Hindawi Publishing Corporation
EURASIP Journal on Advances in Signal Processing
Volume 2008, Article ID 360912, 10 pages
doi:10.1155/2008/360912

Research Article

Tracking Objects with Networked Scattered Directional Sensors

Kurt Plarre[1] and P. R. Kumar[2]

[1] *Department of Mechanical Engineering and Center for Control, Dynamical Systems and Computation, University of California, Santa Barbara, CA 93106, USA*
[2] *Department of Electrical and Computer Engineering and Coordinated Science Laboratory, University of Illinois at Urbana-Champaign, 1308 W. Main St., Urbana, IL 61801, USA*

Correspondence should be addressed to P. R. Kumar, plarre@engineering.ucsb.edu

Received 19 April 2007; Accepted 4 August 2007

Recommended by Damien B. Jourdan

We study the problem of object tracking using highly directional sensors—sensors whose field of vision is a line or a line segment. A network of such sensors monitors a certain region of the plane. Sporadically, objects moving in straight lines and at a constant speed cross the region. A sensor detects an object when it crosses its line of sight, and records the time of the detection. No distance or angle measurements are available. The task of the sensors is to estimate the directions and speeds of the objects, and the sensor lines, which are unknown a priori. This estimation problem involves the minimization of a highly nonconvex cost function. To overcome this difficulty, we introduce an algorithm, which we call "adaptive basis algorithm." This algorithm is divided into three phases: in the first phase, the algorithm is initialized using data from six sensors and four objects; in the second phase, the estimates are updated as data from more sensors and objects are incorporated. The third phase is an optional coordinated transformation. The estimation is done in an "ad-hoc" coordinate system, which we call "adaptive coordinate system." When more information is available, for example, the location of six sensors, the estimates can be transformed to the "real-world" coordinate system. This constitutes the third phase.

1. INTRODUCTION

One of the most widely envisaged applications of sensor networks is surveillance. A sensor network can be used to monitor a certain region, and determine the presence, number, identity, and behavior of objects in that region. Thus, surveillance applications must be able to detect, classify, and track a target [1]. Examples of surveillance applications include wildlife monitoring, heat source detection, water quality monitoring, gas or nuclear plume detection and tracking, security, defense, and so forth.

Sensors can be classified according to different criteria. Here we classify them as omnidirectional or directional. An omnidirectional sensor can detect its environment equally in any direction, while a directional sensor can measure only in a given "field of vision," that is, the sensing area is a sector rather than a disk. The two types of sensors pose different problems and require different solutions.

In this paper, we consider the problem of tracking objects using highly directional sensors, that is, sensors whose field of vision is a very narrow sector or a line. Sensors that fall into this class are lasers and highly directional infrared temperature sensors. Figure 1 compares the possible field of vision of omnidirectional, directional, and highly directional sensors. Although the ideas introduced in this paper are applicable to highly directional sensors, in Section 8 we discuss how they can be extended to other types of sensors.

Target tracking in sensor networks has received much attention in the literature; see, for example, [1–13]. The use of information provided by detected objects to improve the accuracy of sensor localization schemes has also been proposed, although in a different context. In [14], connectivity information and information provided by objects detected by omnidirectional sensors are used to determine, for each sensor, a region in which it is located.

In this paper, we treat the problem of estimating the trajectory of objects moving in straight lines using highly

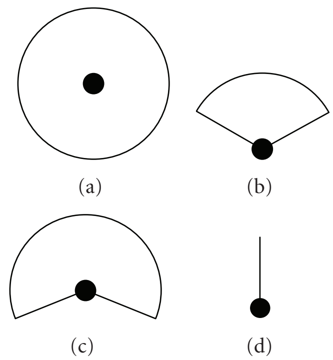


Figure 1: Sensor types: omnidirectional (a), directional (b) and (c), and highly directional (d).

directional sensors. A network of highly directional sensors monitors a region of the plane. The location of the sensors and the directions of their fields of vision are unknown a priori. Sporadically, objects moving in straight lines and constant speed cross the region. We assume that only one object is in the region at any given time. We are not concerned with identity management.

A sensor detects an object when it crosses its field of vision. Sensors cannot measure distance or angle. The only information available to the sensors are the detection times. The estimation of the trajectories and the sensor lines must be done from this time information only. This estimation problem involves the minimization of a highly nonconvex cost function, as is often the case in many such inference problems.

To find the global minimum of such cost function we introduce an algorithm, which we call an "adaptive basis algorithm." This algorithm is divided into three phases. In the first phase, the algorithm is initialized using the detection times of four objects and six sensors. The algorithm estimates the directions and speeds of the four objects, and the sensing lines of the six sensors in an "ad Hoc" coordinate system, which we call "adaptive coordinate system." The reason for this name will become clear in the sequel. In the second phase, the estimates are updated, as new data is collected from new sensors or objects. The third phase is an optional coordinate transformation, to obtain the estimates in the real-world coordinate system.

In the next section, we give an overview of the problem we study, and in Section 3 we give the formal setup. Section 4 contains the main ideas behind the adaptive basis algorithm, while Section 5 describes the algorithm in detail. We provide the results of simulations in Section 6, and an implementation in Section 7. Finally Section 8 contains conclusions and comments.

2. OVERVIEW OF TRAJECTORY ESTIMATION PROBLEM USING DIRECTIONAL SENSORS

A certain region is monitored by a network of directional sensors whose positions and orientations themselves are initially unknown. The region is crossed sporadically by objects assumed to be moving at constant velocity, at least within a bounded domain of interest. We assume that only one object is in the region at a time. There is no need to keep track of the identity of the objects. The task of the network is to detect each object and determine its motion trajectory.

The algorithm developed in this paper uses minimal information, namely, only the detection times of the objects. No distance or angle measurements are needed. We will consider the extreme situation where nothing is known a priori: even the locations of the sensors and the directions at which the sensors point are unknown a priori. The sensor directions are also to be estimated as part of the problem. The central issue is how to estimate both trajectories and sensor lines from time measurements only.

We model objects as points, and the "line of sight" of each sensor simply as a straightline. A sensor detects an object when it crosses its line of sight. Thus the data and input to the algorithm are the object detection times. Such a system requires a clock synchronization algorithm, and in our system the algorithm developed in [15] was used.

A detailed description of the setup for this application is given in Section 3.

In Section 7, we show an implementation of this scenario using lasers. Lasers are pointed at motes equipped with light sensors which detect the presence of a passing vehicle. Detection times are used to estimate the speed and direction of the car, as well as the straightlines formed by the laser beams.

The estimation of the trajectories as well as the sensor lines involves the minimization of a nonconvex cost function. This cost function presents a large number of local minima. We need to find the global minimum of this cost in order to accurately estimate the parameters. In Section 5, we present an algorithm to do so.

Equation (6) in the sequel, which shows the cost for just three objects and two sensors, clearly illustrates the difficulty of this problem. We are, however, able to exploit the specific structure of this problem to solve it. The algorithm can be divided into three phases.

(1) In phase 1, an initial solution is found using the detection times of the first four objects and six sensors (see Section 4). It is surprising that this problem can be solved in closed form. For this, we first need to find an adequate coordinate system in which to express the geometric relationships of the objects and sensors. We call this an "adaptive basis." The key to our solution is that when expressed in the adaptive basis, this initial problem can be solved in closed form. Any other fixed coordinate system does not have such a property.

(2) In phase 2, as new objects arrive, the parameters of the new objects are estimated, and all other earlier parameters are updated. Similarly, if more than six sensors are available, their observed crossing times can be incorporated progressively into the algorithm.

(3) Phase 3 is optional, and involves a coordinate transformation to obtain the parameter estimates in the real-world coordinate system. For this, additional information, such as the location of six sensors or the trajectories of two objects in the desired real-world coordinate system is needed.

In the next section, we give the formal setup of the problem.

3. PROBLEM SETUP

Let us suppose that the equation of the line of sight of sensor s_i is

$$\frac{x_{s_i}}{a_{s_i}} + \frac{y_{s_i}}{b_{s_i}} = 1 \qquad \text{or} \qquad \overline{a}_{s_i} x_{s_i} + \overline{b}_{s_i} y_{s_i} = 1, \tag{1}$$

where a_{s_i} and b_{s_i} are the intercepts of the sensing line of s_i with the horizontal and vertical axis, respectively. Also suppose that the motion of object o_j is described by the following equations, where t denotes time:

$$\begin{aligned} x_{o_j}(t) &= v_{o_j}^x t + x_{o_j}^0, \\ y_{o_j}(t) &= v_{o_j}^y t + y_{o_j}^0. \end{aligned} \tag{2}$$

Here, $v_{o_j}^x$ and $v_{o_j}^y$ are the horizontal and vertical speeds of o_j, respectively, and $(x_{o_j}^0, y_{o_j}^0)$ is the location of o_j at time zero.

The parameters (a_{s_i}, b_{s_i}) for the sensors s_i, and the parameters $(v_{o_j}^x, x_{o_j}^0, v_{o_j}^y, y_{o_j}^0)$ for the various objects o_j are all unknown a priori, and it is desired to estimate them.

The time at which sensor s_i detects object o_j is then

$$t_{s_i}^{o_j} = \frac{1 - \overline{a}_{s_i} x_{o_j}^0 - \overline{b}_{s_i} y_{o_j}^0}{\overline{a}_{s_i} v_{o_j}^x + \overline{b}_{s_i} v_{o_j}^y} + \nu_{s_i}^{o_j}, \tag{3}$$

where we assume that $\nu_{s_i}^{o_j}$ is zero-mean noise, and $\nu_{s_i}^{o_j}$ is independent of $\nu_{s_k}^{o_j}$ for $(s_i, o_j) \neq (s_k, o_l)$.

Corresponding to $t_{s_i}^{o_j}$ associate the "equation error,"

$$\tau_{s_i}^{o_j} := (\overline{a}_{s_i} v_{o_j}^x + \overline{b}_{s_i} v_{o_j}^y) t_{s_i}^{o_j} + (\overline{a}_{s_i} x_{o_j}^0 + \overline{b}_{s_i} y_{o_j}^0) - 1. \tag{4}$$

The estimation of the object motion and sensor direction parameters will be based on the minimization of the cost function that is the sum of the squares of the errors:

$$J = \sum_{i,j} (\tau_{s_i}^{o_j})^2, \tag{5}$$

over the parameters (a, b) of the sensors and (v^x, x^0, v^y, y^0). For simplicity, the arguments of J, which are all the unknown parameters, are not shown explicitly.

To see the difficulty of minimizing (5), we detail the expanded form of J, for just three sensors and two objects:

$$\begin{aligned} J = {} & [(\overline{a}_{s_1} v_{o_1}^x + \overline{b}_{s_1} v_{o_1}^y) t_{s_1}^{o_1} + (\overline{a}_{s_1} x_{o_1}^0 + \overline{b}_{s_1} y_{o_1}^0) - 1]^2 \\ & + [(\overline{a}_{s_1} v_{o_2}^x + \overline{b}_{s_1} v_{o_2}^y) t_{s_1}^{o_2} + (\overline{a}_{s_1} x_{o_2}^0 + \overline{b}_{s_1} y_{o_2}^0) - 1]^2 \\ & + [(\overline{a}_{s_2} v_{o_1}^x + \overline{b}_{s_2} v_{o_1}^y) t_{s_2}^{o_1} + (\overline{a}_{s_2} x_{o_1}^0 + \overline{b}_{s_2} y_{o_1}^0) - 1]^2 \\ & + [(\overline{a}_{s_2} v_{o_2}^x + \overline{b}_{s_2} v_{o_2}^y) t_{s_2}^{o_2} + (\overline{a}_{s_2} x_{o_2}^0 + \overline{b}_{s_2} y_{o_2}^0) - 1]^2 \\ & + [(\overline{a}_{s_3} v_{o_1}^x + \overline{b}_{s_3} v_{o_1}^y) t_{s_3}^{o_1} + (\overline{a}_{s_3} x_{o_1}^0 + \overline{b}_{s_3} y_{o_1}^0) - 1]^2 \\ & + [(\overline{a}_{s_3} v_{o_2}^x + \overline{b}_{s_3} v_{o_2}^y) t_{s_3}^{o_2} + (\overline{a}_{s_3} x_{o_2}^0 + \overline{b}_{s_3} y_{o_2}^0) - 1]^2. \end{aligned} \tag{6}$$

Note that (5) is a nonconvex function of

$$\{(\overline{a}_{s_i}, \overline{b}_{s_i}, v_{o_j}^x, x_{o_j}^0, v_{o_j}^y, y_{o_j}^0);\ 1 < i < 3,\ 1 < j < 2\}, \tag{7}$$

the sensor and object parameters. Note also that even for just four objects and six sensors, the number of unknown parameters is $4 \times 4 + 6 \times 2 = 28$. Only the global minimum is an acceptable solution, not local minima, and only an exhaustive search could ensure that one finds it; but such a search would be too computationally expensive.

We will develop a recursive algorithm by which the data provided by four objects and six sensors is used to determine an initial solution. The data provided by other sensors and objects is subsequently recursively incorporated into the algorithm, thus improving the accuracy of the solution.

To determine the minimum of (5), we devise a novel two-phase algorithm, with an optional third phase that corresponds to the final coordinate transformation.

It is important to mention that there are certain "degenerate" cases that cannot be handled by the algorithm. For example, if the first two objects travel in the same direction, or all sensors lines are parallel. We assume that such cases will not happen in practice (or have a small probability of happening), and do not consider them.

4. THE MAIN IDEAS

The central issue is how to circumvent the problem of finding the global minimum of the nonconvex cost function (5). Our key idea to overcome this is to choose an adaptive basis, which can be optionally transformed at a later phase. We note that since we do not know the real-world coordinate system, we must choose a "custom" system in which to state the equations and thus localize the sensor rotations and the motions of the objects. Later on, we will use the locations of six sensors, if known, to transform the so-obtained parameters to the correct representation. This can be done at any point of the algorithm.

Since we are free to choose our coordinate system, we will choose it in such a way that it simplifies the expressions. In fact, if the coordinate system is not carefully chosen, the resulting equations cannot be solved in closed form. We thus have the task of finding the right coordinate system in which to write the equations, and then finding a procedure to solve them.

We choose the adaptive coordinate system in the following way.

(1) The motion of the first object is used to fix the "horizontal" axis, with its position at time $t = 0$ defined as the origin, and speed normalized to 1. As will be shown in Section 5, this fixes all parameters of o_1 in the custom system.

(2) The motion of the second object is used to fix the "vertical" axis, with its speed also normalized to 1. However, since its position at time $t = 0$ is unknown, two parameters corresponding to o_2, its two coordinates at time $t = 0$, will be undetermined (as detailed in Section 5).

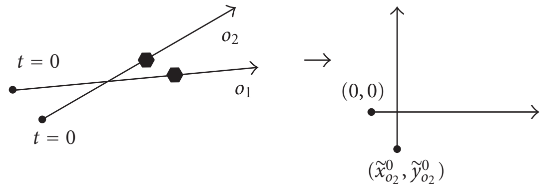


Figure 2: Adaptive coordinate system obtained from the trajectories of the first two objects.

We then divide the process into two-phases. In the first phase, we use the data obtained from only m sensors and n objects, where m and n are chosen in such a way that (5) can be set exactly to zero, independent of the noise. Solving the resulting equation provides an initial estimate of the parameters. In the second phase, as new data are incorporated into the problem, the sensor and object parameter estimates are refined, using a local improvement algorithm.

To determine the number of sensors and object measurements needed to determine the initial estimates, that is, n and m, we reason in the following way.

(1) Each remaining object o_j used in the first phase will add four unknown parameters to the problem: $v^x_{o_j}$, $x^0_{o_j}$, $v^y_{o_j}$, and $y^0_{o_j}$.

(2) Each sensor s_i included in this phase will add two unknown parameters to define its "line."

(3) On the other hand, the number of data measurements obtained from the detection of the first n objects by m sensors is nm.

Considering that we need at least the same number of data variables as the number of unknown parameters to solve the equations, we need

$$nm \geq 4(n-2) + 2 + 2m, \tag{8}$$

which is satisfied by $m = 6$, and $n = 3$. We thus need at least six sensors and three objects to initialize the system. However, we will see in Section 5.1 that the resulting equation is quadratic, and we will need the data from a fourth object to resolve the sign of the root.

5. THE ALGORITHM

In this section, we present the estimation algorithm.

5.1. First phase

During the first phase, after deployment, all sensors are awake, waiting for the first four objects. The data collected from these objects is used to form an initial estimate of the object and sensor parameters. As mentioned before, the first object is used to fix the "horizontal" axis of the adaptive coordinate system (see Figure 2). The point on the plane at which o_1 was at time $t = 0$ represents the origin of the coordinate system. The direction of motion determines the axis, and the scale is given by assuming that the speed of o_1 is 1.

The second object fixes the vertical axis (see Figure 2). The direction of motion of o_2 determines the axis, while the scale is given by assuming that its speed is 1. The point at which o_2 is at time $t = 0$ is unknown. We call this point $(\widetilde{x}^0_{o_2}, \widetilde{y}^0_{o_2})$. These two parameters $\widetilde{x}^0_{o_2}$ and $\widetilde{y}^0_{o_2}$ are unknown even with respect to the adaptive basis and must be estimated as part of the problem.

In our coordinate system, we know that the line corresponding to sensor s_i passes through the points $(t^{o_1}_{s_i}, 0)$ and $(\widetilde{x}^0_{o_2}, \widetilde{y}^0_{o_2} + t^{o_2}_{s_i})$. Thus, the equation for s_i in this system is determined as

$$\frac{\widetilde{y}_{s_i}}{\widetilde{x}_{s_i} - t^{o_1}_{s_i}} = \frac{\widetilde{y}^0_{o_2} + t^{o_2}_{s_i}}{\widetilde{x}^0_{o_2} - t^{o_1}_{s_i}}. \tag{9}$$

Hence, subject only to $(\widetilde{x}^0_{o_2}, \widetilde{y}^0_{o_2})$ being unknown, each sensor's line is determined.

Now we turn to the second object. Reordering (9), we obtain

$$(\widetilde{x}^0_{o_2} - t^{o_1}_{s_i})\widetilde{y}_{s_i} = (\widetilde{y}^0_{o_2} + t^{o_2}_{s_i})\widetilde{x}_{s_i} - t^{o_2}_{s_i}\widetilde{y}^0_{o_2} - t^{o_1}_{s_i}t^{o_2}_{s_i}. \tag{10}$$

Consider now the third object o_3. Assume that the equation for o_3 in our coordinate system is

$$\widetilde{x}_{o_3}(t) = \widetilde{v}^x_{o_3}t + \widetilde{x}^0_{o_3}, \qquad \widetilde{y}_{o_3}(t) = \widetilde{v}^y_{o_3}t + \widetilde{y}^0_{o_3}. \tag{11}$$

We know o_3 is detected by sensor s_i at time $t^{o_3}_{s_i}$. Combining this information with (10), we obtain

$$\begin{aligned}(\widetilde{x}^0_{o_2} - t^{o_1}_{s_i})&(\widetilde{y}^0_{o_3} + \widetilde{v}^y_{o_3}t^{o_3}_{s_i})\\ &= (\widetilde{y}^0_{o_2} + t^{o_2}_{s_i})(\widetilde{x}^0_{o_3} + \widetilde{v}^x_{o_3}t^{o_3}_{s_i}) - t^{o_1}_{s_i}\widetilde{y}^0_{o_2} - t^{o_1}_{s_i}t^{o_2}_{s_i}.\end{aligned} \tag{12}$$

Let $\mathbf{M}$ be a matrix such that its ith row is

$$\begin{aligned}[\mathbf{M}]_{i,*} := [&\widetilde{x}^0_{o_2} - t^{o_1}_{s_i},\ t^{o_3}_{s_i}(\widetilde{x}^0_{o_2} - t^{o_2}_{s_i}),\ -(\widetilde{y}^0_{o_2} + t^{o_2}_{s_i}),\\ &-t^{o_3}_{s_i}(\widetilde{y}^0_{o_2} + t^{o_2}_{s_i}),\ (t^{o_1}_{s_i}\widetilde{y}^0_{o_2} + t^{o_1}_{s_i}t^{o_2}_{s_i})].\end{aligned} \tag{13}$$

Likewise, let $\mathbf{v} := [\widetilde{y}^0_{o_3}, \widetilde{v}^y_{o_3}, \widetilde{x}^0_{o_3}, \widetilde{v}^x_{o_3}, 1]^T$. Then, from (12), we can write the linear system as $\mathbf{Mv} = 0$. If $\mathbf{M}$ was not column-rank deficient, then the unique solution to this system would be $\mathbf{v} = (\mathbf{M}^T\mathbf{M})^{-1}0 = 0$. However, since this system has a nontrivial solution, $\mathbf{M}$ is column-rank deficient. Let us rewrite $\mathbf{M}$ in term of its columns. For this, let us first define the following:

$$\begin{aligned}\bar{e} &:= [1, 1, \ldots, 1]^T,\\ \overline{T}_{o_1} &:= [t^{o_1}_{s_1}, t^{o_1}_{s_2}, \ldots, t^{o_1}_{s_m}]^T,\\ \overline{T}_{o_2} &:= [t^{o_2}_{s_1}, t^{o_2}_{s_2}, \ldots, t^{o_2}_{s_m}]^T,\\ \overline{T}_{o_3} &:= [t^{o_3}_{s_1}, t^{o_3}_{s_2}, \ldots, t^{o_3}_{s_m}]^T,\\ \overline{T}^{o_2}_{o_1} &:= [t^{o_1}_{s_1}t^{o_2}_{s_1}, t^{o_1}_{s_2}t^{o_2}_{s_2}, \ldots, t^{o_1}_{s_m}t^{o_2}_{s_m}]^T,\\ \overline{T}^{o_3}_{o_1} &:= [t^{o_1}_{s_1}t^{o_3}_{s_1}, t^{o_1}_{s_2}t^{o_3}_{s_2}, \ldots, t^{o_1}_{s_m}t^{o_3}_{s_m}]^T,\\ \overline{T}^{o_3}_{o_2} &:= [t^{o_2}_{s_1}t^{o_3}_{s_1}, t^{o_2}_{s_2}t^{o_3}_{s_2}, \ldots, t^{o_2}_{s_m}t^{o_3}_{s_m}]^T.\end{aligned} \tag{14}$$

With these definitions we can write $\mathbf{M}$ as

$$\begin{aligned}\mathbf{M} = [\, & \widetilde{x}^0_{o_2}\,\overline{e} - \overline{T}_{o_1}, \widetilde{x}^0_{o_2}\overline{T}_{o_3} - \overline{T}^{o_3}_{o_1}, -\widetilde{y}^0_{o_2}\overline{e} - \overline{T}_{o_2}, \\ & -\widetilde{y}^0_{o_2}\overline{T}_{o_3} - \overline{T}^{o_3}_{o_2}, \widetilde{y}^0_{o_2}\overline{T}_{o_1} + \overline{T}^{o_2}_{o_1}].\end{aligned} \tag{15}$$

Since $\mathbf{M}$ is column-rank deficient, there exist real numbers $\alpha_1, \alpha_2, \alpha_3, \alpha_4, \alpha_5$, such that

$$\begin{aligned}\alpha_1\left(\widetilde{x}^0_{o_2}\overline{e} - \overline{T}_{o_1}\right) + \alpha_2\left(\widetilde{x}^0_{o_2}\overline{T}_{o_3} - \overline{T}^{o_3}_{o_1}\right) + \alpha_3\left(-\widetilde{y}^0_{o_2}\overline{e} - \overline{T}_{o_2}\right) \\ + \alpha_4\left(-\widetilde{y}^0_{o_2}\overline{T}_{o_3} - \overline{T}^{o_3}_{o_2}\right) + \alpha_5\left(\widetilde{y}^0_{o_2}\overline{T}_{o_1} + \overline{T}^{o_2}_{o_1}\right) = 0.\end{aligned} \tag{16}$$

Collecting terms, and defining

$$\begin{aligned}\overline{\mathbf{M}} &:= [\overline{e}, \overline{T}_{o_1}, \overline{T}_{o_3}, \overline{T}_{o_2}, \overline{T}^{o_3}_{o_1}, \overline{T}^{o_3}_{o_2}, \overline{T}^{o_2}_{o_1}], \\ \overline{\mathbf{v}} &:= [\alpha_1\widetilde{x}^0_{o_2} - \alpha_3\widetilde{y}^0_{o_2}, \alpha_5\widetilde{y}^0_{o_2} - \alpha_1, \alpha_2\widetilde{x}^0_{o_2} - \alpha_4\widetilde{y}^0_{o_2}, \\ &\quad -\alpha_3, -\alpha_2, -\alpha_2, -\alpha_4, \alpha_5]^T,\end{aligned} \tag{17}$$

we can rewrite (16) as

$$\overline{\mathbf{M}}\overline{\mathbf{v}} = 0. \tag{18}$$

Let $[\theta_1, \theta_2, \theta_3, \theta_4, \theta_5, \theta_6]^T$ be the solution to (18), with $\alpha_5 = 1$. Then

$$\begin{aligned}\alpha_1\widetilde{x}^0_{o_2} - \alpha_3\widetilde{y}^0_{o_2} &= \theta_1, \\ \alpha_5\widetilde{y}^0_{o_2} - \alpha_1 &= \theta_2, \\ \alpha_2\widetilde{x}^0_{o_2} - \alpha_4\widetilde{y}^0_{o_2} &= \theta_3, \\ -\alpha_3 &= \theta_4, \\ -\alpha_2 &= \theta_5, \\ -\alpha_4 &= \theta_6, \\ \alpha_5 &= 1.\end{aligned} \tag{19}$$

Solving this nonlinear system, one obtains

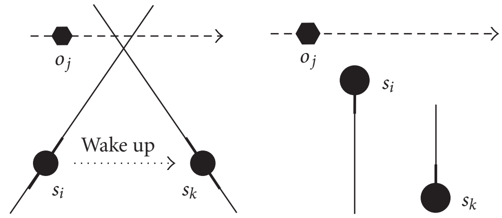
$$\begin{aligned}\widetilde{x}^0_{o_2} = {} & \frac{\alpha_5\theta_3 + \alpha_4\theta_2 + \alpha_2\alpha_3}{2\alpha_2\alpha_5} \\ & \pm \frac{\sqrt{(\alpha_5\theta_3 + \alpha_4\theta_2 + \alpha_2\alpha_3)^2 + 4\alpha_2\alpha_5(\alpha_4\theta_1 - \alpha_3\theta_3)}}{2\alpha_2\alpha_5}.\end{aligned} \tag{20}$$

To resolve the sign in (20) we make use of the data provided by the fourth object o_4. We simply choose the sign that conforms to the detection times $t^{o_4}_{s_i}$.

Once the value of $\widetilde{x}^0_{o_2}$ is known, the rest of the parameters can be easily computed.

5.2. *Second phase*

Once the parameters for the first four objects and six sensors have been estimated, most sensors go to sleep. A few sentinel sensors stay awake and sensing. When a sentinel sensor detects an object, it wakes up the complete sensor network. All sensors then wait for the object and register the time at which they detect it. It is important to note that some sensors will not detect a given object, since they may wake up too late. This is illustrated in Figure 3.



(a) (b)

Figure 3: Some objects are not detected by all sensors: (a) s_k wakes up too late to detect o_j, (b) s_i only covers a half-line, while s_k has a limited range.

	s_1	s_2	s_3	s_4	s_5	s_6	s_7	s_8	s_9
o_1	1	1	1	1	1	1	1		
o_2	1	1	1	1	1	1			1
o_3	1	1	1	1	1	1	1		
o_4	1	1	1	1	1	1			
o_5		1		1		1	1		1
o_6					1	1		1	

Figure 4: Example of a matrix Ω^{s_i} indicating the measurements known to s_i.

Each sensor has at most one detection time for the new object. To form an estimate of the trajectory of this object, at least four measurements are necessary. To gather this information, sensors share their measurements (if they have any), and collect measurements from other nodes. The obtained data are used to refine the estimates of all parameters.

To organize the computations, for each node s_i, we define a matrix Ω^{s_i}, such that

$$\Omega^{s_i}_{k,l} := \begin{cases} 1 & \text{if } s_i \text{ knows } t^{o_l}_{s_k}, \\ 0 & \text{otherwise}. \end{cases} \tag{21}$$

An example matrix Ω^{s_i} is shown in Figure 4.

For each s_k, and object o_l, let $\mathcal{O}^{s_i}_{s_k}$ and $\mathcal{S}^{s_i}_{o_l}$ be defined as

$$\mathcal{O}^{s_i}_{s_k} := \{l \mid \Omega^{s_i}_{k,l} = 1\}, \qquad \mathcal{S}^{s_i}_{o_l} := \{k \mid \Omega^{s_i}_{k,l} = 1\}. \tag{22}$$

The cost corresponding to sensor s_i is then given by

$$J_{s_i} = \sum_k \sum_{l \in \mathcal{O}^{s_i}_{s_k}} \left[\overline{a}^{s_i}_{s_k} v^{x,s_i}_{o_l} t^{o_l}_{s_k} + \overline{a}^{s_i}_{s_k} x^{0,s_i}_{o_l} + \overline{b}^{s_i}_{s_k} v^{y,s_i}_{o_l} t^{o_l}_{s_k} + \overline{b}^{s_i}_{s_k} y^{0,s_i}_{o_l} - 1\right]^2, \tag{23}$$

where $\overline{a}^{s_i}_{s_k}$, $\overline{b}^{s_i}_{s_k}$, $v^{x,s_i}_{o_l}$, $x^{0,s_i}_{o_l}$, $v^{y,s_i}_{o_l}$, and $y^{0,s_i}_{o_l}$ are the estimated parameters at s_i. We use a block coordinate descent method (see [16]) to minimize J_{s_i}. Sensor s_i performs one phase of Newton's algorithm for each row and column of Ω^{s_i}s for which there is enough data. This is done cyclically.

Let us first define

$$
\begin{aligned}
&A^{s_i}_{s_k,o_l} := v^{x,s_i}_{o_l} t^{o_l}_{s_k} + x^{0,s_i}_{o_l},\\
&B^{s_i}_{s_k,o_l} := v^{y,s_i}_{o_l} t^{o_l}_{s_k} + y^{0,s_i}_{o_l},\\
&C^{s_i}_{o_k,o_l} := \overline{a}^{s_i}_{s_k},\\
&D^{s_i}_{o_k,o_l} := \overline{a}^{s_i}_{s_k} t^{o_l}_{s_k},\\
&E^{s_i}_{o_k,o_l} := \overline{b}^{s_i}_{s_k},\\
&F^{s_i}_{o_k,o_l} := \overline{b}^{s_i}_{s_k} t^{o_l}_{s_k},\\
&J_{s_i} = \sum_k \sum_{l \in \mathcal{O}^{s_i}_{s_k}} \left[A^{s_i}_{s_k,o_l}\overline{a}^{s_i}_{s_k} + B^{s_i}_{s_k,o_l}\overline{b}^{s_i}_{s_k} - 1\right]^2\\
&\quad = \sum_l \sum_{k \in \mathcal{S}^{s_i}_{o_l}} \left[D^{s_i}_{o_k,o_l} v^{x,s_i}_{o_l} + C^{s_i}_{o_k,o_l} x^{0,s_i}_{o_l} + F^{s_i}_{o_k,o_l} v^{y,s_i}_{o_l}\right.\\
&\qquad\qquad \left. + E^{s_i}_{o_k,o_l} y^{0,s_i}_{o_l} - 1\right]^2.
\end{aligned}
\tag{24}
$$

To simplify the expressions, let us also define $\mathbf{v}^{s_i}_{s_k} := [\overline{a}^{s_i}_{s_k}, \overline{b}^{s_i}_{s_k}]^T$,

$$
\begin{aligned}
g^{s_i}_{s_k} &:= \begin{bmatrix} \sum_{l \in \mathcal{O}^{s_i}_{s_k}} \left(A^{s_i}_{s_k,o_l}\overline{a}^{s_i}_{s_k} + B^{s_i}_{s_k,o_l}\overline{b}^{s_i}_{s_k} - 1\right) A^{s_i}_{s_k,o_l} \\ \sum_{l \in \mathcal{O}^{s_i}_{s_k}} \left(A^{s_i}_{s_k,o_l}\overline{a}^{s_i}_{s_k} + B^{s_i}_{s_k,o_l}\overline{b}^{s_i}_{s_k} - 1\right) B^{s_i}_{s_k,o_l} \end{bmatrix},\\
\mathbf{H}^{s_i}_{s_k} &:= \begin{bmatrix} \sum_{l \in \mathcal{O}^{s_i}_{s_k}} \left(A^{s_i}_{s_k,o_l}\right)^2 & \sum_{l \in \mathcal{O}^{s_i}_{s_k}} A^{s_i}_{s_k,o_l} B^{s_i}_{s_k,o_l} \\ \sum_{l \in \mathcal{O}^{s_i}_{s_k}} B^{s_i}_{s_k,o_l} A^{s_i}_{s_k,o_l} & \sum_{l \in \mathcal{O}^{s_i}_{s_k}} \left(B^{s_i}_{s_k,o_l}\right)^2 \end{bmatrix}.
\end{aligned}
\tag{25}
$$

Applying Newton's method to (23) with respect to $\overline{a}_{s_k}$ and $\overline{b}_{s_k}$, we obtain the recursion

$$
\mathbf{v}^{s_i}_{s_k} \mathbf{v}^{s_i}_{s_k} - \left(\mathbf{H}^{s_i}_{s_k}\right)^{-1} g^{s_i}_{s_k}. \tag{26}
$$

Similar expressions are obtained by Newton's method applied to (23), with respect to $v^{x,s_i}_{o_l}$, $x^{0,s_i}_{o_l}$, $v^{y,s_i}_{o_l}$, and $y^{0,s_i}_{o_l}$.

5.3. Third phase: coordinate transformation

Once the parameters of the sensors and objects have been estimated in the adaptive coordinate system, they can be transformed into the real-world system if the locations of six sensors are known. The linear coordinate transformation can be represented as

$$
\begin{bmatrix} x^{\text{adaptive}} \\ y^{\text{adaptive}} \end{bmatrix} = \begin{bmatrix} A_{1,1} & A_{1,2} \\ A_{2,1} & A_{2,2} \end{bmatrix} \begin{bmatrix} x^{\text{real}} \\ y^{\text{real}} \end{bmatrix} + \begin{bmatrix} d_x \\ d_y \end{bmatrix}. \tag{27}
$$

Let us assume, without loss of generality, that we know the locations of sensors s_1 to s_6. In the adaptive system, each sensor satisfies the equation corresponding to its line of sight. We can thus write

$$
\frac{x^{\text{adaptive}}_{s_i}}{a_{s_i}} + \frac{y^{\text{adaptive}}_{s_i}}{b_{s_i}} = 1 \tag{28}
$$

for $i = 1, 2, \ldots, 6$, or

$$
\frac{A_{1,1}x^{\text{real}}_{s_i} + a_{1,2}y^{\text{real}}_{s_i} + d_x}{a_{s_i}} + \frac{A_{2,1}x^{\text{real}}_{s_i} + A_{2,2}y^{\text{real}}_{s_i} + d_y}{b_{s_i}} = 1. \tag{29}
$$

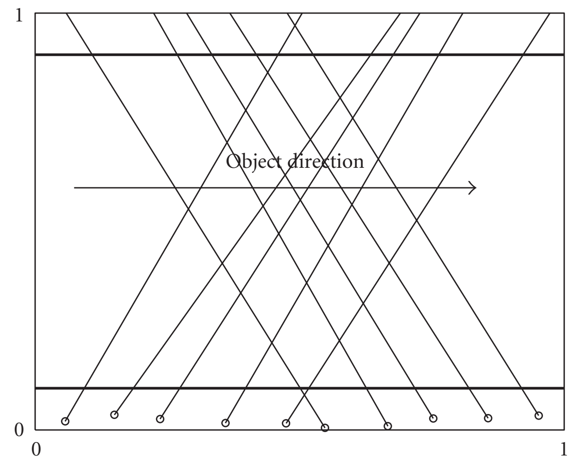


Figure 5: Setup for simulations. Sensors are shown as circles along the bottom of the figure; their directions are shown by lines. The dark parallel horizontal lines indicate the boundaries of the region of interest.

This 6×6 system of equations can be solved for $A_{1,1}$, $A_{1,2}$, $A_{2,1}$, $A_{2,2}$, d_x, and d_y. Once the transformation is known, we can use (29) to recover the lines of sight of the sensors in the real-world system. Grouping terms in (29) we obtain

$$
a^{\text{real}} = \frac{1 - d_x/\widehat{a} - d_y/\widehat{b}}{A_{1,1}/\widehat{a} + A_{2,1}/\widehat{b}}, \qquad b^{\text{real}} = \frac{1 - d_x/\widehat{a} - d_y/\widehat{b}}{A_{2,1}/\widehat{a} + A_{2,1}/\widehat{b}}. \tag{30}
$$

6. A SIMULATION STUDY

We first present the results of a preliminary simulation study that was conducted prior to an actual implementation, which we shall describe in the sequel.

Figure 5 shows the setup for the simulations. A section of a passage (e.g., a road, bridge, tunnel, etc.) is monitored by a collection of m sensors located along the sides of the passage. The length of the section is L, and its width is W. In the simulations, $L = W = 1$. Sensors located on the left side of the section are pointed to the right, while those located at the right side are pointed to the left. Sensors are located regularly, except for noise in their positions, and the angles of their lines of sight are approximately 63^o. Notice that, although in the simulations in this section and the implementation presented in Section 7 the sensors are placed regularly, the actual location of each sensor is irrelevant to the performance of the algorithm. It is the direction of the sensor lines, not the location of the sensors on those lines, what determines the behavior of the algorithm.

The exact angles of the sensors must be recovered from the measurements, as part of the problem. We have purposely avoided situations in which sensors are "close to vertical" or "close to horizontal," since such situations produce numerical problems. The measurement errors are uniformly distributed in $[-0.01, 0.01]$. Objects enter the section from the left and exit it from the right. The speed of the objects is chosen uniformly and independently in the range $[0.01, 0.1]$,

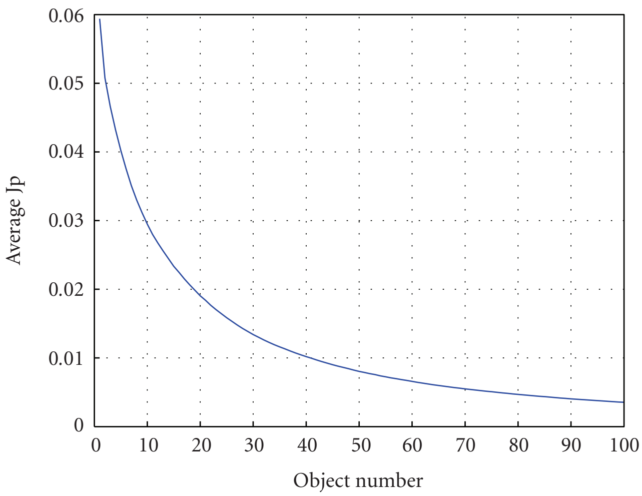


FIGURE 6: Average estimation error (J_p), as a function of the number of detected objects, for 100 different runs of the algorithm.

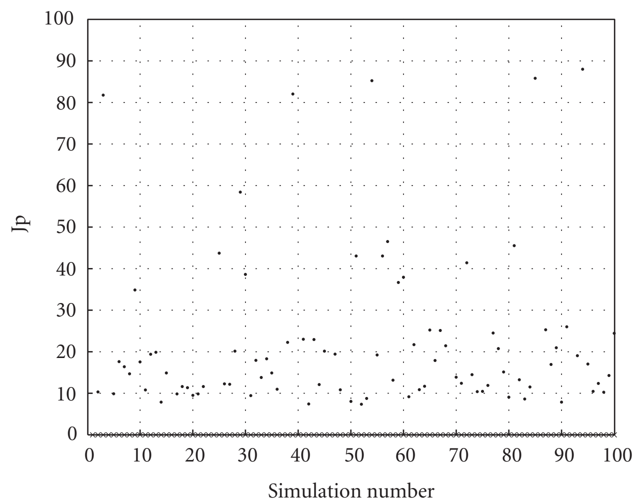


FIGURE 7: Error in parameter estimates given by the adaptive basis algorithm (crosses), and a randomly restarted local improvement algorithm (dots).

while their trajectories are fixed by choosing random entry and exit points. To ensure that the two first trajectories are not parallel, they are fixed: the first trajectory entering and exiting at the bottom, and the second trajectory entering at the bottom and exiting at the top (thus maximizing the angle between them).

The estimation of the sensor and object parameters is done by minimizing the quadratic cost function (5), although the quality of the resulting estimates is assessed by the cost defined by

$$(2\overline{m}+4\overline{n})J_p := \sum_{i=1}^{\overline{m}}\left[(\hat{a}_{s_i}-a_{s_i})^2+(\hat{b}_{s_i}-b_{s_i})^2\right] + \sum_{j=1}^{\overline{n}}\left[(\hat{v}^x_{o_j}-v^x_{o_j})^2+(\hat{x}^0_{o_j}-x^0_{o_j})^2+(\hat{v}^y_{o_j}-v^y_{o_j})^2+(\hat{y}^0_{o_j}-y^0_{o_j})^2\right], \tag{31}$$

where $\overline{m}$, and $\overline{n}$ are the number of sensors and objects, respectively. The behavior of J_p for the first 100 objects (after the passage of the initial four objects necessary to initialize the algorithm) for 100 different runs of the algorithm is shown in Figure 6. The curve shown corresponds to an average over the 100 runs of the simulation.

It is clear from Figure 6 that the quality of the estimation improves with the number of detected objects, which is as desired. It is important to mention the importance of the refining phase, phase 2, to improve the performance of the algorithm when measurements are noisy.

To illustrate the importance of the first phase of the algorithm, we compare in Figure 7 the error in the parameter estimates J_p for the first six sensors and four objects, given by the adaptive basis algorithm (crosses), versus that of a randomly restarted local improvement algorithm (dots). In each simulation, the local improvement algorithm was restarted at 100 different points, and the best parameter estimates chosen as the ones minimizing (5). The random initialization points were obtained in the same fashion as the actual parameters of sensors and objects. No noise in the data was considered. It can be seen in Figure 7 that the local improvement algorithm is unable to find the optimum parameter estimates, in contrast to the adaptive basis algorithm. This is due to the non-convexity of the cost function (5), that is, the local improvement algorithm is able to find only local minima of the cost function. The adaptive basis algorithm finds the global minimum.

7. IMPLEMENTATION

The system described in the previous sections was implemented using Berkeley mica2 motes provided with light sensors. The directional sensors were implemented using laser pointers, pointed directly at the light sensors. A toy car was used to simulate the objects.

7.1. *Setup for the experiments*

The setup for the experiments is shown in Figure 8. Six light sensors and six lasers were placed on different sides of a track of length 16 foot and width 8 foot. The speed of the car was approximately constant equal to 1.41 ft/s.

A picture of the testbed is shown in Figure 9. The car is the object positioned between the sensors and the lasers.

As the car runs through the laser field, it interrupts the lasers. The motes detect the interruption times. The times are transmitted to a seventh mote, which runs the algorithm. After the car has passed four times, the seventh mote estimates the entry and exit points of the fourth car. Then, for each subsequent pass, the estimated parameters are updated, and the entry and exit points of the current pass are estimated.

To perform the coordinate transformation, the trajectories of the two first objects were fixed. The first object entered at 0 and exited at 0, while the second object entered at 0 and

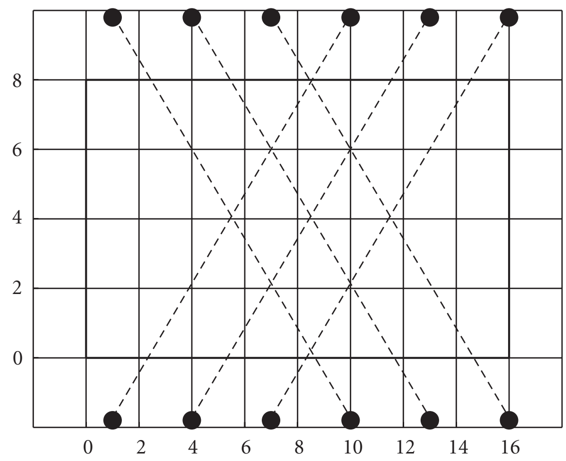


Figure 8: Setup for experiments. Sensors are shown as the black disks at the bottom of the figure. Lasers are represented by disks at the top of the figure.

Figure 9: Picture of the testbed. Sensors can be seen on the left, lasers on the right. The car that was used as an "object" can be seen in the middle.

exited at 8. This was done because the locations of the sensors were hard to measure. This also improved the estimation accuracy, because it maximized the angle between the first two sensors.

Let v denote the speed of the car. The coordinate transformation can be obtained, from the following:

(1) Point $(1, 0)$ in the adaptive basis corresponds to point $(v, 0)$ in the real-world.

(2) Point $(0, 1)$ in the adaptive basis corresponds to $(v(16/\sqrt{16^2 + 8^2}), v(8/\sqrt{16^2 + 8^2}))$ in the realworld.

The conversion is found from

$$\mathbf{A}\begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} = \begin{bmatrix} 1.41 & 1.26 \\ 0 & 0.63 \end{bmatrix}. \tag{32}$$

We then have that

$$\begin{aligned} a^{\text{real}} &= \frac{1}{[\mathbf{A}]_{1,1}/\hat{a} + [\mathbf{A}]_{2,1}/\hat{b}}, \\ b^{\text{real}} &= \frac{1}{[\mathbf{A}]_{1,2}/\hat{a} + [\mathbf{A}]_{2,2}/\hat{b}}. \end{aligned} \tag{33}$$

7.2. Results

We discuss here the results of the experiments. We focus on one experiment with 32 runs, although we performed experiments with up to 40 runs.

Figure 10 shows the actual and estimated entry and exit points for four runs out of 32 runs. It is important to note that the algorithm is able to estimate the entry and exit points with good accuracy, and that it remains stable, even after a large number of objects have passed. The histograms for the errors in entry and exit points for 4–32 runs are shown in Figure 11. The maximum number of objects in one single experiment was 40. After each run, all parameters from previous runs, and all sensor parameters were updated. The number of iterations of Newton's method was fixed to 5, rather than checking for convergence.

Figure 11 shows a histogram of the estimation errors in entry and exit points. Again, we can see that the algorithm was able to accurately estimate the trajectories of the objects.

8. CONCLUSIONS

We considered the problem of tracking objects moving in straightlines, using a network of highly directional sensors. This estimation problem involves a highly nonconvex optimization problem. To overcome this difficulty we introduced a three phase algorithm, which we call the adaptive basis algorithm. We simulated the algorithm and have implemented it in a laboratory setting.

The adaptive basis algorithm assumes that the field of vision of the sensors are straightlines, but it might be possible to extend this algorithm to handle omnidirectional sensors and directional sensors with a field of vision given by a convex sector, rather than a line. We discuss here such possibilities. This is matter of future work.

Assume that two omnidirectional sensors are located on a plane, and measure the intensity of a signal produced by an object. Suppose also that the object is small, and the fields of vision of the sensors are perfect discs. If the object is located closer to one sensor than the other, such sensor will measure a higher intensity. If the two sensors compare their measurements, they can determine the moment at which the object crosses the bisector line between them. Collecting such crossing times from different objects and sensor pairs would provide data that could be used to estimate the trajectories of the objects, and the bisector lines of the sensors.

From Figure 1(b) we notice that although the field of vision of a directional sensor might be a convex sector rather than a line, the edges of such sector are lines. Sensors might record the times at which an object enters or exits their field of vision. An additional difficulty that must be overcome in this case is to determine in each case, on which "side" of the sector the object entered, and on which it exited, and to eliminate the data of objects entering through the "front."

The adaptive basis algorithm uses minimal information. Nothing is known a priori. If more information is available, for example, the trajectories of some of the objects or the directions of some of the sensor lines, and so forth, this could be used to improve the estimates or simplify the estimation.

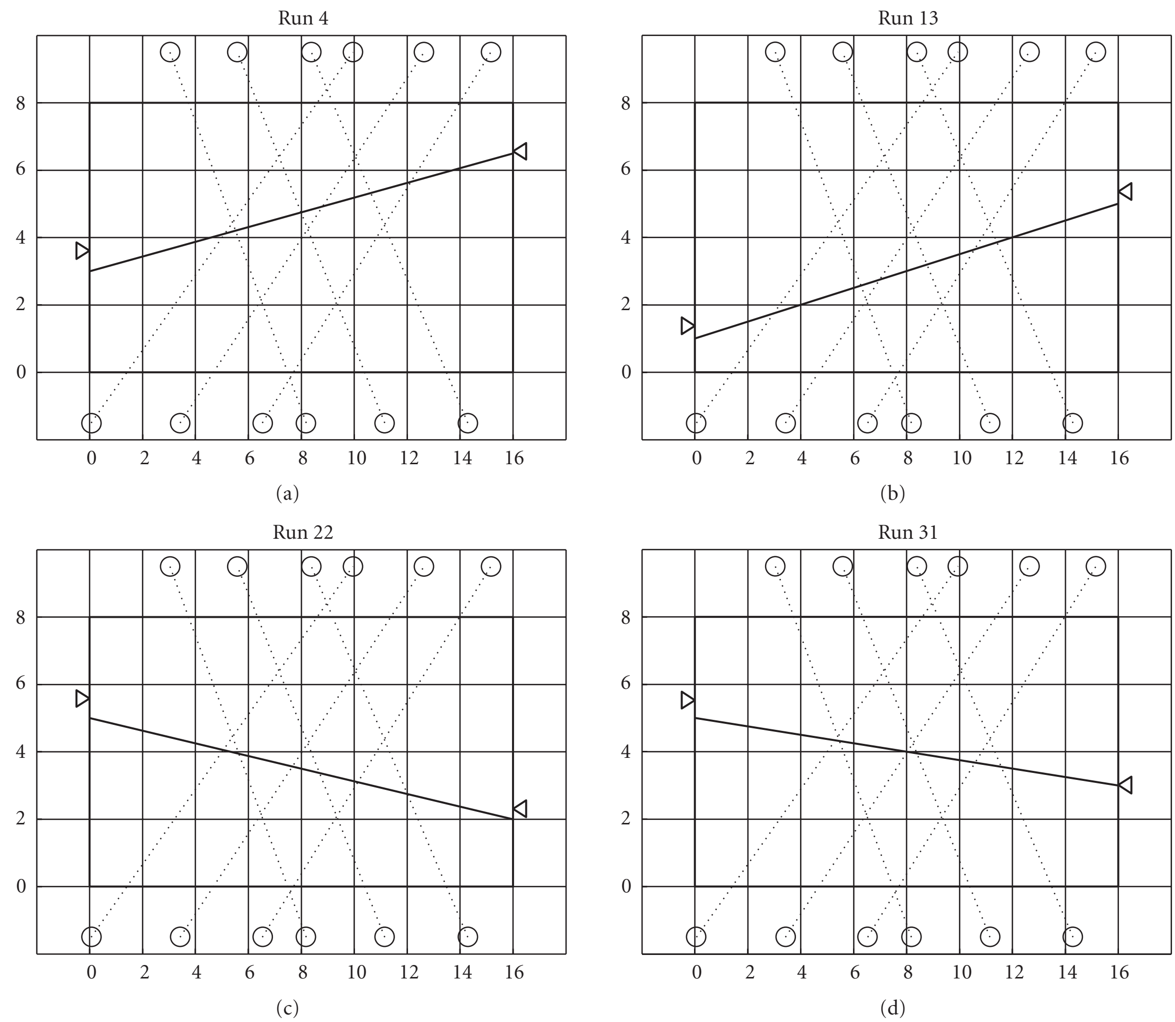


Figure 10: Runs 4, 13, 22, and 31 from an experiment with a total of 32 runs. Top circles are lasers, bottom dark circles are sensors. Sensor lines are shown with dotted lines. Note that the sensor lines shown were estimated from the data. The domain is a rectangle marked with a thick borderline. The actual trajectory is shown as a left-to-right thick line. Estimated entry and exit points are indicated with triangles.

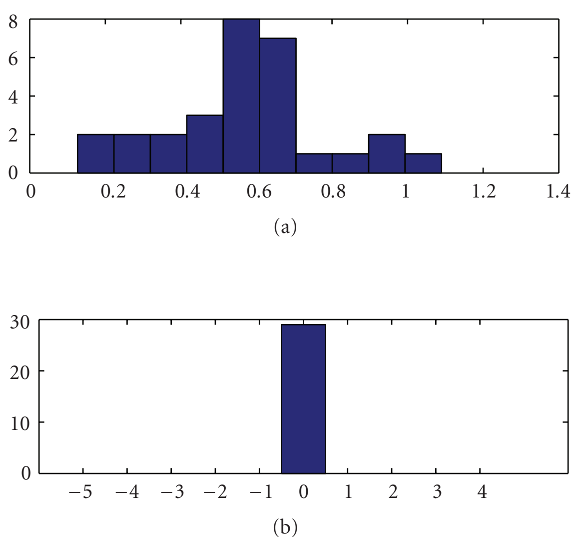


Figure 11: Histograms for errors in entry and exit points for a 32-run (objects) experiment.

ACKNOWLEDGMENTS

This material is based upon work partially supported by NSF under Contracts nos. NSF ANI 02-21357 and CCR-0325716, USARO under Contracts nos. DAAD19-00-1-0466 and DAAD19-01010-465, DARPA/AFOSR under Contract no. F49620-02-1-0325, DARPA under Contracts nos. N00014-0-1-1-0576 and F33615-0-1-C-1905, and AFOSR under Contract no. F49620-02-1-0217.

REFERENCES

[1] A. Arora, P. Dutta, S. Bapat, et al., "A line in the sand: a wireless sensor network for target detection, classification, and tracking," *Computer Networks*, vol. 46, no. 5, pp. 605–634, 2004.

[2] C. Gui and P. Mohapatra, "Power conservation and quality of surveillance in target tracking sensor networks," in *Proceedings of the 10th Annual International Conference on Mobile Computing and Networking (MobiCom '04)*, pp. 129–143, Philadelphia, Pa, USA, September-October 2004.

[3] W.-P. Chen, J. Hou, and L. Sha, "Dynamic clustering for acoustic target tracking in wireless sensor networks," *IEEE Transactions on Mobile Computing*, vol. 3, no. 3, pp. 258–271, 2004.

[4] T. Vercauteren, D. Guo, and X. Wang, "Joint multiple target tracking and classification in collaborative sensor networks," *IEEE Journal on Selected Areas in Communications*, vol. 23, no. 4, pp. 714–723, 2005.

[5] Y. He and K. P. Chong, "Sensor scheduling for target tracking in sensor networks," in *Proceedings of the 43th IEEE Conference on Decision and Control (CDC '04)*, vol. 1, pp. 743–748, Atlantis, Bahamas, December 2004.

[6] J. E. Bevington, "Distributed sensor management and target tracking for unattended ground sensor networks," in *Battlespace Digitization and Network-Centric Systems IV*, vol. 5441 of *Proceedings of SPIE*, pp. 25–35, Orlando, Fla, USA, April 2004.

[7] R. R. Brooks, P. Ramanathan, and A. Sayeed, "Distributed target classification and tracking in sensor networks," *Proceedings of the IEEE*, vol. 91, no. 8, pp. 1163–1171, 2003.

[8] J. Liu, M. Chu, J. Liu, J. Reich, and F. Zhao, "Distributed state representation for tracking problems in sensor networks," in *Proceedings of the 3rd International Symposium on Information Processing in Sensor Networks (IPSN '04)*, pp. 234–242, Berkeley, Calif, USA, April 2004.

[9] J. Liu, P. Cheung, L. Guibas, and F. Zhao, "A dual-space approach to tracking and sensor management in wireless sensor networks," in *Proceedings of the 1st ACM International Workshop on Wireless Sensor Networks and Applications (WSNA '02)*, pp. 131–139, Atlanta, Ga, USA, September 2002.

[10] M. Horton, A. Broad, M. Grimmer, et al., "Deployment ready multimode micropower wireless sensor networks for intrusion detection, classification, and tracking," in *Unattended Ground Sensor Technologies and Applications IV*, vol. 4743 of *Proceedings of SPIE*, pp. 307–312, Orlando, Fla, USA, April 2002.

[11] J. Liu, J. Reich, and F. Zhao, "Collaborative in-network processing for target tracking," *EURASIP Journal on Applied Signal Processing*, vol. 2003, no. 4, pp. 378–391, 2003.

[12] F. Zhao, J. Shin, and J. Reich, "Information-driven dynamic sensor collaboration," *IEEE Signal Processing Magazine*, vol. 19, no. 2, pp. 61–72, 2002.

[13] J. Liu, J. Reich, P. Cheung, and F. Zhao, "Distributed group management for track initiation and maintenance in target localization applications," in *Proceedings of the 2nd International Workshop on Information Processing in Sensor Networks (IPSN '03)*, vol. 2634 of *Lecture Notes in Computer Science*, pp. 113–128, Palo Alto, Calif, USA, April 2003.

[14] A. Galstyan, B. Krishnamachari, K. Lerman, and S. Pattem, "Distributed online localization in sensor-networks using a moving target," in *Proceedings of the 3rd International Conference on Information Processing in Sensor Networks (IPSN '04)*, pp. 61–70, Berkeley, Calif, USA, April 2004.

[15] R. Solis, V. S. Borkar, and P. R. Kumar, "A new distributed time synchronization protocol for multihop wireless networks," in *Proceedings of the 45th IEEE Conference on Decision and Control (CDC '06)*, pp. 2734–2739, Morgan Kaufmann, San Diego, Calif, USA, December 2006.

[16] D. P. Bertsekas, *Nonlinear Programming*, Athena Scientific, Belmont, Mass, USA, 1995.