Target Localization and Tracking with Motion Sensors

Daizhan Cheng[a], Bijoy K. Ghosh[b] and Xiaoming Hu[c]

[a]Institute of Systems Science, Chinese Academy of Sciences
Beijing 100080, China
dcheng@iss.ac.cn
[b]Dept. of Mathematics and Statistics, Texas Tech University
Lubbock, TX 79409, USA
bijoy.ghosh@ttu.edu
[c]Optimization and Systems Theory, Royal Institute of Technology
Stockholm, Sweden
hu@kth.se

1 Introduction

Sensor networks provide unprecedented opportunities to sense and monitor the physical environment, see for example [1] and the references therein. A sensor network consists of a large number of, possibly heterogeneous, sensor nodes, deployed over a region of interest. Each node is typically equipped with limited communication capacity and power supply, and has a variety of sensor modules such as light, sound, temperature and infra-red.

Intrusion detection and tracking has been a research problem of many practical applications and has hence attracted quite some attention in the signal processing and communication community [9, 8, 10, 4, 11]. In most of the literature, an information theoretic approach is taken for collaborative signal processing, where, for example, an entropy function is used [8].

In this paper we focus on the problem of target localization and tracking with infra-red motion-detection sensors, namely sensors that can detect motion (velocity) along certain direction. Naturally several such sensors are needed for localization. Moreover, we develop collaborative localization algorithms based on a control theoretic approach. By the nature of the sensors, the target has to be in constant motion in order to get detected. Thus the algorithm needs to handle dynamically the flow of sensor data. In particular, we take a dynamical system approach to handle measurement noise.

Comparing with the the existing methods, our approach has the following advantages: (a) Most existing methods would only work with linear sensor models or are based on linearization. Our approach is based on a nonlinear model, which eliminates the model approximation (linearization) error that would be very significant due to the particular characteristics of the sensors we consider; (b) unlike most existing methods, which can deal with only particles (solid bodies) in $\mathbb{R}^2$ or $\mathbb{R}^3$, we use an n dimensional model, which means that the objects observed can be a flexible body with n degrees of freedom.

The paper is organized as follows. In Section 2 we formulate the problem and address some fundamental issues regarding detectability of the target. In Section 3 some basic tracking algorithms are given. In Sections 4 issues such as optimal sensor selection, robustness and scalability are discussed. Finally in Section 5 some experimental results are given.

2 Geometry of the sensor network and detectability

In this section we formulate the tracking problem and discuss the placement of sensors such that the tracking problem can be solved.

We assume that a flexible target is moving along the trajectory $r(t) \in \mathbb{R}^n$, which means the target is of n degree of freedom.

We consider sensors that measure the target velocity tangential to the sphere centered at the sensor and with the distance from the sensor to the target as radius. Such sensors can be made of from, for example, infra-red arrays. We suppose that a set of sensors are located at s_i, $i = 1, \cdots, N$ in an initially fixed coordinate system.

Our task is to detect and track the position of the target $r(t)$ based on sensor readings. For this purpose, we are facing two questions: (1) At least how many sensors are necessary to determine the position of $r(t)$ of the particle? (2) How to compute the position via measurements?

We begin by modeling the sensor. For sensor i located at s_i, let $n_i = r - s_i$, $\dot{r}_{n_i}$ be the projection of $\dot{r}$ onto n_i, and $\dot{r}_\sigma$ be the component of $\dot{r}$ perpendicular to n_i. Then,

$$\dot{r}_\sigma = \dot{r} - \dot{r}_{n_i} = \dot{r} - \langle \dot{r}, n_i \rangle \frac{n_i}{\|n_i\|^2} \tag{1}$$

$\dot{r}_\sigma$ is basically what a sensor array can measure. Physically such readings decay over distance, so we assume the output from sensor i is:

$$y_i = \frac{\dot{r}_\sigma}{\|n_i\|^p} = \frac{\|r - s_i\|^2 \dot{r} - \langle r - s_i, \dot{r} \rangle (r - s_i)}{\|r - s_i\|^{p+2}}, \tag{2}$$

where p is a positive number. For convenience, we call sensors that give (2) *vector sensors*. For the time being, we assume the measurement is noise-free.

In particular, when $p = 1$, we have

$$y_i = \dot{\theta}_i,$$

where θ_i is the angle between $r(t) - s_i$ and $r(t_0) - s_i$.

If we assume only the norm of the right hand side of (2) can be measured, then we have in this case

$$y_i = \frac{\left[\|r(t)-s_i\|^2\|\dot r(t)\|^2 - \langle r(t)-s_i, \dot r(t)\rangle^2\right]^{1/2}}{\|r(t)-s_i\|^{p+1}}, \tag{3}$$

where $i = 1, \cdots, N$.

For convenience, we call sensors that give (3) *scalar sensors.*

Proposition 2.1

1. *Generically, n sensors are enough for determining the differential equation of $r(t)$ (i.e., $\dot r(t)$) for scalar sensors.*
2. *In order to reconstruct $r(t)$, at least three vector sensors are necessary, or at least $n+1$ scalar sensors are necessary in R^n.*

To prove this proposition we need some preparation:

Lemma 2.2 *Let $\xi_i \in \mathbb{R}^n$, $i = 1, \cdots, n$, with $\|\xi_i\| = 1$, be linearly independent. If*

$$\Psi = \begin{bmatrix} 0 & 1- < \xi_2, \xi_1 > & \cdots & 1- < \xi_n, \xi_1 > \\ 1- < \xi_2, \xi_1 > & 0 & \cdots & 1- < \xi_n, \xi_2 > \\ \vdots & & & \\ 1- < \xi_n, \xi_1 > & \cdots & 1- < \xi_{n-1}, \xi_n > & 0 \end{bmatrix} \tag{4}$$

is nonsingular, then there exists $c \in \mathbb{R}^n$ such that $A_i c$, $i = 1, \cdots, n$ are linearly independent, where

$$A_i = \|\xi_i\|^2 I_n - \xi_i \xi_i^T, \quad i = 1, \cdots, n.$$

Proof. With linear independence the hyperplane $px = b$ that contains the $\xi_i's$ does not pass through the origin, i.e. $b \neq 0$, where p is the unique vector satisfying

$$\begin{array}{l} \langle p, \xi_i - \xi_j\rangle = 0, \quad i \neq j \\ \langle p, \xi_i\rangle = b, \quad \forall i. \end{array}$$

Simply choose $c = p^T$. Then a straightforward computation shows that

$$\begin{bmatrix} c^T A_1 \\ \vdots \\ c^T A_n \end{bmatrix} = \begin{bmatrix} p - b\xi_1^T \\ \vdots \\ p - b\xi_n^T \end{bmatrix}.$$

Suppose there exists $\alpha = (\alpha_1, \cdots, \alpha_n)^T$, such that

$$\sum_{i=1}^n \alpha_i (p - b\xi_i^T) = 0.$$

Then by right-multiplying both sides by ξ_i, $i = 1, \cdots, n$, we obtain

$$\Psi\alpha = 0.$$

The conclusion follows. □

Remark 1 *It is straightforward to verify that when $n \leq 3$ the matrix Ψ defined by (4) is always nonsingular under the assumptions on ξ_i, $i = 1, \cdots, n$. But it is not true for $n \geq 4$.*

Lemma 2.3 *Let $\xi_i \in \mathbb{R}^n$, $i = 1, \cdots, n$, and define A_i as before. Then for a.e $\{\xi_i\}$ (i.e., except a zero-measure set) there exists $c \in \mathbb{R}^n$ such that $\{c^T A_i\}$ are linearly independent.*

Proof. First, we assume ξ_i are linearly independent. This property holds except on a zero-measure set. Next, without loss of generality, we assume $\|\xi_i\| = 1$. In fact, we can re-scale ξ_i to $\xi_i/\|\xi_i\|$, which does not change the linear independence of $\{c^T A_i\}$.

Finally, we only need to show that $\det(\Psi)$ is not identically zero. Then only on a zero-measure set it can be zero since it is an algebraic equation of the matrix elements.

To show this we choose orthogonal $\{\xi_i\}$. Then

$$\Psi = \begin{bmatrix} 0 \; 1 \cdots \; 1 \\ 1 \; 0 \cdots \; 1 \\ \vdots \\ 1 \; 1 \cdots \; 0 \end{bmatrix}.$$

A straightforward computation shows that

$$det(\Psi) = (-1)^{i-1}(i-1) \neq 0, \quad i \geq 2,$$

which completes the proof. □

Lemma 2.4 *Let $\xi \neq 0$, and $M(\xi) := \xi^T \xi I_n - \xi \xi^T$. Then*

$$\text{rank}(M(\xi)) = n - 1. \tag{5}$$

Proof. Since

$$M(\xi)\xi = 0,$$

$\text{rank}(M(\xi)) \leq n-1$. But $\xi\xi^T$ has 0 as an $(n-1)$-multiple eigenvalue, so $\|\xi\|^2$ is the $(n-1)$-multiple eigenvalue of $M(\xi)$. Hence $\text{rank}(M(\xi)) \geq n-1$. □

Proof of Proposition 2.1. Define a set of functions as

$$\begin{aligned} F_i^v(r, \dot{r}, t) &:= \|r - s_i\|^2 \dot{r} - \langle r - s_i, \dot{r} \rangle (r - s_i) - y_i \|r - s_i\|^{p+2}, \\ F_i^s(r, \dot{r}, t) &:= \|r(t) - s_i\|^2 \|\dot{r}(t)\|^2 - \langle r(t) - s_i, \dot{r}(t) \rangle^2 \\ &- \; (y_i(t))^2 \|r(t) - s_i\|^{2p+2}, \\ & \qquad\qquad\qquad\qquad i = 1, \cdots, N. \end{aligned}$$

We consider the case of using scalar sensors first. Namely, to reconstruct $r(t)$ is to find $r(t)$ such that $F_i^s(r, \dot{r}, t) = 0$. It is easy to calculate that

$$\nabla_{\dot{r}} F_i^s = \left[\|r - s_i\|^2 I_n - (r - s_i)(r - s_i)^T\right] \dot{r},\ i = 1, \cdots, N.$$

We define a set of matrices accordingly as

$$A_i := \|r - s_i\|^2 I_n - (r - s_i)(r - s_i)^T, \quad i = 1, \cdots, N. \tag{6}$$

And denote

$$J_F^s(\dot{r}) = \begin{bmatrix} \nabla_{\dot{r}} F_1^{sT} \\ \vdots \\ \nabla_{\dot{r}} F_N^{sT} \end{bmatrix} = \begin{bmatrix} \dot{r}^T A_1 \\ \vdots \\ \dot{r}^T A_N \end{bmatrix} \tag{7}$$

Clearly N must be at least equal to n in order for J_F^s to have full rank, we therefore assume $N = n$.

By Lemma 2.2, J_F^s is nonsingular a.e. Then by the implicit function theorem we can locally solve $\dot{r}$ out around $\dot{r}_0$ as

$$\dot{r}(t) = \psi(r, t). \tag{8}$$

Otherwise, the ODE of $r(t)$ can not be uniquely determined from $F_i = 0$, $i = 1, \cdots, n$. Note that solutions however always exist (at lease the one that is from the real trajectory of the moving target). So multi-solutions of $r(t)$ exist in this case.

Obviously, there is no way to determine the initial value $r(t_0) = r_0$ since with any initial value the n equations that determine $\dot{r}$ will be satisfied.

Now for the case of using vector sensors, to reconstruct $r(t)$ is to find $r(t)$ such that $F_i^v(r, \dot{r}, t) = 0$. Since

$$\nabla_{\dot{r}} F_i^v = \|r - s_i\|^2 I_n - (r - s_i)(r - s_i)^T, \quad i = 1, \cdots, N,$$

which, according to (i) of above lemma, has rank $n - 1$. A similar argument as in the case of using scalar sensors, one sees that a necessary condition is

$$k(n - 1) \geq n + 1,$$

where k is the number of sensors. So when $n > 2$, $k \geq 2$ and for planar case, $k \geq 3$. □

Next, let us investigate how many sensors are enough for reconstructing $r(t)$. We focus on the case of scalar sensors, which is more complex. Assume the differential equation (8) can be solved from n equations $F_i(r, \dot{r}, t) = 0$, $i = 1, \cdots, n$. Next we define

$$h_i(r, \dot{r}, t) := \frac{\left[\|r(t) - s_i\|^2 \|\dot{r}(t)\|^2 - \langle r(t) - s_i, \dot{r}(t)\rangle^2\right]^{1/2}}{\|r(t) - s_i\|^{p+1}},$$

where $i = 1, \cdots, N$. Then from the nonlinear control theory we have the following:

Proposition 2.4 *$N \geq n+1$ sensors are enough to detect $r(t)$ iff the following system is observable.*

$$\begin{cases} \dot{r}(t) = \psi(r,t) \\ y_j = h_{n+j}(r, \psi(r,t), t), \quad j = 1, \cdots, N-n. \end{cases} \tag{9}$$

Remark 2

• *Define*

$$L_\psi \xi = d\xi \psi + \frac{\partial \xi}{\partial t}.$$

Then the observability of (9) means there exists a $T \geq 0$ such that at T the co-distribution

$$\Omega(T) = Span\left\{ dL_\psi^k h_{n+j} \| \big| \, j = 1, \cdots, N-n, k \geq 0 \right\}$$

has dimension n. Using small perturbation on s_{n+j}, one sees that the rank condition of $\Omega(T)$ can be satisfied genetically as $N = n+1$. So in general, $n+1$ sensors are enough.

• *Tracking the "output" y_i continuously for a time period is inconvenient. Particularly, in practical use we have only discrete observing data, so using the observability to determine the moving trajectory may not be practical. To observe $r(t_0)$ and $\dot{r}(t_0)$ at real time, it is obvious that at least $2n$ sensors are necessary.*

Finally, we consider the following problem: Assume the locations of sensors are fixed in advance, when the velocity is detectable via n sensors (a.e)? For practical purposes, we consider only the cases $1 < n \leq 3$.

Proposition 2.4 *In the planar case, the velocity is observable (a.e), iff $\xi_i = r - s_i$, $i = 1, 2$ are linearly independent. That is, the object is not moving on the line of $\overline{s_1 s_2}$.*

Proof. The necessity can be easily seen from the proof of Proposition 2.1. We can show the sufficiency by applying Lemma 2.1, since in this case ξ_1 and ξ_2 satisfy the assumptions of Lemma 2.1 and for $n = 2$ the matrix Ψ is automatically nonsingular. □

Proposition 2.5 *In the three dimensional case, the velocity is observable, iff $\xi_i = r - s_i$, $i = 1, 2, 3$ are pairwise linearly independent. That is, the object is not moving on one of the three lines: $\overline{s_1 s_2}$, $\overline{s_1 s_3}$, or $\overline{s_2 s_3}$.*

Proof. We only need to show the sufficiency.

Case 1. Assume $\xi_i = r - s_i$, $i = 1, 2, 3$ are linearly independent. Then they satisfy the assumptions of Lemma 2.1 and for $n = 3$ the matrix Ψ is automatically nonsingular.

Case 2. Assume $\xi_i = r - s_i$, $i = 1, 2, 3$ are linearly dependent. Since they are pairwise linearly independent, then we assume that ξ_1 and ξ_2 are linearly independent and $\xi_3 = \mu_1 \xi_1 + \mu_2 \xi_2$. Now pairwise linear independence is equivalent to $\mu_1 \neq 0$, $\mu_2 \neq 0$. Now we can choose an $0 \neq \eta \in (Span\{\xi_1, \xi_2\})^\perp$. Then ξ_1, ξ_2, and η are linearly independent.

Choosing $x = \xi_3 + \eta/\|\eta\|$ yields

$$J_F^s(x) =$$
$$\begin{bmatrix} -\mu_2 < \xi_1, \xi_2 > & \|\xi_1\|^2\mu_2 & \|\xi_1\|^2 \\ \|\xi_2\|^2\mu_1 & -\mu_1 < \xi_1, \xi_2 > & \|\xi_2\|^2 \\ 0 & 0 & \|\xi_2\|^2 \end{bmatrix} \begin{bmatrix} \xi_1^T \\ \xi_2^T \\ \eta^T/\|\eta\| \end{bmatrix}$$
$$:= A_0 \begin{bmatrix} \xi_1^T \\ \xi_2^T \\ \eta^T/\|\eta\| \end{bmatrix}.$$

It follows that

$$\det(A_0) = \mu_1\mu_2 \left(< \xi_1, \xi_2 >^2 - \|\xi_1\|^2\|\xi_2\|^2\right) \neq 0.$$

□

Remark 3

• *Proposition 2.5 applies even when s_1, s_2, and s_3 are on one line, as long as there are no coincides.*

• *Proposition 2.5 also suggests that if there are 4 sensors and no any three are on one line then the velocity is always observable.*

3 Target tracking algorithms

In this section we present two algorithms for reconstructing $r(t)$ based on sensor measurements. The first method is based on solving equation (3). The second method is based on solving (2) but uses the first method to provide an estimation of the initial state.

3.1 Basic numerical algorithms and initial localization

For computational ease, we rewrite (3) as

$$\|x - s_i\|^2\|y\|^2 - \langle x - s_i, y\rangle^2 - \|x - s_i\|^{2p+2} b_i^2 = 0, \tag{10}$$

Where $x = r(t)$, $y = \dot{r}(x)$, and $b_i = y_i(t)$. To simplify the notation, we assume now $p = 1$.

The following two methods are used to solve equation (10).

Method 1. Least Square and Gradient Approach

Define the square error, E, as

$$E = \sum_{i=1}^{N} \left(\|x - s_i\|^2\|y\|^2 - \langle x - s_i, y\rangle^2 - \|x - s_i\|^4 b_i^2\right)^2. \tag{11}$$

Then we have

$$\begin{cases} \frac{\partial E}{\partial x} = 4\sum\limits_{i=1}^{N} \left\{ \left[\|x-s_i\|^2\|y\|^2 - \langle x-s_i, y\rangle^2 - \|x-s_i\|^4 b_i^2 \right] \right. \\ \qquad \left. \left[\|y\|^2 (x-s_i) - \langle x-s_i, y\rangle\, y - 2b_i^2 \|x-s_i\|^2 (x-s_i) \right] \right\} \\ \frac{\partial E}{\partial y} = 4\sum\limits_{i=1}^{N} \left\{ \left[\|x-s_i\|^2\|y\|^2 - \langle x-s_i, y\rangle^2 - \|x-s_i\|^4 b_i^2 \right] \right. \\ \qquad \left. \left[\|x-s_i\|^2 y - \langle x-s_i, y\rangle\, (x-s_i) \right] \right\}. \end{cases} \tag{12}$$

The iterative equation is

$$\begin{bmatrix} x_{t+1} \\ y_{t+1} \end{bmatrix} = \begin{bmatrix} x_t + \Delta x_t \\ y_t + \Delta y_t \end{bmatrix}, \tag{13}$$

where

$$\begin{bmatrix} \Delta x_t \\ \Delta y_t \end{bmatrix} = \begin{bmatrix} \frac{\partial E}{\partial x}\Big|_{(x_t,y_t)} h_x \\ \frac{\partial E}{\partial y}\Big|_{(x_t,y_t)} h_y \end{bmatrix} \tag{14}$$

with h_x and h_y as the constant step lengths.
Method 2. Generalized Newton's Method

Since for the sensor network we assume in most cases $N > 2 * n$, the conventional Newton's method [6] is not applicable. We propose the following generalized Newton's method.

Assume we have the following over-determined nonlinear equations

$$f_i(x_1, \cdots, x_n) = 0, \quad i = 1, \cdots, N, \tag{15}$$

where $N > n$. Like in the standard Newton's method, we use the following approximation:

$$f_i(x^{k+1}) \approx f_i(x^k) + \frac{\partial f_i(x^k)}{\partial x}\,(x^{k+1} - x^k),\ i = 1, \cdots, N. \tag{16}$$

Assume x^{k+1} is the solution, i.e., $f_i(x^{k+1}) = 0$, $i = 1, \cdots, N$, we then have

$$\frac{\partial f_i(x)}{\partial x}\bigg|_{x=x^k} (x^{k+1} - x^k) \approx -f_i(x^k), \quad i = 1, \cdots, N. \tag{17}$$

Then the least square error solution of (17) is

$$x^{k+1} = x^k - J_f^+(x^k) f(x^k), \tag{18}$$

where the pseudo-inverse is defined as

$$J_f^+(x) = \left[J_f^T(x) J_f(x) \right]^{-1} J_f^T(x) := [g_{ij}(x)]$$

with

$$J_f(x) = \begin{bmatrix} \frac{\partial f_1}{\partial x_1}(x) & \cdots & \frac{\partial f_1}{\partial x_n}(x) \\ \vdots & & \\ \frac{\partial f_N}{\partial x_1}(x) & \cdots & \frac{\partial f_N}{\partial x_n}(x) \end{bmatrix}.$$

We need the following formulas for numerical computation:

$$\begin{cases} \frac{\partial F_i}{\partial x} = 2\left[\|y\|^2(x-s_i) - \langle x-s_i, y\rangle\, y - 2b_i^2\|x-s_i\|^2(x-s_i)\right] \\ \frac{\partial F_i}{\partial y} = 2\left[\|x-s_i\|^2 y - \langle x-s_i, y\rangle\,(x-s_i)\right]. \end{cases} \tag{19}$$

Mimicking the proof for the standard Newton's method [7], we can derive the following sufficient condition for the convergence of (18).

Theorem 3.1 *If the functions $f_i(x)$, $i = 1, \cdots, N$ have in a region G second order derivatives that are not greater than some $L > 0$ in the absolute magnitude, if the matrix $\left[J_f^T(x)J_f(x)\right]$ is nonsingular at $x^0 \in G$, and if the following condition is also satisfied:*

$$h = M^2 L\delta n N \le \frac{1}{2} \tag{20}$$

where

$$\left|f_i(x^0)\right| \le \delta,$$

and

$$\left\|J_f^+(x^0)\right\| = \|[g_{ij}(x^0)]\| := \max_i \sum_{j=1}^N |g_{ij}(x^0)| \le M,$$

then the sequence $\{x^k\}$ generated by (18) converges to the least square solution of (15). Namely,

$$\lim_{k\to\infty} x^k = x^*, \tag{21}$$

where x^* satisfies

$$\sum_{i=1}^N f_i^2(x) \ge \sum_{i=1}^N f_i^2(x^*), \quad \forall x \in G. \tag{22}$$

We now apply the generalized Newton's method to our problem. We define

$$F_i(x,y) = \|x-s_i\|^2\|y\|^2 - \langle x-s_i, y\rangle^2 - \|x-s_i\|^4 b_i^2,$$

where $i = 1, \cdots, N$. Then for generalized Newton's method the iterative equation becomes

$$\begin{bmatrix} x^{k+1} \\ y^{k+1} \end{bmatrix} = \begin{bmatrix} x_k \\ y_k \end{bmatrix} - [(J_F^T J_F)^{-1} J_F^T](x^k, y^k) \begin{bmatrix} F_1(x^k, y^k) \\ \vdots \\ f_N(x^k, y^k) \end{bmatrix}, \tag{23}$$

where

$$J_F = \begin{bmatrix} \frac{\partial F_1}{\partial x_1} & \cdots & \frac{\partial F_1}{\partial x_n} & \frac{\partial F_1}{\partial y_1} & \cdots & \frac{\partial F_1}{\partial y_n} \\ \vdots & & & & & \\ \frac{\partial F_N}{\partial x_1} & \cdots & \frac{\partial F_N}{\partial x_n} & \frac{\partial F_N}{\partial y_1} & \cdots & \frac{\partial F_N}{\partial y_n} \end{bmatrix}.$$

The advantage of the least square approach is that it can converge to a given isolated solution x^* as long for the starting point x_0 x^* is the unique solution inside the searching region

$$R = \{x | E(x) \leq E(x_0)\},$$

and the searching algorithm is proper. Its disadvantage is time-consuming in general.

The advantage of Newton's method is its speed, particularly in our case where the functions involved are only polynomial. The disadvantage is, as shown in the above, in general for multi-unknown cases the convergence condition is very rigorous. In general, it requires very precise original guess.

Our proposed algorithm can be summarized as follows.

Target Tracking Algorithm:

Step 1. *Find initial values:* $r(0)$, $\dot{r}(0)$*:*

1.1 Choose initial guess: $(\hat{x}_0, \hat{y}_0) = (\alpha_0, \beta_0)$*;*

1.2 Choose a step length Δh *and using the gradient method to find* (α_1, β_1)*, which minimize square error* E*;*

1.3 Shrink Δh*, for example,* $\Delta h/100$ *and using* (α_1, β_1) *as the initial values, repeat step 1.2 to find refined optimal solutions* (α_1, β_1)*. (Such refined researches can be repeated several times.)*

Step 2. *Find sequence values:* $r(t)$, $\dot{r}(t)$*, where* $t = k(\Delta t)$*:*

2.1 Choose initial value as

i. Case 1: as $k = 0$ *set*

$$\begin{bmatrix} \hat{x}_k \\ \hat{y}_k \end{bmatrix} = \begin{bmatrix} \alpha \\ \beta \end{bmatrix},$$

where (α, β) *are obtained in Step 1.*

ii. Case 2: as $k = 1$ *set*

$$\begin{bmatrix} \hat{x}_k \\ \hat{y}_k \end{bmatrix} = \begin{bmatrix} x_{k-1} + y_{k-1}\Delta t \\ y_{k-1}. \end{bmatrix}.$$

iii. Case 3: as $k > 2$ *set*

$$\begin{bmatrix} \hat{x}_k \\ \hat{y}_k \end{bmatrix} = \begin{bmatrix} x_{k-1} + y_{k-1}\Delta t \\ y_{k-1} + \frac{x_k - 2x_{k-1} + x_{k-2}}{\Delta t} \end{bmatrix}.$$

2.2 Use Newton's method to solve (x_k, y_k) *and use square error,* E*, to stop the iterative process (as* E *stops decreasing, stop the iterative process).*

3.2 A system theoretic approach to target tracking

In this section we focus on the case of using vector sensors.

As we have discussed before, the output of a sensor can be expressed as

$$y_i = \frac{\dot{\boldsymbol{\theta}}_{\mathbf{i}}}{\|r - s_i\|^{p-1}},$$

where

$$\dot{\boldsymbol{\theta}}_{\mathbf{i}} = \frac{(\dot{r} - \dot{s}_i)\|r - s_i\|^2 - \langle \dot{r} - \dot{s}_i, r - s_i \rangle \,(r - s_i)}{\|r - s_i\|^2} \tag{24}$$

For the sake of generality, here we do not assume that sensors are fixed in location. It is well known that the n components of $\boldsymbol{\theta}_{\mathbf{i}}$ are not independent. It can be merged onto the projective space $P(n, \mathbb{R})$, which is of dimension $n-1$. To get a physical decomposition of (24), we use polar coordinate frame. Denoted by $\theta_i = \theta_i(a_1^i, a_2^i, \cdots, a_{n-1}^i)$, $s_i = (s_1^i, s_2^i, \cdots, s_n^i)^T$, the coordinate transformation from Cartesian coordinates to polar coordinates is:

$$\begin{array}{ll} x_1 - s_1^i & = r\cos(a_{n-1}^i)\cos(a_{n-2}^i)\cdots\cos(a_2^i)\cos(a_1^i) \\ x_2 - s_2^i & = r\cos(a_{n-1}^i)\cos(a_{n-2}^i)\cdots\cos(a_2^i)\sin(a_1^i) \\ x_3 - s_3^i & = rcos(a_{n-1}^i)cos(a_{n-2}^i)\cdots cos(a_3^i)\sin(a_2^i) \\ \vdots & \\ x_{n-1} - s_{n-1}^i & = r\cos(a_{n-1}^i)\sin(a_{n-2}^i) \\ x_n - s_n^i & = r\sin(a_{n-1}^i) \end{array} \tag{25}$$

It follows that

$$\tan(a_k^i) = \pm\frac{x_{k+1} - s_{k+1}^i}{\sqrt{(x_1 - s_1^i)^2 + \cdots + (x_k - s_k^i)^2}}, \tag{26}$$

where $k = 1, \cdots, n-1$. We assume a_k^i is designed in such a way that the sign $\pm$ becomes $+$. Differentiating both sides of (26) yields

$$\dot{a}_k^i = \frac{(\dot{x}_{k+1} - \dot{s}_{k+1}^i)Q_k - (x_{k+1} - s_{k+1}^i)P_k}{Q_{k+1}\sqrt{Q_k}}, \tag{27}$$

where

$$Q_j = \sum_{k=1}^{j}(x_k - s_k^i)^2, \quad P_j = \sum_{k=1}^{j}(x_i - s_k^i)(\dot{x}_i - \dot{s}_k^i).$$

In the $\mathbb{R}^2$ case (by choosing a proper sign) (27) becomes

$$\dot{a}_1^i = \frac{(x - s_1^i)(\dot{y} - \dot{s}_2^i) - (y - s_2^i)(\dot{x} - \dot{s}_1^i)}{(x - s_1^i)^2 + (y - s_2^i)^2} \tag{28}$$

In the $\mathbb{R}^3$ case in addition to (28) we have

$$\dot{a}_2^i = \frac{(\dot{z} - \dot{s}_3^i)\left[(x - s_1^i)^2 + (y - s_2^i)^2\right] - (z - s_3^i)P_3}{\sqrt{(x - s_1^i)^2 + (y - s_2^i)^2}\left[(x - s_1^i)^2 + (y - s_2^i)^2 + (z - s_3^i)^2\right]} \tag{29}$$

where $P_3 = \left[(x - s_1^i)(\dot{x} - \dot{s}_1^i) + (y - s_2^i)(\dot{y} - \dot{s}_3^i) + (z - s_3^i)(\dot{z} - \dot{s}_3^i)\right].$

Note that in the $\mathbb{R}^2$ case a_1^i is ϕ in the polar coordinate frame (r, ϕ), that is: $x = r\cos(\phi)$, and $y = r\sin(\phi)$; and in $\mathbb{R}^3$ case a_1 is ϕ and a_2 is θ in (r, ϕ, θ), that is, $x = r\cos(\theta)\cos(\phi)$, $y = r\cos(\theta)\sin(\phi)$, and $z = r\sin(\theta)$.

Using the method proposed in the previous section we can always solve for the state (position) and velocity.

However, in order to filter out the effect of measurement errors we propose the following method.

In fact, from (27) we can derive the following linear equation about $\dot{x}$

$$E\dot{x} = B \tag{30}$$

where

$$E = \left[\begin{array}{cccccc} -(x_2 - s_2^i)(x_1 - s_1^i) & S_1 - (x_2 - s_2^i)^2 & 0 & \cdots & 0 & 0 \\ -(x_3 - s_3^i)(x_1 - s_1^i) & -(x_3 - s_3^i)(x_2 - s_2^i) & S_2 - (x_3 - s_3^i)^2 & \cdots & 0 & 0 \\ \vdots & & & \vdots & & \\ -(x_n - s_n^i)(x_1 - s_1^i) & -(x_n - s_n^i)(x_2 - s_2^i) & -(x_n - s_n^i)(x_3 - s_3^i) & \cdots & -(x_n - s_n^i)(x_{n-1} - s_{n-1}^i) & S_{n-1} - (x_n - s_n^i)^2 \end{array}\right]\Bigg|_{i=1,\cdots,N} \tag{31}$$

$$B = \begin{bmatrix} S_2\dot{s}_2^i - (x_2 - s_2^i)\left[(x_1 - s_1^i)\dot{s}_1^i + (x_2 - s_2^i)\dot{s}_2^i\right] + \dot{a}_1^i\sqrt{S_1}S_2 \\ S_3\dot{s}_3^i - (x_3 - s_3^i)\left[(x_1 - s_1^i)\dot{s}_1^i + (x_2 - s_2^i)\dot{s}_2^i + (x_3 - s_3^i)\dot{s}_3^i\right] + \dot{a}_2^i\sqrt{S_2}S_3 \\ \vdots \\ S_{n-1}\dot{s}_n^i - (x_n - s_n^i)\sum_{k=1}^{n}(x_i - s_k^i)\dot{s}_k^i + \dot{a}_{n-1}^i\sqrt{S_{n-1}}S_n \end{bmatrix} \tag{32}$$

where $i = 1, \cdots, N$. Then the least square solution of (31) provides a differential equation of the states as

$$\dot{x} = (E^T E)^{-1} E^T B. \tag{33}$$

Now all the numerical methods for ODE could be used to solve this differential equation. We can use the target tracking algorithm given in the previous section for estimating the initial condition.

Remark 4

1. *In fact, it is easy to see that (28) implies the anti-clockwise direction is considered as the increment of a_1. Otherwise, negative sign should be added.*

2. *(29) implies the positive direction of* a_2^i *is the same as* $z - s_3$. *This is reasonable and coincides with the conventional assumption in polar coordinates.*
3. *In (31) and (32) the first equation of (30) is expressed in a "uniform way". To specify it by using (28), the first rows of* E *and* B, *denoted by* E^1 *and* B^1 *respectively, can be rewritten as*

$$\begin{aligned} E^1 &= \left[-(x_2 - s_2^i), \quad (x_1 - s_1^i), \quad \cdots\right], \\ B^1 &= (x_1 - s_1^i)\dot{s}_2^i - (x_2 - s_2^i)\dot{s}_1^i + \dot{a}_1^i\left[(x_1 - s_1^i)^2 + (x_2 - s_2^i)^2\right] \end{aligned} \tag{34}$$

So in the planar case, we have the equations as

$$\begin{aligned} &\left[-(y - s_2^i)\ (x - s_1^i)\right]\begin{bmatrix}\dot{x}\\ \dot{y}\end{bmatrix} = \\ &(x - s_1^i)\dot{s}_2^i - (y - s_2^i)\dot{s}_1^i + \dot{\theta}^i\left[(x - s_1^i)^2 + (y - s_2^i)^2\right], \\ &i = 1, \cdots, N. \end{aligned}$$

For numerical purposes we define in $\mathbb{R}^2$

$$F_1^i = (x - s_1^i)(\dot{y} - \dot{s}_2^i) - (y - s_2^i)(\dot{x} - \dot{s}_1^i) - \dot{a}_1^s\left[(x - s_1^i)^2 + (y - s_2^i)^2\right], \tag{35}$$

and in $\mathbb{R}^3$

$$\begin{aligned} &F_2^i = (\dot{z} - \dot{s}_3^i)\left[(x - s_1^i)^2 + (y - s_2^i)^2\right] - (z - s_3^i)I_3 \\ &-\dot{a}_2^i\sqrt{(x - s_1^i)^2 + (y - s_2^i)^2}\left[(x - s_1^i)^2 + (y - s_2^i)^2 + (z - s_3^i)^2\right]. \end{aligned} \tag{36}$$

They are alternative expressions of (28) and (29).

Then we have in $\mathbb{R}^2$

$$\begin{cases} \frac{\partial F_1^i}{\partial x} = (\dot{y} - \dot{s}_2^i) - 2\dot{a}_1^s(x - s_1^i) \\ \frac{\partial F_1^i}{\partial y} = -(\dot{x} - \dot{s}_1^i) - 2\dot{a}_1^s(y - s_2^i) \\ \frac{\partial F_1^i}{\partial \dot{x}} = -(y - s_2^i) \\ \frac{\partial F_1^i}{\partial \dot{y}} = x - s_1^i, \end{cases} \tag{37}$$

and in $\mathbb{R}^3$

$$\begin{cases} \frac{\partial F_2^i}{\partial x} = 2(\dot{z} - \dot{s}_3^i)(x - s_1^i) - (z - s_3^i)(\dot{x} - \dot{s}_1^i) \\ -\dot{a}_2^i\frac{(x - s_1^i)}{\sqrt{(x - s_1^i)^2 + (y - s_2^i)^2}}S_3 - 2\dot{a}_2^i\sqrt{(x - s_1^i)^2 + (y - s_2^i)^2}(x - s_1^i) \\ \frac{\partial F_2^i}{\partial y} = 2(\dot{z} - \dot{s}_3^i)(y - s_2^i)) - (z - s_3^i)(\dot{y} - \dot{s}_2^i) \\ -\dot{a}_2^i\frac{(y - s_2^i}{\sqrt{(x - s_1^i)^2 + (y - s_2^i)^2}}S_3 - 2\dot{a}_2^i\sqrt{(x - s_1^i)^2 + (y - s_2^i)^2}(y - s_2^i) \\ \frac{\partial F_2^i}{\partial z} = -I - (z - s_3^i)(\dot{z} - \dot{s}_3^i) \\ -2\dot{a}_2^i\sqrt{(x - s_1^i)^2 + (y - s_2^i)^2}(z - s_3^i) \\ \frac{\partial F_2^i}{\partial \dot{x}} = -(z - s_3^i)(x - s_1^i) \\ \frac{\partial F_2^i}{\partial \dot{y}} = -(z - s_3^i)(y - s_2^i) \\ \frac{\partial F_2^i}{\partial \dot{z}} = (x - s_1^i)^2 + (y - s_2^i)^2 - (z - s_3^i)^2. \end{cases} \tag{38}$$

Now for vector sensors the components of the angular velocity, $\dot{a}_j^i$, $i = 1, 2, 3$, $j = 1, 2$ ($j = 1$ for planar case) are measurable. So (37) and (38) can be used directly for the computation of generalized Newton's method.

4 Sensor selection, robustness and scalability

We begin this section by considering the issue of how to choose and activate sensors as the target moves. As is discussed in [8], several criteria can be used for sensor querying and data routing, such as to minimize the number of active sensors for a given accuracy, to optimize the use of multi-modality sensor information or to optimally cover the unknown target dynamics.

In this paper we focus on the criterion to choose sensors such that the observability of the target is assured. Here we only discuss sensor selection for the case of vector sensors.

With vector sensors, the key for target tracking is that (33) is well defined and easy to integrate numerically. Suppose at the present time sensors $k1, \cdots, k_m$ are active, and at $t = t_1$ with the estimated target position at $\tilde{r}(t_1)$, we want to update the active sensors. We use $q_1, \cdots, q_r$ to indicate the new active sensors. Then obviously the new sensors should be chosen such that the E matrix has full rank at $\tilde{r}(t_1)$. Further more the right hand side of (33) should not have a big change at the switch of sensors. Thus we can formulate the sensor switching criterion as follows.

$$\begin{aligned} &\min_{q_1, \cdots, q_r} \|E^T E(s_{q_1}, \cdots, s_{q_r}, \tilde{r}(t_1)) - E^T E(s_{k_1}, \cdots, s_{k_m}, \tilde{r}(t_1))\| \\ &s.t. \;\; rank \; E(s_{q_1}, \cdots, s_{q_r}, \tilde{r}(t_1)) = n. \end{aligned}$$

In practice, one should of course also add in the range constraints.

Since the key computation in the method is to solve the differential equation (33), one can expect that the method will be robust with respect to non-biased noise in the sensor measurements. In practice, one can first use the least square method to estimate r_0, the initial state, then integrate over (33) to track the target.

With the method introduced in the previous section, when more sensors are added to the network, computationally it means that more rows are added to the matrix E in (33). However, the matrix product $E^T E$ remains the same dimension. In fact, more sensors imply easier to compute the inverse of $E^T E$. Thus, the method is well scalable in this respect. With this method, one can easily set a threshold on each sensor, as long as the reading from a sensor is below the threshold, the sensor will be turned off in order to save energy, which just means that there is an all-zero row in the E matrix.

5 An Experiment

In this section we use one experiment to illustrate our algorithms.

Example 5.1 In this example we show some experimental results done by using Scatterweb sensor nodes. Since they measure only the magnitude of motion change, they are the so-called scalar sensors in our definition.

In the experiments four such nodes (only the motion detection system on each of them is used) are used to observe a Khepera mobile robot moving at a constant velocity of 0.12 [m/s] along the x-axis. The nodes are positioned on the vertices of a rectangle of dimension $0.3m \times 0.6m$. They are all oriented toward the center. A cubic spline is used to obtain the continuous representation. The least square approach is applied to estimate the initial value, whereas the sequence values are computed using the generalized Newton Method. The square error distribution at the initial time is shown in Figure 1. The origin is chosen as the initial guess to start the iterative least square process that converges toward the position marked with a dot. The accuracy of the estimated position is best near the middle of the field of view of the sensors. The estimated velocities in this region are shown in Figure 2. There, the target was tracked with a maximal error of below *10cm*. Figure 3 shows the reconstructed trajectory at the points where the square error is below a threshold of 10.3cm. The exact position error is hard to determine as there is significant error in the positioning of the sensors and, more importantly, in the real trajectory of the robot. However, in the four test runs the error never exceeded 10cm. As expected, the Newtons method is much faster than the least square approach.

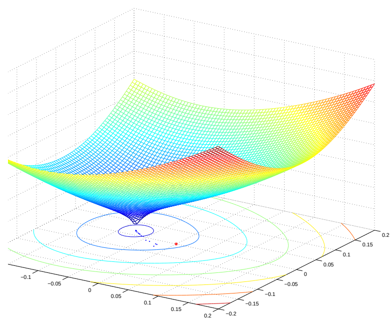

Fig. 1. Estimated square error at the initial time.

6 Conclusion

In this paper we studied the problem of target localization and tracking using network of nonlinear sensors that can only detect the motion of a moving target, which could be a flexible body of n degrees of freedom. Some fundamental

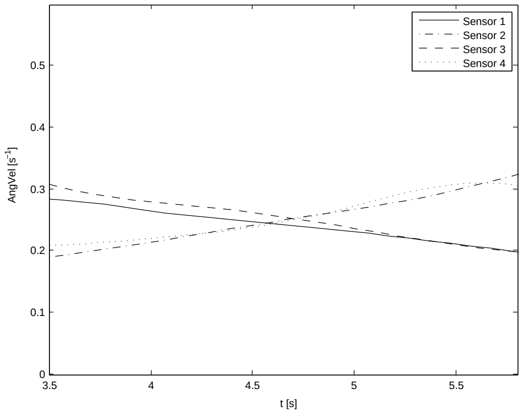

Fig. 2. Estimated velocities in the mid field.

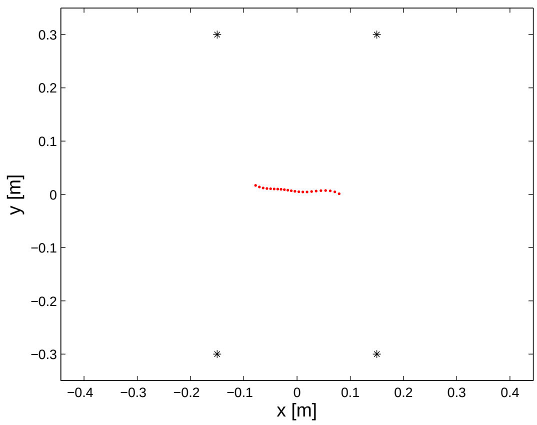

Fig. 3. Reconstructed trajectory in the mid field.

properties about higher dimensional target tracking with motion sensors were studied and some tracking algorithms based on a control theoretic approach were developed. Because of the nonlinearity of the sensors and the higher dimensional objects, the theoretical frame work becomes quite difficult. Thus, the robustness of the algorithms with respect to measurement noises, which is a very important aspect of the problem, has to be left for further investigation.

Acknowledgment This work was supported partly by Sida-VR Swedish Research Links Grant 348-2002-6936, partly by NNSF 60736022, and 60821091 of China, and partly by the Gustafsson Foundation.

References

1. Special issue on sensor networks and applications, *Proceedings of the IEEE*, vol. 91, no. 8, 2003.
2. D. Cheng, B. Ghosh and X. Hu, Distributed Sensor Network for Target Tracking, Proc. of MTNS 2006, Kyoto, July, 2006.
3. N. Hugi, Target tracking using embedded sensor networks, MS thesis, KTH, 2006.
4. D. Reid, An algorithm for tracking multiple targets, *IEEE Trans. Automat. Contr.*, vol. AC-24, Dec. 1979, pp. 843-854.
5. Y.Bar-Shalom and X. R.Li, *Multitarget-Multisensor Tracking: Principles and Techniques*, Storrs, CT: YBS, 1995.
6. I.S. Berezin, N.P. Zhidkov, *Computing Metnods*, translated by O.M. Blunn, Vol. 2, Pergamon Press, Oxford, 1965.
7. L.V. Kamtprpvoch, functional analysis and applied mathematics, *Funktsional'nyi analiz prikladnaya matematika UMN*, Vol. 3, No. 6, 1948.
8. Feng Zhao, Jaewon Shin and J. Reich, Information-driven dynamic sensor collaboration, *IEEE Signal Processing Magazine*, Volume 19, Issue 2, March 2002, pp 61 - 72.
9. R. Brooks, P. Ramanathan and A. Sayeed, Distributed target classification and tracking in sensor networks, *Proceedings of the IEEE*, vol. 91, no. 8, 2003, pp. 1163-1171.
10. F. Zhao, Jie Liu, Juan Liu, J.Guibas and J. Reich, Collaborative signal and information processing: an information-directed approach, *Proceedings of the IEEE*, vol. 91, no. 8, 2003, pp. 1199-1209.
11. Arora, A., Dutta, P., Bapat, S., Kulathumani, V., Zhang, H., Naik, V., Mittal, V., Cao, H., Demirbas, M., Gouda, M., Choi, Y., Herman, T., Kulkarni, S., Arumugam, U., Nesterenko, M., Vora, A., Miyashita, M., A line in the sand: A wireless sensor network for target detection, classification, and tracking, *Computer Networks*, v 46, n 5, Dec 5, 2004, pp.605-634.