
Optimal Motion Strategies for Range-only Distributed Target Tracking

Ke Zhou and Stergios I. Roumeliotis
{kezhou|stergios}@cs.umn.edu

Department of Computer Science & Engineering
University of Minnesota

MARS LAB

Multiple Autonomous
Robotic Systems Laboratory

Technical Report
Number -2006-004
April 2006

Dept. of Computer Science & Engineering
University of Minnesota
4-192 EE/CS Building
200 Union St. S.E.
Minneapolis, MN 55455
Tel: (612) 625-2217
Fax: (612) 625-0572

Optimal Motion Strategies for Range-only Constrained Multi-sensor Target Tracking

Ke Zhou[†] and Stergios I. Roumeliotis[‡]

Abstract—In this paper, we study the problem of *optimal trajectory generation* for a team of mobile sensors tracking a moving target using distance-only measurements. This problem is shown to be *NP-Hard*, in general, when constraints are imposed on the speed of the sensors. We propose two algorithms, *modified Gauss-Seidel-relaxation* and *LP-relaxation*, for determining the set of feasible locations that each sensor should move to in order to collect the most *informative* measurements; i.e., distance measurements that minimize the uncertainty about the position of the target. Furthermore, we prove that the motion strategy that minimizes the trace of the position error covariance matrix is equivalent to the one that maximizes the minimum eigenvalue of its inverse. The two proposed algorithms are applicable regardless of the process model that is employed for describing the motion of the target, while the computational complexity of both methods is *linear* in the number of sensors. Extensive simulation results are presented demonstrating that the performance attained with the proposed methods is comparable to that obtained with *grid-based exhaustive search*, whose computational cost is exponential in the number of sensors, and significantly better than that of a *random*, towards the target, motion strategy.

Index Terms—Mobile Sensor, Target Tracking, Distance Measurement, Modified Gauss-Seidel Relaxation, LP Relaxation.

I. INTRODUCTION

Target tracking has recently attracted significant interest in the research community because of its importance in a variety of applications, such as environmental monitoring [1], surveillance [2], [3], human-robot interaction [4], as well as defence applications [5]. In order to obtain increased tracking accuracy and monitor extensive areas, a large number of sensors are often utilized for tracking, while communicating over a wireless sensor network. When multiple nodes obtain measurements of a target of interest, the acquired data can be processed, either at a fusion center, or in a distributed fashion, in order to estimate the target's trajectory.

As an alternative to using *static* sensors, the deployment of *mobile* sensors (i.e., *robots*) for tracking offers significant advantages. By providing mobility to the sensors, a larger area can be covered without the need to increase the number of nodes in the sensing network [6]. Additionally, the spatial distribution of the sensors can be changed dynamically in order to adapt to the motion of the target. For example, a team of sensors can actively pursue a target, to avert the target's evasion from the sensors' visibility range [7].

Regardless of the estimation algorithm employed in a given application, the processing of every new measurement by a

networked tracking system incurs a penalty in terms of use of communication bandwidth and CPU time, as well as in terms of power consumption. Since these resources are inevitably limited, it is necessary to devise *active sensing* algorithms that guarantee their optimal utilization. Moreover, in many tracking applications the time needed for determining the trajectory of a target is critically important (e.g., when tracking a hostile target). Sensors that actively pursue a target and move to locations where they collect the most informative measurements, can achieve optimal tracking performance. That is, they will minimize the uncertainty about the position of the target significantly faster compared to a random motion strategy.

In this paper, we study the problem of determining *optimal trajectories* for a team of sensors that track a moving target using *range* (distance) measurements. Since the measurement model is non-linear, the locations where distance measurements are collected have a profound effect on the estimation accuracy. Consider, for example, the simple case of a single sensor tracking a target using *distance* measurements corrupted by Gaussian noise (cf. Fig. 1(a) and Fig. 1(b)). In this scenario, the *prior* uncertainty for the position of the target $P_{k+1|k}$ is depicted by the solid-line 3σ ellipse shown in Fig. 1(a). If the sensor remains still and measures the distance to the target, then based solely on this measurement, the sensor believes that the target is within the dotted-line circular ring with probability 99.7%. Combining the *prior* estimate with this measurement, the *posterior* uncertainty $P_{k+1|k+1}$ is only slightly reduced (dashed-line ellipse in Fig. 1(a)). As evident, by remaining in the same position, the sensor's measurement provides limited information for the target's position along the x direction. If instead, the sensor moves to a new location (cf. Fig. 1(b)), then combining this new measurement with the prior estimate will result in significant reduction of the uncertainty in both directions, but primarily along the x -axis. The improved confidence in the target-position estimate after this informative measurement is processed, is depicted by the small dashed-line ellipse in Fig. 1(b).

In this work, we extend this intuitive strategy to the case of one *moving target* and multiple *moving sensors* and determine the optimal trajectories that the sensors should follow to minimize the error in the posterior estimate of the position of the target [8]. Here, optimality is sought with respect to the accuracy of the target's localization, i.e., we seek to minimize the trace of the covariance matrix of the target's position estimate. We show that regardless of the target's motion model, this optimization problem can be exactly reformulated as that of minimizing the norm of the sum of a set of vectors of known length (cf. Section III). The motion direction of each

[†]Dept. of Electrical & Computer Engineering, University of Minnesota, Minneapolis, MN, 55455. Email: kezhou@cs.umn.edu. [‡]Dept. of Computer Science & Engineering, University of Minnesota, Minneapolis, MN, 55455. Email: stergios@cs.umn.edu.

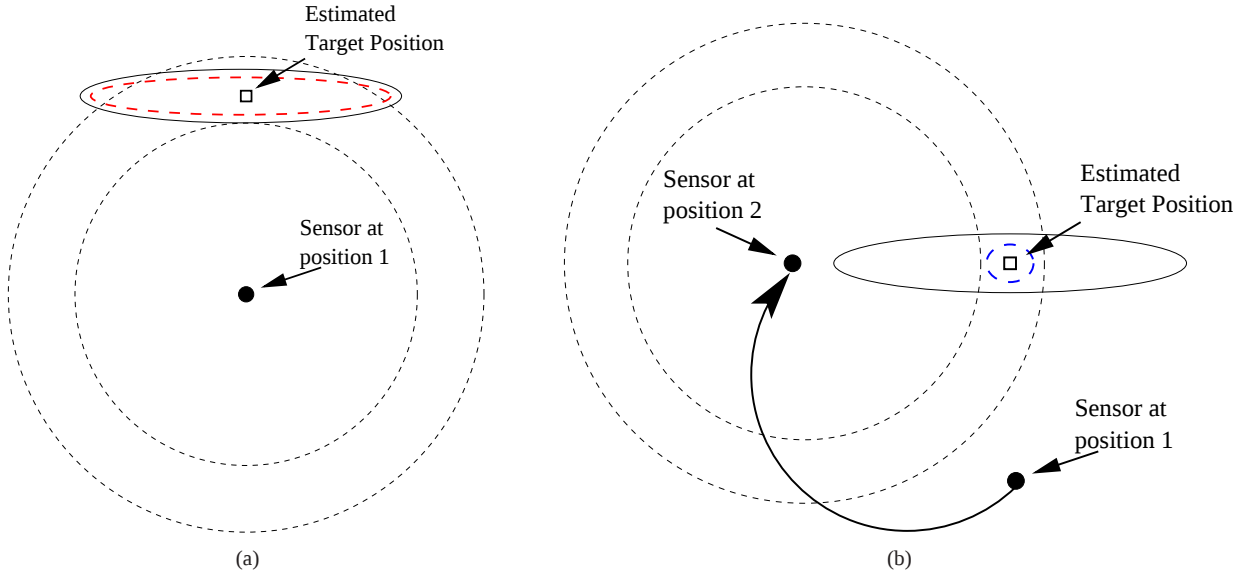


Fig. 1. (a) Suboptimal target tracking: The sensor remains in the same location. (b) Optimal target tracking: The sensor moves to the position that minimizes the uncertainty for the target's position along the x -axis. In both plots, the *prior uncertainty* (3σ) is denoted by a solid-line ellipse, the *posterior* by a dashed-line ellipse, while the *measurement uncertainty* is depicted as a circular ring (dotted-line) with center the location of the sensor.

sensor affects the direction of the corresponding vector, while the speed of motion determines the range of possible angles (constraints) for each vector. We also prove that this optimization problem is indeed *NP-Hard* in general (cf. Section IV) and show that minimizing the trace of the covariance matrix is *equivalent* to maximizing the minimum eigenvalue of its inverse (cf. Section III).

Two novel *relaxation* algorithms, *modified Gauss-Seidel relaxation* and *Linear Programming (LP) relaxation*, are proposed for solving this problem (cf. Section V), and it is shown through extensive simulation studies that the performance attainable with each of them is comparable to that obtained with a *grid-based exhaustive search* algorithm. While the computational complexity of exhaustive search is prohibitively large (exponential in the number of sensors), both proposed relaxation methods have complexity only linear in the number of sensors, and are thus well-suited for real-time implementations. Additionally, the accuracy achieved by both *modified Gauss-Seidel relaxation* and *LP relaxation* is significantly better than that obtained when following a “random” motion strategy (cf. Section VI).

Following a brief review of related literature in Section II, we present the formulation of the target tracking problem in Section III. In Section IV, we show that the problem is NP-hard. We describe two proposed relaxation algorithms in Section V. Extensive simulation results are presented in Section VI. Finally, in Section VII the conclusions of this work are drawn and future research directions are suggested.

II. LITERATURE REVIEW

Target tracking has received considerable attention in the literature (e.g., [9], [10]). In most cases, however, the sensors involved are *static* and the emphasis is on the optimal processing of the available information rather than the placement or repositioning of the sensors. The idea of choosing sensing locations in order to maximize information gain (also known

as adaptive sensing or active perception [11]) has been applied to the problems of cooperative localization [12], Simultaneous Localization and Mapping (SLAM) [11], parameter detection [13], [14], and optimal sensor selection [15]. In particular for single-sensor target tracking using *bearing-only* measurements, significant work has been presented in [16], [17], [18], [19], [20]. In these cases, the most common optimization criterion used is the determinant of the Fisher Information Matrix (FIM) over a horizon of one or multiple time steps.

Due to the key differences in the observation model when distance, instead of bearing, measurements are used and the implications on the selection process of the next best sensing location, we hereafter limit our discussion to single- and multi-sensor active sensing approaches that use *distance and bearing* or *distance-only* measurements to the target. The latter case is the main focus of our work.

A. Active target tracking - distance and bearing measurements

Stroupe and Balch [21] propose an approximate tracking behavior, in which the mobile sensors attempt to minimize the target location uncertainty using distance and bearing measurements. The objective function is the *determinant* of the target position estimates' covariance matrix. The optimization process in this case does not consider the set of all possible trajectories. Instead, a *greedy search* is performed over the discretized set of possible headings, separately for each sensor. Additionally, the proposed algorithm approximates the expected information gain from the teammates' actions by assuming that the other sensors' measurements in the next time step will be the same as these recorded at their current locations.

Olfati-Saber [22] addresses the problem of distributed target tracking for mobile sensor networks with a dynamic communication topology. The author tackles the network connectivity issue using a flocking-based mobility model, and presents a

modified version of the distributed Kalman filter algorithm for estimating the target's state. In this case, the sensors use both distance and bearing measurements to a target that moves in 2D with constant velocity driven by zero-mean Gaussian noise. The objective of the proposed algorithm is to *minimize the distance* between the sensor and the target, while at the same time avoid collisions. The proposed algorithm does not consider positioning information from previous steps.

Chung *et al.* [23] present a decentralized motion planning algorithm for solving the multi-sensor target tracking problem using both distance and bearing measurements. The authors employ the *determinant* of the target's position covariance matrix as the cost function. The decentralized control law in this case is based on the *gradient* of the cost function with respect to each of the sensor's coordinates with constant step-size of 1. The authors, however, do not account for the speed constraints on the motion of sensors. In addition, the convergence rate of the gradient-based method and the existence of local minima are not considered.

B. Active target tracking - distance-only measurements

Contrary to [23], where sensors can transmit/receive information to/from all their teammates, in [24] they are confined to communicate with one-hop neighbors only. In this case, distance-only measurements are used, while both the trace and the determinant of the covariance matrix for the target's position estimates are considered as objective functions. The control law, with constant step size, is computed from the *gradient* of the cost function with respect to each sensor's coordinates. However, as is the case for [23], physical constraints on the motion of the sensors are not accounted for. Furthermore, the impact of the step-size selection on the convergence of the algorithm and the existence of local minima are not considered.

In [25], Martínez and Bullo address the problem of optimal sensor placement and motion coordination strategies for mobile sensor networks using distance-only measurements. In this case, all the sensors are assumed to be identical (i.e., same level of accuracy in the distance measurements). The authors consider the optimal sensor placement for (non random) *static* target position estimation. The objective is to maximize the *determinant* of the FIM, or equivalently minimize the determinant of the covariance matrix. However, the optimization process does not address the *dynamic* target case. Instead, the authors argue that the optimal sensor placement derived for the static target scenario is also expected to have good performance in the dynamic case. By not considering the prior estimates and assuming a homogeneous sensor team with no motion constraints, the optimal placement of the sensors can be computed analytically. The resulting control law requires that the sensors move on a polygon surrounding the static target so as the vectors from the target to each sensor are uniformly (in terms of direction) spaced.

The main drawback of the previous approaches is that no constraints on the speed of the sensors are considered. Furthermore, their impact on the computational complexity of the optimization algorithm used is not examined. The only

exception is the work presented in [21]. In that case, however, these constraints are used only to define the discretized region over which the heading of each sensor is optimized *independently* (i.e., each sensor determines its next sensing location without considering the constraints on the motion of its teammates).

In this paper, we address the problem of constrained optimal target tracking using distance measurements collected from teams of *heterogeneous* sensors. In the problem formulation, we account for the existence of *prior information*, the impact of which can be appreciated from the simple example shown in Figs. 1(a) and 1(b). Furthermore, we consider constraints on the speed of the sensors and prove that their inclusion makes the problem *NP-Hard*. Based on appropriate relaxations of the original problem, we propose two algorithms, *modified Gauss-Seidel relaxation* and *LP relaxation*, that minimize the trace of the target's position estimate covariance matrix with respect to the motion of *all* sensors concurrently. Both algorithms have computational complexity *linear* in the number of sensors and achieve tracking accuracy indistinguishable of that of an exhaustive search over all possible combinations of the sensors' directions.

III. PROBLEM FORMULATION

Consider a group of mobile sensors (or robots) moving in a plane and tracking the position of a moving target by processing distance measurements. In this paper, we study the case of global tracking, i.e., the position of the target is determined with respect to a fixed (global) frame of reference, instead of a relative *group-centered* one. Hence, we hereafter employ the assumption that the position and orientation (pose) of each of the tracking sensors are known with high accuracy within the global frame of reference; this is the case, for example, when all sensors have access to precise GPS and compass measurements.

Furthermore, we consider the case where each sensor can move in 2D with speed v_i , which is *upper bounded* by v_{imax} , $i = 1, \dots, M$, where M is the number of sensors. Therefore, at time-step $k + 1$, sensor- i can only move within a circular region centered at its position at time-step k with radius $r = v_{imax} \delta t$ (cf. Fig. 2), where δt is the time step. Note also that since the motion of the target can be reliably predicted for the next time step only, our objective is to determine the next best sensing locations for all sensors at every time step.

In the next two sections, we present the target's state propagation equations and the sensors' measurement model.

A. State Propagation

In this work, we employ the Extended Kalman Filter (EKF) for recursively estimating the target's state, $\mathbf{x}_T(k)$. This is defined as a vector of dimension $2N$, where N is the highest order time derivative of the position described by the motion model, and can include components such as position, velocity, and acceleration:

$$\mathbf{x}_T(k) = [x_T(k) \ y_T(k) \ \dot{x}_T(k) \ \dot{y}_T(k) \ \ddot{x}_T(k) \ \ddot{y}_T(k) \ \dots]^T \quad (1)$$

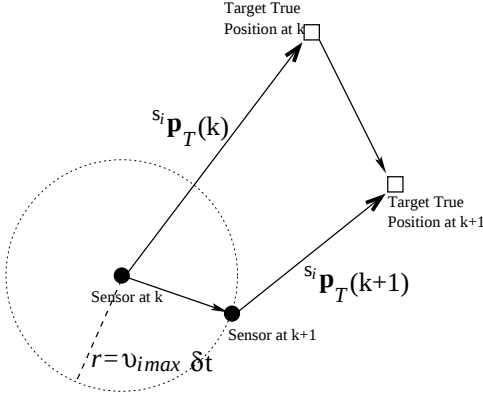


Fig. 2. Illustration of sensor's and target's motion: Each sensor moves in 2D with speed v_i , which is bounded by v_{imax} . From time-step k to $k+1$, the sensor can only move within a circular region centered at its position at time-step k with radius $r = v_{imax}\delta t$. ${}^{s_i}\mathbf{p}_T$ is the target's position with respect to sensor- i . The distance measurement is the norm of ${}^{s_i}\mathbf{p}_T$ plus noise.

We consider the case that the target moves randomly and assume that we know the stochastic model describing the motion of the target (e.g., constant-acceleration or constant-velocity, etc). However, as it will become evident later on, neither of our sensing strategies depends on the particular selection of the target's motion model.

The discrete-time state propagation equation is:

$$\mathbf{x}_T(k+1) = \Phi_k \mathbf{x}_T(k) + G_k \mathbf{w}_d(k) \quad (2)$$

where $\mathbf{w}_d$ is a zero-mean white Gaussian noise process with covariance $Q_d = E[\mathbf{w}_d(k)\mathbf{w}_d^T(k)]$.

The estimate of the target's state is propagated by:¹

$$\hat{\mathbf{x}}_T(k+1|k) = \Phi_k \hat{\mathbf{x}}_T(k|k) \quad (3)$$

where $\hat{\mathbf{x}}_T(\ell|j)$ is the state estimate at time-step ℓ , after measurements up to time-step j have been processed.

The error-state covariance matrix is propagated as:

$$P_{k+1|k} = \Phi_k P_{k|k} \Phi_k^T + G_k Q_d G_k^T$$

where $P_{\ell|j}$ is the covariance of the error, $\tilde{\mathbf{x}}_T(\ell|j)$, in the state estimate. The state transition matrix, Φ_k , and the process noise Jacobian, G_k , that appear in the preceding expressions depend on the motion model used [26]. In our work, these can be arbitrary matrices, since no assumptions on their properties are imposed.

B. Measurement model

At time-step $k+1$, each sensor of the team measures its distance to the target, as shown in Fig. 2, and therefore the measurement equation is:

$$\mathbf{z}(k+1) = \begin{bmatrix} d_1(k+1) \\ \vdots \\ d_M(k+1) \end{bmatrix} + \begin{bmatrix} n_1(k+1) \\ \vdots \\ n_M(k+1) \end{bmatrix} = \mathbf{d}(k+1) + \mathbf{n}(k+1) \quad (4)$$

¹In the remainder of the paper, the “hat” symbol, $\hat{\cdot}$, is used to denote the estimated value of a quantity, while the “tilde” symbol, $\tilde{\cdot}$, is used to signify the error between the actual value of a quantity and its estimate. The relationship between a variable, x , and its estimate, $\hat{x}$, is $\tilde{x} = x - \hat{x}$.

with (for $i = 1, \dots, M$)

$$\begin{aligned} d_i(k+1) &= \sqrt{{}^{s_i}\mathbf{p}_T(k+1)^T {}^{s_i}\mathbf{p}_T(k+1)} \\ &= \sqrt{(\mathbf{p}_T(k+1) - \mathbf{p}_i(k+1))^T (\mathbf{p}_T(k+1) - \mathbf{p}_i(k+1))} \\ &= \sqrt{(x_T(k+1) - x_i(k+1))^2 + (y_T(k+1) - y_i(k+1))^2} \end{aligned}$$

where ${}^{s_i}\mathbf{p}_T(k+1)$ is the position of the target with respect to sensor- i , and $\mathbf{p}_T(k+1) = [x_T(k+1) \ y_T(k+1)]^T$, $\mathbf{p}_i(k+1) = [x_i(k+1) \ y_i(k+1)]^T$ are the positions of the target and the sensor respectively, expressed in the global frame of reference. Note also that $n_i(k+1)$ is the noise in the i -th sensor's distance measurement, which is a zero-mean white Gaussian process, independent of the noise in other sensors, with variance $E[n_i(k+1)n_j(k+1)] = \sigma_i^2 \delta_{ij}$, where δ_{ij} is the Kronecker delta.

The measurement equation (4) is a nonlinear function of the state variables. The measurement-error equation, obtained by linearizing (4) is:

$$\begin{aligned} \tilde{\mathbf{z}}(k+1|k) &= \mathbf{z}(k+1) - \hat{\mathbf{z}}(k+1|k) \\ &\simeq H_{k+1} \tilde{\mathbf{x}}_T(k+1|k) + \mathbf{n}(k+1) \end{aligned} \quad (5)$$

where

$$\begin{aligned} \hat{\mathbf{z}}(k+1|k) &= [\hat{d}_1(k+1|k) \ \dots \ \hat{d}_M(k+1|k)]^T \\ \hat{d}_i(k+1|k) &= \sqrt{\widehat{\Delta x_{T_i}^2}(k+1|k) + \widehat{\Delta y_{T_i}^2}(k+1|k)} \\ \widehat{\Delta x_{T_i}}(k+1|k) &= \hat{x}_T(k+1|k) - x_i(k+1) \\ \widehat{\Delta y_{T_i}}(k+1|k) &= \hat{y}_T(k+1|k) - y_i(k+1) \\ \tilde{\mathbf{x}}_T(k+1|k) &= \mathbf{x}_T(k+1) - \hat{\mathbf{x}}_T(k+1|k) \end{aligned}$$

Note that the measurement matrix in (5) has a block column structure, which is given by the following expression:

$$H_{k+1} = [H_{e,k+1} \ \mathbf{0}_{M \times (2N-2)}] \quad (6)$$

where $2N$ is the dimension of the state vector and

$$H_{e,k+1}^T = \begin{bmatrix} \cos \theta_1(k+1) & \dots & \cos \theta_M(k+1) \\ \sin \theta_1(k+1) & \dots & \sin \theta_M(k+1) \end{bmatrix} \quad (7)$$

$$\cos \theta_i(k+1) = \frac{\widehat{\Delta x_{T_i}}(k+1|k)}{\hat{d}_i(k+1|k)} \quad (8)$$

$$\sin \theta_i(k+1) = \frac{\widehat{\Delta y_{T_i}}(k+1|k)}{\hat{d}_i(k+1|k)} \quad (9)$$

The angle θ_i that appears in the preceding equations represents the *bearing angle* of sensor- i towards the estimated position of the target, expressed in global coordinates (cf. Fig. 3).

C. State and Covariance Update

Once the distance measurements, $\mathbf{z}(k+1)$, from all the sensors are available, the target's state estimate and its covariance are updated as:

$$\begin{aligned} \hat{\mathbf{x}}_T(k+1|k+1) &= \hat{\mathbf{x}}_T(k+1|k) + K_{k+1} \tilde{\mathbf{z}}(k+1|k) \\ P_{k+1|k+1} &= P_{k+1|k} - K_{k+1} S_{k+1} K_{k+1}^T \end{aligned} \quad (10)$$

where $K_{k+1} = P_{k+1|k} H_{k+1}^T S_{k+1}^{-1}$ is the Kalman gain, $S_{k+1} = H_{k+1} P_{k+1|k} H_{k+1}^T + R$ is the measurement residual covariance,

and $R = \text{diag}(\sigma_i^2)$ is the measurement noise covariance.

Our objective in this work is to determine the active sensing strategy that minimizes the uncertainty for the *position* estimate of the target. In order to account for the impact of the prior state estimates on the motion of the sensors, we first prove the following lemma.

Lemma 1: The posterior (updated) covariance for the target's position estimate depends on (i) the prior (propagated) covariance sub-matrix of the target's *position* (i.e., it is independent of the uncertainty in the estimates of higher-order time derivatives of the position such as velocity, acceleration, etc, and hence it is independent of the target's motion model) and (ii) the measurement information matrix corresponding to the target's *position*, i.e.,

$$P_{k+1|k+1,11} = \left((P_{k+1|k,11})^{-1} + H_{e,k+1}^T R^{-1} H_{e,k+1} \right)^{-1} \quad (11)$$

Proof: The covariance matrices appearing in (11) are defined based on the following partition:

$$P_{\ell|j} = \begin{bmatrix} P_{\ell|j,11} & P_{\ell|j,12} \\ P_{\ell|j,12}^T & P_{\ell|j,22} \end{bmatrix} \quad (12)$$

where the 2×2 matrix $P_{\ell|j,11}$ denotes the covariance for the target's *position* estimate, $\hat{\mathbf{p}}_T = [\hat{x}_T \ \hat{y}_T]^T$, at time-step ℓ given measurements up to time-step j .

Employing the matrix inversion lemma, the covariance update equation, (cf. (10)) can be written as

$$P_{k+1|k+1}^{-1} = P_{k+1|k}^{-1} + H_{k+1}^T R^{-1} H_{k+1} \quad (13)$$

Note that if the state vector contains only the position of the target, then (11) is identical to (13).

In the general case, when the state vector also contains higher order derivatives of the position (e.g., velocity, acceleration, etc), substituting

$$P_{k+1|k}^{-1} = \begin{bmatrix} A_{11} & A_{12} \\ A_{12}^T & A_{22} \end{bmatrix} \quad (14)$$

and

$$H_{k+1}^T R^{-1} H_{k+1} = \begin{bmatrix} H_{e,k+1}^T R^{-1} H_{e,k+1} & \mathbf{0}_{2 \times (2N-2)} \\ \mathbf{0}_{(2N-2) \times 2} & \mathbf{0}_{(2N-2) \times (2N-2)} \end{bmatrix}$$

on the right-hand side of (13) yields:

$$P_{k+1|k+1} = \begin{bmatrix} A_{11} + H_{e,k+1}^T R^{-1} H_{e,k+1} & A_{12} \\ A_{12}^T & A_{22} \end{bmatrix}^{-1} \quad (15)$$

Employing the properties of the Schur complement [27] for the inversion of a partitioned matrix, in (15), we obtain

$$\begin{aligned} P_{k+1|k+1,11} &= \left(A_{11} + H_{e,k+1}^T R^{-1} H_{e,k+1} - A_{12} A_{22}^{-1} A_{12}^T \right)^{-1} \\ &= \left((P_{k+1|k,11})^{-1} + H_{e,k+1}^T R^{-1} H_{e,k+1} \right)^{-1} \end{aligned}$$

where in the second equality we used the Schur complement for the inversion of the partitioned matrix $P_{k+1|k}$ (cf. (14)). ■

The importance of this lemma is that both optimization algorithms presented in Section V can be derived based on (11) for the position covariance update – instead of (10) or (13) for the whole state covariance update – regardless of the stochastic

process model employed for describing the target's motion.

In the next section, we formulate the sensors' one-step-ahead *optimal motion strategy* as a constrained optimization problem and show that it can be exactly reformulated as that of minimizing the norm of the sum of a set of vectors of known length with constraints imposed on their directions.

D. Problem Statement and Reformulation

As evident from (7)-(9) and (11), after each update step the target's position covariance matrix will depend on all the next sensors' positions $\mathbf{p}_i(k+1) = [x_i(k+1) \ y_i(k+1)]^T$, $i = 1, \dots, M$. Assuming that at time-step k , sensor- i is at location $\mathbf{p}_i(k) = [x_i(k) \ y_i(k)]^T$ and moves with speed $v_i(k)$, at time-step $k+1$ its position will be:

$$x_i(k+1) = x_i(k) + v_i(k) \delta t \cos \varphi_i(k) \quad (16)$$

$$y_i(k+1) = y_i(k) + v_i(k) \delta t \sin \varphi_i(k) \quad (17)$$

where $\varphi_i(k) \in [0, 2\pi)$ is the heading direction of the sensor. We thus see that given the current sensor positions, $\mathbf{p}_i(k)$, the covariance for the target's position estimate after the update (cf. (11)) is a function of the sensors' speeds, $v_i(k)$, and motion directions $\varphi_i(k)$.

The problem we address in this work is that of determining the sensors' *optimal motion strategy*, i.e., the set $\mathcal{C}(k) = \{(v_i(k), \varphi_i(k)), i = 1, \dots, M\}$, that minimizes the *trace* of the target's position estimate covariance matrix. Based on the following lemma, we first show that minimizing the trace of the (posterior) covariance matrix requires optimization with respect to the *bearing directions* of the sensors towards the estimated position of the target, while the speed of each sensor only affects the constraints imposed on this problem.

Lemma 2: The following two optimization problems are equivalent:

- OPTIMIZATION PROBLEM 1 (Π_1)

$$\begin{aligned} &\text{minimize}_{\varphi_1(k), \dots, \varphi_M(k), v_1(k), \dots, v_M(k)} \quad \text{tr}(P_{k+1|k+1,11}) \\ &\text{s.t.} \quad 0 \leq v_i(k) \leq v_{imax} \quad \forall i = 1, \dots, M \end{aligned}$$

- OPTIMIZATION PROBLEM 2 (Π_2)

$$\begin{aligned} &\text{minimize}_{\theta_1(k+1), \dots, \theta_M(k+1)} \quad \text{tr}(P_{k+1|k+1,11}) \\ &\text{s.t.} \quad |\theta_i(k+1) - \theta'_i(k)| \leq \eta_{imax}(k) \quad \forall i = 1, \dots, M \end{aligned} \quad (18)$$

with²

$$\eta_{imax}(k) = \arcsin \left(\frac{v_{imax} \delta t}{\hat{d}'_i(k)} \right) \quad (19)$$

$$\hat{d}'_i(k) = \sqrt{(\hat{x}_T(k+1|k) - x_i(k))^2 + (\hat{y}_T(k+1|k) - y_i(k))^2} \quad (20)$$

$$\theta'_i(k) = \text{Atan2}(\hat{y}_T(k+1|k) - y_i(k), \hat{x}_T(k+1|k) - x_i(k)) \quad (21)$$

where (cf. Fig. 3) $\hat{d}'_i(k)$ and $\theta'_i(k)$ are the distance and bearing angle from the *current* location of sensor- i , $\mathbf{p}_i(k)$, to the *next* (predicted) position of the target $\hat{\mathbf{p}}_T(k+1|k)$.

²Note that if $\hat{d}'_i(k) < v_{imax} \delta t$, then $\theta_i(k+1) \in [0, 2\pi)$, i.e., no constraint is imposed on the bearing angle to the target. We hereafter consider the most challenging case, when all bearing angles are constrained.

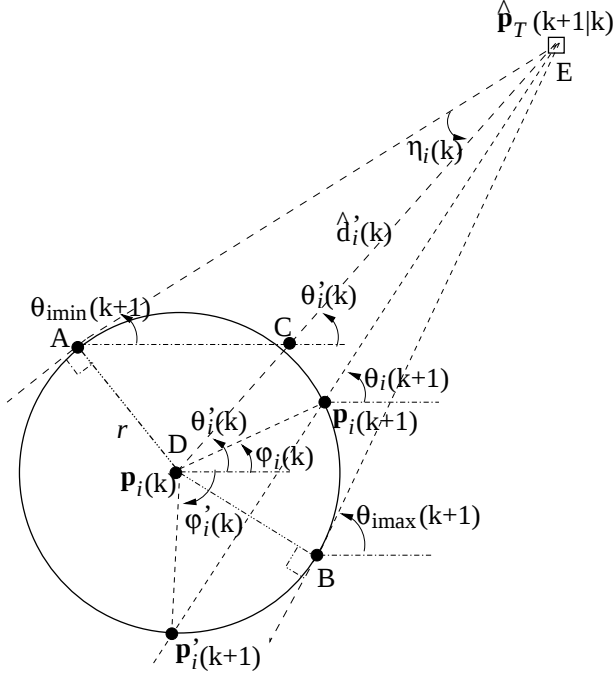


Fig. 3. Geometric interpretation of the bearing angle constraints: Since the speed $v_i(k)$ of each sensor is bounded by v_{imax} , the bearing angle $\theta_i(k+1)$ is constrained in the interval $[\theta_{imin}(k+1), \theta_{imax}(k+1)]$.

Proof: Since the measurement matrix $H_{e,k+1}$ (cf. (7)), and hence the posterior covariance matrix (cf. (11)), has an explicit form in terms of the bearing angles, $\theta_i(k+1)$, towards the estimated target position, minimizing the trace of the covariance matrix can be performed using the $\theta_i(k+1)$, $i = 1, \dots, M$, as the *optimization variables*, instead of the heading direction, $\varphi_i(k)$, or speed, $v_i(k)$, of each sensor. Note, however, that although the variables $\{\varphi_1(k), \dots, \varphi_M(k)\}$ are unconstrained, the bearing angles, $\{\theta_1(k+1), \dots, \theta_M(k+1)\}$, are constrained by the fact that the speed, $v_i(k)$, of each sensor, is bounded by v_{imax} . Our objective here is to determine the constraints on the new optimization variables $\theta_i(k+1)$ and reveal their relation to v_{imax} .

Consider the geometry of this problem shown in Fig. 3. At time-step k , sensor- i is located at $\mathbf{p}_i(k) = [x_i(k) \ y_i(k)]^T$ and predicts, based on the motion model (cf. (3)), that the target will move to $\hat{\mathbf{p}}_T(k+1|k) = [\hat{x}_T(k+1|k) \ \hat{y}_T(k+1|k)]^T$. Assume that sensor- i moves with speed v_i and reaches a point $\mathbf{p}_i(k+1) = [x_i(k+1) \ y_i(k+1)]^T$ located on a circle of radius $r = v_i \delta t$, centered at its previous position $\mathbf{p}_i(k)$ (cf. Fig. 3, for $v_i = v_{imax}$), which does *not* include the target. From point E (i.e., the target's estimated location at time-step $k+1$, cf. Fig. 3), we draw two lines tangent to the circle where sensor- i will move to. The two tangent points A and B correspond to the two extreme values of the bearing angle that define the constraints on $\theta_i(k+1)$, i.e., $\theta_{imin}(k+1) \leq \theta_i(k+1) \leq \theta_{imax}(k+1)$, with

$$\theta_{imin}(k+1) = \theta'_i(k) - \eta_i(k) \quad (22)$$

$$\theta_{imax}(k+1) = \theta'_i(k) + \eta_i(k) \quad (23)$$

$$\eta_i(k) = \arcsin \left(\frac{v_i(k) \delta t}{\hat{d}'_i(k)} \right) \quad (24)$$

where (24) results from the sine relation in the right triangle ADE , while (22) is derived from the relation for the external to the triangle ACE angle $\theta'_i(k)$ (note that (23) can be easily derived in a similar manner based on the geometry of the problem).

Since the inverse-sine function (cf. (24)) is monotonically increasing within the interval of concern ($0 < \eta_i(k) < \pi/2$), the angle $\eta_i(k)$ is maximized when $r = r_{imax}$, which corresponds to $v_i = v_{imax}$ for sensors moving with bounded speed. For $\eta_i(k) = \eta_{imax}(k)$ (cf. (19)), the range of values of the bearing angles $\theta_i(k+1)$ is maximized (i.e., the constraints on the bearing angles are most relaxed), which leads to a smaller or equal minimum value for the objective function (covariance trace) compared to when $\eta_i(k) < \eta_{imax}(k)$. Therefore, the speeds of all sensors are set to their maximum values and optimization is performed with respect to the bearing angles $\theta_i(k+1)$ within the constraints defined by (22) and (23). ■

Once the optimal value for the bearing angle, $\theta_i(k+1)$, of sensor- i has been determined, its optimal heading directions, $\varphi_i(k)$ and $\varphi'_i(k)$ (cf. Fig. 3), can be computed in closed form.

Corollary 1: Given the optimal bearing angle $\theta_i(k+1)$, the optimal heading directions, $\varphi_i(k)$ and $\varphi'_i(k)$, of sensor- i (cf. Fig. 3) are computed from the following relations:

$$\varphi_i(k) = \theta_i(k+1) + \xi_i(k) \quad (25)$$

$$\varphi'_i(k) = \theta_i(k+1) + \pi - \xi_i(k) \quad (26)$$

where

$$\xi_i(k) = \arcsin \left(\frac{(\hat{y}_T(k+1|k) - y_i(k)) \cos \theta_i(k+1)}{v_i(k) \delta t} - \frac{(\hat{x}_T(k+1|k) - x_i(k)) \sin \theta_i(k+1)}{v_i(k) \delta t} \right) \quad (27)$$

Note that between these two equivalent solutions, sensor- i will choose the one that brings it closer to the target so as to increase the probability of re-detection later on.

Proof: The proof is described in Appendix A. ■

At this point, we should note that the preceding analysis is not limited to the case of sensors moving with constant speed during each time step. In fact, Lemma 2 can be directly applied to any higher-order sensor motion model. For example, if a second-order model with bounded acceleration $a_i(k) \leq a_{imax}$ was used to describe the sensors' motion, then maximizing $\eta_i(k)$, or equivalently $r = v_i(k) \delta t + \frac{1}{2} a_i(k) \delta t^2$, would require that the sensors move with maximum acceleration.

From here on, we turn our attention to determining the optimal bearing angles to the estimated target position given the constraints of Lemma 2. Before showing the final result of this section, we first prove the following properties for the objective function of the optimization problem.

Lemma 3: In the optimal target tracking problem using distance-only measurements, minimizing the trace of the target

position estimates' covariance matrix is equivalent to:³

- (i) maximizing the determinant of its inverse
- (ii) maximizing the minimum eigenvalue of its inverse
- (iii) minimizing the difference of its eigenvalues

$$\begin{aligned}
 & \text{minimize}_{\theta_1, \dots, \theta_M} \text{tr}(P_{k+1|k+1,11}) \\
 \Leftrightarrow & \text{maximize}_{\theta_1, \dots, \theta_M} \det((P_{k+1|k+1,11})^{-1}) \\
 \Leftrightarrow & \text{maximize}_{\theta_1, \dots, \theta_M} \mu_{\min}((P_{k+1|k+1,11})^{-1}) \\
 \Leftrightarrow & \text{minimize}_{\theta_1, \dots, \theta_M} (\mu_{\max}(P_{k+1|k+1,11}) - \mu_{\min}(P_{k+1|k+1,11}))
 \end{aligned}$$

where $\bar{\theta}_i = \theta_i - \theta_0$, $i = 1, \dots, M$, θ_0 is a constant defined from the 2×2 unitary (rotational) matrix appearing in the singular value decomposition of $P_{k+1|k,11}$ (cf. (29) and (30)), and $\mu_{\min}(\cdot)$ and $\mu_{\max}(\cdot)$ denote the minimum and the maximum eigenvalues of their matrix arguments, respectively.

Proof: (i) Since $P_{k+1|k+1,11}$ is a 2×2 matrix, it is trivial to prove that

$$\text{tr}(P_{k+1|k+1,11}) = \frac{\text{tr}((P_{k+1|k+1,11})^{-1})}{\det((P_{k+1|k+1,11})^{-1})} \quad (28)$$

Thus for completing the proof of (i), it suffices to compute the inverse of the position covariance matrix $P_{k+1|k+1,11}$ and show that its trace is constant.

Note that since the covariance matrix $P_{k+1|k}$ for the state estimates is positive semi-definite, so is the covariance matrix $P_{k+1|k,11}$ of the target's position estimates. The singular value decomposition of $(P_{k+1|k,11})^{-1}$ yields:

$$(P_{k+1|k,11})^{-1} = U \Sigma^{-1} U^T \quad (29)$$

where $\Sigma^{-1} = \text{diag}(\mu'_1, \mu'_2)$, $\mu'_1 \geq \mu'_2 \geq 0$, and

$$U = \begin{bmatrix} \cos \theta_0 & -\sin \theta_0 \\ \sin \theta_0 & \cos \theta_0 \end{bmatrix} \text{ with } UU^T = U^T U = I_{2 \times 2} \quad (30)$$

Substituting (29) in the right-hand side of (11), we have:

$$\begin{aligned}
 P_{k+1|k+1,11} &= (U \Sigma^{-1} U^T + H_{e,k+1}^T R^{-1} H_{e,k+1})^{-1} \\
 &= U (\Sigma^{-1} + H_{n,k+1}^T R^{-1} H_{n,k+1})^{-1} U^T \\
 &= U \mathcal{I}^{-1} U^T
 \end{aligned}$$

or equivalently

$$(P_{k+1|k+1,11})^{-1} = U \mathcal{I} U^T \quad (31)$$

where

$$H_{n,k+1} = H_{e,k+1} U = \begin{bmatrix} \cos \bar{\theta}_1 & \dots & \cos \bar{\theta}_M \\ \sin \bar{\theta}_1 & \dots & \sin \bar{\theta}_M \end{bmatrix}^T$$

with $\bar{\theta}_i = \theta_i - \theta_0$, $i = 1, \dots, M$, and

$$\mathcal{I} = \begin{bmatrix} \mu'_1 + \sum_{i=1}^M \sigma_i^{-2} \cos^2 \bar{\theta}_i & \sum_{i=1}^M \sigma_i^{-2} \cos \bar{\theta}_i \sin \bar{\theta}_i \\ \sum_{i=1}^M \sigma_i^{-2} \cos \bar{\theta}_i \sin \bar{\theta}_i & \mu'_2 + \sum_{i=1}^M \sigma_i^{-2} \sin^2 \bar{\theta}_i \end{bmatrix} \quad (32)$$

³For clarity, from here on we drop the time indices from the bearing angles $\theta_i(k+1)$ and $\bar{\theta}_i(k+1)$.

Substituting (32) in (31) and noting that similarity transformations do not change the trace of a matrix, yields

$$\text{tr}((P_{k+1|k+1,11})^{-1}) = \text{tr}(\mathcal{I}) = \mu'_1 + \mu'_2 + \sum_{i=1}^M \sigma_i^{-2} = c \quad (33)$$

which is constant.

(ii) Let $\mu_2 := \mu_{\min}((P_{k+1|k+1,11})^{-1}) \leq \mu_1 := \mu_{\max}((P_{k+1|k+1,11})^{-1})$, be the minimum and maximum eigenvalues of the inverse covariance matrix for the position estimates. Based on the relations

$$\text{tr}((P_{k+1|k+1,11})^{-1}) = \mu_1 + \mu_2 = c \quad (34)$$

$$\det((P_{k+1|k+1,11})^{-1}) = \mu_1 \mu_2 \quad (35)$$

we have

$$\begin{aligned}
 \text{maximize } \det((P_{k+1|k+1,11})^{-1}) &\Leftrightarrow \text{maximize } (\mu_1 \mu_2) \\
 &\Leftrightarrow \text{minimize } (-4\mu_1 \mu_2) \Leftrightarrow \text{minimize } (c^2 - 4\mu_1 \mu_2) \\
 &\Leftrightarrow \text{minimize } (\mu_1 - \mu_2)^2 \Leftrightarrow \text{minimize } (\mu_1 - \mu_2) \\
 &\Leftrightarrow \text{minimize } (2\mu_1 - c) \Leftrightarrow \text{minimize } (\mu_1)
 \end{aligned}$$

(iii) Note that $\mu_{\max}(P_{k+1|k+1,11}) = \frac{1}{\mu_2}$ and $\mu_{\min}(P_{k+1|k+1,11}) = \frac{1}{\mu_1}$ and (cf. (34))

$$\begin{aligned}
 \text{minimize } \left(\frac{1}{\mu_2} - \frac{1}{\mu_1} \right) &\Leftrightarrow \text{minimize } \frac{\mu_1 - \mu_2}{\mu_1 \mu_2} \\
 &\Leftrightarrow \text{minimize } \frac{2\mu_1 - c}{-\mu_1^2 + c\mu_1}
 \end{aligned}$$

However, this last quantity is a monotonically increasing function of μ_1 within the interval of concern $[c/2, c]$ (from (34), it is $\mu_2 \leq c/2 \leq \mu_1 \leq c$). Therefore, minimizing it is equivalent to minimizing μ_1 which, based on the result of (ii), is equivalent to maximizing the determinant of the inverse covariance matrix. ■

The key result of this section is described by the following lemma.

Lemma 4: The optimal motions of a group of sensors estimating the position of a moving target can be determined by solving the following constrained optimization problem:

- OPTIMIZATION PROBLEM 3 (Π_3)

$$\text{minimize}_{\bar{\theta}_1, \dots, \bar{\theta}_M} \left\| \lambda_0 + \sum_{i=1}^M \lambda_i \exp(j2\bar{\theta}_i) \right\|_2 \quad (36)$$

$$\text{s.t. } \bar{\theta}_{\min} \leq \bar{\theta}_i \leq \bar{\theta}_{\max}, \quad \forall i = 1, \dots, M \quad (37)$$

with $j = \sqrt{-1}$ and (cf. (29) and (30))

$$\lambda_0 = \mu'_1 - \mu'_2 \geq 0, \quad \lambda_i = \sigma_i^{-2} > 0, \quad i = 1, \dots, M \quad (38)$$

$$\bar{\theta}_{\min} = \theta_{\min} - \theta_0, \quad \bar{\theta}_{\max} = \theta_{\max} - \theta_0$$

or equivalently:

$$\text{minimize}_{\bar{\theta}_1, \dots, \bar{\theta}_M} \left\| \sum_{i=0}^M \mathbf{v}_i \right\|_2 \quad (39)$$

$$\text{s.t. } \bar{\theta}_{\min} \leq \bar{\theta}_i \leq \bar{\theta}_{\max}, \quad \forall i = 1, \dots, M \quad (40)$$

with (for $i = 1, \dots, M$)

$$\mathbf{v}_0 = [\lambda_0 \ 0]^T, \quad \mathbf{v}_i = [\lambda_i \cos 2\bar{\theta}_i \ \lambda_i \sin 2\bar{\theta}_i]^T$$

Proof: We first note, that the constraints of (37) are the same as the ones for the variables θ_i of the second optimization problem in Lemma 2, transformed for the new variables $\bar{\theta}_i = \theta_i - \theta_0$. To prove the equivalence between the objectives functions in (36) and (18), we rely on the equivalence between minimizing the trace of the covariance matrix and maximizing the determinant of the inverse covariance matrix, shown in Lemma 3, and proceed as follows.

Substituting (32) in (31), and employing the trigonometric identities $\cos^2 \bar{\theta}_i = (1 + \cos 2\bar{\theta}_i)/2$, $\sin^2 \bar{\theta}_i = (1 - \cos 2\bar{\theta}_i)/2$, $\cos \bar{\theta}_i \sin \bar{\theta}_i = (\sin 2\bar{\theta}_i)/2$ we have:

$$\det((P_{k+1|k+1,11})^{-1}) = \det(\mathcal{I}) = d_c - \frac{1}{4}d_{\bar{\theta}} \quad (41)$$

where

$$\begin{aligned} d_c &= (\mu'_1 + \frac{1}{2} \sum_{i=1}^M \sigma_i^{-2})(\mu'_2 + \frac{1}{2} \sum_{i=1}^M \sigma_i^{-2}) + \frac{1}{4}(\mu'_1 - \mu'_2)^2 \\ &\text{is constant, and} \\ d_{\bar{\theta}} &= \left((\mu'_1 - \mu'_2) + \sum_{i=1}^M \sigma_i^{-2} \cos 2\bar{\theta}_i \right)^2 + \left(\sum_{i=1}^M \sigma_i^{-2} \sin 2\bar{\theta}_i \right)^2 \\ &= \left\| \lambda_0 + \sum_{i=1}^M \lambda_i \exp(j2\bar{\theta}_i) \right\|_2^2 = \left\| \sum_{i=0}^M \mathbf{v}_i \right\|_2^2 \end{aligned} \quad (42)$$

From (41), we conclude that maximizing the determinant of the inverse covariance matrix is equivalent to minimizing the quantity $d_{\bar{\theta}}$, i.e., the norm of the sum of the vectors $\mathbf{v}_i$, $i = 0, \dots, M$. ■

We thus see that the original problem of minimizing the trace of the covariance matrix of the target's position estimate (cf. Lemma 2) is *exactly reformulated* to that of *minimizing the norm of the sum of $M+1$ vectors in 2D* (cf. Lemma 4). Note that although the vector $\mathbf{v}_0 = [\lambda_0 \ 0]^T$ remains constant (affixed to the positive x semi-axis), each of the vectors $\mathbf{v}_i$, $i = 1, \dots, M$, has fixed length λ_i but its direction can vary under the constraints described by (37). This geometric interpretation is depicted in Fig. 4.

IV. COMPUTATIONAL COMPLEXITY

We now analyze the complexity of the optimization problem described in Lemma 4. The main result of this section is that the problem of determining the optimal constrained motion for a team of $M > 1$ mobile sensors tracking a moving target using distance-only measurements is *NP-Hard* in general (cf. Section IV-B).

Before considering the general case of multiple sensors, however, we first focus on determining the optimal solution when only a single sensor tracks the target. The main reason for this is that the closed-form solution derived for this case is extended and generalized to form the basis of the *modified Gauss-Seidel relaxation* algorithm presented in Section V.

A. Single-sensor Target Tracking: Closed-form Solution

For $M = 1$, the optimization problem described by (36) is simplified to:

$$\underset{\bar{\theta}_1}{\text{minimize}} \quad \|\lambda_0 + \lambda_1 \exp(j2\bar{\theta}_1)\|_2 \Leftrightarrow \underset{\bar{\theta}_1}{\text{minimize}} \quad \|\mathbf{v}_0 + \mathbf{v}_1\|_2$$

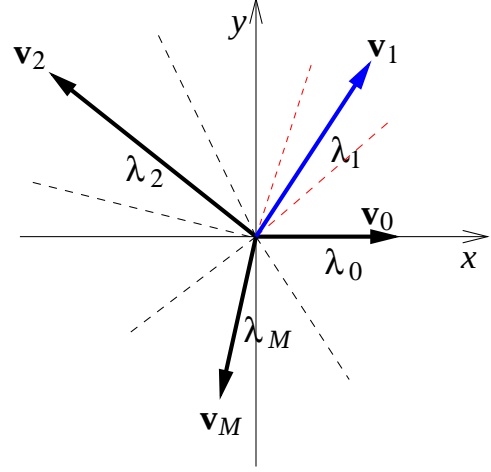


Fig. 4. Geometric interpretation of the optimal motion strategy problem: The $M+1$ vectors shown have fixed lengths λ_i , $i = 0, \dots, M$. The vector $\mathbf{v}_0$ is affixed to the positive x semi-axis, while the direction of each of the vectors $\mathbf{v}_i$, $i = 1, \dots, M$, can change, within the interval denoted by the enclosing dashed lines, based on the motion of the corresponding sensor. The objective is to find the directions of the vectors $\mathbf{v}_i$, $i = 1, \dots, M$ – directly related to the optimal heading directions of the sensors – that minimize the Euclidean norm of $\sum_{i=0}^M \mathbf{v}_i$.

$$\text{s.t. } \bar{\theta}_{1min} \leq \bar{\theta}_1 \leq \bar{\theta}_{1max}$$

with $\mathbf{v}_0 = \lambda_0[1 \ 0]^T$, and $\mathbf{v}_1 = \lambda_1[\cos 2\bar{\theta}_1 \ \sin 2\bar{\theta}_1]^T$.

This norm-minimization problem can be solved trivially by maximizing the angle between the two vectors (i.e., setting $2\bar{\theta}_1$ as close to π as possible, while satisfying the constraints on it). The closed-form solution for the optimal value is:

$$\bar{\theta}_1^* = \begin{cases} \frac{n\pi}{2} & \text{if } \frac{n\pi}{2} \in [\bar{\theta}_{1min}, \bar{\theta}_{1max}], \ n \text{ is odd} \\ \bar{\theta}_{1min} & \text{if } \frac{n\pi}{2} \notin [\bar{\theta}_{1min}, \bar{\theta}_{1max}] \text{ and } |\frac{n\pi}{2} - \bar{\theta}_{1min}| \leq |\frac{n\pi}{2} - \bar{\theta}_{1max}| \\ \bar{\theta}_{1max} & \text{otherwise} \end{cases} \quad (43)$$

Once $\bar{\theta}_1^*$ is determined, the optimal θ_1 is computed as $\theta_1^* = \theta_0 + \bar{\theta}_1^*$.

Intuitively, the result of (43) can be explained as follows: Recall that θ_0 is the direction of the eigenvector $\mathbf{u}_1 = [\cos \theta_0 \ \sin \theta_0]^T$ corresponding to the *maximum* eigenvalue, μ'_1 , of the *prior* information matrix $(P_{k+1|k,11})^{-1}$, while $(\theta_0 + \frac{\pi}{2})$ is the direction of eigenvector $\mathbf{u}_2 = [\cos(\theta_0 + \frac{\pi}{2}) \ \sin(\theta_0 + \frac{\pi}{2})]^T$ corresponding to the *minimum* eigenvalue, μ'_2 , of $(P_{k+1|k,11})^{-1}$ (cf. (29) and (30)). When only one sensor is available, it should always move so as the new measurement contributes information along (or as close as possible to) the direction where the least information is available. This is best achieved when $\bar{\theta}_1^* = \frac{\pi}{2}$ and hence $\theta_1^* = \theta_0 + \frac{\pi}{2}$.

Interestingly, the minimization of the trace of the covariance matrix for the case of a single sensor can also be shown to be exactly equivalent to the maximization of the resulting Rayleigh quotient. The proof is described in Appendix B.

At this point, we should note that the solution described by (43) for one sensor can be adapted and generalized to determine the motion of multiple sensors. In such case, the objective function (cf. (36)) will be sequentially minimized over each variable $\bar{\theta}_i$ separately, while considering the remaining

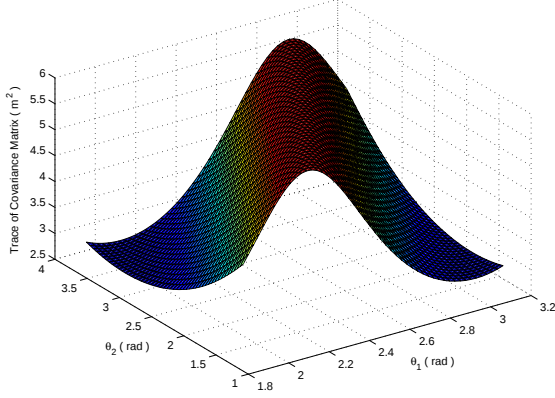


Fig. 5. Illustration of non-convexity: A simple scenario in which 2 sensors track a target. We plot the trace of the posterior covariance matrix, $P_{k+1|k+1,11}$, of the target's position estimates for the following values of the sensors' bearing directions: $(\theta_1(k), \theta_2(k)) \in [\theta_{1min}, \theta_{1max}] \times [\theta_{2min}, \theta_{2max}] = [\frac{3}{5}\pi, \pi] \times [\frac{2}{5}\pi, \frac{7}{5}\pi]$. Note that the bearing directions' intervals are determined based on the sensors' motion constraints. The sensors' speed is set to $v_1 = v_2 = 6$ m/sec, the time step is $\delta t = 1$ sec, their current positions are given by $\mathbf{p}_1(k) = [8, -6]^T$, and $\mathbf{p}_2(k) = [5, -4]^T$, the covariance of the range measurements is $R = 2I_{2 \times 2}$, the estimated target position is $\hat{\mathbf{p}}_T(k+1|k) = [0, 0]^T$ and the prior position estimates' covariance matrix is $P_{k+1|k,11} = 5I_{2 \times 2}$.

bearings (i.e., $\bar{\theta}_j, j \neq i$) as constant during that step. In fact, our *modified Gauss-Seidel relaxation* algorithm follows this idea and its solution at each iteration has similar closed form as (43).

B. Multi-sensor Target Tracking: NP-Hardness

The objective function in (18), and equivalently in (36), is non-convex in the optimization variables $\bar{\theta}_i, i = 1, \dots, M$. This is clearly demonstrated in Fig. 5 for $M = 2$, where the trace of the posterior covariance matrix is plotted against the possible values of the two sensors' bearing directions. More importantly, in this section we show that the problem of determining the optimal constrained motion of multiple sensors tracking a moving target with range measurements, is *NP-Hard* in general (cf. Theorem 1).

We proceed by first considering the following well-known *NP-Complete* problem [28, Ch. 3],

- PARTITION PROBLEM

Given M positive integers $\lambda_1, \dots, \lambda_M$, determine whether there exist $\zeta_i \in \{-1, +1\}, i = 1, \dots, M$, such that $\sum_{i=1}^M \lambda_i \zeta_i = 0$.

and

- OPTIMIZATION PROBLEM 3' (Π'_3)

$$\begin{aligned} & \underset{\bar{\theta}_1, \dots, \bar{\theta}_M}{\text{minimize}} \quad \left(\left(\sum_{i=1}^M \lambda_i \cos 2\bar{\theta}_i \right)^2 + \left(\sum_{i=1}^M \lambda_i \sin 2\bar{\theta}_i \right)^2 \right)^{\frac{1}{2}} \\ & \text{s.t.} \quad 0 \leq \bar{\theta}_i \leq \pi/2, \forall i = 1, \dots, M \\ & \quad \lambda_i \in \mathbb{Z}^+, \forall i = 1, \dots, M \end{aligned} \quad (44)$$

which is an instance of optimization problem Π_3 described by (36)-(37), for $\lambda_0 = 0, \theta_{imin} = 0, \theta_{imax} = \pi/2$ and $\lambda_i \in \mathbb{Z}^+$.

Proving by restriction [28, Ch. 3] that Π_3 is *NP-Hard*, in general, requires to show that solving⁴ Π'_3 , which is a special case of Π_3 , is *equivalent* to solving the partition problem. Since the partition problem is *NP-Complete*, it will follow that the general problem Π_3 is *at least* as hard as that, i.e., Π_3 is *NP-Hard*. We first prove that the answer to the partition problem is positive ("yes"), if and only if Π'_3 achieves optimal value of zero.

Lemma 5: For positive integers $\lambda_1, \dots, \lambda_M$, there exist $\zeta_i \in \{-1, +1\}, i = 1, \dots, M$, such that $\sum_{i=1}^M \lambda_i \zeta_i = 0$, if and only if, the optimal value of Π'_3 is 0.

Proof:

(Necessary): Assume $\exists \zeta_i \in \{-1, +1\}, i = 1, \dots, M$, such that

$$\sum_{i=1}^M \lambda_i \zeta_i = 0 \quad (46)$$

Based on these, consider the following choice of $\bar{\theta}_i$ for Π'_3

$$\bar{\theta}_i^* = \begin{cases} 0 & \text{if } \zeta_i = 1 \\ \pi/2 & \text{if } \zeta_i = -1 \end{cases} \quad (47)$$

Note that $\bar{\theta}_i^*, i = 1, \dots, M$, satisfies the constraints of Π'_3 (cf. (45)). Additionally, it is easy to verify that $\cos 2\bar{\theta}_i^* = \zeta_i$ and $\sin 2\bar{\theta}_i^* = 0, \forall i = 1, \dots, M$. Substituting in the objective function (squared) of Π'_3 (cf. (44)) yields

$$\left(\sum_{i=1}^M \lambda_i \cos 2\bar{\theta}_i^* \right)^2 + \left(\sum_{i=1}^M \lambda_i \sin 2\bar{\theta}_i^* \right)^2 = \left(\sum_{i=1}^M \lambda_i \zeta_i \right)^2 = 0$$

where the last equality follows from (46).

Since the objective function of Π'_3 is always nonnegative and the choice of $\bar{\theta}_i^*$ (cf. (47)) based on ζ_i achieves zero, the set $\{\bar{\theta}_i^*, i = 1, \dots, M\}$ is the optimal solution of Π'_3 .

(Sufficient): Suppose $\exists \bar{\theta}_i^*$, with $0 \leq \bar{\theta}_i^* \leq \pi/2, \forall i = 1, \dots, M$, and

$$\left(\left(\sum_{i=1}^M \lambda_i \cos 2\bar{\theta}_i^* \right)^2 + \left(\sum_{i=1}^M \lambda_i \sin 2\bar{\theta}_i^* \right)^2 \right)^{\frac{1}{2}} = 0 \quad (48)$$

This last equality for the objective function of Π'_3 requires

$$\sum_{i=1}^M \lambda_i \sin 2\bar{\theta}_i^* = 0 \quad \text{and} \quad (49)$$

$$\sum_{i=1}^M \lambda_i \cos 2\bar{\theta}_i^* = 0 \quad (50)$$

Note that the constraints on $\bar{\theta}_i$ (cf. (45)) imply that $\sin 2\bar{\theta}_i^* \geq 0, \forall i = 1, \dots, M$. Additionally, since $\lambda_i > 0$, it follows from (49) that $\sin 2\bar{\theta}_i^* = 0 \Rightarrow \cos 2\bar{\theta}_i^* = \pm 1, \forall i = 1, \dots, M$. Thus, there exist $\zeta_i = \cos 2\bar{\theta}_i^* \in \{-1, +1\}$, such that $\sum_{i=1}^M \lambda_i \zeta_i = \sum_{i=1}^M \lambda_i \cos 2\bar{\theta}_i^* = 0$ (cf. (50)). ■

Lemma 5, establishes a one-to-one correspondence between

⁴Here "solve" means to find the *global* optimal solution and the optimal value.

every instance⁵ of Π'_3 and that of the partition problem. In particular, if we are able to solve the optimization problem Π'_3 , then by examining its optimal value, we can answer the partition problem, i.e., zero (vs. positive) optimal value for the objective function of Π'_3 corresponds to positive (vs. negative) answer to the partition problem. Based on the result of Lemma 5, we hereafter state and prove the main result of this section.

Theorem 1: The problem of determining the optimal constrained motion of a team of mobile sensors tracking a moving target using distance-only measurements is *NP-Hard* in general.

Proof: Assume that the general problem Π_3 is *not NP-Hard*. Then there exists a polynomial-time algorithm that can solve all instances of Π_3 , and hence Π'_3 . From Lemma 5, however, the answer to the partition problem can be determined based on the optimal value of Π'_3 . This implies that the partition problem can be solved in polynomial time, which is a contradiction. ■

V. PROBLEM SOLUTION

As shown in the previous section, the problem of *optimal trajectory generation* for multiple sensors with mobility constraints that track a moving target using range-only measurements is *NP-Hard* in general. Hence, finding the *global* optimal solution for the original optimization problem, or for its equivalent formulations (cf. $\Pi_1 \Leftrightarrow \Pi_2 \Leftrightarrow \Pi_3$), becomes extremely difficult. Ideally, the optimal solution can be determined if one discretizes the space of possible heading directions of all sensors and performs an exhaustive search. This approach, however, has computational complexity *exponential* in the number of sensors which is of limited practical use given realistic processing constraints.

In order to design algorithms that can operate in real time, appropriate relaxations of the original optimization problem become necessary. In the next two sections, we present two methods for solving the problem under consideration, namely *modified Gauss-Seidel relaxation* and *LP Relaxation*. Both algorithms have computational complexity *linear* in the number of sensors, which ensures real-time implementations even for a large number of sensors. Furthermore, as shown in Section VI, they both achieve tracking accuracy indistinguishable of that of exhaustive search.

A. Modified Gauss-Seidel Relaxation

Motivated by the simplicity of the closed-form solution for the case of one sensor (cf. Section IV-A), a straightforward approach to finding a minimum of the optimization problem Π_3 would be to iteratively minimize its objective function (cf. (36)) for each optimization variable separately, i.e., [29, Ch. 3]

- NONLINEAR GAUSS-SEIDEL ALGORITHM

$$\min_{\bar{\theta}_i^{(\ell+1)}} \left\| \lambda_0 + \sum_{\kappa=1}^{i-1} \left(\lambda_\kappa \exp \left(j2\bar{\theta}_\kappa^{(\ell+1)} \right) \right) \right\|$$

⁵Note that the parameters for both the partition problem and the optimization problem Π_3 are $\lambda_1, \dots, \lambda_M$. An instance of these two problems is obtained by specifying particular values for $\lambda_1, \dots, \lambda_M$.

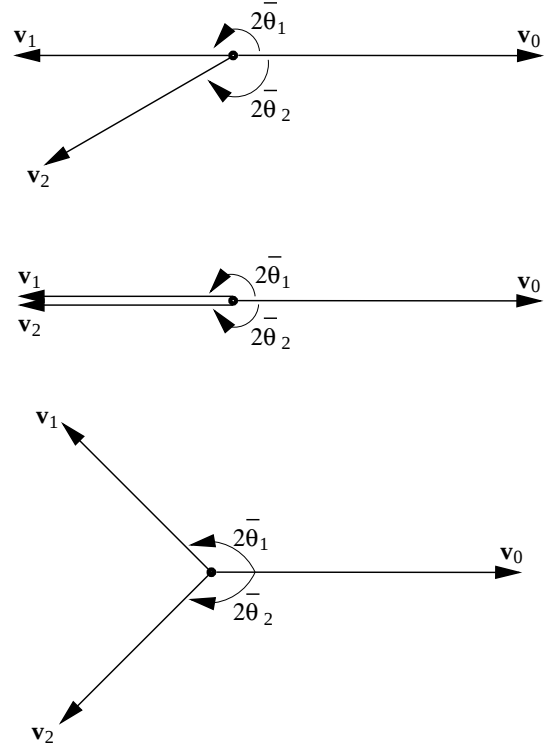


Fig. 6. Norm minimization example for the sum of the vectors with norms $\|\mathbf{v}_0\|_2 = \sqrt{2}$, $\|\mathbf{v}_1\|_2 = \|\mathbf{v}_2\|_2 = 1$. The direction of $\mathbf{v}_0$ is fixed while the directions of $\mathbf{v}_1$ and $\mathbf{v}_2$ are the optimization variables. (Top) Initial vector directions: $2\bar{\theta}_1 = \pi$ and $2\bar{\theta}_2 = -5\pi/6$. (Middle) Final vector directions computed by the nonlinear Gauss-Seidel algorithm: $2\bar{\theta}_1 = -2\bar{\theta}_2 = 3.1416 = \pi$. The norm of the sum in this case is $(2 - \sqrt{2})$ which corresponds to a local minimum. (Bottom) Final vector directions computed by the modified Gauss-Seidel relaxation algorithm: $2\bar{\theta}_1 = -2\bar{\theta}_2 = 2.3562 = 3\pi/4$. The norm of the sum in this case is 0 which corresponds to the global minimum.

$$+ \sum_{\kappa=i+1}^M \left(\lambda_\kappa \exp \left(j2\bar{\theta}_\kappa^{(\ell)} \right) \right) + \lambda_i \exp \left(j2\bar{\theta}_i^{(\ell+1)} \right) \Big\|_2$$

s.t. $\bar{\theta}_{imin} \leq \bar{\theta}_i^{(\ell+1)} \leq \bar{\theta}_{imax}$

where $\bar{\theta}_i^{(\ell+1)}$ is the new optimal value of $\bar{\theta}_i$, $\bar{\theta}_\kappa^{(\ell+1)}$, $\kappa = 1, \dots, i-1$, and $\bar{\theta}_\kappa^{(\ell)}$, $\kappa = i+1, \dots, M$, are the remaining vector directions, considered fixed during this step, computed sequentially during the previous iterations.

However, it is easy to demonstrate that this sequential gradient-based approach is prone to being trapped in *local minima*. For example, consider the simple case of two sensors, with no constraints imposed on $\bar{\theta}_i$, $i = 1, 2$. For $\lambda_0 = \sqrt{2}$, $\lambda_1 = \lambda_2 = 1$, and initial bearing directions $2\bar{\theta}_1 = \pi$, $2\bar{\theta}_2 = -5\pi/6$ (cf. Fig. 6), the optimal values after the first 3 iterations are:

$$\begin{aligned} \text{Initial Condition : } 2\bar{\theta}_2 &= -2.6180, 2\bar{\theta}_1 = 3.1416 \\ \text{1st Iteration : } 2\bar{\theta}_2 &= 3.1416, 2\bar{\theta}_1 = -3.1416 \\ \text{2nd Iteration : } 2\bar{\theta}_2 &= -3.1416, 2\bar{\theta}_1 = 3.1416 \\ \text{3rd Iteration : } 2\bar{\theta}_2 &= -3.1416, 2\bar{\theta}_1 = 3.1416 \end{aligned}$$

As evident this algorithm converges to a local minimum $2\bar{\theta}_1 = -2\bar{\theta}_2 = 3.1416 = \pi$. The objective function value in this case is $(2 - \sqrt{2})$, while the true global minimum is 0, obtained for $2\bar{\theta}_1 = -2\bar{\theta}_2 = 3\pi/4$.

To overcome this limitation, we propose the following

modification:

- MODIFIED GAUSS-SEIDEL RELAXATION

$$\begin{aligned} \min_{\bar{\theta}_i^{(\ell+1)}} & \left\| \lambda_0 + \sum_{\kappa=1}^{i-1} \left(\lambda_{\kappa} \exp \left(j2\bar{\theta}_{\kappa}^{(\ell+1)} \right) \right) \right. \\ & \left. + \sum_{\kappa=i+1}^M \left(\lambda_{\kappa} \exp \left(j2\bar{\theta}_{\kappa}^{(\ell)} \right) \right) + \lambda_i \exp \left(j2\bar{\theta}_i^{(\ell+1)} \right) + \mathbf{v}_{M+1} \right\|_2 \\ \text{s.t. } & \bar{\theta}_{imin} \leq \bar{\theta}_i^{(\ell+1)} \leq \bar{\theta}_{imax} \end{aligned} \quad (51)$$

$$\begin{aligned} \text{with } \mathbf{v}_{M+1} := & -\alpha \left(\lambda_0 + \sum_{\kappa=1}^{i-1} \left(\lambda_{\kappa} \exp \left(j2\bar{\theta}_{\kappa}^{(\ell+1)} \right) \right) \right. \\ & \left. + \sum_{\kappa=i}^M \left(\lambda_{\kappa} \exp \left(j2\bar{\theta}_{\kappa}^{(\ell)} \right) \right) \right) \end{aligned} \quad (52)$$

where we have introduced the perturbation vector $\mathbf{v}_{M+1}$, which is proportional to the sum of the vectors computed in the previous iteration. The parameter $\alpha \in [0, 1]$ is termed the *relaxation factor*. When $\alpha = 0$, this method becomes identical with the nonlinear Gauss-Seidel algorithm, while for $\alpha = 1$, it results in $\bar{\theta}_i^{(\ell+1)} = \bar{\theta}_i^{(\ell)}$, $i = 1, \dots, M$, and therefore the solution does not change between iterations. We thus see that the perturbation vector $\mathbf{v}_{M+1}$ reduces the convergence rate of the modified Gauss-Seidel relaxation algorithm by smoothing the cost function. This makes the algorithm less sensitive to local minima at the expense of increasing the number of iterations required to converge.

This is demonstrated for the previously mentioned two-sensor example (cf. Fig. 6). In this case, the optimal values computed by the modified Gauss-Seidel relaxation algorithm are:

$$\begin{aligned} \text{Initial Condition : } & 2\bar{\theta}_2 = -2.6180, 2\bar{\theta}_1 = 3.1416 \\ \text{1st Iteration : } & 2\bar{\theta}_2 = -2.3625, 2\bar{\theta}_1 = 2.4927 \\ & \dots \quad \dots \quad \dots \\ \text{4th Iteration : } & 2\bar{\theta}_2 = -2.3562, 2\bar{\theta}_1 = 2.3563 \\ \text{5th Iteration : } & 2\bar{\theta}_2 = -2.3562, 2\bar{\theta}_1 = 2.3562 \end{aligned}$$

Noting that $2.3562 = 3\pi/4$, we see that the modified Gauss-Seidel relaxation method returns the global minimum.

The optimization process in the modified Gauss-Seidel relaxation algorithm is carried out only for one variable (i.e., $\bar{\theta}_i$) at every step using a similar closed-form solution as the one used in the single-sensor case (cf. Section IV-A). Thus, the modified Gauss-Seidel relaxation process has computational complexity, per iteration step, only *linear* in the number of sensors. Furthermore, it is easily implemented, has low memory requirements and, as demonstrated in Section VI, it achieves the same level of positioning accuracy as the exhaustive search approach. For clarity, we present the basic steps of the modified Gauss-Seidel relaxation process in Algorithm 1.

B. Linear Programming (LP) Relaxation

In this section, an alternative relaxation is introduced that leads to the formulation of a Linear Programming (LP) algorithm for solving the constrained optimal motion generation problem.

We start by defining the following problem:

Algorithm 1 Modified Gauss-Seidel Relaxation Algorithm

Require: $\bar{\theta}_i^0 = \bar{\theta}_i(k)$, $i = 1 : M$, $\alpha \in [0, 1]$

Ensure: $\bar{\theta}_i(k+1) = \bar{\theta}_i^{(\ell+1)}$, $i = 1 : M$ {Minimize (36)}

- 1: **repeat**
 - 2: Calculate $\mathbf{v}_{M+1}$ from (52)
 - 3: **for** $i = 1 : M$ **do**
 - 4: Determine $\bar{\theta}_i^{(\ell+1)}$ from (51)
 - 5: $\bar{\theta}_i^{(\ell)} \leftarrow \bar{\theta}_i^{(\ell+1)}$ {Update $\bar{\theta}_i$ }
 - 6: Update $\mathbf{v}_{M+1}$ from (52)
 - 7: **end for**
 - 8: **until** max. number of iterations is reached or change in the objective function is less than 1%
-

- OPTIMIZATION PROBLEM 4 (Π_4)

$$\begin{aligned} \text{maximize } & \mu_{min}(\mathcal{I}) \\ & \bar{\theta}_1, \dots, \bar{\theta}_M \end{aligned} \quad (53)$$

$$\text{s.t. } \bar{\theta}_{imin} \leq \bar{\theta}_i \leq \bar{\theta}_{imax}, \quad \forall i = 1, \dots, M$$

with $\mathcal{I}$ defined in (32), which (cf. Lemma 3) is exactly equivalent to the optimization problems $\Pi_1 - \Pi_3$, and proceed to show the remaining of the following relations:⁶

$$\Pi_1 \Leftrightarrow \Pi_2 \Leftrightarrow \Pi_3 \Leftrightarrow \Pi_4 \Leftrightarrow \Pi_5 \leftarrow \Pi_6 \leftarrow \Pi_7 \Leftrightarrow \Pi_8 \Leftrightarrow \Pi_9$$

where $\Pi_i \leftarrow \Pi_j$ denotes that the optimization problem Π_j is a relaxation of Π_i , i.e., the feasible set of Π_i is a subset of that of Π_j . The NP-Hard problem Π_5 , the Semi-Definite Programming (SDP) problems $\Pi_6 - \Pi_8$, and the LP problem Π_9 whose solution is the basis of the LP relaxation algorithm, are defined hereafter.

Lemma 6: The optimization problems Π_4 and Π_5 are equivalent, where

- OPTIMIZATION PROBLEM 5 (Π_5)

$$\text{maximize } \beta \quad (54)$$

$$\text{s.t. } \Sigma^{-1} + \sum_{i=1}^M \lambda_i X_i - \beta I_{2 \times 2} \succeq 0 \quad (55)$$

$$X_i = \begin{bmatrix} x_i & z_i \\ z_i & y_i \end{bmatrix} \succeq 0, \quad \forall i = 1, \dots, M \quad (56)$$

$$\text{rank}(X_i) = 1, \quad \forall i = 1, \dots, M \quad (57)$$

$$\text{tr}(X_i) = 1, \quad \forall i = 1, \dots, M \quad (58)$$

$$X_{i,11}^{(l)} \leq x_i \leq X_{i,11}^{(r)}, \quad \forall i = 1, \dots, M \quad (59)$$

$$\frac{\cos 2\eta_{imax}}{2} \leq \begin{bmatrix} \cos 2\check{\theta}_i \\ \sin 2\check{\theta}_i \end{bmatrix}^T \begin{bmatrix} x_i - 1/2 \\ z_i \end{bmatrix} \leq \frac{1}{2}, \quad \forall i = 1, \dots, M \quad (60)$$

with η_{imax} and Σ^{-1} defined in (19) and (29) respectively,

$$\check{\theta}_i := (\bar{\theta}_{imin} + \bar{\theta}_{imax})/2 \quad (61)$$

$$X_{i,11}^{(l)} := \min_{\bar{\theta}_i \in [\bar{\theta}_{imin}, \bar{\theta}_{imax}]} \cos^2 \bar{\theta}_i \quad (62)$$

$$X_{i,11}^{(r)} := \max_{\bar{\theta}_i \in [\bar{\theta}_{imin}, \bar{\theta}_{imax}]} \cos^2 \bar{\theta}_i \quad (63)$$

⁶Note that the equivalence relations between the previously defined optimization problems $\Pi_1 - \Pi_4$ has already been established based on the results of Lemmas 2, 3, and 4.

Proof: The proof proceeds in four steps.

(i) *Modification of the objective function and introduction of constraint (55):* Since $\mu_{\min}(\mathcal{I}) \geq \beta \Leftrightarrow \mathcal{I} \succeq \beta I_{2 \times 2}$, where “ $\succeq$ ” denotes that $\mathcal{I} - \beta I_{2 \times 2}$ is positive semi-definite, it follows that

$$\begin{aligned} \text{maximize } \mu_{\min}(\mathcal{I}) &\Leftrightarrow \text{maximize } \beta \\ \text{s.t. } \mathcal{I} - \beta I_{2 \times 2} &\succeq 0 \end{aligned} \quad (64)$$

Defining

$$X_i := \begin{bmatrix} \cos^2 \bar{\theta}_i & \cos \bar{\theta}_i \sin \bar{\theta}_i \\ \cos \bar{\theta}_i \sin \bar{\theta}_i & \sin^2 \bar{\theta}_i \end{bmatrix}, \quad i = 1, \dots, M \quad (65)$$

and substituting in (32), yields

$$\mathcal{I} = \Sigma^{-1} + \sum_{i=1}^M \lambda_i X_i \quad (66)$$

with $\lambda_i := \sigma_i^{-2}$. Finally, substituting (66) in (64) results in the constraint (55).

(ii) *Constraints (56)-(58):* From (65), it is evident that X_i has the following properties:

$$X_i \succeq 0, \quad \text{rank}(X_i) = 1, \quad \text{and} \quad \text{tr}(X_i) = 1 \quad (67)$$

Conversely, it is easy to show that any 2×2 matrix X_i satisfying the above constraints can be written in the form of (65). Hence, we conclude that requiring a matrix X_i to be of the form of (65) is equivalent to X_i satisfying the constraints in (56)-(58).

(iii) *Constraint (59):* This is a direct result of the constraint $\bar{\theta}_{\min} \leq \bar{\theta}_i \leq \bar{\theta}_{\max}$ and the definition $x_i := \cos^2 \bar{\theta}_i$.

(iv) *Constraint (60):* Since $z_i := \cos \bar{\theta}_i \sin \bar{\theta}_i = (\sin 2\bar{\theta}_i)/2$ and $x_i := \cos^2 \bar{\theta}_i = (1 + \cos 2\bar{\theta}_i)/2$, the constraint on z_i could be determined based on the constraint (59) on x_i and the trigonometric relation $(\sin 2\bar{\theta}_i)^2 + (\cos 2\bar{\theta}_i)^2 = (2z_i)^2 + (2x_i - 1)^2 = 1$ between x_i and z_i . However, this would result in two feasible regions for z_i and complicate the process of recovering $\bar{\theta}_i$. Instead, we hereafter determine a linear inequality constraint on z_i based on x_i .

Substituting (56) and (65) in the following relation, yields

$$2 \begin{bmatrix} \cos 2\bar{\theta}_i \\ \sin 2\bar{\theta}_i \end{bmatrix}^T \begin{bmatrix} x_i - 1/2 \\ z_i \end{bmatrix} = \cos 2(\bar{\theta}_i - \check{\theta}_i) \quad (68)$$

with $\check{\theta}_i$ defined in (61).

Our objective now is to determine the range of feasible values of $\cos 2|\bar{\theta}_i - \check{\theta}_i|$. Subtracting (22) from (23), we have

$$\begin{aligned} \eta_{\max} &= \frac{\bar{\theta}_{\max} - \bar{\theta}_{\min}}{2} = \check{\theta}_i - \bar{\theta}_{\min} = \bar{\theta}_{\max} - \check{\theta}_i \\ \Rightarrow \bar{\theta}_{\min} &= \check{\theta}_i - \eta_{\max}, \quad \bar{\theta}_{\max} = \check{\theta}_i + \eta_{\max} \end{aligned} \quad (69)$$

Substituting these last two relations on both sides of the inequality $\bar{\theta}_{\min} \leq \bar{\theta}_i \leq \bar{\theta}_{\max}$ and rearranging terms, yields

$$0 \leq |\bar{\theta}_i - \check{\theta}_i| \leq \eta_{\max} \leq \pi/2 \quad (70)$$

where the right-most inequality is due to the geometry of the problem (cf. Fig. 3). Since the cosine function is monotonically decreasing within the interval $[0, \pi]$, from (70), we have

$$0 \leq 2|\bar{\theta}_i - \check{\theta}_i| \leq 2\eta_{\max} \leq \pi$$

$$\cos 2\eta_{\max} \leq \cos 2|\bar{\theta}_i - \check{\theta}_i| \leq 1 \quad (71)$$

Noting that $\cos 2|\bar{\theta}_i - \check{\theta}_i| = \cos 2(\bar{\theta}_i - \check{\theta}_i)$ and substituting the left hand-side of (68) in (71), results in the affine constraint (60). ■

Note that based on the equivalence relation of Lemma 6, Π_5 has the same computational complexity as $\Pi_1 - \Pi_4$ and thus it cannot be solved in polynomial time. In order to devise an efficient algorithm that will support a real-time implementation, we need to modify Π_5 so that it becomes convex. Dropping the rank constraints in (57), yields the following *relaxed* version of Π_5 :

- SDP OPTIMIZATION PROBLEM 6 (Π_6)

$$\text{maximize } \beta \quad (72)$$

$$\begin{aligned} \text{s.t. } \Sigma^{-1} + \sum_{i=1}^M \lambda_i X_i - \beta I_{2 \times 2} &\succeq 0 \\ X_i &\succeq 0, \quad \forall i = 1, \dots, M \\ \text{tr}(X_i) &= 1, \quad \forall i = 1, \dots, M \\ X_{i,11}^{(l)} &\leq x_i \leq X_{i,11}^{(r)}, \quad \forall i = 1, \dots, M \\ \frac{\cos 2\eta_{\max}}{2} &\leq \begin{bmatrix} \cos 2\check{\theta}_i \\ \sin 2\check{\theta}_i \end{bmatrix}^T \begin{bmatrix} x_i - 1/2 \\ z_i \end{bmatrix} \leq \frac{1}{2}, \\ &\quad \forall i = 1, \dots, M \end{aligned}$$

It is clear that Π_6 is a relaxation of Π_5 because the feasible set of Π_5 is a subset of that of Π_6 . Moreover, with respect to the variables $X_i, i = 1, \dots, M$, and β , Π_6 is an SDP problem which can be solved using a *polynomial-time* algorithm [30, Ch. 4]. However, solving Π_6 requires computations at least in the order of $\mathcal{O}(M^3)$ [30, Ch. 11], which makes real-time implementations prohibitive when M is large (note that this solution approach is not considered in the results shown in Section VI). In order to further reduce the computational complexity, we make a second modification to Π_5 by dropping the constraints in (57) and (60) simultaneously, to obtain the following *relaxed* version of Π_5 :

- SDP OPTIMIZATION PROBLEM 7 (Π_7)

$$\text{maximize } \beta \quad (73)$$

$$\begin{aligned} \text{s.t. } \Sigma^{-1} + \sum_{i=1}^M \lambda_i X_i - \beta I_{2 \times 2} &\succeq 0 \\ X_i &\succeq 0, \quad \forall i = 1, \dots, M \\ \text{tr}(X_i) &= 1, \quad \forall i = 1, \dots, M \\ X_{i,11}^{(l)} &\leq x_i \leq X_{i,11}^{(r)}, \quad \forall i = 1, \dots, M \end{aligned} \quad (74)$$

Note again, that the feasible sets of Π_5 and Π_6 are subsets of that of Π_7 , hence Π_7 is a relaxation of both Π_5 and Π_6 .

Although Π_7 is also an SDP problem, we will show that it is exactly equivalent to an LP problem whose solution has computational complexity linear in the number of sensors. We proceed by first proving the following lemma:

Lemma 7: The SDP optimization problems Π_7 and Π_8 are equivalent in the optimal value, where

- SDP OPTIMIZATION PROBLEM 8 (Π_8)

$$\text{maximize } \beta \quad (75)$$

$$\text{s.t. } \Sigma^{-1} + \sum_{i=1}^M \lambda_i X_i - \beta I_{2 \times 2} \succeq 0 \quad (76)$$

$$X_i \succeq 0, \quad \forall i = 1, \dots, M \quad (77)$$

$$\text{tr}(X_i) = 1, \quad \forall i = 1, \dots, M \quad (78)$$

$$X_{i,11}^{(l)} \leq x_i \leq X_{i,11}^{(r)}, \quad \forall i = 1, \dots, M$$

$$z_i = 0, \quad \forall i = 1, \dots, M \quad (79)$$

Proof: In order to prove the equivalence of Π_7 and Π_8 , it suffices to show that both problems have the same optimal value. Denoting as β_7^* and β_8^* the optimal values of Π_7 and Π_8 respectively, we will prove the equality $\beta_7^* = \beta_8^*$ by showing that $\beta_8^* \leq \beta_7^*$ and $\beta_7^* \leq \beta_8^*$.

(i) $\beta_8^* \leq \beta_7^*$: Note that the feasible set of Π_8 is contained in that of Π_7 and both problems have the same objective function; hence $\beta_8^* \leq \beta_7^*$.

(ii) $\beta_7^* \leq \beta_8^*$: We denote as $\{X_1^*, \dots, X_M^*\}$ one of the optimal solution(s) corresponding to Π_7 , and define:

$$C^* := \Sigma^{-1} + \sum_{i=1}^M \lambda_i X_i^* = \begin{bmatrix} a^* & b^* \\ b^* & d^* \end{bmatrix} \succeq \beta_7^* I_{2 \times 2}$$

where the last inequality follows from optimality and the constraint (74), and yields

$$\beta_7^* \leq \min\{a^*, d^*\} \quad (80)$$

We further define as

$$X_i' := \text{diag}(X_i^*), \quad i = 1, \dots, M$$

$$C' := \Sigma^{-1} + \sum_{i=1}^M \lambda_i X_i' = \begin{bmatrix} a^* & 0 \\ 0 & d^* \end{bmatrix}$$

$$\beta_8' := \max \beta$$

$$\text{s.t. } C' - \beta I_{2 \times 2} \succeq 0 \quad (81)$$

where from (81), it is evident that

$$\beta_8' = \min\{a^*, d^*\} \quad (82)$$

Furthermore, note that X_i' satisfies all constraints of Π_8 (i.e., $\{X_i', i = 1, 2, \dots, M\}$ is in the feasible set of Π_8), and therefore

$$\beta_8' \leq \beta_8^* \quad (83)$$

Combining (80), (82), and (83), we have:

$$\beta_7^* \leq \min\{a^*, d^*\} = \beta_8' \leq \beta_8^*.$$

The final result of this section is provided by the following lemma.

Lemma 8: The SDP optimization problem Π_8 is equivalent to the Linear Programming (LP) problem Π_9 , where

- LP OPTIMIZATION PROBLEM 9 (Π_9)

$$\text{maximize } \beta \quad (84)$$

$$\text{s.t. } \mu_1' + \sum_{i=1}^M \lambda_i x_i - \beta \geq 0 \quad (85)$$

$$\mu_2' + \sum_{i=1}^M \lambda_i - \sum_{i=1}^M \lambda_i x_i - \beta \geq 0 \quad (86)$$

$$X_{i,11}^{(l)} \leq x_i \leq X_{i,11}^{(r)} \quad \forall i = 1, \dots, M \quad (87)$$

with μ_1' and μ_2' defined in (29).

Proof: Note that in the formulation of the SDP problem Π_8 , the off-diagonal constraints (79) are satisfied by forcing all matrices X_i to become diagonal. Hence the linear matrix inequality constraint in (76) can be decomposed into the following two *linear scalar inequalities*:

$$\mu_1' + \sum_{i=1}^M \lambda_i x_i - \beta \geq 0 \quad (88)$$

$$\mu_2' + \sum_{i=1}^M \lambda_i y_i - \beta \geq 0 \quad (89)$$

where (88) is the same as the constraint in (85). Additionally, solving for y_i from the constraint $\text{tr}(X_i) = x_i + y_i = 1$ (cf. (78)), and substituting in (89), yields the constraint (86).

Finally, from the definitions of $X_{i,11}^{(l)}$ and $X_{i,11}^{(r)}$ (cf. (62) and (63)), it is evident that the constraint in (87) makes the one in (77) redundant. ■

The LP problem Π_9 , which is a relaxation of the NP-Hard problems $\Pi_1 \Leftrightarrow \dots \Leftrightarrow \Pi_5$, can be solved efficiently using *linprog* from MATLAB® [31]. Note also that the relaxations employed for deriving Π_9 do not affect the feasibility of the solution (i.e., any solution of Π_9 is within the feasible set of Π_5). Once the optimal solution $x_i^*, i = 1, \dots, M$ of Π_9 is computed, the optimal bearing directions $\theta_i^*, i = 1, \dots, M$ are calculated from $\cos^2 \theta_i^* = x_i^*$ (cf. (56) and (65)). If multiple solutions exist for θ_i^* , we choose the one that brings the sensor closer to the target.

Finally, we should note that although the computational cost for solving an LP problem can be in the order of $\mathcal{O}(M^3)$, it can be shown that the solution of Π_9 requires only $\mathcal{O}(M)$ operations (i.e., linear in the number of sensors) due to the special structure of the matrices involved. The proof is described in Appendix C.

VI. SIMULATION RESULTS

In order to evaluate the two presented *constrained* optimal motion strategies, Modified Gauss-Seidel Relaxation (MGSR) and LP Relaxation (LPR), we have conducted extensive simulation experiments and compared the performance of MGSR and LPR to the following methods:

- *Grid-Based Exhaustive Search* (GBES). In this case, we discretize the space of the sensors' heading directions and perform an exhaustive search over all possible combinations of these to find the one that minimizes the trace of the covariance matrix for the target's position estimates. Ideally, the GBES should return the global optimal solution and it could be used as a benchmark for evaluating the MGSR and the LPR, if the grid size is sufficiently small. However, this is difficult to guarantee in practice since its computational complexity and memory requirements are exponential in the number of sensors. Hence implementing the GBES becomes prohibitive when the number of sensors, M , increases and/or when the size of the grid cells decreases.

- *Random Motion* (RM). This is a modification of an intuitive strategy that would require the sensors to move towards the target. In this case, however, and in order to ensure

that the sensors do not converge to the same point (i.e., zero baseline), we require that at every time step sensor- i selects its heading direction with uniform probability towards points within the arc ACB shown in Fig. 3, i.e., each sensor is required to move towards the target at a random angle.

- *Gradient Descent with Constant step-size* (GDC). In [24], the authors considered the case where distance-only measurements are available and implemented the gradient descent algorithm with a *constant step-size*. However, *no motion constraints* were imposed on the sensors. To overcome this limitation, instead of using the bearings, we chose the headings of the sensors as the optimization variables (which are unconstrained) and implemented the GDC algorithm on the reformulated unconstrained optimization problem.

- *Gradient Descent with Successive step-size reduction* (GDS). In order to improve the performance of the GDC, and avoid the difficulty of selecting the optimal constant step-size, we also implemented the gradient descent method with *successive step-size reduction*. In this case, the step size was selected using backtracking line search (i.e., Armijo's rule, [32, Ch. 1], [30, Ch. 9]). The procedure to calculate the gradient descent direction and the algorithms to select the step-size are described in Appendix D.

A. Simulation Setup

For the purposes of this simulation, we adopt a zero-acceleration target motion model

$$\dot{\mathbf{x}}_T(t) = F \mathbf{x}_T(t) + G \mathbf{w}(t) \quad (90)$$

where

$$F = \begin{bmatrix} 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}, G = \begin{bmatrix} 0 & 0 \\ 0 & 0 \\ 1 & 0 \\ 0 & 1 \end{bmatrix}, \mathbf{x}_T(t) = \begin{bmatrix} x_T(t) \\ y_T(t) \\ \dot{x}_T(t) \\ \dot{y}_T(t) \end{bmatrix},$$

and $\mathbf{w}(t) = [w_x(t) \ w_y(t)]^T$ is a zero-mean white Gaussian noise vector with covariance $E[\mathbf{w}(t)\mathbf{w}^T(\tau)] = qI_2\delta(t-\tau)$, $q = 10$, and $\delta(t-\tau)$ is the Dirac delta. In our implementation, we discretize the continuous-time system model (cf. (90)) with time step $\delta t = 0.1$ sec.

The initial true state of the target is $\mathbf{x}_T(0) = [0, 0, -8, 4]^T$. The initial estimate for the target's state is $\hat{\mathbf{x}}_T(0) = [2, -2, 0, 0]^T$. This can be obtained by processing the first measurements from the sensors at time-step 0. At the beginning of the experiment, the sensors are randomly distributed within a circle of radius 5 m, which is at a distance of about 20 m from the target's initial position. The maximum speed for each sensor is set to 10 m/s, i.e., the largest distance that a sensor can travel during any time step is 1 m. The duration of the simulations is 10 sec (i.e., 100 time steps). At every time step, we employ the methods described (i.e., GBES, MGSR, LPR, RM, GDC and GDS) to calculate the next heading direction of each sensor. Throughout the simulations, we set the grid size for the GBES method to $\pi/200$ and the relaxation factor for the MGSR strategy to $\alpha = 0.5$.

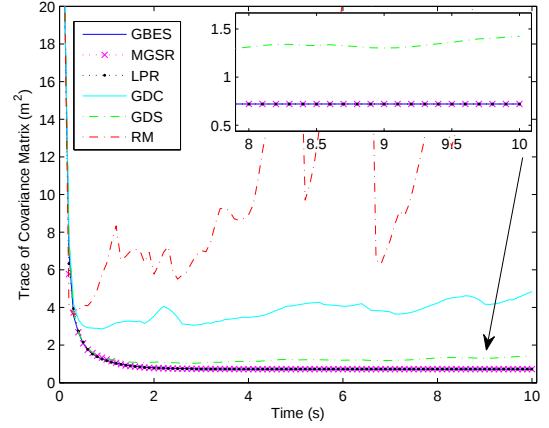


Fig. 7. [2-Sensors case] Trace of the target's position covariance matrix. Comparison between GBES, MGSR, LPR, RM, GDC, and GDS.

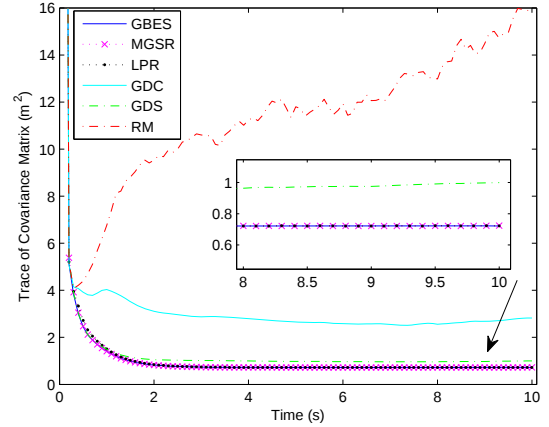


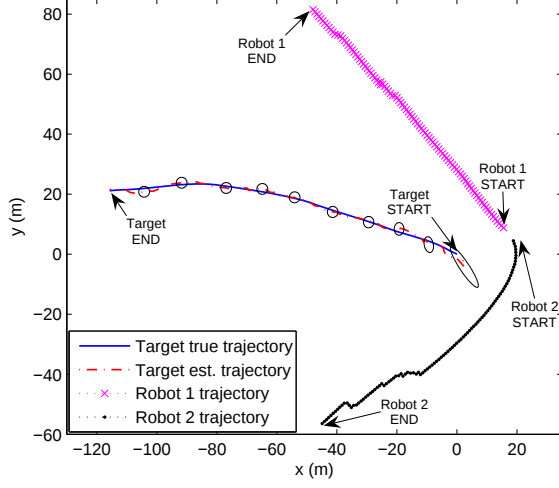
Fig. 8. [2-Sensors case, Monte Carlo simulations] Average trace of the target's position covariance matrix in 100 experiments.

B. Target Tracking with 2 Sensors (homogeneous team)

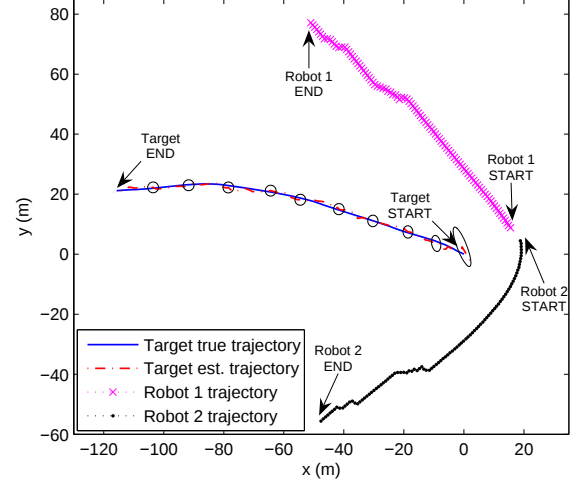
We first investigate the scenario where 2 identical sensors track a moving target, i.e., the covariance matrix of the noise in the distance measurements is $R = \sigma^2 I_{2 \times 2}$ with $\sigma = 1$.

The time evolution of the trace of the target's position covariance in a typical simulation is shown in Fig. 7. As expected, the performance of any of the optimized approaches (i.e., MGSR, LPR, GBES, and GDS) is significantly better compared to that of the non-optimized cases (i.e., RM and GDC). Additionally, the uncertainty in the target's position estimates (trace of the covariance matrix) achieved by either of the two proposed motion strategies, MGSR and LPR, is indistinguishable of that of the GBES, at a cost linear, instead of exponential, in the number of sensors. Note also that although the GDS algorithm performs significantly better than the GDC, it is approximately 50-100% worse when compared to the MGSR and LPR optimal motion strategies. These results are typical for all experiments conducted and are summarized, for the average of 100 trials, in Fig. 8.

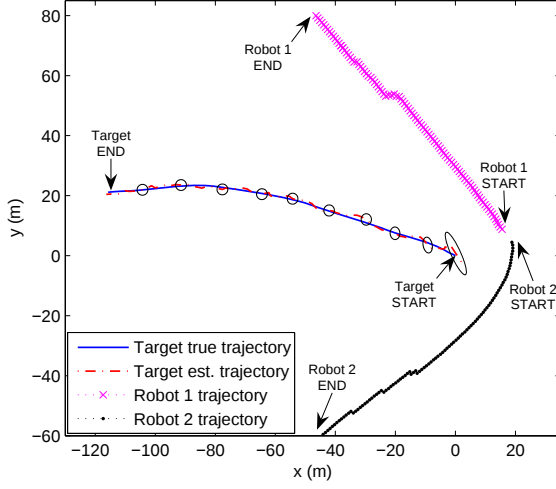
Figs 10(a), 10(b), 10(c), 10(d), 10(e), and 10(f), depict the actual and estimated trajectories of the target, along



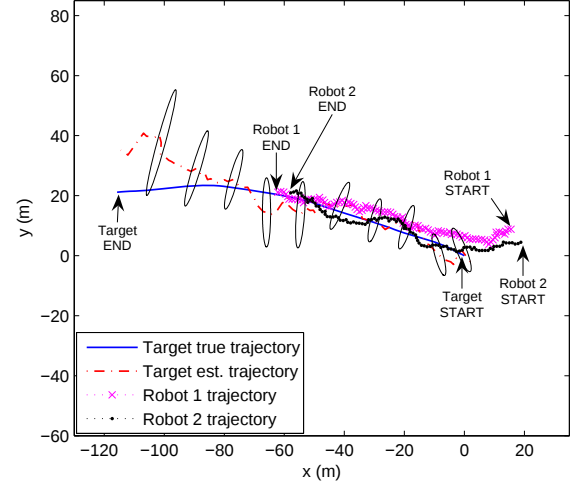
(a)



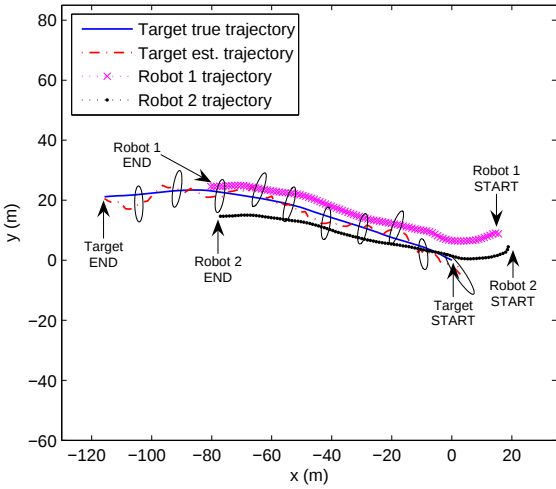
(b)



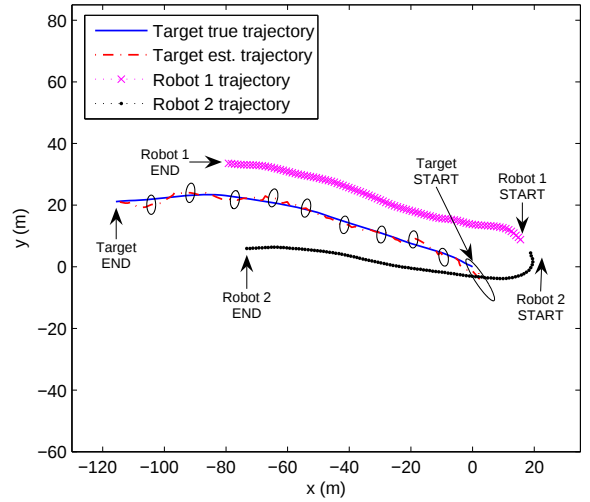
(c)



(d)



(e)



(f)

Fig. 10. [2-Sensors case] Trajectories of the two sensors, and the actual and estimated trajectories of the target, when employing as motion strategy (a) MGSR, (b) LPR, (c) GBES, (d) RM, (e) GDC, and (f) GDS. The ellipses denote the 3σ bounds for the target's position uncertainty at the corresponding time steps.

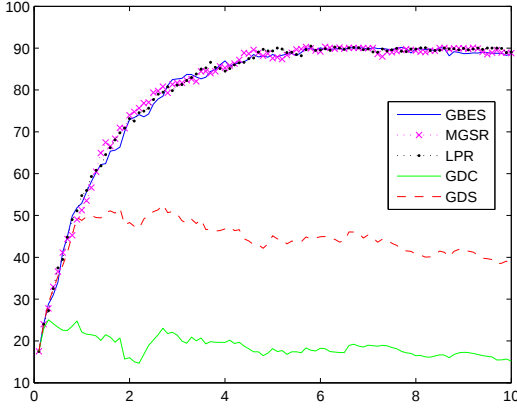


Fig. 9. [2-Sensors case] The angle formed by sensor-1, the target, and sensor-2 vs. time. As time increases this angle approaches 90 degrees.

with the trajectories of the two sensors, when employing as motion strategy MGSR, LPR, GBES, RM, GDC, and GDS, respectively. As evident, the accuracy of the target's position estimates for both MGSR and LPR are significantly better than the case of RM, GDC, and GDS, and almost identical to that of GBES. Additionally, for both MGSR and LPR, the EKF produces consistent estimates, i.e., the real target's position is within the 3σ ellipse centered at the target's estimated position. This is not the case for the RM strategy where the inconsistency is due to the large errors in the state estimates used for approximating the measurement Jacobian.

Note also that for both MGSR and LPR (cf. Fig.s 10(a) and 10(b)), although the two sensors start close to each other, they immediately move in separate directions and eventually form a right angle with vertex the location of the target (cf. Fig. 9). This interesting result is explained as follows: Based on Lemma 3, the optimal motion strategy for the two sensors minimizes the difference between the maximum and the minimum eigenvalue of the covariance matrix. Once this difference approaches zero, the eigenvalues of the prior covariance matrix are almost identical and the uncertainty ellipse becomes a circle. In this case, for $M = 2$ we have (cf. (38)): $\lambda_0 = \mu'_1 - \mu'_2 \simeq 0$, $\lambda_1 = \lambda_2 = 1/\sigma^2 = 1$, $\mathbf{v}_0 \simeq [0 \ 0]^T$, $\mathbf{v}_1 = [\cos 2\theta_1 \ \sin 2\theta_1]^T$, and $\mathbf{v}_2 = [\cos 2\theta_2 \ \sin 2\theta_2]^T$. Hence the optimal solution to (39) is $|\bar{\theta}_2 - \bar{\theta}_1| \simeq \pi/2$, which requires that the two sensors should move so as to measure their distances to the target from perpendicular directions.

C. Target Tracking with 4 Sensors (heterogeneous team)

We hereafter examine the performance of the MGSR and LPR motion strategies for a heterogeneous team of 4 sensors tracking a moving target. In this case, the covariance matrix of the noise in the distance measurements is set to $R = \text{diag}(\sigma_i^2)$, with $\sigma_1^2 = 1$ and $\sigma_2^2 = \sigma_3^2 = \sigma_4^2 = 3$.

Fig.s 11(a), 11(b), 11(c), and 11(d) depict the actual and estimated trajectories of the target, along with the trajectories of the four sensors, when employing as motion strategy MGSR, LPR, GBES, and RM, respectively. As for the case of 2 sensors, the accuracy of the target's position estimates for both MGSR and LPR are significantly better than that of RM

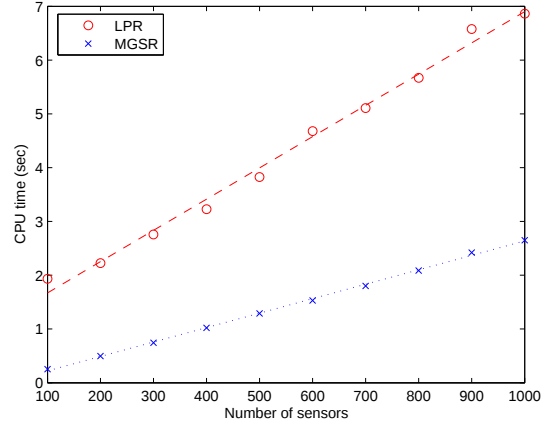


Fig. 12. [2-Sensors case, Monte Carlo simulations] Average run-time of LPR (circles) and MGSR (x's) in 100 experiments. The dashed and dotted lines show the best linear fit for the time required by each of the two methods.

and almost identical to that of GBES. Furthermore, the EKF estimates from the MGSR, LPR, and GBES are consistent.

Interestingly, in this case the heterogeneous sensor team splits into two groups. Sensor-1 (the most accurate one with distance measurement noise variance $\sigma_1^2 = 1$) follows the target from the left, while sensors 2, 3, and 4 form a separate cluster approaching the target from the right while moving very close to each other. The reason for this is the following: As sensors 2, 3, and 4 measure their distances to the target from approximately the same location at every time step, their independent distance measurements become equivalent, in terms of accuracy, to one with variance

$$\frac{1}{\sigma_{2,3,4}^2} \simeq \frac{1}{\sigma_2^2} + \frac{1}{\sigma_3^2} + \frac{1}{\sigma_4^2} = 1, \text{ or } \sigma_{2,3,4}^2 \simeq 1$$

Hence, this problem becomes equivalent to that of 2 sensors with equal noise variances (cf. Section VI-B), with the difference that in this case the “second” sensor is realized by requiring sensors 2, 3, and 4 to move close to each other.

Finally, we should note that for this case, the time evolution of the trace of the target's position covariance matrix is almost identical to that of Fig. 7.

D. Scalability and Run-Time

Contrary to the GBES method, which has computational and memory requirements exponential in the number of sensors, the complexity of the MGSR and LPR algorithms is only linear. In order to corroborate our theoretical analysis, we have evaluated the computation time required by these two algorithms⁷ for the case of a homogeneous sensor team ($R = \text{diag}(\sigma_i^2)$, with $\sigma_i^2 = 1$ for $i = 1, \dots, M$) tracking a moving target. Specifically, we have examined the scalability of our algorithms by varying M from 100 to 1000, and for every value of M , we have conducted 100 simulations. These results are summarized in Fig. 12. Furthermore, after performing linear regression on these data we have determined that the

⁷Note that due to its exponential complexity, it was not possible to implement GBES for teams of more than 5 sensors.

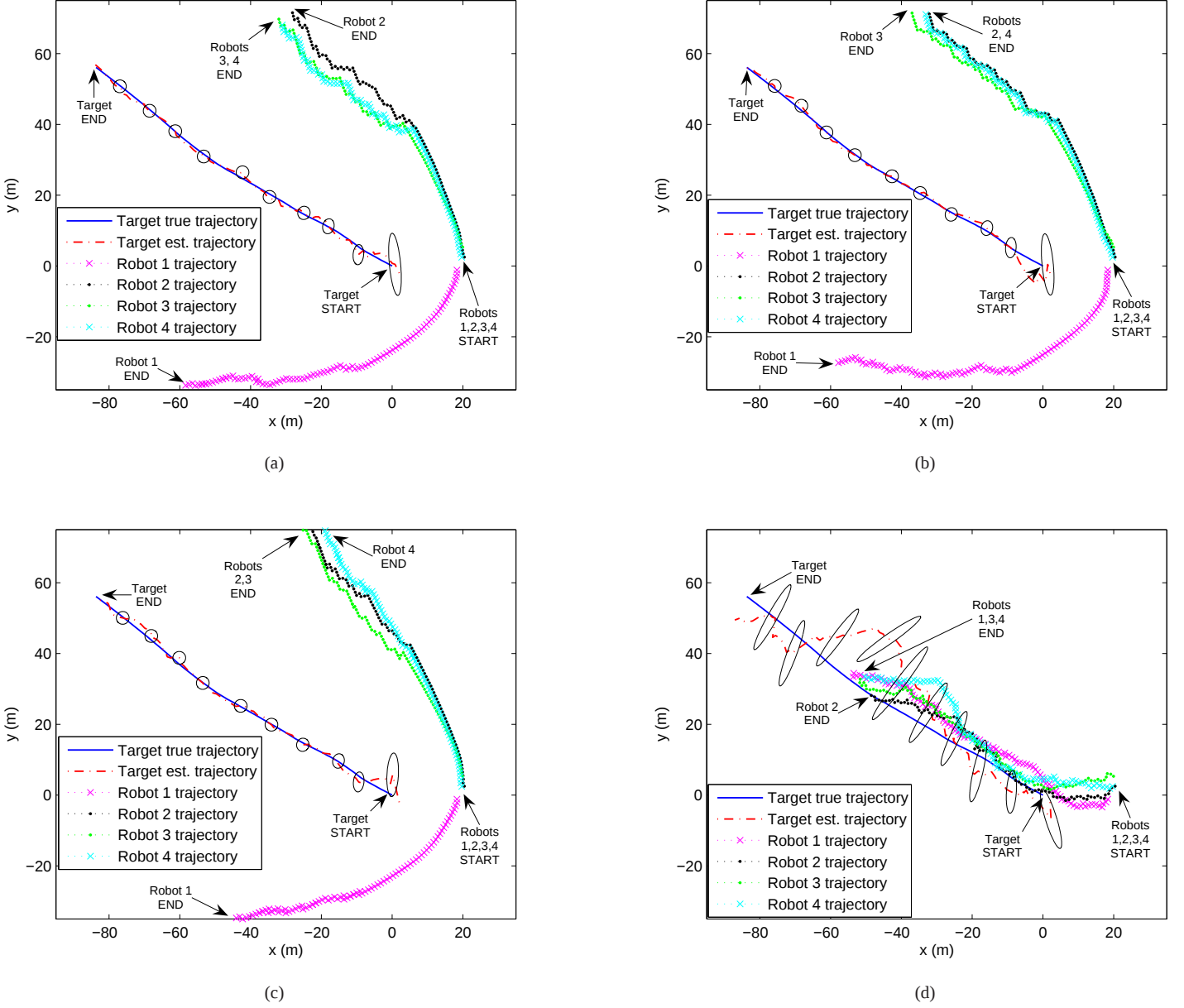


Fig. 11. [4-Sensors case] Trajectories of the four sensors, and the actual and estimated trajectories of the target, when employing as motion strategy (a) MGSR, (b) LPR, (c) GBES, and (d) RM. The ellipses denote the 3σ bounds for the uncertainty of the target's position estimates at the corresponding time steps.

average time (in sec) required by each of these algorithms is:

$$t_{MGSR} = 0.0027M - 0.0480$$

$$t_{LPR} = 0.0058M + 1.0907$$

As evident from the precise linear fit shown in Fig. 12, both algorithms have linear, in the number of sensors computational complexity. Finally, we should note that the main reason for the slower performance of the LPR algorithm (when compared to the MGSR) is that we employ the MATLAB built-in function “linprog” to solve the linear program. This function improves the numerical accuracy of the LPR algorithm at the expense of additional preprocessing steps.

VII. CONCLUSIONS AND FUTURE WORK

In this paper, we address the problem of constrained optimal motion strategies for heterogeneous teams of mobile sensors tracking a moving target using range-only measurements. Our objective is to determine the best locations that the sensors should move to at every time step in order to collect the most informative distance measurements, i.e., the measurements that minimize the trace of the target's position covariance matrix. We have shown that this problem can be exactly reformulated to that of minimizing the norm of the sum of vectors of different lengths with constraints imposed on their directions. These constraints, which result from limitations on the maximum speed of each sensor, make the problem NP-Hard, in general.

In order to provide solutions that can be implemented in

equivalence:⁸

$$\begin{aligned} & \{\theta_1^*, \dots, \theta_M^*\} \\ &= \arg \max_{\theta_1(k+1), \dots, \theta_M(k+1)} \text{tr}(PH_e^T(H_ePH_e^T + R)^{-1}H_eP) \end{aligned} \quad (96)$$

Now let us restrict the above optimization problem (96) to the case of a single sensor, i.e.,

$$\theta_1^* = \arg \max_{\theta_1} \text{tr}(PH_e^T(H_ePH_e^T + R)^{-1}H_eP) \quad (97)$$

Here, the measurement matrix H_e is a row vector with dimension 1×2 , the measurement noise covariance R becomes a scalar. We denote:

$$\begin{aligned} \mathbf{h} &:= H_e^T = [\cos \theta_1, \sin \theta_1]^T \\ r &:= R = \sigma^2 \end{aligned}$$

and notice that the term $\mathbf{h}^T P \mathbf{h} + r$ is a scalar, thus (97) can be rewritten as:

$$\theta_1^* = \arg \max_{\theta_1} \left(\frac{\text{tr}(P \mathbf{h} \mathbf{h}^T P)}{\mathbf{h}^T P \mathbf{h} + r} \right) \quad (98)$$

Applying the identity $\text{tr}(AB) = \text{tr}(BA)$, we have:

$$\text{tr}(P \mathbf{h} \mathbf{h}^T P) = \text{tr}(\mathbf{h}^T P P \mathbf{h}) = \mathbf{h}^T P^2 \mathbf{h} \quad (99)$$

The second equality of (99) holds since $\mathbf{h}^T P P \mathbf{h}$ is a scalar.

Substituting (99) into (98) and applying the identity $\mathbf{h}^T \mathbf{h} = 1$ to write $r = \mathbf{h}^T (r I_{2 \times 2}) \mathbf{h}$, we derive:

$$\theta_1^* = \arg \max_{\theta_1} \left(\frac{\mathbf{h}^T P^2 \mathbf{h}}{\mathbf{h}^T (P + r I_{2 \times 2}) \mathbf{h}} \right) \quad (100)$$

In order to transform (100) into the standard Rayleigh quotient form, we employ the Singular Value Decomposition (SVD) of P (cf. (29)):

$$P = U \Sigma U^T$$

where $\Sigma = \text{diag}(\nu'_1, \nu'_2)$, $\nu'_1 = (\mu'_1)^{-1}$, $\nu'_2 = (\mu'_2)^{-1}$ and $0 \leq \nu'_1 \leq \nu'_2$, and define $\bar{\mathbf{h}} := [\cos(\theta_1 - \theta_0), \sin(\theta_1 - \theta_0)]^T$, we obtain:

$$\frac{\mathbf{h}^T P^2 \mathbf{h}}{\mathbf{h}^T (P + r I_{2 \times 2}) \mathbf{h}} = \frac{\bar{\mathbf{h}}^T \Sigma^2 \bar{\mathbf{h}}}{\bar{\mathbf{h}}^T (\Sigma + r I_{2 \times 2}) \bar{\mathbf{h}}} \quad (101)$$

Next, let us define $\check{\mathbf{h}} := (\Sigma + r I_{2 \times 2})^{1/2} \bar{\mathbf{h}}$, and (101) can be rewritten as:

$$\begin{aligned} \frac{\bar{\mathbf{h}}^T \Sigma^2 \bar{\mathbf{h}}}{\bar{\mathbf{h}}^T (\Sigma + r I_{2 \times 2}) \bar{\mathbf{h}}} &= \frac{\check{\mathbf{h}}^T ((\Sigma + r I_{2 \times 2})^{-1/2} \Sigma^2 (\Sigma + r I_{2 \times 2})^{-1/2}) \check{\mathbf{h}}}{\check{\mathbf{h}}^T \check{\mathbf{h}}} \\ &:= \frac{\check{\mathbf{h}}^T \check{A} \check{\mathbf{h}}}{\check{\mathbf{h}}^T \check{\mathbf{h}}} \end{aligned} \quad (102)$$

which is the standard form of the Rayleigh quotient, with $\check{A} = (\Sigma + r I_{2 \times 2})^{-1/2} \Sigma^2 (\Sigma + r I_{2 \times 2})^{-1/2}$.

Since

$$\begin{aligned} & (\Sigma + r I_{2 \times 2})^{-1/2} \Sigma^2 (\Sigma + r I_{2 \times 2})^{-1/2} \\ &= \begin{bmatrix} (\nu'_1)^2 / (\nu'_1 + r) & 0 \\ 0 & (\nu'_2)^2 / (\nu'_2 + r) \end{bmatrix} \end{aligned}$$

⁸For clarity, we drop the time index from the covariance matrix and the measurement matrix, i.e., $P = P_{k+1|k,11}$ and $H_e = H_{e,k+1}$.

and it is easy to verify that when $0 \leq \nu'_1 \leq \nu'_2$:

$$(\nu'_1)^2 / (\nu'_1 + r) \leq (\nu'_2)^2 / (\nu'_2 + r)$$

so the unconstrained optimization problem

$$\max_{\check{\mathbf{h}}} \left(\frac{\check{\mathbf{h}}^T ((\Sigma + r I_{2 \times 2})^{-1/2} \Sigma^2 (\Sigma + r I_{2 \times 2})^{-1/2}) \check{\mathbf{h}}}{\check{\mathbf{h}}^T \check{\mathbf{h}}} \right)$$

achieve the maximum value at $\check{\mathbf{h}}^* = (\nu'_2 + r)^{1/2} [0, 1]^T$, i.e., $\bar{\mathbf{h}}^* = [0, 1]^T$, or equivalently, $\theta_1^* = \theta_0 + \pi/2$.

For the original constrained optimization problem, the above result shows that the sensor should always move towards the direction of the eigenvector corresponding to the maximum eigenvalue of $P_{k+1|k,11}$, which is exactly the same result as we have obtained in Section IV-A.

APPENDIX C

COMPUTATIONAL COMPLEXITY ANALYSIS OF THE LP RELAXATION METHOD

We hereafter adopt the notation and methodology of [37, Ch. 20] to evaluate the computational complexity of solving the LP problem Π_9 .

Most LP problems take the following form:

$$\min_{\mathbf{x}} \{ \mathbf{c}^T \mathbf{x} ; A \mathbf{x} \geq \mathbf{b}, \mathbf{x} \leq \mathbf{b}_u, \mathbf{x} \geq \mathbf{0} \} \quad (103)$$

where $\mathbf{c}, \mathbf{x}, \mathbf{b}_u \in \mathbb{R}^m$, $\mathbf{b} \in \mathbb{R}^n$, and $A \in \mathbb{R}^{n \times m}$. Additionally, the dual problem of (103), used in the solution process, is written as:

$$\max_{\mathbf{y}, \mathbf{y}_u} \{ \mathbf{b}^T \mathbf{y} - \mathbf{b}_u^T \mathbf{y}_u ; A^T \mathbf{y} - \mathbf{y}_u \leq \mathbf{c}, \mathbf{y} \geq \mathbf{0}, \mathbf{y}_u \geq \mathbf{0} \}$$

where $\mathbf{y} \in \mathbb{R}^n$, $\mathbf{y}_u \in \mathbb{R}^m$.

To transform the LP problem Π_9 to the standard form of (103), we define:

$$\begin{aligned} \mathbf{x} &= [\zeta_1, \zeta_2, \dots, \zeta_M, \zeta_{M+1}]^T \\ &:= [x_1 - X_{1,11}^{(l)}, x_2 - X_{2,11}^{(l)}, \dots, x_M - X_{M,11}^{(l)}, \beta]^T \\ A &:= \begin{bmatrix} \lambda_1 & \lambda_2 & \dots & \lambda_M & -1 \\ -\lambda_1 & -\lambda_2 & \dots & -\lambda_M & -1 \end{bmatrix} \\ \mathbf{b} &:= [-\mu'_1 - \sum_{i=1}^M \lambda_i X_{i,11}^{(l)}, -\mu'_2 + \sum_{i=1}^M \lambda_i X_{i,11}^{(l)} - \sum_{i=1}^M \lambda_i]^T \\ \mathbf{b}_u &:= [X_{1,11}^{(r)} - X_{1,11}^{(l)}, \dots, X_{M,11}^{(r)} - X_{M,11}^{(l)}, \mu'_1 + \mu'_2 + \sum_{i=1}^M \lambda_i]^T \\ \mathbf{c} &:= [\mathbf{0}_{1 \times M}, -1] \end{aligned} \quad (104)$$

with $n = 2$ and $m = M + 1$.

In order to solve (103) using the Interior Point Method (IPM), we need to introduce the slack variables

$$\mathbf{z} := A \mathbf{x} - \mathbf{b}, \mathbf{z}_u := \mathbf{b}_u - \mathbf{x}, \mathbf{s} := \mathbf{c} + \mathbf{y}_u - A^T \mathbf{y}$$

and define the following diagonal matrices

$$\begin{aligned} X &:= \text{diag}(\mathbf{x}), \quad Y := \text{diag}(\mathbf{y}), \quad Z := \text{diag}(\mathbf{z}) \\ Y_u &:= \text{diag}(\mathbf{y}_u), \quad Z_u := \text{diag}(\mathbf{z}_u), \quad S := \text{diag}(\mathbf{s}) \end{aligned}$$

The Newton direction, $\mathbf{y}_d$, can be calculated by solving the

following *normal equation* [37, Ch. 20]:

$$(AD_2A^T + D_1)\mathbf{y}_d = \mathbf{r} + AD_2\mathbf{h} \quad (105)$$

where

$$D_1 = ZX^{-1}, \quad D_2 = (SX^{-1} + Y_u Z_u^{-1})^{-1} \\ \mathbf{r} = Y^{-1}(\gamma\mathbf{1} - Y\mathbf{z}), \quad \mathbf{h} = Z_u^{-1}(\gamma\mathbf{1} - Z_u\mathbf{y}_u) - X^{-1}(\gamma\mathbf{1} - X\mathbf{s})$$

with $\mathbf{1}$ denoting the $M \times 1$ vector of ones and $\gamma > 0$ representing the duality gap.

Solving (105) is often the most computationally intensive step of the IPM since employing Gaussian elimination requires $\mathcal{O}(n^2m + n^3)$ operations [38, Ch. 3]. In the case of Π_9 , however, this can be done in *constant time* – the matrix $(AD_2A^T + D_1)$ and the vector $(\mathbf{r} + AD_2\mathbf{h})$ in (105) are of dimensions 2×2 and 2×1 , respectively. Instead, the most computationally demanding process for Π_9 is calculating these two quantities. Noting though that matrix A has dimensions $2 \times (M+1)$ (cf. (104)) and the remaining quantities involved are either $M \times M$ *diagonal* matrices or $M \times 1$ vectors, this requires only $\mathcal{O}(M)$ operations to calculate the matrix $(AD_2A^T + D_1)$ and the vector $(\mathbf{r} + AD_2\mathbf{h})$.

APPENDIX D

GRADIENT DESCENT METHOD AND IMPLEMENTATIONS

Gradient descent method is widely applied to nonlinear optimization problems [32, Ch. 1]. It has the advantage of an explicit form and the gradient vector can be easily obtained for most differentiable objective function. The authors of [23] and [24] generate the optimal motion strategy, based on the gradient descent algorithm with *constant* step-size, for a team of mobile sensors tracking moving targets. In this section, we will describe the implementations of two gradient descent algorithms, one employing the constant step-size rule (called “GDC” in Section VI), the other selecting the step-size successively (called “GDS” in Section VI), for the target tracking problem using range-only measurements.

Before we proceed to the calculation of the gradient vector and the selection of the step-size, it is worthy noting that the gradient descent algorithms are often employed to solve *unconstrained* nonlinear optimization problem [30, Ch. 9]. Although some techniques, such as *Gradient Projection Methods* [32, Ch. 2], can be adopted in the gradient descent algorithm and applied to solve *constrained* optimization problem, it is sometimes computationally expensive to calculate the projection directions or vectors. Recall the original formulation of the optimal motion strategy problem (cf. Π_2 in Lemma 2 in Section III-D), while in terms of the bearings $\{\theta_i, i = 1, \dots, M\}$ as the optimization variables, is a *constrained* optimization problem (i.e., the box constraint on θ_i in Π_2 due to the speed constraint on sensor- i). On the other hand, it is worthy noticing that φ_i , the heading of sensor- i , is *unconstrained*. Moreover, φ_i and θ_i are related through (25). Therefore, to avoid the complication of calculating the projection directions for constrained optimization problem Π_2 , we replace the bearings $\{\theta_i, i = 1, \dots, M\}$ with the headings $\{\varphi_i, i = 1, \dots, M\}$ as the optimization variables and consider the following equivalent optimization problem:

• OPTIMIZATION PROBLEM 10 (Π_{10})

$$\underset{\varphi_1, \dots, \varphi_M}{\text{minimize}} \quad \text{tr}(P_{k+1|k+1,11}) = f(\varphi_1, \dots, \varphi_M)$$

There are two key steps to implement the gradient descent method. The first step is to calculate the gradient vector, which we will discuss in detail next (in Appendix D-A). The second step is to determine the step-size, which we will describe in the following Appendix D-B.

A. Gradient Vector Calculation

The gradient vector for Π_{10} has the following form:

$$\nabla_{\varphi} f = \left[\frac{\partial f}{\partial \varphi_1}, \dots, \frac{\partial f}{\partial \varphi_M} \right]$$

where $\varphi = [\varphi_1, \dots, \varphi_M]^T$ and $f(\varphi) := f(\varphi_1, \dots, \varphi_M)$.

Our goal is to obtain the explicit form for $\frac{\partial f}{\partial \varphi_i}$, $i = 1, \dots, M$.

Let us first apply the chain rule and since $\text{tr}(P_{k+1|k+1,11})$ can be written in a closed form in terms of θ_i , $i = 1, \dots, M$, we can immediately have:

$$\frac{\partial f}{\partial \varphi_i} = \sum_{j=1}^M \frac{\partial f}{\partial \theta_j} \frac{\partial \theta_j}{\partial \varphi_i} \quad (106)$$

Since θ_i is only dependent on φ_i , and independent on φ_j for $j \neq i$, which implies $\frac{\partial \theta_j}{\partial \varphi_i} = 0$ for $j \neq i$, hence we can immediately simplify (106) to:

$$\frac{\partial f}{\partial \varphi_i} = \frac{\partial f}{\partial \theta_i} \frac{d\theta_i}{d\varphi_i} \quad (107)$$

Let us first consider the term $\frac{\partial f}{\partial \theta_i}$. To find the closed-form expression, we need to combine (28), (33) and (41) into:

$$\text{tr}(P_{k+1|k+1,11}) = \frac{c}{d_c - \frac{1}{4}d_{\bar{\theta}}} \quad (108)$$

where c and d_c are constants defined in (33) and (41) respectively, and the $d_{\bar{\theta}}$ is a function of $\bar{\theta}$ (i.e., θ) expressed in (42).

From (108), we obtain:

$$\frac{\partial f}{\partial \theta_i} = \frac{1}{4} \frac{c}{(d_c - \frac{1}{4}d_{\bar{\theta}})^2} \frac{\partial d_{\bar{\theta}}}{\partial \theta_i} = \frac{1}{4} \frac{(\text{tr}(P_{k+1|k+1,11}))^2}{c} \frac{\partial d_{\bar{\theta}}}{\partial \theta_i} \quad (109)$$

The last equality follows from (108).

The closed form of $\frac{\partial d_{\bar{\theta}}}{\partial \theta_i}$ can be found from (42):

$$\frac{\partial d_{\bar{\theta}}}{\partial \theta_i} = 4\lambda_i \\ \times \left[\left(\sum_{j=1}^M \lambda_j \sin 2\bar{\theta}_j \right) \cos 2\bar{\theta}_i - \left(\lambda_0 + \sum_{j=1}^M \lambda_j \cos 2\bar{\theta}_j \right) \sin 2\bar{\theta}_i \right] \quad (110)$$

We combine (109) and (110) together and obtain:

$$\frac{\partial f}{\partial \theta_i} = \lambda_i \frac{(\text{tr}(P_{k+1|k+1,11}))^2}{c} \\ \times \left[\left(\sum_{j=1}^M \lambda_j \sin 2\bar{\theta}_j \right) \cos 2\bar{\theta}_i - \left(\lambda_0 + \sum_{j=1}^M \lambda_j \cos 2\bar{\theta}_j \right) \sin 2\bar{\theta}_i \right] \quad (111)$$

Our next step is to find the closed form expression for $\frac{d\theta_i}{d\varphi_i}$. This can be obtained through the geometric relation between θ_i and φ_i from Fig. 13 (For notational brevity, we will drop

all the time index (i.e., k or $k+1$) from all variables in Fig. 13 and in Appendix A).

From Fig. 13, we have:

$$\widehat{EOA} = \widehat{DOA} - \widehat{DOE} = \varphi_i - \theta'_i \quad (112)$$

$$\widehat{AEO} = \eta'_i = \theta'_i - \theta_i \quad (113)$$

where θ'_i is defined in (21).

Hence, we obtain $\widehat{OAE}$ from (112) and (113):

$$\widehat{OAE} = \pi - \widehat{EOA} - \widehat{AEO} = \pi - \varphi_i + \theta_i \quad (114)$$

Let us apply the rule of sin to the triangle $\triangle OAE$:

$$\frac{|OA|}{\sin(\widehat{AEO})} = \frac{|OE|}{\sin(\widehat{OAE})} \quad (115)$$

where $|OA| = v_i \delta t$ and $|OE| = \tilde{d}'_i$ denote the length of vectors $\vec{OA}$ and $\vec{OE}$ respectively. Substituting (113) and (114) into (115), we have:

$$\begin{aligned} \frac{v_i \delta t}{\sin(\theta'_i - \theta_i)} &= \frac{\tilde{d}'_i}{\sin(\varphi_i - \theta_i)} \\ \Rightarrow r_i \sin(\varphi_i - \theta_i) &= \tilde{d}'_i \sin(\theta'_i - \theta_i) \end{aligned} \quad (116)$$

where $r_i := v_i \delta t$ and $\tilde{d}'_i$ is defined in (20).

Since r_i , θ'_i and $\tilde{d}'_i$ are constant, from (116) we have:

$$\begin{aligned} r_i \cos(\varphi_i - \theta_i) (d\varphi_i - d\theta_i) &= -\tilde{d}'_i \cos(\theta'_i - \theta_i) d\theta_i \\ \Rightarrow \frac{d\theta_i}{d\varphi_i} &= \frac{r_i \cos(\varphi_i - \theta_i)}{r_i \cos(\varphi_i - \theta_i) - \tilde{d}'_i \cos(\theta'_i - \theta_i)} \end{aligned} \quad (117)$$

Hence, the explicit closed form of $\frac{\partial f}{\partial \varphi_i}$ is acquired by combining (107), (111) and (117):

$$\begin{aligned} \frac{\partial f}{\partial \varphi_i} &= \lambda_i \frac{(\text{tr}(P_{k+1|k+1,11}))^2}{c} \frac{r_i \cos(\varphi_i - \theta_i)}{r_i \cos(\varphi_i - \theta_i) - \tilde{d}'_i \cos(\theta'_i - \theta_i)} \\ &\times \left[\left(\sum_{j=1}^M \lambda_j \sin 2\bar{\theta}_j \right) \cos 2\bar{\theta}_i - \left(\lambda_0 + \sum_{j=1}^M \lambda_j \cos 2\bar{\theta}_j \right) \sin 2\bar{\theta}_i \right] \end{aligned} \quad (118)$$

Once we have the closed-form expression of the gradient vector $\nabla_{\varphi} f$, then the *unconstrained* variable φ_i ($i = 1, \dots, M$) is updated through the gradient descent method as following:

$$\varphi_i^{(\ell+1)} = \varphi_i^{(\ell)} - \alpha_s \frac{\partial f}{\partial \varphi_i} \quad i = 1, \dots, M \quad (119)$$

where $\alpha_s (> 0)$ is the step-size.

It is worthy pointing out that it is fairly easy to obtain the updated θ_i from the updated φ_i . First, from Fig. 13, the coordinate of the point A is: $[x_A, y_A]^T = [x_i(k) + r_i \cos \varphi_i, y_i(k) + r_i \sin \varphi_i]^T$. Hence the bearing from A to E (i.e., θ_i) can be readily obtained:

$$\begin{aligned} \theta_i &= \text{Atan2}(\hat{y}_T(k+1|k) - y_A, \hat{x}_T(k+1|k) - x_A) \\ &= \text{Atan2}(\hat{y}_T(k+1|k) - y_i(k) - r_i \sin \varphi_i, \\ &\quad \hat{x}_T(k+1|k) - x_i(k) - r_i \cos \varphi_i) \end{aligned}$$

B. Step-size Selection

The next step after the acquisition of the gradient descent direction is to select the step-size (i.e., the value of α_s in (119)). It is important to choose the appropriate step-size in gradient descent methods. If α_s is too large, the algorithm might diverge; if α_s is too small, the convergence rate would be slow. In fact, there is no single rule that could determine the optimal α_s for all problems. Next, we will present two strategies (cf. GDC and GDS in Section VI) for selecting the step-size. Their performance and simulation results are demonstrated in Section VI.

1) *Constant Step-size (GDC)*: Here, the step-size α_s is fixed to a constant value and will not change over time. The constant step-size rule is very simple and is applied in both [23] and [24]. However, it is often very difficult to determine the appropriate α_s [32, Ch. 1]. In our implementation of the GDC, we select $\alpha_s = 1$ in (119), i.e.,

$$\varphi_i^{(\ell+1)} = \varphi_i^{(\ell)} - \frac{\partial f}{\partial \varphi_i} \quad i = 1, \dots, M$$

The performance of GDC is compared with the results from the GBES, AR, LPR and GDS in Figs 7, 8 and 9.

2) *Successive Step-size Reduction (GDS)*: To overcome the difficulty of selecting the appropriate step-size in the GDC, we have implemented the successive step-size reduction (i.e., Armijo rule) algorithm to choose the step-size. Here the step-size α_s is successively reduced by a constant factor β_s ($0 < \beta_s < 1$) until a sufficient improvement of the objective function is achieved, i.e.,

$$f(\varphi - \alpha_s \nabla_{\varphi} f) - f(\varphi) \leq -\gamma_s \alpha_s \|\nabla_{\varphi} f\|_2^2$$

where the constant $\gamma_s \in [0, 0.5]$.

Throughout the simulations in Section VI, we set $\beta_s = 0.5$ and $\gamma_s = 0.1$. The performance of GDS is compared with the results from the GBES, AR, LPR and GDC in Figs 7, 8 and 9.

For the completeness, we present the basic steps of the GDS in Algorithm 2. More details and discussions about the convergence properties of this algorithm can be found in [32, Ch. 1] and [30, Ch. 9].

Algorithm 2 GDS Algorithm

Require: $f(\varphi)$, $\nabla_{\varphi} f$, $\beta_s \in (0, 1)$ and $\gamma_s \in [0, 0.5]$

Ensure: $\alpha_s \in (0, 1]$ {Determine the step-size}

- 1: $\alpha_s := 1$
 - 2: **while** $f(\varphi - \alpha_s \nabla_{\varphi} f) - f(\varphi) > -\gamma_s \alpha_s \|\nabla_{\varphi} f\|_2^2$ **do**
 - 3: $\alpha_s := \beta_s \alpha_s$
 - 4: **end while**
 - 5: **Return** α_s
-

REFERENCES

- [1] J. Polastre, "Design and implementation of wireless sensor networks for habitat monitoring," Master's thesis, University of California, Berkeley, Berkeley, CA, May 23 2003.
- [2] L. E. Parker, B. Birch, and C. Reardon, "Indoor target intercept using an acoustic sensor network and dual wavefront path planning," in *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems*, Las Vegas, NV, Oct. 27-31 2003, pp. 278 – 283.

- [3] R. Bodor, R. Morlok, and N. Papanikolopoulos, "Dual-camera system for multi-level activity recognition," in *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems*, Sendai, Japan, Sept. 28 - Oct. 2 2004, pp. 643 - 648.
- [4] J. Pineau, M. Montemerlo, M. Pollack, N. Roy, and S. Thrun, "Towards robotic assistants in nursing homes: Challenges and results," *Robotics and Autonomous Systems*, vol. 42, no. 3-4, pp. 271-281, Mar. 2003.
- [5] G. M. Siouris, G. Chen, and J. Wang, "Tracking an incoming ballistic missile using an extended interval kalman filter," *IEEE Transactions on Aerospace and Electronic Systems*, vol. 33, no. 1, pp. 232-240, Jan. 1997.
- [6] B. Jung and G. S. Sukhatme, "Tracking targets using multiple robots: The effect of environment occlusion," *Autonomous Robots*, vol. 13, no. 3, pp. 191-205, Nov. 2002.
- [7] R. Vidal, O. Shakernia, H. J. Kim, D. H. Shim, and S. Sastry, "Probabilistic pursuit-evasion games: theory, implementation, and experimental evaluation," *IEEE Transactions on Robotics and Automation*, vol. 18, no. 5, pp. 662-669, Oct. 2002.
- [8] K. Zhou and S. I. Roumeliotis, "Optimal motion strategies for range-only distributed target tracking," in *Proceedings of the American Control Conference*, Minneapolis, MN, June 14-16 2006, pp. 5195-5200.
- [9] R. Barsanti and M. Tummala, "Parameter estimation for target tracking with uncertain sensor positions," in *Proceedings of the IEEE International Symposium on Circuits and Systems*, Sydney, Australia, May 6-9 2001, pp. 257-260.
- [10] K. C. Chang, R. K. Saha, and Y. Bar-Shalom, "On optimal track-to-track fusion," *IEEE Transactions on Aerospace and Electronic Systems*, vol. 33, no. 4, pp. 1271-1276, Oct. 1997.
- [11] H. J. S. Feder, J. J. Leonard, and C. M. Smith, "Adaptive mobile robot navigation and mapping," *International Journal of Robotics Research*, vol. 18, no. 7, pp. 650-668, July 1999.
- [12] N. Trawny and T. Barfoot, "Optimized motion strategies for cooperative localization of mobile robots," in *Proceedings of the IEEE International Conference on Robotics and Automation*, New Orleans, LA, Apr. 26 - May 1 2004, pp. 1027-1032.
- [13] V. N. Christopoulos and S. I. Roumeliotis, "Adaptive sensing for instantaneous gas release parameter estimation," in *Proc. IEEE International Conference on Robotics and Automation*, Barcelona, Spain, Apr. 18-22, pp. 4461-4467.
- [14] —, "Multi robot trajectory generation for single source explosion parameter estimation," in *Proceedings of the IEEE International Conference on Robotics and Automation*, Barcelona, Spain, Apr. 18-22 2005, pp. 2803-2809.
- [15] V. Isler and R. Bajcsy, "The sensor selection problem for bounded uncertainty sensing models," *IEEE Transactions on Automation Science and Engineering*, vol. 3, no. 4, pp. 372-381, Oct. 2006.
- [16] J. P. Le Cadre, "Optimization of the observer motion for bearings-only target motion analysis," in *Proceedings of the 36th IEEE Conference on Decision and Control*, San Diego, CA, Dec. 10-12 1997, pp. 3126-3131.
- [17] J. M. Passerieux and D. Van Cappel, "Optimal observer maneuver for bearings-only tracking," *IEEE Transactions on Aerospace and Electronic Systems*, vol. 34, no. 3, pp. 777-788, July 1998.
- [18] A. Logothetis, A. Isaksson, and R. J. Evans, "Comparison of suboptimal strategies for optimal own-ship maneuvers in bearings-only tracking," in *Proceedings of the American Control Conference*, Philadelphia, PA, June 24-26 1998, pp. 3334-3338.
- [19] E. W. Frew, "Trajectory design for target motion estimation using monocular vision," Ph.D. dissertation, Stanford University, Stanford, CA, Aug. 2003.
- [20] J. R. Spletzer and C. J. Taylor, "Dynamic sensor planning and control for optimally tracking targets," *The International Journal of Robotics Research*, vol. 22, no. 1, pp. 7-20, 2003.
- [21] A. W. Stroupe and T. Balch, "Value-based action selection for observation with robot teams using probabilistic techniques," *Robotics and Autonomous Systems*, vol. 50, no. 2-3, pp. 85-97, Feb. 2005.
- [22] R. Olfati-Saber, "Distributed tracking for mobile sensor networks with information-driven mobility," in *Proceedings of the American Control Conference*, New York, NY, July 11-13 2007, pp. 4606-4612.
- [23] T. H. Chung, J. W. Burdick, and R. M. Murray, "A decentralized motion coordination strategy for dynamic target tracking," in *Proceedings of the IEEE International Conference on Robotics and Automation*, Orlando, FL, May 15-19 2006, pp. 2416-2422.
- [24] P. Yang, R. A. Freeman, and K. M. Lynch, "Distributed cooperative active sensing using consensus filters," in *Proceedings of the IEEE International Conference on Robotics and Automation*, Rome, Italy, Apr. 10-14 2007, pp. 405-410.
- [25] S. Martínez and F. Bullo, "Optimal sensor placement and motion coordination for target tracking," *Automatica*, vol. 42, no. 4, pp. 661-668, Apr. 2006.
- [26] Y. Bar-Shalom, X. R. Li, and T. Kirubarajan, *Estimation with Applications to Tracking and Navigation*. New York, NY: Wiley-Interscience, John Wiley & Sons Inc., June 2001.
- [27] R. A. Horn and C. R. Johnson, *Matrix Analysis*. New York, NY: Cambridge University Press, Feb. 1990.
- [28] M. R. Garey and D. S. Johnson, *Computers and Intractability: A Guide to the Theory of NP-Completeness*, ser. Books in the Mathematical Sciences. San Francisco, CA: W. H. Freeman, Jan. 1979.
- [29] D. P. Bertsekas and J. N. Tsitsiklis, *Parallel and Distributed Computation: Numerical Methods*. Belmont, MA: Athena Scientific, Sept. 1997.
- [30] S. Boyd and L. Vandenberghe, *Convex Optimization*. New York, NY: Cambridge University Press, Mar. 2004.
- [31] "Matlab documentations," Online, <http://www.mathworks.com/>.
- [32] D. P. Bertsekas, *Nonlinear Programming*, 2nd ed. Belmont, MA: Athena Scientific, Sept. 1999.
- [33] S. L. Laubach and J. W. Burdick, "An autonomous sensor-based path-planner for planetary microrovers," in *Proceedings of the IEEE International Conference on Robotics and Automation*, Detroit, MI, May 10-15 1999, pp. 347-354.
- [34] A. Ribeiro, G. B. Giannakis, and S. I. Roumeliotis, "SOI-KF: Distributed kalman filtering with low-cost communications using the sign of innovations," *IEEE Transactions on Signal Processing*, vol. 54, no. 12, pp. 4782-4795, Dec. 2006.
- [35] E. J. Msechu, S. I. Roumeliotis, A. Ribeiro, and G. B. Giannakis, "Distributed quantized kalman filtering with scalable communication cost," *IEEE Transactions on Signal Processing*, vol. 56, no. 8, pp. 3727-3741, Aug. 2008.
- [36] I. Schizas, G. B. Giannakis, S. I. Roumeliotis, and A. Ribeiro, "Consensus in ad hoc wsns with noisy links - part ii: Distributed estimation and smoothing of random signals," *IEEE Transactions on Signal Processing*, vol. 56, no. 4, pp. 1650-1666, Apr. 2008.
- [37] C. Roos, T. Terlaky, and J. P. Vial, *Interior Point Methods for Linear Optimization*, 2nd ed. New York, NY: Springer, Sept. 2005.
- [38] G. H. Golub and C. F. Van Loan, *Matrix Computations*, 3rd ed., ser. Johns Hopkins Studies in Mathematical Sciences. Baltimore, MD: The Johns Hopkins University Press, Oct. 1996.