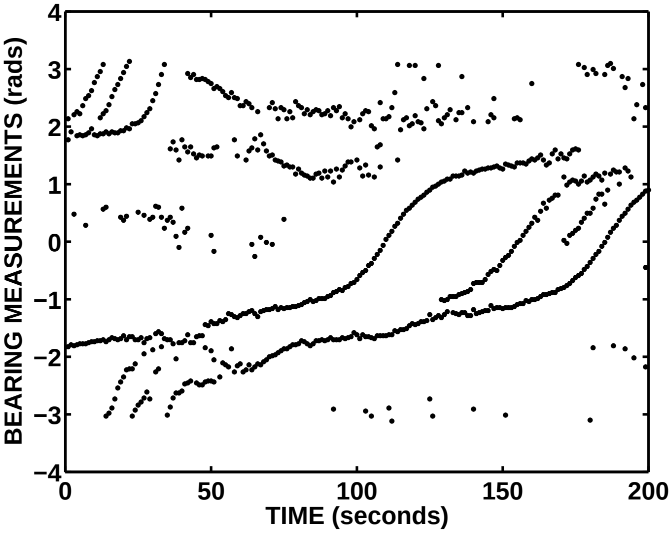


Figure 22: Bearing measurement output of MVDR for the middle left node in Scenario 2.

$\omega \in \{-20°, 0°, 20°\}$, and the fourth mode represents a stationary dynamical model. The initial mode probability $\mu_i(0)$ is set uniformly so that $\mu_i(0) = 1/4$, $i = 1, 2, 3, 4$.

Figure 23 shows the estimated tracks using MM-PDA or MM-JPDA when all nodes are active and the process noise parameter σ_ν is set to a value that minimizes the RMS position error. We considered eleven different values from σ_ν between 1 m^2/s to 21 m^2/s in steps of 2 m^2/s. The target states were initialized via (27). The figure shows that the MM-PDA method has the most severe adaptation delay around the turns while the MM-JPDA has very little adaptation delay except when tracking the middle targets in Scenario 2.

Next, we evaluated the multiple target trackers using the GNS method for different values of N_a. We also considered a simplified node selection method that selects the closest N_a nodes to the predicted target positions. Figs. 24-26 show the average RMS position errors via (36) for the different approaches using either (27) or the true target positions for initialization. Again, the best process noise is used. For Scenario 1, it is clear in Fig. 24 that the MM-JPDA is more effective than the MM-PDA and that GNS outperforms the "closest" selection approach when $N_a = 2$ for different initializations

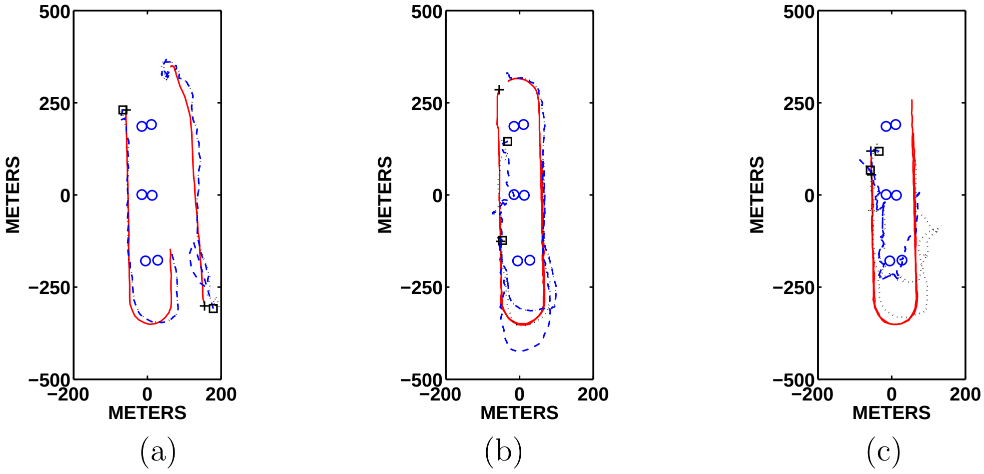
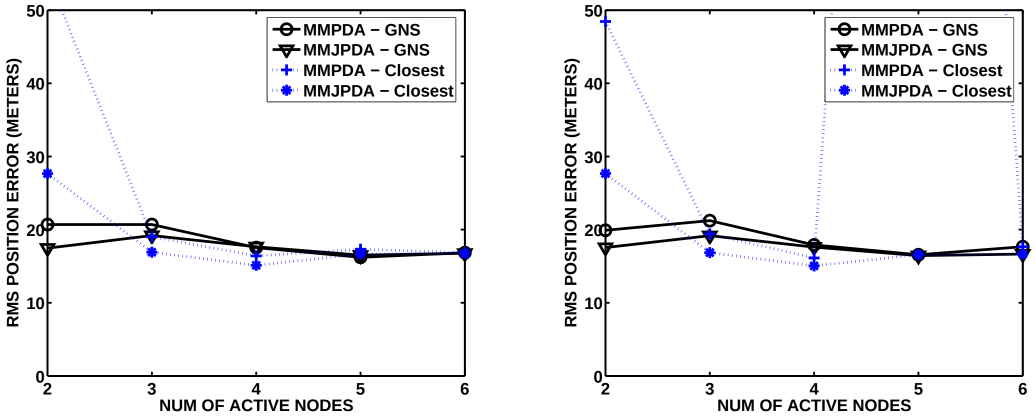


Figure 23: Estimated tracks: (a) Scenario 1, (b) Scenario 2 for the two end targets, and (c) Scenario 2 for the two middle targets. In each case, the dotted and dashed lines represent MM-JPDA and MM-PDA tracks, respectively.

of the filters.

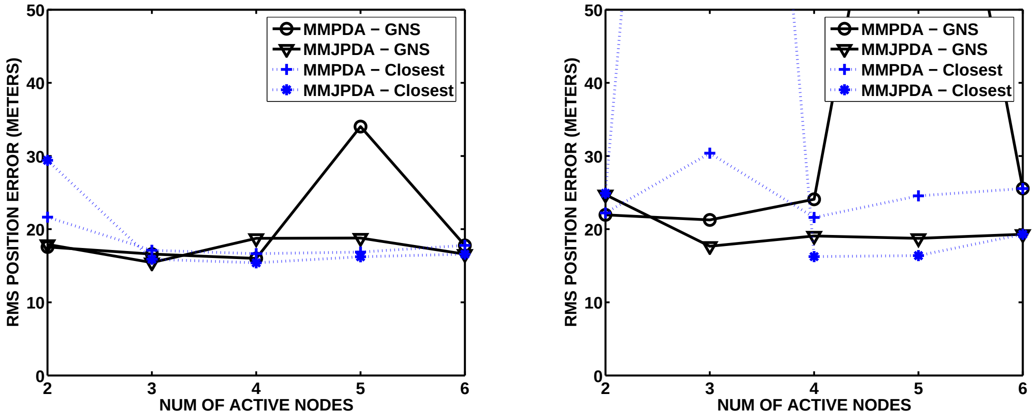
For Scenario 2, we intend to track the first and last targets of the convoy, or the middle two targets along the oval tracks. Figure 25 shows that when tracking the end targets, GNS is able to maintain localization performance as N_a goes to two. MM-JPDA with GNS is robust even when the initial guesses are noisy. However, other combinations of track filters and node selection are poor at some values of N_a. Figure 26 shows that when tracking the middle two targets in Scenario 2, the average RMS errors via (36) are almost always more than 70 meters. Note that the average distances between two adjacent targets from first to last are 107, 77 and 201 meters. Inspection of the tracks actually indicate that track mergence and swaps occur during the lifetime of the tracks while attempting to follow the middle targets. Comparing Fig. 26 and Fig. 27, we could infer that track merges and swaps occur.

Tables 2 and 3 quantify the RMS error via (36) or (37), and track purity performance of MM-PDA or MM-JPDA with different nodes selection methods when



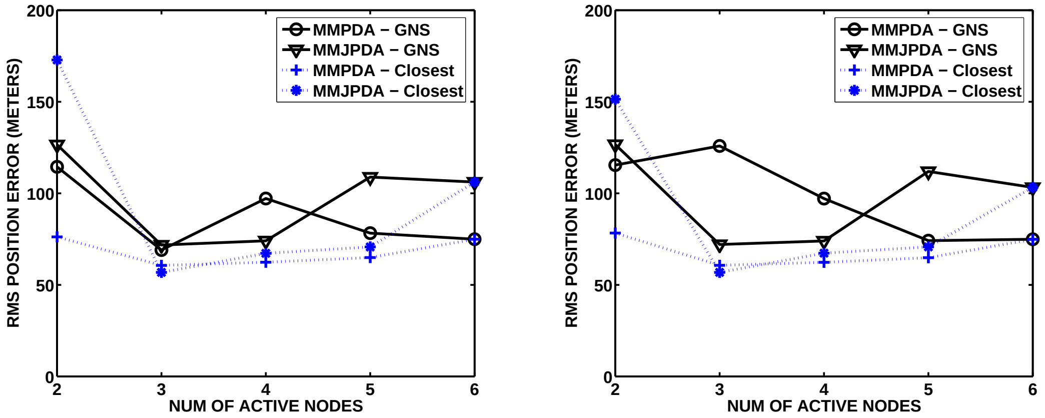
(a) (b)

Figure 24: Average RMS errors via (36) for Scenario 1 by initializing the track filters using (a) the true target positions, or (b) estimated target positions via (27).



(a) (b)

Figure 25: Average RMS errors via (36) for Scenario 2 to track first and bottom targets by initializing the track filters using (a) the true target positions, or (b) estimated target positions via (27).



(a) (b)

Figure 26: Average RMS errors via (36) for Scenario 2 to track the middle two targets by initializing the track filters using (a) the true target positions, or (b) estimated target positions via (27). Note the vertical scale change with respect to Figs. 25 and 26.

$N_a = 2$. Usually, a higher purity score translates to a lower RMS position error. It is noted that a high purity score with large RMS errors is possible because the collection geometry is not well spread out. For example, in Scenario 1 the MM-PDA using the "closest" method leads to the worst estimation performance, but its purity Q_m is above 0.9. The MM-JPDA using GNS has the smallest RMS errors and highest Q_m no matter how the filters are initialized. In Scenario 2, when tracking the first and bottom targets, the purity Q_m is at least 0.8. When tracking the middle two targets, the purity, Q_m is poor and below 0.5. The poor purity explains the poor RMS error values via (36), while smaller RMS error via (37) explains the appearance of the track merges and swaps. The MMPDA or MMJPDA with the GNS node selection to choose even a small number of active nodes would not have difficulties following a maneuvering motion if nodes were distributed around tracks or a better node-target geometry was formed. Since the node-target geometry is set up poorly, the combined tracker has trouble following the intended tracks when turning around the corner. Consequently, trackers could switch, which leads to large RMS errors.

Table 2: Track purity and corresponding average RMS positions errors with $N_a = 2$ when initializing the filters using true target positions.

	Scenario 1			
	PDA GNS	JPDA GNS	PDA closest	JPDA closest
RMS err.(m) via (36)	20.69	17.46	54.93	27.65
RMS err.(m) via (37)	20.69	17.46	54.93	27.65
Q_m	0.970	0.981	0.966	0.969
	Scenario 2 to track first and bottom targets			
	PDA GNS	JPDA GNS	PDA closest	JPDA closest
RMS err.(m) via (36)	17.55	17.91	21.63	29.41
RMS err.(m) via (37)	17.55	17.91	21.63	27.63
Q_m	0.935	0.932	0.929	0.886
	Scenario 2 to track middle two targets			
	PDA GNS	JPDA GNS	PDA closest	JPDA closest
RMS err.(m) via (36)	114.4	126.5	76.19	172.8
RMS err.(m) via (37)	47.46	25.77	41.82	51.32
Q_m	0.337	0.242	0.429	0.108

Table 3: Track purity and corresponding average RMS positions errors with $N_a = 2$ when initializing the filters via (27).

	Scenario 1			
	PDA GNS	JPDA GNS	PDA closest	JPDA closest
RMS err.(m) via (36)	19.90	17.55	48.45	27.66
RMS err.(m) via (37)	19.90	17.55	48.45	27.66
Q_m	0.968	0.981	0.944	0.967
	Scenario 2 to track first and last targets			
	PDA GNS	JPDA GNS	PDA closest	JPDA closest
RMS err.(m) via (36)	21.95	24.71	22.21	24.90
RMS err.(m) via (37)	19.02	22.48	18.67	18.57
Q_m	0.905	0.890	0.923	0.903
	Scenario 2 to track middle two targets			
	PDA GNS	JPDA GNS	PDA closest	JPDA closest
RMS err.(m) via (36)	115.4	126.6	78.28	151.4
RMS err.(m) via (37)	47.65	25.89	58.74	75.63
Q_m	0.339	0.243	0.427	0.138

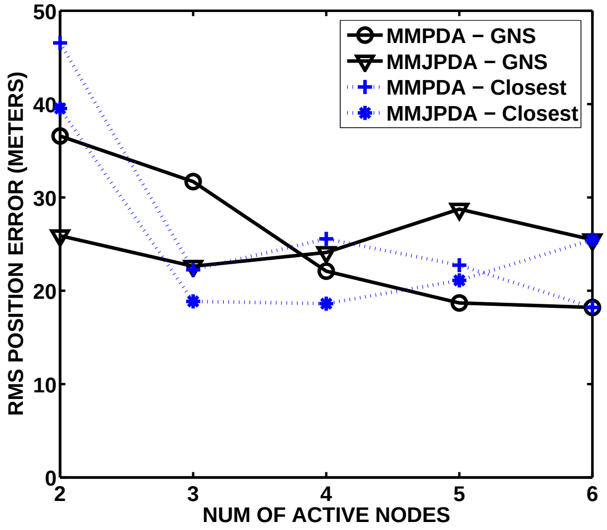


Figure 27: Average RMS errors via (37) for Scenario 2 to track the middle two targets by initializing the track filters using estimated target positions via (27).

4.4.2 Simulated data test

Using real data, if two targets are separated by less than seventy meters, the node configuration where six nodes are placed as in Fig. 20 is the reason why the performance is poor. We can test the situation where there are more available nodes by generating simulated bearing measurements. The objective is to overcome the data association problem by using a more intelligent node placement in the field. In the test below, we consider a node configuration where fourteen nodes are set up as shown in Fig. 28.

The simulated bearing measurements obtained at each node in Fig. 29 may include measurements that truly originated from targets, merged measurements when two targets are close, or false measurements due to noise. In addition, targets may not be detected. The measurement at node i, $\tilde{\theta}_i$, could be:

1. A measurement that truly originated from a detected target t: $\theta_{i,t} + n_i$, where $\theta_{i,t}$ is the noise-free bearing angle for target t and n_i is Gaussian noise with zero mean and variance σ_i . We set $\sigma_i = 2$ for all nodes.

2. A merged measurement when two targets are close. We set the cell resolution equal to 5° . In other words, if $|\theta_{i,1} - \theta_{i,2}| \leq 5^\circ$, then $\tilde{\theta}_i = \frac{\theta_{i,1}+\theta_{i,2}}{2} + n_i$.

3. A false measurement due to noise. The number of false measurements follows a Poisson distribution with density $\lambda = 0.6$, which means in one snapshot of interest, there are on average 0.6 false measurements uniformly distributed in the measurement space.

The simulated measurements obtained at the top right node are shown in Fig. 29. We initially intend to track the two middle targets using the true target position and a nonzero velocity as the initial state guesses. We also artificially design a smaller initial error covariance to emphasize the contribution of the measurements. Otherwise, if the initial error covariance is too large, even a measurement very close to the predicted measurement will not be given the relatively large weight. Clearly, the middle targets pose a greater challenge to the measurement-to-target association portion of the tracker. The tracker parameters P_d and P_g and are fixed to be 0.999. A bank of four mode-matched filters is used where three modes represent the CT model with turn rates as mode parameters and the final mode represents a stationary dynamical model. The initial mode probability is set uniformly. Figures 30-31 show the average RMS position errors computed by (36) or (37) using different node configurations to track the middle two targets. An appropriate process noise is searched over 1 m/s^2 to 21 m/s^2 with grid distance 2 m/s^2 to minimize the RMS error. When tracking the close targets, the MMJPDA with the GNS is able to maintain localization performances as N_a goes to two. As expected, the fourteen-node configuration performs better than the six-node configuration.

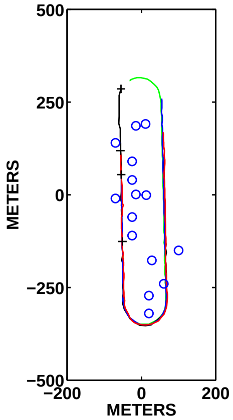


Figure 28: Fourteen-node configuration where the circles represent the node locations, the plus symbols represent the initial positions of the four targets, and the lines represent the target trajectories.

4.4.3 Targets within close proximity

The structure in Fig. 19 combines node selection algorithms with multiple-mode multitarget trackers. We note that the GB metric (20) is a function of the node configuration $\mathcal{N}$ and the predicted state of the target t, i.e., by $\rho(\mathcal{N}, t)$. The node selection algorithm chooses the best node set per target per snapshot:

$$\mathcal{N}_t^* = \arg\min_{\mathcal{N}} \rho(\mathcal{N}, t). \tag{40}$$

Let N_d be the desired number of active nodes per target per snapshot, M_t be the number of detected targets, and D_n be the number of the distinct chosen nodes per snapshot because $\mathcal{N}_1^*$ and $\mathcal{N}_2^*$ have possible overlaps. Then we have the relationship $N_d \leq D_n \leq M_t \times N_d$. Especially when targets are within close proximity, D_n could be reduced to N_d.

In the following paragraph, we introduce a way to conserve node usage while reaching comparable estimation performance by averaging $\rho(\mathcal{N}, t)$ over all the detected targets. The same node set $\mathcal{N}^*$ will end up being chosen when two targets are separated within 50m:

$$\mathcal{N}^* = \arg\min_{\mathcal{N}} \sum_{t=1}^{M_t} \rho(\mathcal{N}, t). \tag{41}$$

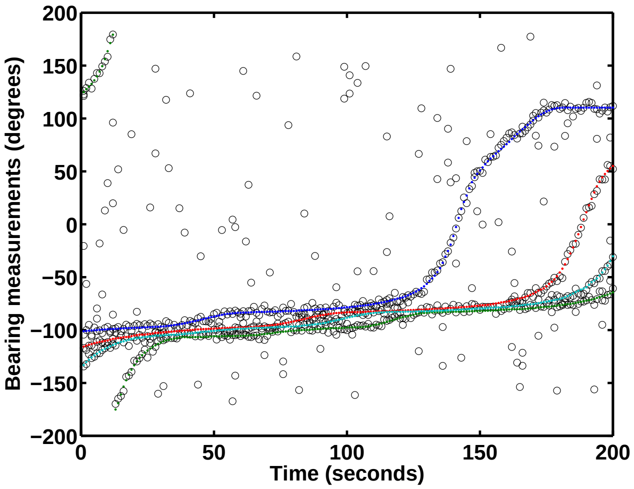
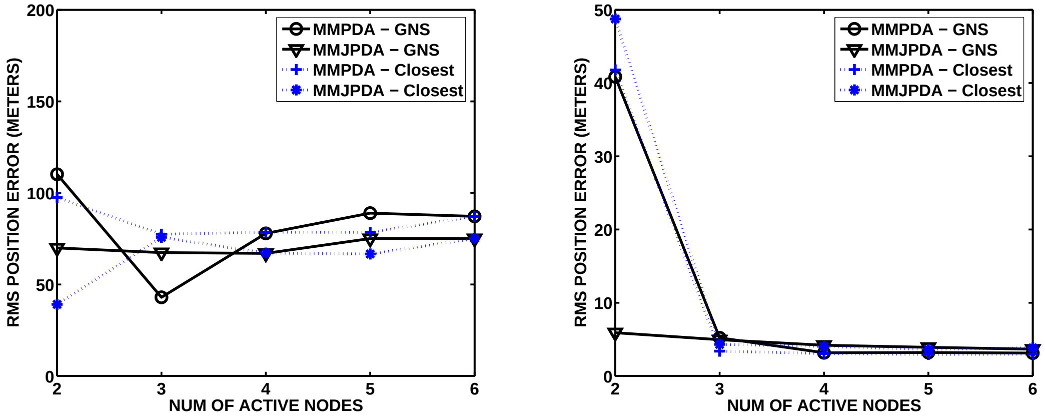


Figure 29: Simulated bearing measurements at the top right node in Fig. 28.

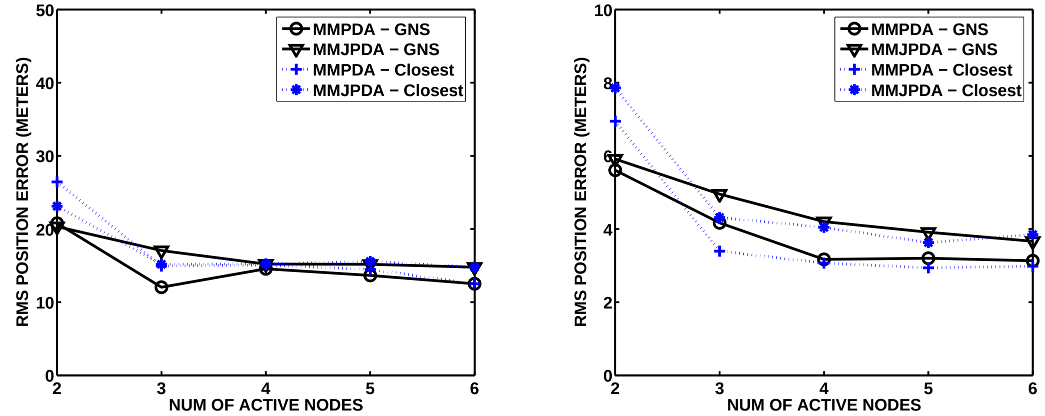
In the simulation, two targets are moving at a constant velocity of 10m/s. For a node configuration with twenty randomly distributed nodes as shown in Fig. 32, we run fifty Monte Carlo simulations with 5-degree bearing errors. The cell resolution is 10 degrees, which means that when two targets are separated by 10 degrees or less, only one measurement will be given at a given node. The Poisson density of the number of false measurements is 0.6. The maximum and the minimum separation between targets is 360m and 5m, respectively. Tracking lasts 100 seconds with 1 second as the time interval to update the trackers. Targets are within close proximity or separated by less than 50m for about 13 seconds. We compare two global node selection algorithms: one choosing nodes per target per snapshot by (40), the other choosing nodes when two estimated targets are separated by less than 50m by (41). Table 4 shows that choosing $N_d = 3$ nodes over all the targets per snapshot for targets within close proximity saves a little bit node usage D_n than choosing N_d nodes per target per snapshot while reaching comparable estimation performance.



(a)

(b)

Figure 30: Average RMS errors via (36) with simulated measurements to track the middle two targets by initializing the track filters using the true target positions using (a) six nodes in Fig. 20 , or (b) fourteen node in Fig. 28. Note the different vertical scales.



(a)

(b)

Figure 31: Average RMS errors via (37) with simulated measurements to track middle two targets by initializing the track filters using the true target positions using (a) six nodes in Fig. 20 , or (b) fourteen node in Fig. 28. Note the different vertical scales.

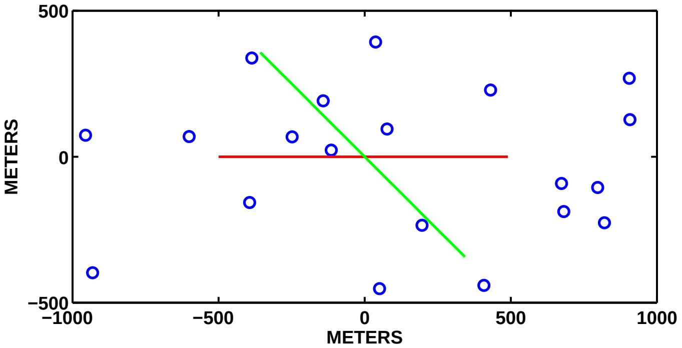


Figure 32: Node configuration with 20 randomly place nodes and two constant-velocity tracks.

Table 4: Conservation of node usage

When two targets are separated within 50m	Avg. RMS error(m) over 100s	Avg.D_n over 100s
Choose $N_d = 3$ nodes per target per snapshot	33.1333	4.3869
Choose $N_d = 3$ nodes over all targets per snapshot	32.3354	4.3358

CHAPTER V

MULTIOBJECTIVE ENERGY-AWARE NODE SELECTION

This chapter develops an energy-aware resource management strategy for a wireless sensor network of bearings-only sensors [41]. Specifically, the resource manager determines which nodes actively sense and communicate during each snapshot in order to achieve a tolerable level of geolocalization accuracy while attempting to maximize the effective lifetime of the network. Unlike other methods that use the total energy consumed for the given snapshot as an energy-based metric, a new energy-based (EB) metric can achieve load balancing of the nodes without resorting to computationally demanding non-myopic optimization. Simulation results show that EB provides longer lifetime than the GB metric discussed in Chapter 3. We consider an adaptive transmission range control based upon the remaining battery level and the knowledge of the physical location of nodes in the network. The activation decision is performed in a decentralized manner over the active set of nodes. Each active node transmits just far enough to reach all the other active nodes for information sharing and the potentially active nodes for information handoff. In determining the active set, both global and local approaches are considered. The global approach assumes each node knows the physical location of every other node in the network. On the other hand, the local approach assumes that a node only knows the location of itself, the previous active set, and neighboring nodes.

5.1 Introduction

Recently, sensor allocation has become a hot topic in resource management [36, 35, 63, 47, 30, 31]. Motivated by information theory, Kreucher proposed myopic (short-term) and non-myopic (long-term) sensor scheduling methods to task the sensor for multiple-target tracking where the action the sensor takes results in the maximum amount of Renyi's information gain by approximating Bellman's equation [36, 35]. Wang [63], on the other hand, proposed an efficient heuristic myopic sensor selection algorithm to choose one sensor at each time step so that the chosen sensor would yield the largest entropy reduction of the target location distribution. In a similar spirit, Liu *et al.* [47] determine the sensor that minimizes the entropy of the posterior target distribution in a sequential Bayesian filter. Kaplan proposed a more computationally efficient geometry-based (GB) node selection algorithm that selects the N_a active nodes that minimize the posterior RMS error derived from the Kalman filter [30, 31].

While researchers in signal processing are engaged in finding node selection methods to minimize the localization errors of trackers, those in communication systems have preferred to minimize transmission energy (MTE) or maximize the lifetime of the sensor network with respect to routing because transmitting information across the network dominates the energy consumption [60, 48, 10]. Maximizing the lifetime of the network is more desirable for routing than minimizing the total energy usage. The reason is that some nodes will be overused and run out of energy very quickly because the MTE metric does not consider the current remaining battery level of each node. Chang [10] experimentally demonstrated that the average gain in the system lifetime is around 50 percent using energy-aware routing. Therefore, maximizing the tracking lifetime using energy-aware metrics, where the lifetime is defined as the first transmission failure, along with minimizing the localization error are competing multiple objectives in sensor allocation.

Combining two different objectives to reach a certain level of estimation performance at a cost of a small amount of energy consumption has also drawn great interest from researchers[15]. Chhetri introduced a composite optimization where a sensor action minimizes the total energy from the current time step k up to a future time step $k + M$ while maintaining the tracking error below a predefined limit M time steps in the future [15]. Although the results show that by increasing the error threshold the average energy usage would decrease, two types of sensor nodes are required, one collecting only bearing measurements, the other acting as the central powerful decision maker to perform scheduling. The work did not consider how much energy remains in the battery of the sensor node during tracking, but did assume that the global topology of the network is available for evaluating the objective functions.

In this work, we first propose an energy-based metric that represents the number of snapshots remaining for the active node set. Unlike other methods that use the total energy consumed for the given snapshot as the metric, the new metric can achieve load balancing of the nodes without resorting to computationally demanding non-myopic optimization. For example, if targets are traveling down a road, the new metric is able to limit the overuse of nodes alongside the road so that communication consumption increases gracefully when nodes further from the road must be employed. In contrast, an energy consumption metric would require a k-horizon optimization to catch the effects of Greedy use of the roadside nodes. Optimization of the EB metric could be implemented via a Greedy search in practice. Simulation results show that the EB metric provides longer lifetime than an existing geometry-based (GB) metric, which was proposed by Kaplan in [30] and minimizes the posterior RMS position error.

How the energy consumption or the tracking lifetime is measured is entangled with how the activation decision is made in a decentralized manner, or how the broadcast range is determined based upon the available location information and current battery information. Unlike other transmission range controls where active nodes transmit signals to other nodes only within a certain fixed range, we consider an adaptive transmission range control based upon the current remaining battery level in the active nodes and the knowledge of physical location of nodes in the network. The activation decision is performed in a decentralized manner over the active set of nodes after the currently obtained information is shared among the active set of nodes. To this end, each active node determines the next active set by evaluating a metric, and decides whether it will remain active or whether it should wake-up and hand off information to inactive nodes that are members of the next active set. Energy is consumed during information sharing over the currently active set and during information handoff to the next active set. Therefore, the broadcast range should be just long enough to reach all the active nodes for information sharing or reach the next active set for information handoff. The next active set must be known, which means that the physical location information of the next active set must be available at the presently active set. It must also be reachable in the sense that the next active set must be within the maximum reach of the active nodes according to their remaining battery level.

Another contribution of this work is to provide practical ways to optimize a constraint metric where one metric is used as the optimization metric while the other obeys a constraint. We discuss three algorithms to search for the best set of nodes given the constraint. Listed in descending order of computational complexity, they are 1) exhaustive search, 2) sequential search, and 3) Greedy search. The simulations indicate that all search methods provide comparable network lifetime and geolocation performance. However, the Greedy search is typically three times faster than sequential search, and the exhaustive search is prohibitively slow. In the simulations, the effective lifetime of the sensor network is measured by two definitions: Type 1) the time when at least one active nodes has too little energy to reach the other active nodes, and Type 2) the time it is not possible to meet the location constraint even when using all the nodes.

The process of determining which nodes are active and how far to communicate is performed at each node. Specifically, each node uses knowledge about the other nodes in the network to search for the best active set. If the node belongs in that best set, it collects and communicates a measurement over the next snapshot. Otherwise, it conserves its battery. In the following sections, we investigate both global and neighborhood-based selection approaches.

5.2 Background and Metrics

The objective of this work is to investigate node selection algorithms that use different metrics and to embed them into existing trackers, such as Kalman filters, so that we can evaluate the localization performance and the tracking lifetime of the system. A node selection algorithm embedded in the resource manager to determine which set of nodes should be active has to be performed at each node in a distributed system. In a distributed architecture, each node must also be capable of implementing tracking filters to extract useful information out of the locally obtained measurements, broadcast the intermediate results, integrate them into a global state and predict the state for the next snapshot. The global predicted information including the predicted target state and the predicted error covariance would then be used in the node selection for the next snapshot.

The bearing measurement obtained at the i-th node for a given snapshot is the true retarded bearing angle embedded in additive white Gaussian noise. The retarded bearing angle model is used to generate measurements in the simulations (see Section 2.1). However, an extended Kalman filter (EKF) assumes that the mean measurement is the non-retarded bearing $\theta_{i,0}$. In the EKF, the target is assumed to follow a constant-velocity dynamic model (see Section 2.3).

5.2.1 Energy-based (EB) metric

In order to describe the lifetime of a sensor network, this thesis models the energy consumed for transmitting l bits over d meters in a multipath environment as

$$E = l \cdot \epsilon_{amp} \cdot d^4,$$

where ϵ_{amp} is a constant that represents the energy expense of engaging the power amplifiers to transmit sufficient signal power for delivery of one bit over a range of d meters. This transmission model was derived from the model used in [23], and it assumes that the energy usage is dominated by the radio transmission rather than the computation [57]. Let C be the capacity of the battery in $Joule$s, and let $p_i(k)$ be the remaining energy at time k. Since radio transmission dominates the energy usage, the i-th node in a hypothesized set $\mathcal{N}_a$ could survive for $\frac{p_i(k)}{\epsilon d^4_{i,\mathcal{N}_a}}$ number of time intervals without further handoff to a different node set, where

$$d_{i,\mathcal{N}_a} = \max_{j \in \mathcal{N}_a} \{d_{i,j}\}, \tag{42}$$

and $d_{i,j}$ is the distance between node i and node j.

The optimization of the lifetime of a sensor network, whether the Type 1 or 2 definition is considered, requires a non-myopic strategy. A simple minimization of the energy usage over one snapshot does not consider whether or not a particular node has been over utilized and is near death. To provide better load balancing, we consider the following energy-based metric for a hypothesized node set $\mathcal{N}_a$,

$$\mathbf{E}(\mathcal{N}_a) = \min_{i \in \mathcal{N}_a} \frac{p_i(k)}{\epsilon d^4_{i,\mathcal{N}_a}}. \tag{43}$$

Equation (43) defines the minimum number of time intervals that the hypothesized node set $\mathcal{N}_a$ could survive with all nodes still operating.

5.2.2 Transmission range control: knowledge pool (KP)

Without defining the transmission range of an active node, we could not measure how much energy is consumed or how long the tracking system could operate. In Chapter 3, we proposed a transmission range control using available network information and called it the knowledge pool. The activation decision is performed in a decentralized manner over the active set of nodes after the currently obtained information is shared among the active set of nodes. To this end, each active node determines the next active set by evaluating a metric, and decides whether it will remain active or whether it should wake up and hand off information to inactive nodes that are members of the next active set. Therefore, the broadcast range should be just long enough to reach all the active nodes for information sharing and to reach the next active set for information handoff.

We mentioned how active nodes hand over the useful information to the potentially active nodes for the next snapshot. The distributed system works collaboratively if every node makes its own active/inactive decision intelligently. Each node in the sensor network is equally powerful and capable of implementing a Kalman filter. The predicted target state and the predicted target error covariance would be broadcast by active nodes for information hand off or evaluation. If a node beyond the transmission range of the active nodes could not hear any useful information including the predicted target state and the predicted state error covariance, it has nothing to evaluate so it would decide to turn off. Furthermore, an active node must be aware of its remaining power level. In other words, if it is beyond the maximum reach of an active node, a node should not be considered as a candidate for the next snapshot. Let

$$\mathcal{N}_j^{mr} = \left\{ i | d_{i,j} \leq \sqrt[4]{\frac{p_j}{\epsilon}} \right\} \tag{44}$$

be the maximum reached node set from the j-th node. A common maximum reached node set by the currently active node set $\mathcal{N}_a$ is determined by the intersection

$$\mathcal{N}^{mr} = \bigcap_{j \in \mathcal{N}_a} \mathcal{N}_j^{mr}. \tag{45}$$

Only nodes that are elements of $\mathcal{N}^{mr}$ can be candidates for activation over the next snapshot. The node selection method chooses a subset $\mathcal{N}_a^{'} \subseteq \mathcal{N}^{mr}$ to be active over the next snapshot. If node i is currently active, it must transmit $d_{i,\mathcal{N}_a'}$ meters (see (42)) for information handoff. Over the next snapshot, if node i is active, it must transmit its measurements $d_{i,\mathcal{N}_a'}$ meters to share with other active nodes.

5.3 Global Network Knowledge

This section first discusses the global node selection method using either the GB or EB metric. Then, the joint metric method and its search strategies are discussed.

Global node selection (GNS) assumes that every node knows the physical location of every other node in the sensor network in order to compute the GB metric. Furthermore, each node keeps a table of the battery level of every node in the network so that it can calculate the EB metric. The active nodes broadcast this table so that all inactive nodes within earshot can update their battery information. Because the battery level is monotonically decreasing with usage, a node will update its battery usage table with the lowest reported level for each node. Using the global knowledge of node locations and battery levels, each node can now determine whether or not to be active by optimizing one of the metrics (GB or EB) or performing a joint optimization as discussed in the following subsections.

5.3.1 Single Metric

Let N_d be the desired number of active nodes per snapshot and $\mathcal{N}_a$ be the currently active node set. Our goal here is to determine the next set of active nodes $\mathcal{N}_a'$. The solution found by evaluating the GB or EB metric is listed in Fig. 33.

1. Determine the maximum reached node set $\mathcal{N}^{mr}$ by (45).
2. Determine the next set of active nodes:

$$\text{EB: } \mathcal{N}_a' = \arg \max_{\mathcal{N} \subseteq \mathcal{N}^{mr}} \mathbf{E}(\mathcal{N}),$$

$$\text{GB: } \mathcal{N}_a' = \arg \min_{\mathcal{N} \subseteq \mathcal{N}^{mr}} \rho(\mathcal{N}),$$

   where $|\mathcal{N}| = N_d$.
3. For $i \in \mathcal{N}_a$, $\epsilon d_{i,\mathcal{N}_a'}^4$ is consumed for the information handoff.
4. For $i \in \mathcal{N}_a'$, $\epsilon d_{i,\mathcal{N}_a'}^4$ is consumed for the information exchange at the next snapshot.

Figure 33: Global EB/GB node selection using KP where a node's maximum reach or battery level is considered in the search space.

Figure 34 shows information handoff using a KP when the GB metric is used to choose nodes. In Fig. 34(a), each node in $\mathcal{N}_a(k)$ shares local information obtained at time k. After the currently active nodes share and process the current information, they know which set of nodes will be active for the next snapshot by evaluating a metric. Then the nodes in $\mathcal{N}_a(k)$ could ping the inactive nodes that are members of the next active set to wake them up by adjusting the broadcast range. The information handoff is dependent on having the global location knowledge and knowing the strength of the currently active nodes (i.e., the search space is denoted by $\mathcal{N}^{mr}$). The problem with the knowledge pool is that the currently active nodes must ping an inactive node to wake up. As a result, the i-th node in $\mathcal{N}_a(k)$ consumes not only $\epsilon d_{i,\mathcal{N}_a(k)}^4$ for the information sharing, but also $\epsilon d_{i,\mathcal{N}_a(k+1)}^4$ for the information handoff to the next set of active nodes, $\mathcal{N}_a(k+1)$.

5.3.2 Joint Metric GB-EB

Two metrics were investigated when selecting nodes: one for minimizing the estimation error unaware of the current battery level, the other for maximizing the number of time intervals used as the lifetime metric unaware of the estimation performance.

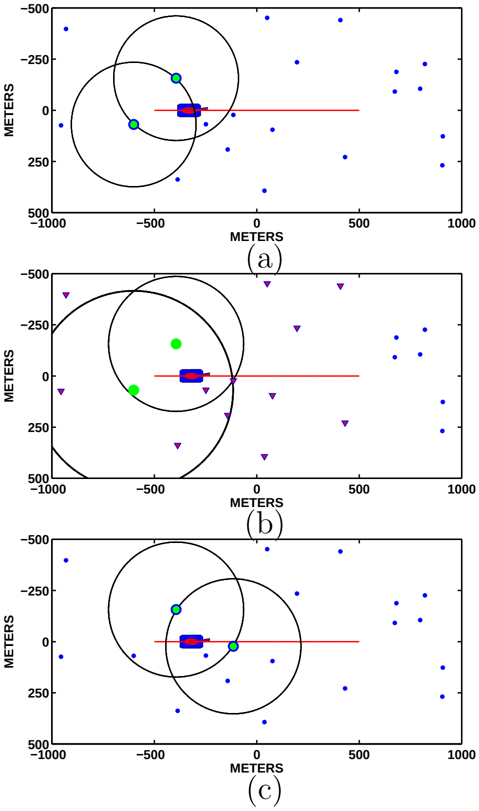


Figure 34: Illustration of the information handoff using KP where the chosen nodes are the origins of the broadcast ranges denoted by the black circles: (a) $\mathcal{N}_a(k)$ (b) $\mathcal{N}_a(k)$ adjusts the broadcast range to transmit the information to the next active nodes where $\mathcal{N}^{mr}$ is denoted by symbol $'\nabla'$ and (c) $\mathcal{N}_a(k+1)$ becomes active.

This gives rise to a more challenging problem: *Is there a composite metric to balance these two goals?*

The EB and GB metrics represent competing goals. Better localization accuracy requires a larger array of active nodes at the cost of more energy consumption. To balance the localization accuracy in lieu of the energy costs, we let the user define an acceptable level of location accuracy, and the algorithm optimizes the EB metric over a subset of nodes that can meet this requirement. Specifically, the conditioned joint metric denoted by GB-EB is written as:

$$\mathcal{N}^* = \arg \max_{\mathcal{N} \in \mathcal{C}_{\rho_0}} \mathbf{E}(\mathcal{N}), \tag{46}$$

where

$$\mathcal{C}_{\rho_0} = \left\{ \mathcal{N} | \mathcal{N} \subseteq \mathcal{N}^{mr} ; \rho(\mathcal{N}) \leq \rho_0 \right\}, \tag{47}$$

and $|\mathcal{N}^{mr}| = N$. The subset $\mathcal{N}^{mr}$ is the common reachable node set of the active nodes (45). The joint metric chooses the node set maximizing the proposed EB metric while maintaining the estimation performance below the desired threshold. Three different methods to determine an active set that approximates (46) are discussed below.

5.3.2.1 Exhaustive Search

Exhaustive search finds the global maximum of the EB metric under the constraint of the error threshold by searching the entire candidate space denoted by (47). Exhaustively enumerating the node sets whose lengths vary from 2 to N to form $\mathcal{C}_{\rho_0}$ is prohibitive and impractical because the computational complexity for metric evaluations is $\mathbf{O}(2^N)$.

5.3.2.2 Sequential Search

We start by choosing two active nodes per snapshot. If a certain number of nodes could not meet the error threshold, the number of active nodes is increased. Suppose $\mathcal{N}^{mr}$ is the maximum reachable node set from all the currently active nodes (45).

```
1. $M_d = 2$, $N = |\mathcal{N}^{mr}|$;
2. while $M_d \le N$,
        enumerate all $M_d$-node subsets to form $\mathcal{C}_{cand}(M_d)$
        as given by (48);
        if $\mathcal{C}_{cand}(M_d)$ is not empty, then
            $\mathcal{C}_s = \mathcal{C}_{cand}(M_d)$;
            $\mathcal{N}^* = \arg\max_{\mathcal{N} \in \mathcal{C}_s} \mathbf{E}(\mathcal{N})$;
            break;
        else
            $M_d = M_d + 1$;
        end
   end
```

Figure 35: Sequential search for the joint GB-EB optimization.

The candidate active sets consisting of M_d nodes are

$$\mathcal{C}_{cand}(M_d) = \{\mathcal{N} \mid |\mathcal{N}| = M_d; \mathcal{N} \subseteq \mathcal{N}^{mr}; \rho(\mathcal{N}) \le \rho_0 \}\,; \tag{48}$$

The steps to find the next active node set, $\mathcal{N}^*$, via Sequential search are enumerated in Fig. 35.

The sequential search starts with two nodes and then adds one node at a time making sure that the chosen node set satisfies the threshold. The drawback of the sequential search is that the search space denoted by (48) is not the complete search space, which was defined in (47). Whether there exists a node set with larger length that has a larger EB value is unknown via the sequential search. The sequential search reduces the search space by considering the minimum number of active nodes first and then adding one node at a time until the error threshold is met. In fact, the sequential search finds the minimum number of active nodes to meet the error threshold. However, enumerating all the possible subsets with a certain length and performing the metric evaluation is still expensive. The computational complexity for metric evaluations is $\mathbf{O}(N^d)$ where d is the minimum number of active nodes that

```
1. $M_d = 2$, $N = |\mathcal{N}^{mr}|$;
2. enumerate all $M_d$-node subsets to form $\mathcal{C} = \{\mathcal{N} | |\mathcal{N}| = M_d; \mathcal{N} \subseteq \mathcal{N}^{mr}\}$,
   and $\mathcal{C}_{cand} = \{\mathcal{N} | \mathcal{N} \in \mathcal{C}; \rho(\mathcal{N}) \leq \rho_0\}$;
3. $\mathcal{N}_m = \arg\min_{\mathcal{N} \in \mathcal{C}} \rho(\mathcal{N})$;
4. while $M_d \leq N$ & $\mathcal{C}_{cand}$ is empty,
       $\mathcal{C} = \{\mathcal{N} | \mathcal{N} = \mathcal{N}_m \bigcup \{j\}; j \in \mathcal{N}^{mr} \setminus \mathcal{N}_m\}$;
       $\mathcal{C}_{cand} = \{\mathcal{N} | \mathcal{N} \in \mathcal{C}; \rho(\mathcal{N}) \leq \rho_0\}$;
       $\mathcal{N}_m = \arg\min_{\mathcal{N} \in \mathcal{C}} \rho(\mathcal{N})$;
       $M_d = |\mathcal{N}_m|$;
   end
5. $\mathcal{C}_g = \mathcal{C}_{cand}$;
6. $\mathcal{N}^* = \arg\max_{\mathcal{N} \in \mathcal{C}_g} \mathbf{E}(\mathcal{N})$;
```

Figure 36: Greedy search for the joint GB-EB optimization.

meets the error threshold ($|\mathcal{N}^*| = d$).

5.3.2.3 Greedy Search

The Greedy search uses an "add one node at a time" strategy to build the candidate space until the space contains sets that meet the threshold constraint. The steps of the search method are given in Fig. 36. Instead of exhaustively enumerating all the node sets with a certain length, say M_d, the Greedy search adds one more node into the existing suboptimal M_d-node set (Step 4) and stops when $\mathcal{C}_{cand}$ is not empty. The computational complexity of the Greedy search consists of exhaustively evaluating 2-node sets, $\mathbf{O}(N^2)$, and adding one at a time, $\mathbf{O}((|\mathcal{N}^*| - 2)N)$. If the Greedy search continued past the point that $\mathcal{C}_{cand}$ is nonempty, it will not find a better solution because the set $\mathcal{N}_m$ will lead to a smaller EB metric according to the following theorem.

Theorem 1 *If $\mathcal{N}_1 \subseteq \mathcal{N}_2$ and $\rho(\mathcal{N}_1) \leq \rho_0$, then $\rho(\mathcal{N}_2) \leq \rho(\mathcal{N}_1) \leq \rho_0$ and $\mathbf{E}(\mathcal{N}_2) \leq$*

$\mathbf{E}(\mathcal{N}_1)$.

Proof: Suppose node $l \in \mathcal{N}_1$. Node l provides the shortest tracking lifetime in $\mathcal{N}_1$, that is, $\mathbf{E}(\mathcal{N}_1) = \frac{p_l}{\epsilon d^4_{l,\mathcal{N}_1}}$. We add one more node m into $\mathcal{N}_1$ to form $\mathcal{N}_2 = \{m, \mathcal{N}_1\}$. Then we have

$$\mathbf{E}(\mathcal{N}_2) = \min\{\frac{p_l}{\epsilon d^4_{l,\mathcal{N}_1}}, \frac{p_l}{\epsilon d^4_{l,m}}, \frac{p_m}{\epsilon d^4_{m,\mathcal{N}_1}}\}.$$

Therefore, $\mathbf{E}(\mathcal{N}_2) \leq \mathbf{E}(\mathcal{N}_1)$.

Suppose $\mathcal{N}_1 = \mathcal{N}$ and $\mathcal{N}_2 = \mathcal{N} \cup \{j\}$ where $j \ni \mathcal{N}$. The posterior squared RMS error $\rho^2(\mathcal{N}) =$ trace$(\mathbf{S}^{-1}_{\mathcal{N}})$ where $\mathbf{S}^{-1}_{\mathcal{N}} = [\mathbf{J}^{-1}_f(\mathcal{N})]_{1:2,1:2}$. Explicitly $\mathbf{S}_{\mathcal{N}} = A - BD^{-1}C + \mathbf{J}_m(\mathcal{N})$ where $\mathbf{J}_p = \begin{bmatrix} A & B \\ C & D \end{bmatrix}$.

$\mathbf{S}_{\mathcal{N}_2} = \mathbf{S}_{\mathcal{N}_1} + \frac{n_j n_j^T}{\sigma_j^2 r_j^2}$ where $n_j = [-\sin(\phi_j), \cos(\phi_j)]^T$. Both $\mathbf{S}_{\mathcal{N}_1}$ and $\mathbf{S}_{\mathcal{N}_2}$ are positive definite and invertible, and $\mathbf{S}_{\mathcal{N}_2} \geq \mathbf{S}_{\mathcal{N}_1}$. Therefore both $\mathbf{S}^{-1}_{\mathcal{N}_2}$ and $\mathbf{S}^{-1}_{\mathcal{N}_1}$ are positive definite, and $\mathbf{S}^{-1}_{\mathcal{N}_2} \leq \mathbf{S}^{-1}_{\mathcal{N}_1}$. Then

$$\text{trace}(\mathbf{S}^{-1}_{\mathcal{N}_2}) \leq \text{trace}(\mathbf{S}^{-1}_{\mathcal{N}_1}).$$

Then, $\rho(\mathcal{N}_2) \leq \rho(\mathcal{N}_1)$. **Q.E.D.**

5.3.3 Joint Metric EB-GB

Another interesting joint metric, denoted by EB-GB, depends on the EB metric, $\mathbf{E}(\mathcal{N})$ and a user-defined threshold, τ_0, to determine the size of search space in the following manner

$$\mathcal{N}^* = \arg \min_{\mathcal{N} \in \mathcal{C}_{cand}} \rho(\mathcal{N}), \tag{49}$$

where

$$\mathcal{C}_{cand} = \{\mathcal{N} | \mathcal{N} \subseteq \mathcal{N}^{mr}; \mathbf{E}(\mathcal{N}) \geq \tau_0\}, \tag{50}$$

and where $|\mathcal{N}^{mr}| = N$ by (45). Recall that the EB metric has units of seconds. This optimization requires that we assume the global location information is available at each node. As the counterpart of the GB-EB, the EB-GB tries to find the node set which provides the best geometric view while its EB value is longer than the user-defined threshold, τ_0.

5.3.3.1 Exhaustive Search

The solution via Exhaustive Search minimizing the GB metric under the constraint of a time threshold is obtained by searching for the candidate space denoted by (50). The candidate search is complete considering the node set whose length is from two to N. We did not provide the simulation results via Exhaustive Search because it is impractical. Instead, we provide two practical ways, Sequential Search and Greedy Search, to reduce the search space.

5.3.3.2 Sequential Search

We start with the maximum number of nodes N where N is the length of the maximum reachable nodes $\mathcal{N}^{mr}$. If a certain number of nodes could not meet the time threshold, the number of active nodes must be adjusted or decreased because a node set with fewer members has a better chance of meeting the time threshold. The steps to find $\mathcal{N}^*$ via Sequential Search are shown in Fig. 37.

In this Sequential Search, we start with N nodes and then we remove one node at a time while making sure that the chosen node set satisfies the threshold. The actual number of chosen nodes cannot be less than two. The Sequential Search finds the maximum number of nodes to meet the time threshold. The drawback of the Sequential Search is that the solution via the Sequential Search may not provide the minimum GB value because the search space defined by (51) in the Sequential Search is not as complete as in the Exhaustive Search. Although the search space in the Sequential Search is reduced to some extent, enumerating all the possible subsets with a certain length for metric evaluation is still painful. The computational complexity for metric evaluations is $\mathbf{O}(N^d)$ where d is the maximum number of active nodes that

```
1. $M_d = N = |\mathcal{N}^{mr}|$.
2. while $M_d \geq 2$,
        enumerate all $M_d$-node subsets to form
            $\mathcal{C}_{cand} = \{\mathcal{N} || \mathcal{N}| = M_d; \mathcal{N} \subseteq \mathcal{N}^{mr}; \mathbf{E}(\mathcal{N}) \geq \tau_0\}.$    (51)
        if $\mathcal{C}_{cand}$ is not empty,
            $\mathcal{C}_s = \mathcal{C}_{cand}$;
            $\mathcal{N}^* = \arg\min_{\mathcal{N} \in \mathcal{C}_s} \rho(\mathcal{N})$;
        else
            $M_d = M_d - 1$;
        end
   end
```

Figure 37: Sequential search for the joint EB-GB optimization.

meet the time threshold, i.e., $d = |\mathcal{N}^*|$.

5.3.3.3 Greedy Search

As opposed to the Greedy Search for the GB-EB, we start with N nodes. Inspired by Theorem 1, if three nodes could meet the time threshold, it is not necessary to remove one more node because three nodes can provide smaller RMS position error than two nodes. The steps to find $\mathcal{N}^*$ are shown in Fig 38. The computational complexity consists of enumerating all two-node sets to check if a two-node set with the maximum EB value could meet the time threshold, $\mathbf{O}(N^2)$, and removing one node at a time from the suboptimal solution $\mathcal{N}_m$, $\mathbf{O}((|\mathcal{N}^*| - 2)N)$.

5.4 Partial Network Knowledge

For large sensor networks, it becomes impractical for each node to store a table containing information about all the other nodes in the network. In practice, a node will only keep a table about its neighbors, i.e., the set of nodes within a distance of r_{nei}. Furthermore, it is reasonable for the nodes within earshot of the active set to be

```
1. $M_d = N = |\mathcal{N}^{mr}|$.
2. enumerate all 2-node subsets to form $\mathcal{C} = \{\mathcal{N} || \mathcal{N}| = 2; \mathcal{N} \subseteq \mathcal{N}^{mr}\}$,
   and $\mathcal{C}^* = \{\mathcal{N} | \mathcal{N} \in \mathcal{C}; \mathbf{E}(\mathcal{N}) \geq \tau_0\}$.
3. if $\mathcal{C}^*$ is not empty,
        while $M_d \geq 2 \& \mathbf{E}(\mathcal{N}_m) \leq \tau_0$,
            $\mathcal{C} = \{\mathcal{N} | \mathcal{N} = \mathcal{N}_m \setminus \{j\}; j \in \mathcal{N}_m\}$;
            $\mathcal{C}_{cand} = \{\mathcal{N} | \mathcal{N} \in \mathcal{C}; \mathbf{E}(\mathcal{N}) \geq \tau_0\}$;
            $\mathcal{N}_m = \arg\max_{\mathcal{N} \in \mathcal{C}} \mathbf{E}(\mathcal{N})$;
            $M_d = |\mathcal{N}_m|$;
        end
   end
4. $\mathcal{C}_g = \mathcal{C}_{cand}$;
5. $\mathcal{N}^* = \arg\max_{\mathcal{N} \in \mathcal{C}_g} \mathbf{E}(\mathcal{N})$;
```

Figure 38: Greedy search for the joint EB-GB optimization.

able to store information about the active nodes. Let $\mathcal{N}_{nei}(i) = \{j | d_{i,j} \leq r_{nei}, i \neq j\}$ be the neighbor node set of the i-th node. Like the global approach, the active nodes will provide battery level updates to the other nodes, but only the neighbors.

The steps of the partial node selection (PNS) algorithm are provided in Fig. 39. PNS exploits the fact that each active node will know the location of the other active nodes. Therefore, each active node can determine which of its neighboring nodes is a neighbor of all the active nodes and reachable by all active nodes (52). Therefore, each node is able to define a candidate space that is consistent over all active nodes. The efficient optimization of the GB-EB metric in Step 1 can be performed by Sequential or Greedy search. Note that as r_{nei} grows to infinity, the PNS becomes equivalent to GNS.

1. Determine the next set of active nodes $\mathcal{N}'_a$ from the point of view of the currently active node set $\mathcal{N}_a$:

   $$\text{GB-EB: } \mathcal{N}'_a = \arg \max_{\mathcal{N} \in \mathcal{C}_{cand}(i)} \mathbf{E}(\mathcal{N}),$$

   where

   $$\mathcal{C}_{cand} = \{\mathcal{N} | |\mathcal{N}| = M_d ; \mathcal{N} \subseteq \mathcal{N}_{known} ; \rho(\mathcal{N}) \leq \rho_0 \},$$

   and

   $$\mathcal{N}_{known} = \{\{\cap_{i \in \mathcal{N}_a} \mathcal{N}_{nei}(i)\} \cup \mathcal{N}_a\} \cap \mathcal{N}^{mr}. \qquad (52)$$

2. For $i \in \mathcal{N}_a$, $\epsilon d^4_{i,\mathcal{N}'_a}$ is consumed for the information handoff.

3. For $i \in \mathcal{N}'_a$, $\epsilon d^4_{i,\mathcal{N}'_a}$ is consumed for the information sharing at the next snapshot.

Figure 39: One cycle of the GB-EB-PNS method.

5.5 Simulations

In the simulations, a target traverses along a straight line at a constant speed of 10m/s as shown in Fig. 40. The goal here is to measure how accurate the tracking estimates are over the lifetime of the network. The time interval for updating the tracker is set to 1 second. We use the knowledge pool to determine the broadcast range. The search space consists of the nodes that are reachable by the active nodes according to their battery level, that is, $\mathcal{N}^{mr}$ in (45). We also use the posterior GB metric denoted by (19) to choose a best set of nodes.

Assuming the global physical location information of nodes is available, Fig. 41 shows the lifetime of EB and GB by averaging ten node configurations. The lifetime for such a single metric optimization is defined as the earliest time point when one or more active nodes has too little energy to reach the other active nodes. The EB method has a longer lifetime than the GB method especially when the number of active nodes is small. We note that the lifetime using the GB method is estimation-dependent while that using the EB method does not depend on the target location or the accuracy of the estimation.

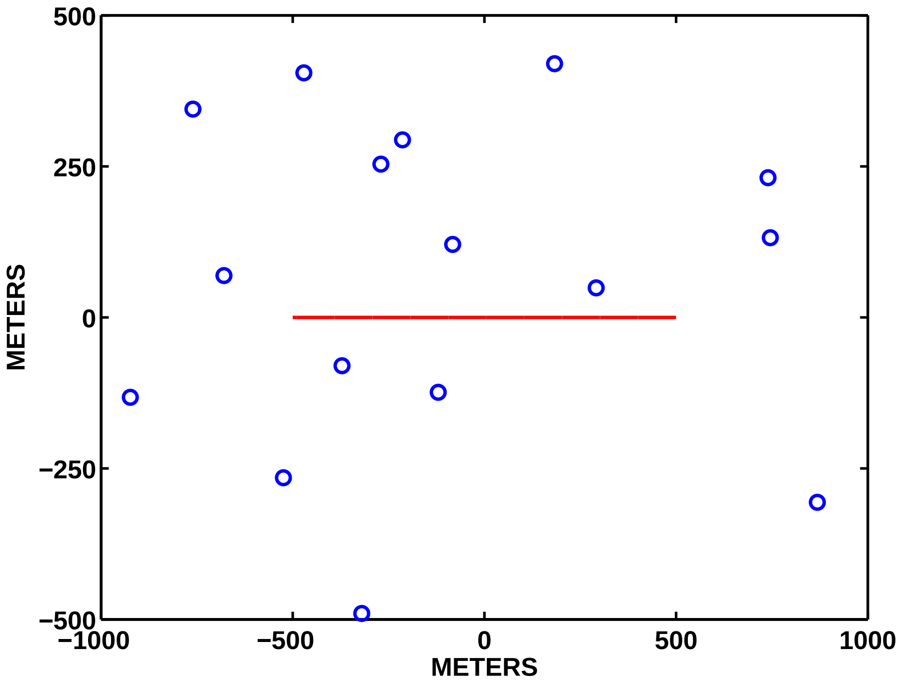
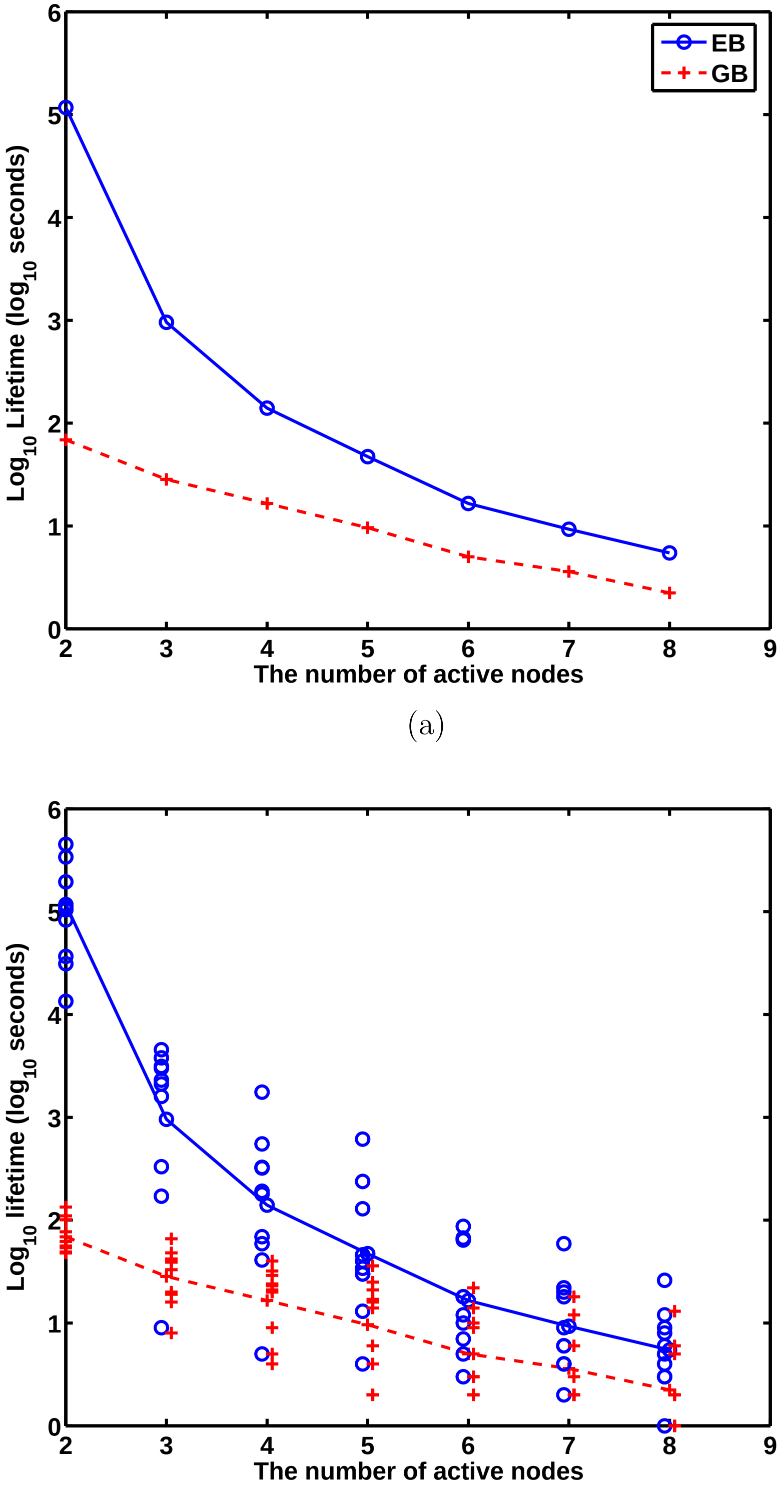


Figure 40: One node configuration with twenty nodes where ∘ denotes the node and the solid line denotes the true target track along which a target goes back and forth for an infinite time.

Defining the tracking lifetime as the earliest time point when the transmission fails is called Type 1. Because it is desirable that the average RMS error over the lifetime be less than the error threshold, we have defined an alternative lifetime, called Type 2, as the earliest time point when even using all the possibly reachable nodes we could not meet the error threshold (i.e., $\rho(\mathcal{N}^{mr}) \geq \rho_0$). If the tracking is governed by Type 1 but Type 2 happens earlier, we can make the tracking continue after the Type 2 failure by making all the possibly reachable nodes, $\mathcal{N}^{mr}$, active for the next snapshot. On the other hand, if the tracking is governed by Type 2 but Type 1 happens earlier, the tracking continues by simply removing the node that fails to communicate.

Given a node configuration and $\rho_0 = 60$m, Fig. 42 and Table 5 demonstrate the performance of GNS over one Monte Carlo run. Specifically, they show that Type 1 lasts longer than Type 2 lifetime for the one Monte Carlo run. Since the joint metric chooses the one with the largest EB value among those candidate sets whose GB value



(a)

(b)

Figure 41: Performance of the global EB and the GB (a) average lifetime (b) scattered points of different trials where the symbol ∘ denotes the lifetime of EB and the symbol + denotes the lifetime of GB.

is below the error threshold, it is likely that the solution would be one whose GB value is a little bit smaller than ρ_0 and whose EB value is large as shown in Table 5. When a handoff occurs, the node selection does not consider the energy required for the handoff. Therefore, while the EB-metric may be high before the handoff, it becomes low after the handoff. For example, the jump in the EB value from snapshot 55 to snapshot 56 or from snapshot 56 to snapshot 57 is due to the fact that nodes remaining active could not pass the error threshold, and they have no choice but to choose the nodes which may provide a worse EB value but meet the GB constraint (see Fig. 43 for the chosen node set and information handoff). After enough handoffs, the chosen active set might not have enough energy to handoff its information to any other nodes. The active nodes will simply collect and share their measurements even though they will no longer satisfy the GB constraint. For example, from snapshot 57 to 61, the active set meets the GB constraint, but does not meet the constraint by snapshot 62 because the active nodes do not have enough energy to handoff to better sensors. Eventually, these nodes exhaust their battery supplies while providing poor localization performance (see Fig. 42 after 60 seconds). As a result, Type 1 lifetime is longer than Type 2. Table 5 shows that after Snapshot 61, the geolocalization constraint can never be met when $\rho_0 = 60$m. By loosening the GB constraint, it would be possible to extend the network lifetime.

Figure 44 shows the tradeoff between the tracking lifetime and the RMS position error averaged over the period of the lifetime using the joint metric via different search algorithms. Figure. 44 shows the standard deviation of the RMS position error versus the lifetime. We run ten node configurations, each with ten or twenty randomly placed nodes. Given a node configuration and a value for ρ_0, we run ten Monte Carlo simulations with five-degree bearing measurement error. The error threshold ρ_0 varies from 10m to 70m with a 10m increment. We omit the simulation results via exhaustive search when the node configuration consists of twenty nodes because it

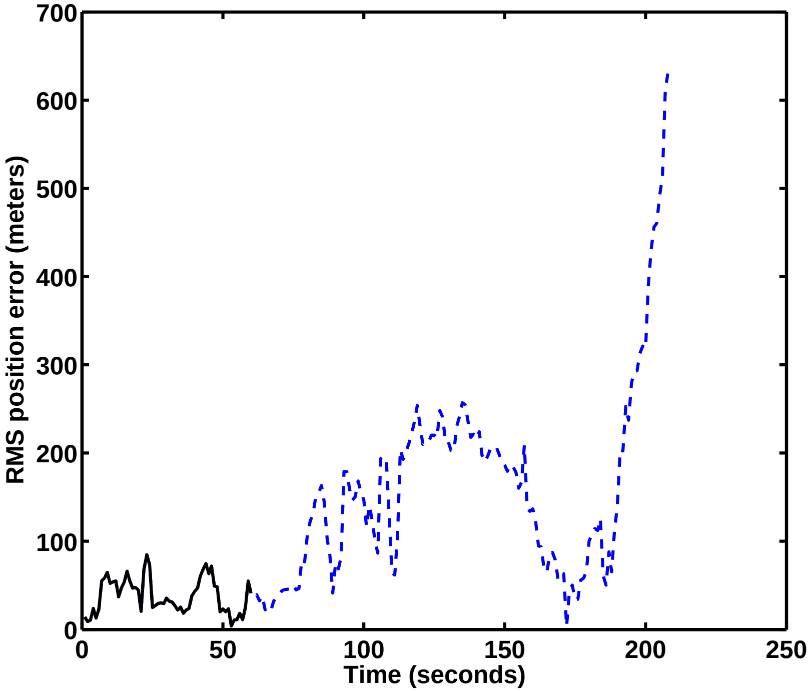


Figure 42: RMS error versus time using the joint metric where the solid line denotes the RMS error over Type 2 and the dotted line denotes the RMS error over Type 1 when ρ_0=60m.

Table 5: Illustration of GB and EB values for some snapshots where $\mathcal{N}^{mr}$ is the maximum reachable node set of previously active nodes $\mathcal{N}_a(k-1)$ and $\mathcal{N}_a(k) \subseteq \mathcal{N}^{mr}$.

Time k	$\rho(\mathcal{N}_a(k-1))$	$\rho(\mathcal{N}^{mr})$	$\rho(\mathcal{N}_a(k))$	$\mathbf{E}(\mathcal{N}_a(k))$
52	56	27	56	297635
53	56	18	56	297634
54	57	19	57	297632
55	59	19	59	297630
56	61	19	59	53783
57	62	19	57	304
58	56	56	56	303
59	56	56	56	301
60	58	58	58	299
61	60	60	60	297
62	62	62	62	295

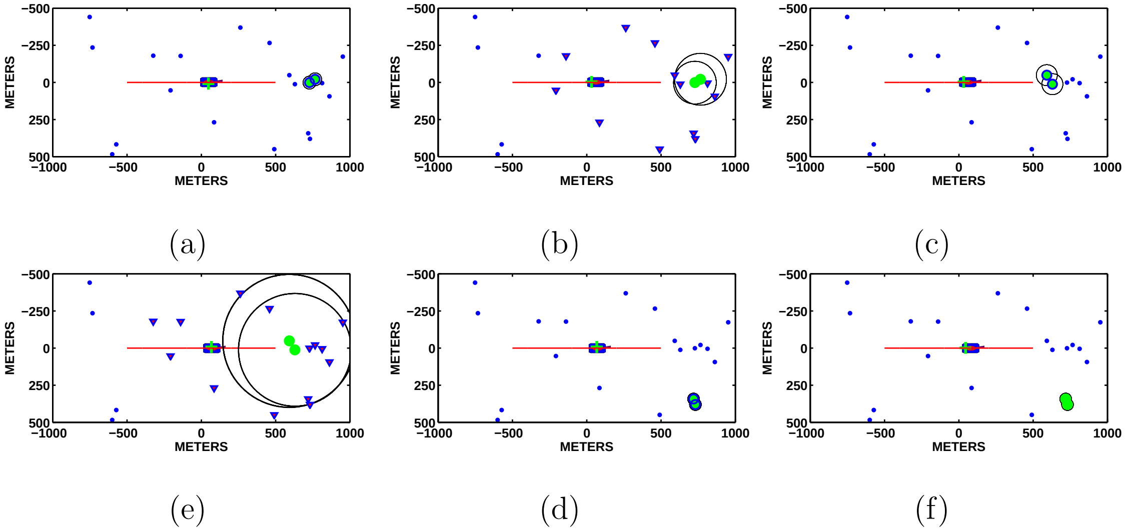


Figure 43: Illustration of global GB-EB where the chosen nodes are the origins of the broadcast ranges denoted by the black circles and $\mathcal{N}^{mr}$ is denoted by the symbol $'\nabla'$: (a) information sharing at snapshot 55 (b) information handoff at snapshot 55 (c) information sharing at snapshot 56 (d) information handoff at snapshot 56 (e) information sharing at snapshot 57 (f) information handoff at snapshot 57.

takes too long to run. There are a few observations for these three search algorithms in Figs. 44 45.

1. The performance differences among these three search algorithms are small although their search spaces are quite different.

2. Type 1 lasts longer than Type 2 lifetime, which means that breaking the error threshold occurs earlier than the first transmission failure. The performance after Type 2 until Type 1 is determined by all the nodes within the maximum power reach, that is $\mathcal{N}^{mr}$ in (45). This is the reason why as the threshold ρ_0 increases, the tracking lifetime or the RMS error does not increase monotonically when the tracking is terminated by a Type 1 failure.

3. In most cases, the exhaustive search, denoted by symbol $'\diamond'$ for Type 1 lifetime and $'\times'$ for Type 2 lifetime, respectively, leads to the longest lifetime. The reason is that its search space is complete.

Table 6: Comparisons among different search algorithms using global GB-EB when $\rho_0 = 60m$.

	Monte Carlo run 1			
	RMS er.	Type 1	Node usage	Processing time
	(meters)	lifetime(s)		per snapshot(ms)
Sequential	72.29	215	2.00	17.46
Greedy	72.29	215	2.00	3.63
Exhaustive	72.29	216	2.00	10746

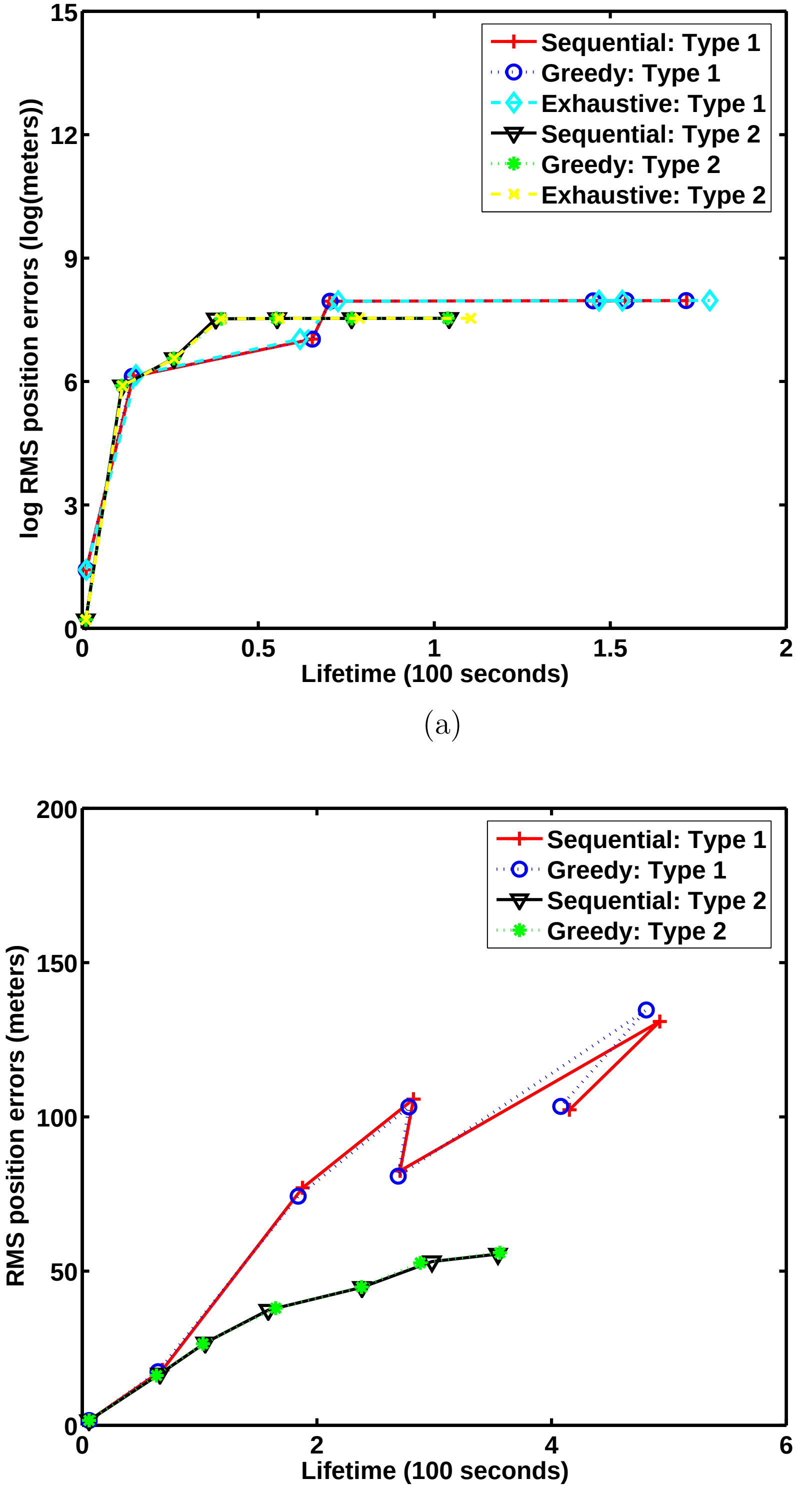
	Monte Carlo run 2			
	RMS er.	Type 1	Node usage	Processing time
	(meters)	lifetime(s)		per snapshot(ms)
Sequential	81.49	178	2.00	15.92
Greedy	81.49	178	2.00	3.54
Exhaustive	117.4	230	2.00	4981.4

4. When the tracking is terminated by Type 2 failures, increasing ρ_0 leads to more sets with large EB values in the candidate space. As a result, the tracking lifetime increases at a cost of higher RMS errors.

5. Fig. 45 shows the error bars and ellipse around the data points for Type 2 lifetime. The error bars are not shown for Type 1 lifetime because it is beyond the text.

Table 6 compares the speed of running different search algorithms given a node configuration, a threshold and a Monte Carlo run. The processing time per snapshot includes running the filter update and the search algorithm. The Greedy search is at least four times faster than sequential search, and exhaustive search is the most time-consuming taking four times longer than one snapshot to process one snapshot's data. The Monte Carlo runs were performed on a Dell 700m laptop with a Intel Pentium 1.6GHz CPU and 512 MB of RAM. We also observe the exhaustive search could provide longest lifetime among those three searches.

Figure 46 shows the lifetime and the standard deviation of the lifetime using the global EB-GB metric to choose nodes. Since the constraint is on EB metric, Type 2 lifetime is defined as the earliest time when any two-node set could not meet the



(a)

(b)

Figure 44: Performance of GB-EB-GNS using different search algorithms when a node configuration consists of (a) ten nodes, (b) twenty nodes.

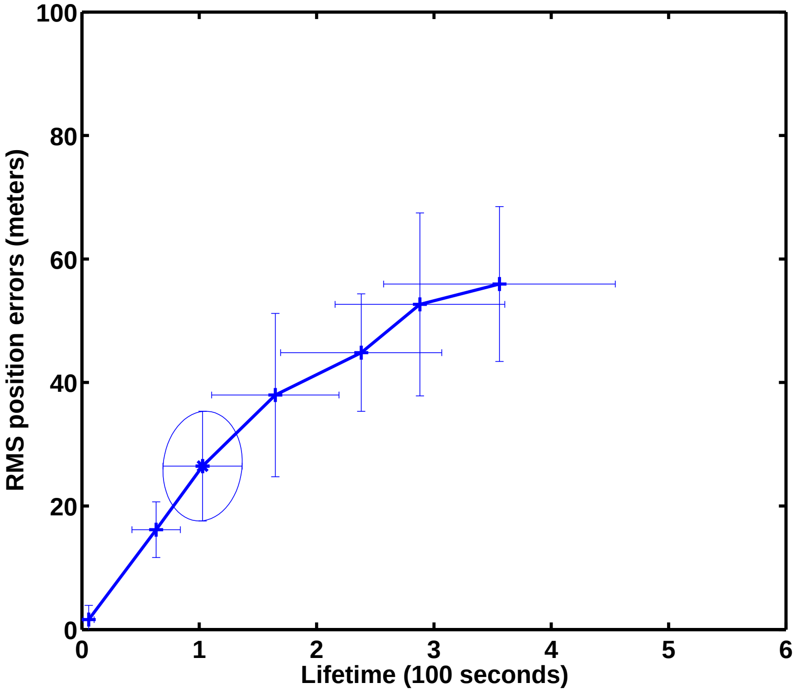


Figure 45: Error bars and error ellipse from 100 runs of GB-EB-GNS using Greedy Search for Type 2 lifetime.

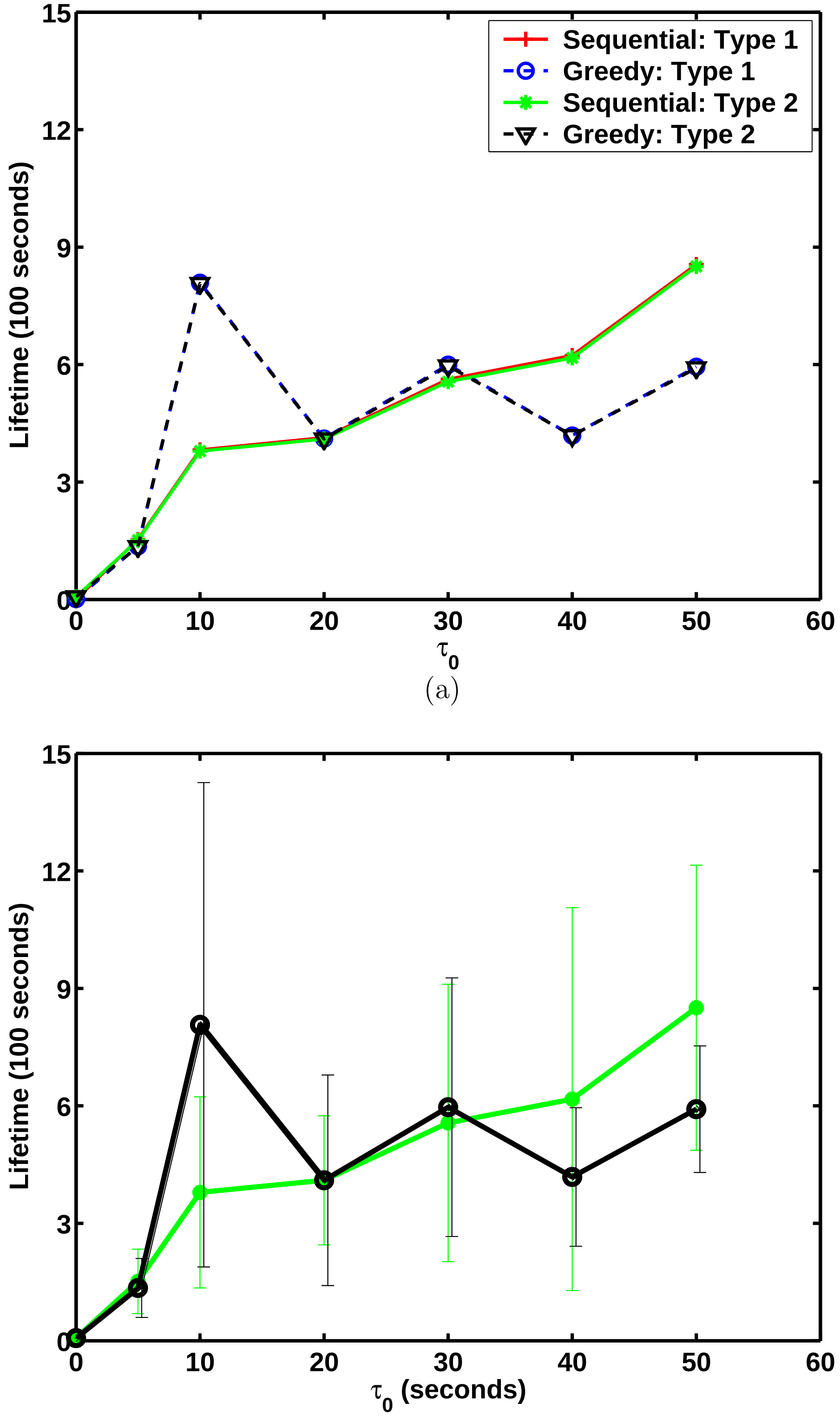
Table 7: Comparisons among different search algorithms using global EB-GB when $\tau_0 = 20$s.

	RMS er. (meters)	Type 1 lifetime(s)	Node usage	Processing time per snapshot(ms)
Sequential	366.1	2822	2.29	115
Greedy	371.4	4679	2.12	9
Exhaustive	14.1	151	2.62	3715

time threshold. We consider ten node configurations, each with ten randomly placed nodes. Given a node configuration and a time threshold, we run ten Monte Carlo experiments with five-degree measurement errors. For some values of τ_0, the Greedy Search gives similar lifetime to the Sequential Sequential search. For other values of τ_0, the Greedy Search falls apart from the Sequential Search because the Greedy Search for EB-GB-GNS utilizes "remove one node at a time" strategy and may be more dependent on the node layout. Another observation is that Type 2 and Type 1 lifetimes are similar. For the Sequential Search, with τ_0 increasing, only node sets with small size could easily pass the strict time threshold, so the result is that the lifetime increases.

Table 7 compares the speed of running different search algorithms given a node configuration with twenty randomly placed nodes and a threshold. The processing time per snapshot includes running the EKF filter update and the search algorithm. The Greedy Search is the fastest and the Exhaustive Search is almost four times longer than one time interval while the Exhaustive Search has the largest node usage and lowest RMS error.

Figure 47 shows the performance of GB-EB-PNS. The RMS position error is averaged over Type 2 lifetime when $\rho(\mathcal{N}_{known}) \geq \rho_0$ and $\mathcal{N}_{known}$ is given by (52). We consider ten configurations, each with twenty nodes and we vary the estimation threshold ρ_0=[10 30 50 70]m. Given a value of ρ_0 and a configuration, we run one hundred Monte Carlo simulations using the Greedy search. From Fig. 47, we make



(a)

(b)

Figure 46: Performance of the global EB-GB (a) lifetime (b) error bars from 100 runs for Type 2 where the symbol ∘ denotes Greedy Search and the symbol ⋆ denotes Sequential Search.

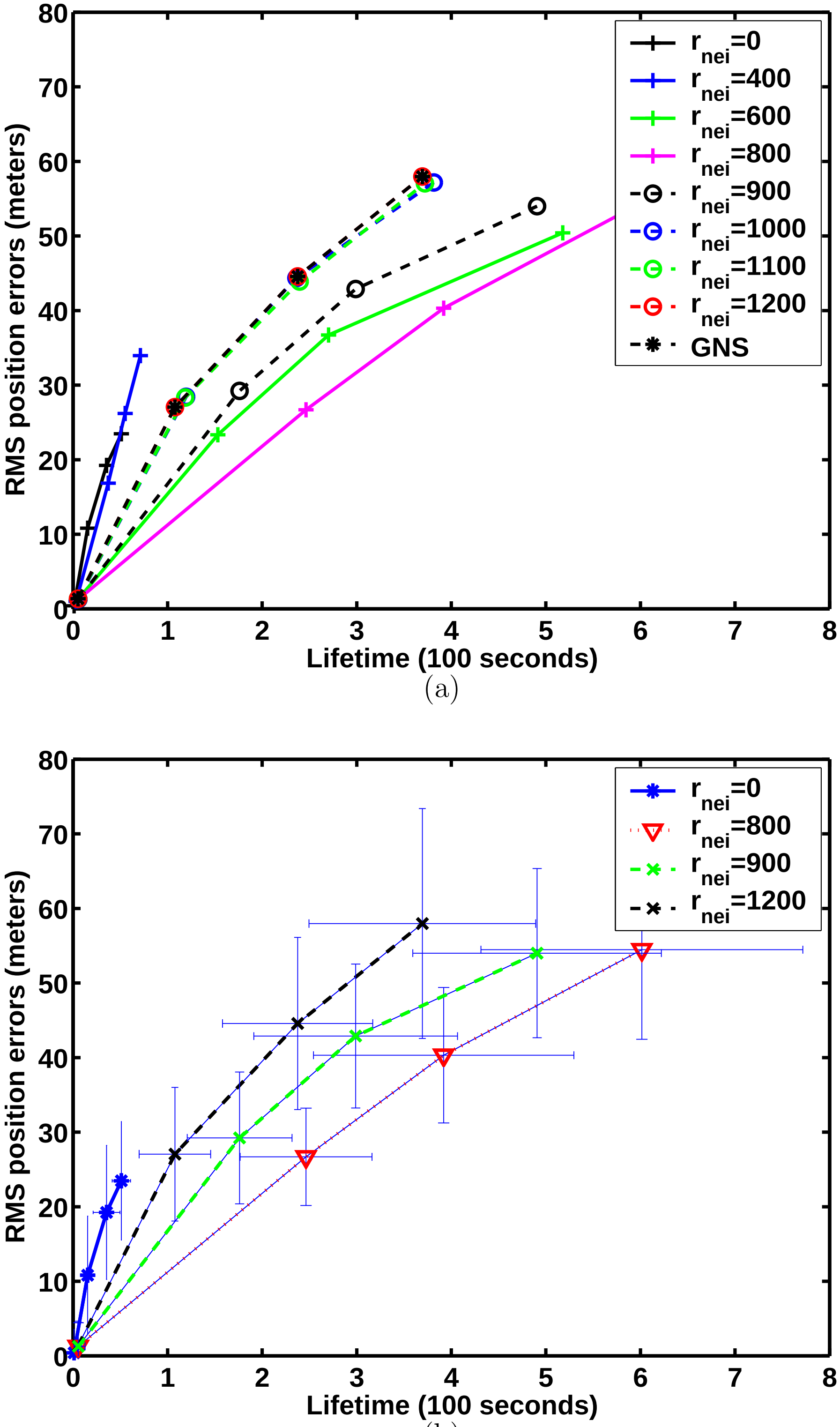
the following observations:

1. PNS with $r_{nei} = 1200$m approaches GNS.

2. For a given r_{nei}, as ρ_0 increases, the RMS error or the lifetime increases.

3. As r_{nei} increases from 0 to 800m, it seems that the lifetime increases while the RMS error gets worse. More neighbor information helps prolong the lifetime. However, for a given ρ_0 ($10 \leq \rho_0 \leq 70$), the lifetime decreases when going from r_{nei}=800m to r_{nei}=1200m. This fact reveals a conflicted role for r_{nei} on the constraint metric. On one hand, more knowledge of the neighbors means that more node sets could pass the error threshold, leading us to choose a node set with a longer EB value. On the other hand, more knowledge of the neighbors could also mean that a far away node set with longer EB value could be chosen for the next snapshot, which costs the currently active node set higher energy for longer information handoff. What is more, that a node set with longer EB value is chosen means more information handoff in the future because a node set with longer EB value may not have smaller GB value. In a word, more knowledge of the neighbors could shorten the lifetime due to longer or more information handoff in the future.In the experiments, we found that nodes, that are close to each other and far away from the active nodes, are chosen at earlier snapshots when r_{nei}=1200m, which might lead to breaking the error threshold earlier than expected.

4. For a given r_{nei}, as ρ_0 increases, the standard deviation of lifetime or RMS error is not monotonously increasing. This may be because the number of Monte Carlo runs, i.e., ten node configurations each with 100 Monte Carlo runs, is not enough.



(a)

(b)

Figure 47: Performance of GB-EB-PNS (a) RMS error versus lifetime (b) error bars from 1000 runs.

5.6 Lifetime for multitarget case

Chapter 4 was dedicated to showing how estimation performance changes when we combine the GB metric with multitarget trackers. Because the EB tries to maximize the lifetime, measuring the lifetime would make more sense when we apply EB or constrained metric to a multitarget case. The following simulation addresses the question: "how long is the tracking lifetime when the system tracks two targets instead of tracking a single target of interest in the presence of two targets?"

Measuring the tracking lifetime in the multitarget scenario is related to which multitarget tracker (PDA or JPDA) is used. The PDA or JPDA is a target-oriented tracker, which means that given a particular target, we calculate how much contribution each measurement could make. Each broadcast packet includes the predicted information for one target. That a node receives two packets belonging to different targets means a node has the predicted information from two targets. A node capable of implementing the PDA algorithm for a particular target does not require the predicted information for other targets while a node capable of implementing the JPDA algorithm does. Suppose clusters are formed where a cluster consists of an established target and its associated node set. Bearing information is shared over the nodes within the same cluster. The PDA tracker naturally simplifies the communication control by treating other targets as noise and saving the energy used for information sharing among clusters, although it has a weakness when tracking close targets. However, the JPDA may spend extra energy in information sharing among clusters. For preliminary research on measuring tracking lifetime in a multitarget case, we use PDA over other multitarget trackers.

In the simulations, we consider two targets moving along straight lines for infinite time in a network with twenty randomly placed nodes as shown in Fig. 48. The minimum separation distance between these two targets is 5 meters. We assume that global information including node location information and battery level is available.

Table 8: Tracking lifetime using different metrics and PDA.

Metric and M_t	Lifetime(s)
GB: $M_t = 1$, $N_d = 3$	91
GB: $M_t = 2$, $N_d = 3$	52
EB: $M_t = 1$, $N_d = 3$	2110
EB: $M_t = 2$, $N_d = 3$	1236
GB-EB $\rho_0 = 50\text{m}$: $M_t = 1$	242
GB-EB $\rho_0 = 50\text{m}$: $M_t = 2$	78

Energy is spent in information sharing among nodes in a cluster and information handoff from the current cluster to the next cluster belonging to the same target. For a single metric optimization, the tracking lifetime is defined as the first transmission failure while for the constrained metric it is defined as the earliest time point when using all reachable nodes could not meet the error threshold. The simulated bearing measurements obtained at each node may include measurements that truly originated from targets, merged measurements when two targets are close, or false measurements due to noise. The detection probability is set equal to 0.98. The bearing measurement model and the PDA tracker were previously discussed in Chapter 4. When there are always two targets present, we could underestimate the number of targets by tracking a single target of interest.

Table 8 shows the tracking lifetime using different metrics and the PDA tracker for fifty Monte Carlo runs where M_t denotes the number of assumed targets and N_d is the predefined number of active nodes. The chosen node set is obtained via the Greedy Search. It is shown that the lifetime for tracking two targets is 2.0-3.0 times shorter than that for tracking one target of interest. For example, the lifetime lasts 242 seconds for tracking one target while it lasts 78 seconds for tracking two targets when we use GB-EB metric with ρ_0=50m to choose nodes.

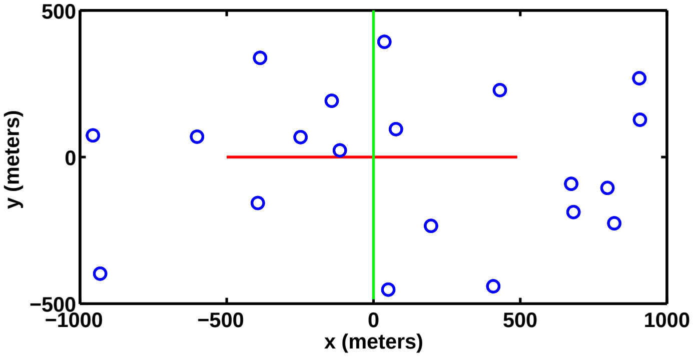


Figure 48: Node configuration and two target tracks.

5.7 Conclusions

We proposed an EB metric to maximize the number of time intervals that a hypothesized active node set could survive without further information handoff to another active set. Simulation results show that the EB metric provides longer lifetime than the existing GB metric. In order to reduce the energy consumption and measure the tracking lifetime, we also proposed a variable-range transmission range control where the transmission range is determined by the knowledge of the network and the strength of the current battery. Each active node transmits just far enough to reach all the active nodes. In other words, the energy consumption is proportional to the physical distance between the active set of nodes and is divided into two stages, one for active nodes to share the currently obtained information, the other for the currently active nodes to hand off the useful information to the next active set. In order to optimize a constrained metric using one metric as the optimization metric under the constraint of the other, we proposed practical search algorithms whose search space is dependent on the knowledge of the network. The tracking is terminated at different time points depending on whether there is a threshold and the kinds of the threshold. In the future, we will further investigate an inconsistency problem resulting from using KP to determine the broadcast range when partial knowledge is available. We will also investigate an energy-based metric accounting for energy consumption due to handoff in one hop.

CHAPTER VI

CONCLUSIONS

6.1 Contributions

In this section, we sum up our contributions of the thesis in the following list.

- In Chapter 5, we proposed an EB metric to maximize the number of time intervals that a hypothesized active node set could survive without further information handoff to another active set. Simulation results show EB provides longer lifetime than an existing GB metric.

- We proposed a transmission range control, called the knowledge pool (KP), where the transmission range is determined by the knowledge of the network and the remaining battery level of nodes. The energy consumption is proportional to the physical distance between the active set of nodes and is divided into two stages, one for active nodes to share the currently obtained information, the other for the currently active nodes to hand off the useful information to the next active set.

- We provided practical search algorithms to optimize a constrained metric using one metric as the optimization metric under the constraint of the other. Simulation results show the performance similarities among Greedy, Sequential and Exhaustive Search, but the Exhaustive Search is computationally expensive and the Greedy Search is three times faster than the Sequential Search. Metrics were applied to partial network knowledge where neighbors' information is available.

- In Chapter 4, we applied the resource management to multitarget tracking on the bearing measurements from the field data where the multitarget target tracker is either PDA or JPDA. The resource manager chooses a node set to minimize GB metric or a node set closest to the predicted target position. Experiments show that for sufficiently separated targets, the GB metric leads to better geolocation performance than the "closest" selection approach when the number of active nodes is set to two per snapshot and global network knowledge is available.

- In Chapter 3, we applied GB to partial network knowledge. Simulation results show that the RMS errors using partial node selection (PNS) and global node selection (GNS) are comparable while GNS is more energy-efficient when the energy is measured using an existing transmission control, critical range (CR). GB was also applied to partial network knowledge. We showed that using KP saves more energy than using CR and reaches the similar estimation performance when the available location information, i.e., r_{nei} is greater than 700m.

- In Chapter 2, we compared the performances of EKF and UKF and showed that UKF only outperforms EKF when the measurement noise is larger than eight degrees. We also investigated MM and IMM to tackle the maneuvering of target using field data test.

6.2 Future work

This section discusses possible future work related to resource management and target tracking.

6.2.1 Energy-based metric

There are some pros and cons of the solution maximizing (43) via the KP communication strategy. The strength is that maximizing (43) is tractable. The solution could be obtained via Greedy Search compared to nonmyopic scheduling which would have to be done via dynamic programming for all but simplest problems. The weakness is that (43) does not account for the energy consumption due to handoff in one hop although the reachable nodes of the active nodes are considered in the search space. Let $\mathcal{N}^\circ$ be the active node set at time $k-1$ and $\mathcal{N}$ be the chosen node set at time k. A better EB metric that includes energy consumption for handoff could be:

$$\mathbf{E}(\mathcal{N}) = \min_{i\in\mathcal{N}} \frac{p_i(k)}{\epsilon d_{i,\mathcal{N}}^4} + \min_{i\in\mathcal{N}^\circ} \frac{p_i(k) - \epsilon d_{i,\mathcal{N}}^4}{\epsilon d_{i,\mathcal{N}^\circ}^4}. \tag{53}$$

The term $\min_{i\in\mathcal{N}^\circ} \frac{p_i(k)-\epsilon d_{i,\mathcal{N}}^4}{\epsilon d_{i,\mathcal{N}^\circ}^4}$ in (53) tries to model how many time intervals the currently active node set $\mathcal{N}^\circ$ could still run after information handoff to $\mathcal{N}$. Note that it is the geometric constraints or the error threshold that forces handoffs to relatively far away nodes (Fig. 43). It is expected that GB-EB with handoff, i.e., maximizing EB (53) under the constraint of GB should prolong the lifetime. The discussion about GB-EB with handoff is worth studying in the future.

6.2.2 Transmission range control

The advantage of using CR is that nodes that provide better geometric views could be covered by active nodes when the available neighbor information is sparse. On the other hand, the strength of using KP is that when the available neighbor information is rich to some extent, using KP could save more energy while reaching similar geolocation performance compared to using CR. The problem of using KP is that active nodes may not be covered by each other especially when network knowledge is meager. Investigating the pros and cons of CR and KP further may lead to a better transmission range control as far as energy consumption and estimation performance are concerned.

6.2.3 Presence of multiple targets

The presence of multiple targets complicates the design of the distributed communication system which enables tracking ability. Clusters are formed where a cluster consists of an established target and its chosen node set. Information is shared within cluster. Information sharing between clusters is not required if PDA is used as a multitarget tracker. However, when two targets are close, a tracking expert prefers JPDA over PDA to track targets because JPDA explicitly considers the existence of multiple targets. As a result, multitarget tracking naturally requires information sharing between clusters. How to design a distributed or hierarchical communication system to address the presence of multiple targets is worthy of more attention.

APPENDIX A

RELEVANT PUBLICATION

1. Q. Le, L.M. Kaplan, and J.H. McClellan "Multiobjective energy-aware node selection," IEEE Aerospace Conference, Mar. 2006.

2. Q. Le, J.H. McClellan, and L.M. Kaplan, "Node selection for unattended ground sensor network when interrogating multiple targets," IEEE Army Science Conference, Nov. 2004.

3. L.M. Kaplan and Q. Le, "On exploiting propagation delays for passive target localization using bearings-only measurements," Journal of Franklin Institute, vol. 342, pp. 193-211, Mar. 2005.

4. Q. Le, J.H. McClellan, and L.M. Kaplan, "Multiple-mode Kalman fitering with node selection for wireless networks of bearings-only sensors," 36th IEEE Southeneastern Symposium on System Theory, Mar. 2004, vol.36, pp185–189.

5. Q. Le, L.M. Kaplan and J.H. McClellan, "Kalman filtering using bearings-only easurements from a network of acoustical arrays," ARL CTA Symposium, College Park, MD, April 29–May 1, 2003.

6. Q. Le, L.M. Kaplan and J.H. McClellan, "Kalman filters using bearings-only measurements from a network of acoustics arrays," 10th Digital Signal Processing Workshop, Pine Mountain, Georgia, October 13-16, 2002.

REFERENCES

[1] Bar-Shalom, Y. and Li, X., *Estimation and Tracking: Principles, Techniques, and Software.* Boston: Artech House, 1993.

[2] Bar-Shalom, Y. and Li, X., *Multitarget-Multisensor Tracking: Principles and Techniques.* YBS, 1995.

[3] Bertsekas, D. P., *Dynamic Programming and Optimal Control.* Belmont, MA: Athena Scientific, 1995.

[4] Bertsekas, D. P. and Castanon, D., "Rollout algorithms for stochastic scheduling problems," *Journal of Heurtistics*, vol. 5, pp. 89–108, 1999.

[5] Bhardwaj, M., Garnett, T., and Chandrakasan, A. P., "Upper boundes on the lifetime of sensor networks," in *Proc. of ICC*, 2001.

[6] Blackman, S., "Multiple hypothesis tracking for multiple target tracking," *IEEE AES Systems magazine*, vol. 19, pp. 5–18, Jan. 2004.

[7] Blackman, S. and Popoli, R., *Design and Analysis of Modern Tracking Systems.* Boston: Artech House, 1999.

[8] Castanon, D. A., "Approximate dynamic programming for sensor management," *Proceedings of the 36th IEEE Conference on Decision and Control*, vol. 2, pp. 1202–1207, 1997.

[9] Cevher, V. and McClellan, J. H., "General directional-of-arrival tracking with acoustic nodes," *to appear in IEEE Trans. on Signal Processing*, Feb. 2003.

[10] Chang, J. H. and Tassiulas, L., "Energy conserving routing in wireless adhoc networks," in *Proc. IEEE INFOCOM*, pp. 22–31, 2000.

[11] Chen, H. and Olson, T., "Adaptive spatiotemporal multiple sensor fusion," *Optical Engineering*, vol. 42, pp. 1481–1495, May 2003.

[12] Chen, P., O'Dea, B., and Callaway, E., "Energy efficient system design with optimum transmission range for wireless ad hoc networks," in *IEEE International Conference on Communications*, pp. 945–952, 2002.

[13] Chhetri, A. S., Morrell, D., and Papandreou-Suppappola, A., "Scheduling multiple sensors using partical filters in target tracking," in *Proc. of IEEE Statistical and Signal Processing Workshop*, pp. 529–532, 2003.

[14] Chhetri, A. S., Morrell, D., and Papandreou-Suppappola, A., "Efficient search strategies for non-myopic sensor scheduling in target tracking," in *Proc.of the Thirty-Eighth Asilomar Conference on Signals, Systems and Computers*, vol. 2, pp. 2106–2110, 2004.

[15] Chhetri, A. S., Morrell, D., and Suppappola, A. P., "Energy efficient target tracking in a sensor network using non-myopic sensor scheduling," in *Proc. of the 8th Intl. Conf. on Information Fusion*, July 2005.

[16] Chong, C. Y., Zhao, F., Mori, S., and Kumar, S., "Distibuted tracking in wireless ad hoc sensor network," *Proceeding of the 6-th international Conference on Information Fusion*, vol. 1, pp. 431–438, 2003.

[17] Chu, M., Haussecker, H., and Zhao, F., "Scalable information-driven sensor querying and routing for ad hoc heterogeneous sensor networks," in *International Journal of High Performance Computing Applications*, vol. 16, Aug. 2002.

[18] Efe, M., "Adaptive Kalman filters for manoeuvring target tracking," in *IEE Colloquium on target tracking and data fusion*, pp. 4/1–4/7, June 1998.

[19] Efe, M. and Atherton, D., "Maneuvering target tracking using adaptive turn rate models in the interacting multiple model algorithm," in *Proc. of the 35th IEEE on Decision and control*, pp. 3151–3156, Dec. 1996.

[20] Farina, A., "Target tracking with bearing-only measurements," *Signal Processing*, vol. 78, pp. 61–78, Jan. 1999.

[21] Farina, A., Golino, G., Capponi, A., and Pilotto, C., "Surveillance by means of a random sensor network: a heterogeneous sensor approach," in *Conf. on Information Fusion*, July 2005.

[22] Gomez, J. and Campbell, A. T., "A case for variable-range transmission power control in wireless multihop networks," in *INFOCOM*, vol. 2, pp. 1425–1436, 2004.

[23] Heinzelman, W. B. and Chandrakasan, A. P., "An application-specific protocol architecture for wireless microsensor networks," *IEEE Trans. on Wireless Communications*, vol. 1, pp. 660–669, Apr. 2002.

[24] Hu, Z. and Li, B., "On the fundamental capacity and lifetime limits of energy-constrained wireless sensor network," in *Proc. of the 10th IEEE Real-Time and Embedded Techonology and Applications Symposium (RTAS/04)*, 2004.

[25] Julier, S., Uhlmann, J., and Durrant-Whyte, H., "A new method for the nonlinear transformation of means and covariances in filters and estimators," *IEEE Trans. on Automatic Control*, vol. 45, pp. 477–482, Mar. 2000.

[26] Kalman, R. E., "A new approach to linear filtering and prediction problems," *AEME. J, Basic Engineering*, vol. 83, pp. 34–45, Mar. 1960.

[27] Kalman, R. E. and Bucy, R., "New results in linear filtering and prediction theory," *AEME. J, Basic Engineering*, vol. 83, pp. 95–108, Mar. 1961.

[28] Kaplan, L. M., "Node selection for target tracking using bearing measurements from unattended ground sensors," in *Proc. of the IEEE Aerospace Conference*, (Big Sky, MT), Mar. 2003.

[29] Kaplan, L. M., "Transmission range control during autonomous node selection forwireless sensor networks," in *Proc. of the 2004 IEEE Aerospace Conference*, 2004.

[30] Kaplan, L. M., "Global selection for target localization in a distributed network of bearings-only sensors." to be published in *IEEE Trans. on Aerospace and Electronic Systems*, Oct. 2005.

[31] Kaplan, L. M., "Local selection for target localization in a distributed network of bearings-only sensors." to be published in *IEEE Trans. on Aerospace and Electronic Systems*, Jan. 2006.

[32] Kaplan, L. M. and Le, Q., "On exploiting propagation delays for passive target localization using bearings-only measurements," *Journal of Franklin Institute*, vol. 342, pp. 193–211, Mar. 2005.

[33] Kaplan, L. M., Molnar, P., and Le, Q., "Bearings-only target localization for an acoustical unattended ground sensor network," in *Proc. of SPIE*, vol. 4393, pp. 40–51, Apr. 2001.

[34] Kaplan, L. M., Molnar, P., Srour, N., and Filipov, A., "Autonomous node selection for wireless networks of bearings-only sensors," in *Proc. of the 23rd Army Science Conference*, (Orlando, FL), Dec. 2002.

[35] Kreucher, C., Hero, A., Kastella, K., and Chang, D., "Efficient methods of non-myopic sensor management for multitarget tracking," in *Proc. of the 43th IEEE Conf. on Decision and Control*, pp. 722–727, 2004.

[36] Kreucher, C., Kastella, K., and Hero, A., "A Bayesian method for intergrated multitarget tracking and sensor management," in *Proc.of the 6th Int. Conf. on Information Fusion*, pp. 704–712, 2003.

[37] Kreucher, C. M. and III, A. O. H., "Non-myopic approaches to scheulding agile sensors for multitarget detection, tracking, and identification," in *ICASSP*, vol. 5, pp. 885–888, 2005.

[38] Kreucher, C. M., Kastella, K., and III, A. O. H., "Information based sensor management for multitarget tracking," in *Proceedings of SPIE Conference on Signal and Data Processing of Small Targets*, 2003.

[39] Krishnamurthy, V. and Evans, R. J., "Hidden Markov model multiarm bandits: a methodology for beam scheduling in multitarget tracking," *IEEE Trans. on Signal Processing*, vol. 49, no. 12, pp. 2893–2908, 2001.

[40] Le, Q., Kaplan, L. M., and McClellan, J. H., "Kalman filtering using bearings-only measurements from a network of acoustical arrays," in *ARL CTA Symposium*, (College Park, MD), 2003.

[41] Le, Q., Kaplan, L. M., and McClellan, J. H., "Multiobjective energy-aware node selection," in *IEEE Aerospace Conference*, (Big Sky, MT), 2006.

[42] Le, Q., McClellan, J. H., and Kaplan, L. M., "Multiple-mode Kalman fitering with node selection for wireless networks of bearings-only sensors," in *36th IEEE Southeneastern Symposium on System Theory*, vol. 36, pp. 185–189, 2004.

[43] Le, Q., McClellan, J. H., and Kaplan, L. M., "Node selection for unattended ground sensor network when interrogating multiple targets," in *Proc. of the Army Science Conference*, 2004.

[44] Le Cadre, J.-P. and Tremois, O., "Bearings-only tracking for maneuvering sources," *IEEE Trans. on Aerospace and Electronic Systems*, vol. 34, pp. 179–193, Jan. 1998.

[45] Li, X. R. and Bar-Shalom, Y., "Multiple-model estimation with variable structure," *IEEE Trans. on Automatic Control*, vol. 41, pp. 478–493, Apr. 1996.

[46] Li, X. R. and Zhang, Y., "Multiple-model estimation with variable structure. V. Likely-model set algorithm," *IEEE Trans. on Aerospace and Electronic Systems*, vol. 36, pp. 448–466, Apr. 2000.

[47] Liu, J., Reich, J., and Zhao, F., "Collaborative in-network processing for target tracking," *EURASIP Journal on Applied Signal Processing*, vol. 2003, pp. 378–391, Mar. 2003.

[48] Madan, R. and Lall, S., "Distributed algorithms for maximum lifetime routing in wireless sensor networks," in *IEEE Communications Society Globecom*, 2004.

[49] Magill, D., "Optimal adaptive estimation of sampled stochastic processes," *IEEE Trans. on Automatic Control*, vol. 10, pp. 434–439, Oct. 1965.

[50] Mahler, R., "Global optimal sensor allocation," in *Proceedings of the Ninth National Symposium on Sensor Fusion*, vol. 1, pp. 167–172, 1996.

[51] Mehra, R. E., "On the identification of variances and adaptive Kalman filtering," *IEEE Trans. on Automatic Control*, vol. 15, pp. 175–184, Apr. 1970.

[52] Myers, K., "Adaptive sequential estimation with unknown noise statistics," *IEEE Trans. on Automatic Control*, vol. 21, pp. 520–523, Aug. 1976.

[53] Nardone, S., Lindgren, A., and Gong, K., "Fundamental properties and performance of conventional bearings-only target motion analysis," *IEEE Trans. on Automatic Control*, vol. 29, pp. 775–787, Sept. 1984.

[54] Orton, M. and Fitzgerald, W., "A Bayesian approach to tracking multiple targets using sensor arrays and particle filters," vol. 50, no. 2, pp. 216–223, 2002.

[55] Oshman, Y. and Davidson, P., "Optimization of observer trajectories for bearings-only target localization," *IEEE Trans. on Aerospace and Electronic Systems*, vol. 35, pp. 892–902, July 1999.

[56] Poor, H., *An Introduction to Signal Detection and Estimation.* Springer-Verlag, 1994.

[57] Raghunathan, V., Schurgers, C., Park, S., and Srivastava, M., "Energy-aware wireless microsensor networks," *IEEE Signal Processing Magazine*, vol. 19, pp. 40–50, Mar. 2002.

[58] Rao, B. S. and Durrant-Whyte, H. F., "Fully decentralised algorithm for multisensor Kalman filtering," in *IEE Proceedings D*, vol. 138, pp. 413–420, Sept. 1991.

[59] Reid, D. B., "An algorithm for tracking multiple targets," *IEEE Trans. on Automatic Control*, vol. 24, pp. 843–854, Dec. 1979.

[60] Rodoplu, V. and Meng, T. H., "Minimum energy mobile wireless networks," *IEEE Journal on Selected Areas in Communications*, vol. 17, pp. 1333–1344, Aug. 1999.

[61] Roecker, J. A., "A class of near optimal JPDA algorithms," *IEEE Trans. on Aerospace and Electronic Systems*, vol. 30, pp. 504–510, Apr. 1994.

[62] Shepard, T., "Decentralized channel management in scalable multihop spread spectrum packet radio networks," in *Laboratory for Computer Science,Massachusetts Institute of Technology, Cambridge, Tech. Rep.MIT/LCS/TR-670*, July 1995.

[63] Wang, H., Yao, K., Pottie, G., and Estrin, D., "Entropy-based sensor selection heuristic for target tracking localization," in *Proc. of the 3rd Intl. Symp. on Information Processing in Sensor Networks (IPSN)*, pp. 36–45, 2004.

[64] Watson, G. A., "Multisensor ESA resource management," in *IEEE Aerospace Conference*, vol. 5, pp. 13–27, 1998.

[65] Williams, J. L., Fisher III, J. W., and Willsky, A. S., "An approximate dynamic programming approach for communication constrained inference," in *Proc. IEEE Workshop on Statistical Signal Processing*, July 2005.

[66] Williams, J. L., Fisher III, J. W., and Willsky, A. S., "An approximate dynamic programming approach to a communication constrained sensor management problem," in *Proc. Eighth International Conference of Information Fusion*, July 2005.

[67] Wilson, D. K., Sandler, B. M., and Pham, T., "Simulation of detection and beamforming with acousitcal ground sensors," in *Proc. of SPIE*, vol. 4743, pp. 50–61, 2002.

[68] Wu, J. and Dai, F., "Efficient broadcasting with guaranteed coverage in mobile ad hoc networks," *IEEE Transactions on Mobile Computing*, vol. 4, pp. 259–270, May 2005.

[69] Xiong, N. and Svensson, P., "Multi-sensor management for information fusion: Issues and approaches," *Information fusion*, vol. 3, pp. 163–186, 2002.

[70] Zhang, X. and Maxemchuk, N. F., "A generalized energy consumption analysis in multihop wireless networks," in *Wireless Communications and Networking Conference*, pp. 1476–1481, 2004.

[71] Zhao, F., Shin, J., and Reich, J., "Information-driven dynamic sensor collaboration," *IEEE Signal Processing Magazine*, vol. 19, pp. 61–72, Mar. 2002.

[72] Zou, Y. and Chakrabarty, K., "Energy-aware target localization in wireless sensor networks," in *Proceedings of the First IEEE International Conference on Pervasive Computing and Communications*, pp. 60–67, 2003.

[73] Zou, Y. and Chakrabarty, K., "Target localization based on energy considerations in distributed sensor networks," in *Proceedings of SPIE Conference on Signal and Data Processing of Small Targets*, 2003.

VITA

Qiang Le was born in a central southern town of People's Republic of China. She had lived there for eighteen years. During her pursuing high education in Beijing, she fortunately met her future husband. Then she came to USA and continued her education.