A New Approach for Target Motion Analysis in a Binary Sensor Network

Adrien Ickowicz, Jean-Pierre Le Cadre

A New Approach for Target Motion Analysis in a Binary Sensor Network

Adrien Ickowicz and Jean-Pierre Le Cadre

IRISA/CNRS, 35042 Rennes, France
ickowicz, lecadre@irisa.fr

Abstract. The aim of this paper is to present a new concept for target motion analysis within a binary sensor network. For the sake of simplicity, we focus on a constant target motion. The binary information represents a very rough information about the perception of the target motion by an elementary sensor, i.e. is the target approaching or going away. Collecting these binary informations, a first step is to determine the information we can extract at the network level about target motion. Then, based on this step, new concepts are introduced for infering the target motion parameters. One is based upon the separation properties and relies on the SVM formalism; while the other one uses the concept of the velocity plane and the PPR (Projection Pursuit Regression) framework. Moreover, theoretical results about the convergence of this method are also presented.

1 Introduction

We consider a sensor network, made with N sensors (e.g. video),with (known) positions. Each sensor can only gives us a binary $\{-,+\}$ information [1], i.e. whether the target-sensor distance is decreasing $(-)$ or increasing $(+)$. This "choice" can result from severe communication requirements or from the difficulties from fusing inhomogeneous data. Even if many important works deal with proximity sensors [6], [5], we decide here to focus on the binary $\{-,+\}$ information [1]. Here, the aim is to estimate the parameters defining the target trajectory. Even if our methods can be rather easily extended to more complex models of target motion, we decide to focus here on a constant velocity movement. Actually, this framework is sufficiently general to present the main problems we have to face, as well as the foundations of the methods we have to develop for dealing with these binary data. See fig. 1 for an example.

In a first time, the observability requirements are considered. Then, we turn toward the development of specific estimation methods. Especially, the new concept of the velocity plane is introduced as an exhaustive representation of the spatio-temporal sequence of binary data. It is then used both in a separation-oriented framework (SVM) and in a projection pursuit regression (PPR) one. The corresponding methods are carefully presented and analyzed. Simulation results illustrate the behavior of these methods.

Fig. 1. *A view of a getting closer car*

2 Binary Sensor Network Observability Properties

Let us denote s_i a sensor whose position is represented by the the vector $\mathbf{t}_i$ Similarly, the vector $\mathbf{x}_t$ represents the position vector of the target at the time-period t. Let us denote $d_i(t)$ the (time-varying) distance from sensor s_i to the target at time t. Then, we have that:

$$d_i(t) \searrow \Longleftrightarrow \dot{d}_i(t) < 0 \text{ , or: } \langle \mathbf{x}_t - \mathbf{t}_i, \mathbf{v}_t \rangle < 0 \ , \tag{1}$$

where $\mathbf{v}_t$ is the instantaneous target velocity. We thus have the following lemma.

Lemma 1 *Let s_i (resp. s_j) a sensor whose the target distance is decreasing (resp. increasing) at the time-period t, then we have:*

$$\langle \mathbf{t}_j, \mathbf{v}_t \rangle \ < \ \langle \mathbf{x}_t, \mathbf{v}_t \rangle \ < \ \langle \mathbf{t}_i, \mathbf{v}_t \rangle \ . \tag{2}$$

If we restrict to binary motion information, we consider that the output $s_i(t)$ of a sensor (at time t) is $+1$ or -1 according to the distance $d_i(t)$ is decreasing or increasing, so that we have:

$$\begin{cases} s_i(t) = +1 \ \text{ if } \ \dot{d}_i(t) < 0 \ , \\ s_j(t) = -1 \ \text{ if } \ \dot{d}_j(t) > 0 \ . \end{cases} \tag{3}$$

Let us denote A the subset of sensor whose output is $+1$ and B the subset of sensors whose output is -1, i.e. $A = \{\mathsf{s}_i | \ s_i(t) = +1\}$ and $B = \{\mathsf{s}_j | \ s_j(t) = -1\}$ and $C(A)$ and $C(B)$ their convex hulls, then we have [1]:

Proposition 1 $C(A) \bigcap C(B) = \emptyset$ *and* $\mathbf{x}_t \notin C(A) \bigcup C(B)$.

Proof: The proof is quite simple is reproduced here only for the sake of completeness. First assume that $C(A) \bigcap C(B) \neq \emptyset$, this means that there exists an element of $C(B)$, lying in $C(A)$. Let s be this element (and $\mathbf{t}$ its associated position), then we have ($t \in C(B)$):

$$\begin{array}{l}
\mathbf{t} = \sum_{j\in B} \beta_j \ \mathbf{t}_j \ , \ \beta_j \geq 0 \text{ and } \sum_{j\in B} \beta_j = 1 \\
\text{so that we have on the first hand:} \\
\langle \mathbf{t}, \mathbf{v}_t \rangle = \sum_{j\in B} \beta_j \ \langle \mathbf{t}_j, \mathbf{v}_t \rangle \ < \ \langle \mathbf{x}_t, \mathbf{v}_t \rangle \ \text{ (see eq. 2),} \\
\text{and, on the other one } (t \in C(A))\text{:} \\
\langle \mathbf{t}, \mathbf{v}_t \rangle = \sum_{i\in A} \alpha_i \ \langle \mathbf{t}_i, \mathbf{v}_t \rangle \ \geq \left(\sum_{i\in A} \alpha_i \right) \ \min_i \ \{ \langle \mathbf{t}_i, \mathbf{v}(t) \rangle \} > \langle \mathbf{x}_t, \mathbf{v}_t \rangle \ .
\end{array} \tag{4}$$

Thus a contradiction which shows that $C(A) \bigcap C(B) = \emptyset$. For the second part, we have simply to assume that $\mathbf{x}(t) \in C(A)$ ($\mathbf{x}_t = \sum_{i\in A} \alpha_i \ \mathbf{t}_i, \ \alpha_i \geq 0$), which yields:

$$\langle \mathbf{x}_t, \mathbf{v}_t \rangle = \sum_{i\in A} \alpha_i \ \langle \mathbf{t}_i, \mathbf{v}_t \rangle \geq \min_{i\in A} \langle \mathbf{t}_i, \mathbf{v}_t \rangle, \tag{5}$$

which is clearly a contradiction, idem if $X(t) \in C(B)$.
□□□

So, $C(A)$ and $C(B)$ being two disjoint convex subsets, we know that there exists an hyperplane (here a line) separating them. Then, let s_k be a generic sensor, we can write $\mathbf{t}_k = \lambda \ \mathbf{v}_t + \mu \ \mathbf{v}_t^{\perp}$, so that:

$$\langle \mathbf{t}_k, \mathbf{v}_t \rangle = \lambda \left\| \mathbf{v}_t \right\|^2 > 0 \Longleftrightarrow \lambda > 0 \ . \tag{6}$$

This means that the line spanned by the vector $\mathbf{v}_t^{\perp}$ separates $C(A)$ and $C(B)$. Without considering the translation and considering again the $\left\{ \mathbf{v}_t, \mathbf{v}_t^{\perp} \right\}$ basis , we have :

$$\begin{cases} \mathbf{t}_k \in A \Longleftrightarrow \lambda \left\| \mathbf{v}_t \right\|^2 > \langle \mathbf{x}_t, \mathbf{v}_t \rangle \ , \\ \mathbf{t}_k \in B \Longleftrightarrow \lambda \left\| \mathbf{v}_t \right\|^2 < \langle \mathbf{x}_t, \mathbf{v}_t \rangle \ . \end{cases} \tag{7}$$

Thus in the basis $(\mathbf{v}_t, \mathbf{v}_t^{\perp})$, the line passing by the point $\left(\frac{\langle \mathbf{x}_t, \mathbf{v}_t \rangle}{\left\| \mathbf{v}_t \right\|^2}, 0 \right)$ and whose direction is given by $\mathbf{v}_t^{\perp}$ is separating $C(A)$ and $C(B)$. We have now to turn toward the indistinguishability conditions for two trajectories. Two trajectories are said indistinguishable if they induce the same outputs from the sensor network. We have then the following property [1].

Proposition 2 *Assume that the sensor network is dense, then two target trajectories (say $\mathbf{x}_t$ and $\mathbf{y}_t$) are indistinguishable iff the following conditions hold true:*

$$\begin{cases} \dot{\mathbf{y}}_t = \lambda_t \ \dot{\mathbf{x}}_t \ \ (\lambda_t > 0) \ \ \forall t \ , \\ \langle \mathbf{x}_t - \mathbf{y}_t, \dot{\mathbf{x}}_t \rangle = 0 \ \ \forall t \ . \end{cases} \tag{8}$$

Proof: First, we shall consider the implications of the indistinguishability. Actually, the two trajectories are indistinguishable iff the following condition holds:

$$\langle \mathbf{t}_j - \mathbf{t}_i, \dot{\mathbf{x}}_t \rangle \leq 0 \Longleftrightarrow \langle \mathbf{t}_j - \mathbf{t}_i, \dot{\mathbf{y}}_t \rangle \leq 0 \ \ \forall t \ \ \forall (\mathbf{t}_i, \mathbf{t}_j) \ . \tag{9}$$

We then *choose* $\mathbf{t}_j - \mathbf{t}_i = \alpha \, \dot{\mathbf{x}}_t^{\perp}$ (i.e. $\mathbf{t}_i$ and $\mathbf{t}_j$ both belongs to the line separating A and B) and consider the following decomposition of the $\dot{\mathbf{y}}_t$ vector:

$$\dot{\mathbf{y}}_t = \lambda_t \, \dot{\mathbf{x}}_t + \mu_t \, \dot{\mathbf{x}}_t^{\perp} \ ,$$

so that we have:

$$\langle \mathbf{t}_j - \mathbf{t}_i, \dot{\mathbf{y}}_t \rangle = \alpha \mu_t \, \|\dot{\mathbf{x}}_t^{\perp}\|^2 \leq 0 \ . \tag{10}$$

Now, it is always possible to choose a scalar α of the same sign than μ_t. So, we conclude that the scalar μ_t is necessarily equal to zero. Thus , if the trajectories $\mathbf{x}_t$ and $\mathbf{y}_t$ are indistinguishable we have necessarily:

$$\dot{\mathbf{y}}_t = \lambda_t \, \dot{\mathbf{x}}_t \ , \ \forall t \ .$$

Furthermore, the scalar λ_t is necessarily positive (see eq. 9). Then, the lemma 1 inequalities yield:

$$\langle \mathbf{t}_j - \mathbf{t}_i, \dot{\mathbf{x}}_t \rangle \ < \ \langle \mathbf{x}_t - \mathbf{y}_t, \dot{\mathbf{x}}_t \rangle \ < \ \langle \mathbf{t}_i - \mathbf{t}_j, \dot{\mathbf{x}}_t \rangle \ . \tag{11}$$

Choosing once again $\mathbf{t}_j - \mathbf{t}_i = \alpha \, \dot{\mathbf{x}}_t^{\perp}$, we deduce from eq. 11 the second part of prop. 2, i.e. $\langle \mathbf{x}_t - \mathbf{y}_t, \dot{\mathbf{x}}_t \rangle = 0 \ \ \forall t$. Considering now the distance between the two indistinguishable trajectories, we have ($\dot{\mathbf{y}}_t = \lambda_t \, \dot{\mathbf{x}}_t$) :

$$\frac{d}{dt} \, \|\mathbf{x}_t - \mathbf{y}_t\|^2 = 2 \, \langle \mathbf{x}_t - \mathbf{y}_t, \dot{\mathbf{x}}_t - \dot{\mathbf{y}}_t \rangle = 0 \ , \tag{12}$$

so that we have $\|\mathbf{x}_t - \mathbf{y}_t\| = \mathsf{cst}$.
Reciprocally, assume that the two conditions $\dot{\mathbf{y}}_t = \lambda_t \, \dot{\mathbf{x}}_t$ and $\langle \mathbf{x}_t - \mathbf{y}_t, \dot{\mathbf{x}}_t \rangle = 0$ hold true $\forall t$, are the two trajectories then indistinguishable? It is sufficient to remark that:

$$\begin{aligned} &\langle \mathbf{y}_t, \dot{\mathbf{y}}_t \rangle = \langle \mathbf{x}_t + (\mathbf{y}_t - \mathbf{x}_t), \dot{\mathbf{y}}_t \rangle = \langle \mathbf{x}_t, \dot{\mathbf{y}}_t \rangle = \lambda_t \, \langle \mathbf{x}_t, \dot{\mathbf{x}}_t \rangle \ , \\ &\langle \mathbf{t}_i, \dot{\mathbf{y}}_t \rangle = \lambda_t \, \langle \mathbf{t}_i, \dot{\mathbf{x}}_t \rangle \ . \end{aligned} \tag{13}$$

Since the scalar λ_t is positive this ends the proof.
□□□

Let us now consider the practical applications of the above general results.

Rectilinear and uniform motion

Admitting now that the target motions are rectilinear and uniform (i.e. $\mathbf{x}_t = \mathbf{x}_0 + t \, \dot{\mathbf{x}}$). Then prop. 2 yields $\dot{\mathbf{y}} = \lambda \, \dot{\mathbf{x}} \ \ (\lambda > 0)$ and:

$$\langle \mathbf{y}_t - \mathbf{x}_t, \dot{\mathbf{x}} \rangle = \langle \mathbf{y}_0 - \mathbf{x}_0, \dot{\mathbf{x}} \rangle + t \, (1 - \lambda) \, \|\dot{\mathbf{x}}\|^2 = 0 \ \forall t \ . \tag{14}$$

Then, from eq. 14 we deduce that $\lambda = 1$ and $\mathbf{y}_0 = \mathbf{x}_0 + \alpha \, \dot{\mathbf{x}}^{\perp}$. So that, the target velocity is fully observable while the position is uniquely determined modulo a $\alpha \, \dot{\mathbf{x}}^{\perp}$ translation.

leg-by-leg trajectory

Consider now a leg-by-leg trajectory modeling. For a 2-leg one, we have for two indistinguishable trajectories:

$$\begin{cases} \mathbf{x}_t = \mathbf{x}_0 + t_1 \ \mathbf{v}_x^1 + (t - t_1) \ \mathbf{v}_x^2 \ , \\ \mathbf{y}_t = \mathbf{y}_0 + t_1^{'} \ \mathbf{v}_y^1 + (t - t_1^{'}) \ \mathbf{v}_y^2 \ , \end{cases} \tag{15}$$

where $\mathbf{v}_x^i$ is the velocity of the $\mathbf{x}(t)$ trajectory on the i-th leg and t_i is the epoch of maneuver. Furthermore, we can assume that $t_1 < t_1^{'}$. Considering the implications of prop. 2 both for $t < t_1$ and for $t > t_1^{'}$, we know that if the trajectories are indistinguishable we must have:

$$\mathbf{v}_x^1 = \mathbf{v}_y^1 \ \text{ and: } \ \mathbf{v}_x^2 = \mathbf{v}_y^2 \ . \tag{16}$$

So, our objective is now to prove that we have also $t_1 = t_1^{'}$. Considering prop. 2, we thus have the following system of equations :

$$\begin{cases} \langle \mathbf{y}_0 - \mathbf{x}_0 + (t - t_1) \ \left(\mathbf{v}_x^1 - \mathbf{v}_x^2\right), \mathbf{v}_x^1 \rangle = 0 & \text{for} : t_1 < t < t_1^{'} \ \ (a), \\ \langle \mathbf{y}_0 - \mathbf{x}_0 + (t - t_1) \ \left(\mathbf{v}_x^1 - \mathbf{v}_x^2\right), \mathbf{v}_x^2 \rangle = 0 & \text{for} : t_1 < t < t_1^{'} \ \ (b), \\ \langle \mathbf{y}_0 - \mathbf{x}_0 + (t_1^{'} - t_1) \ \left(\mathbf{v}_x^1 - \mathbf{v}_x^2\right), \mathbf{v}_x^2 \rangle = 0 & \text{for} : t_1^{'} < t \qquad\quad (c) \ . \end{cases} \tag{17}$$

Now, on the 1-st leg we have also $\langle \mathbf{y}_0 - \mathbf{x}_0, \mathbf{v}_x^1 \rangle = 0$ (see prop. 2 for $t = 0$), so that eqs 17a,b yield:

$$\langle \left(\mathbf{v}_x^1 - \mathbf{v}_x^2\right), \mathbf{v}_x^1 \rangle = \langle \left(\mathbf{v}_x^1 - \mathbf{v}_x^2\right), \mathbf{v}_x^1 \rangle = 0 \ . \tag{18}$$

This means that $\mathbf{v}_x^1$ and $\mathbf{v}_x^2$ are both orthogonal to the same vector $(\mathbf{v}_x^1 - \mathbf{v}_x^2)$, so they are collinear, and we straightforwardly deduce from eq. 18 that $\mathbf{v}_x^1 = \mathbf{v}_x^2$. Finally, it has thus been proved that $t_1 = t_1^{'}$ and this reasoning can be extended to any leg number. The observability requirements having been considered, we turn now toward the development of the algorithmic approaches. Let us first introduce the following functional.

3 The stairwise functional

Our fist aim is to estimate the target velocity, within a batch processing framework. We assume that N binary ($\{-,+\}$ sensors are uniformly distributed on the field of interest (see fig. 2).

Each sensor will be coupled with a counter, that will be increased by a *unity* each time-period the sensor gives us a $\{+\}$, an will keep its value each time the sensor gives us a $\{-\}$. Then, at the end of the trajectory, each sensor has a entire value representing the number of periods the target was approaching. Within a given batch, the outputs of the sensor counters can be represented by a stairwise functional (see fig. 3).

Then, once this stair is built, we can define what we call the velocity plane. This plane is the tangent plane of the stairwise functional, which means that its direction gives the direction of the stair, while its angle θ gives the slope. The

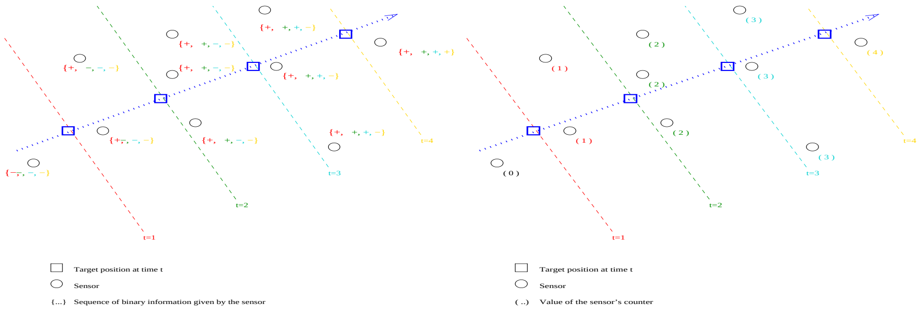

Fig. 2. *A scenario of target evolution and sensor network information*

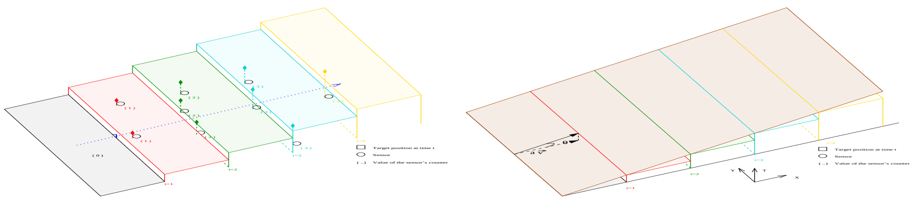

Fig. 3. *The theoretical stairway of the trajectory.*

direction of the plane gives us the target heading, while the target speed v is given by:

$$v = \frac{1}{\tan(\theta)} \ . \tag{19}$$

Thus, estimating the velocity is equivalent to estimating the velocity plane parameters. Mathematical justifications are then presented. The target moves with a constant velocity $\mathbf{v}$. Considering the results of section 2, its starting position is given by the following equation:

$$\begin{aligned} &\mathbf{x}(0) = \mathbf{x}_0 + \lambda \ \mathbf{v}^{\perp} \ \ , \ \lambda \in \mathbb{R} \ , \\ &\text{so, that :} \\ &\mathbf{x}(t) = \mathbf{x}_0 + \lambda \ \mathbf{v}^{\perp} + \mathbf{t} \ \mathbf{v} \end{aligned} \tag{20}$$

This means that at each time period $t \in \mathbb{R}_+$, the possible positions $\mathbf{x}(t)$ define a (moving) straight line, whose direction is $\mathbf{v}^{\perp}$. Let us consider now the scalar product $\langle \mathbf{x}(t), \mathbf{v} \rangle$, then we have:

$$\frac{\partial}{\partial \, t} \langle \mathbf{x}(t), \mathbf{v} \rangle = v^2 \ . \tag{21}$$

This is clearly constant, which means that the surface is a plane. The conclusion follows: the stairwise plane is an exhaustive information for the velocity vector. We provide in the next section two solutions to estimate the velocity plane from the observed data, and give some asymptotic results about the estimation.

4 Statistical Methods to Estimate the Velocity Plane

We showed that estimating the velocity plane allows us to estimate the velocity vector. Wile there exists several methods to do that, we shall focus on two of them.

4.1 The Support Vector Machine (SVM) approach [2]

As seen previously, the problem we have to face is to optimally separate the two classes of sensors (i.e. the $+$ and $-$). So, we can use the general framework of SVM, widely used in the classification context. The set of labeled patterns $\{(y_1, \mathbf{x}_1, \cdots, y_l, \mathbf{x}_l\}$ ($y_i \in \{-1, 1\}$ and $\mathbf{x}_i$ sensor positions) is said to be linearly separable if there exists a vector $\mathbf{w}$ and a scalar b such that the following inequalities hold true:

$$\begin{cases} \langle \mathbf{w}, \mathbf{x}_i \rangle + b \geq 1 \quad \text{if} : y_i = 1 \;, \\ \langle \mathbf{w}, \mathbf{x}_i \rangle + b \leq -1 \quad \text{if} : y_i = -1 \;. \end{cases} \tag{22}$$

Let $\mathcal{H}(\mathbf{w}, b) \stackrel{\Delta}{=} \{\mathbf{x} | \langle \mathbf{w}, \mathbf{x} \rangle + b = 0\}$ ($\mathbf{w}$: normal vector) be this optimal separation plane. and define the margin (marg) as the distance of the closest point $\mathbf{x}_i$ to $\mathcal{H}$, then it is easily seen that $\mathsf{marg} = \frac{1}{\|\mathbf{w}\|}$. Thus, maximizing the margin lead to consider the following problem:

$$\left| \begin{array}{l} \min\limits_{\mathbf{w}, b} \; \tau(\mathbf{w}) \stackrel{\delta}{=} \|\mathbf{w}\|^2 \;, \\ \text{s.t.} \;\; : y_i \left(\langle \mathbf{w}, \mathbf{x}_i \rangle + b \right) \geq 1 \;\; \forall \, i = 1, \cdots, l \;\; y_i = \pm 1 \;. \end{array} \right. \tag{23}$$

Denoting Λ the vector of Lagrange multipliers, dualization of eq. 23 leads to consider again a quadratic problem, but with more explicit constraints [2], i.e. :

$$\left| \begin{array}{l} \max\limits_{\Lambda} W(\Lambda) = -\dfrac{1}{2} \, \Lambda^T \; D \; \Lambda + \Lambda^T \; \mathbf{1} \;, \\ \text{s.t.} : \Lambda \geq 0 \;, \;\; \Lambda^T Y = 0 \;, \end{array} \right. \tag{24}$$

where $\mathbf{1}$ is a vector made of 1 and $Y^T = (y_1, \cdots, y_l)$ is the l-dimensional vector of labels, and D is the Gram matrix:

$$D_{i,j} = \langle y_i \mathbf{x}_i, y_j \mathbf{x}_j \rangle \;. \tag{25}$$

The dualized problem can be efficiently solved by classical quadratic programming methods. The less-perfect case consider the case when data cannot be separated without errors and lead to replace the constraints of eq. 23 by the following ones:

$$y_i \left(\langle \mathbf{w}, \mathbf{x}_i \rangle + b \right) \geq 1 - \xi_i \;, \; \xi_i \geq 0 \;, \; i = 1, \cdots, l \;. \tag{26}$$

Consider now a multiperiod extension of the previous analysis. Let us restrict first to a two-period analysis, we shall consider two separating hyperplanes (say $\mathcal{H}_1, \mathcal{H}_2$) defined by:

$$\begin{cases} \langle \mathbf{w}, x_l^1 \rangle + b_1 \gtreqless \pm c_1 \;\; \text{according to:} \;\; y_l^1 = \pm 1 \;, \\ \langle \mathbf{w}, x_l^2 \rangle + b_2 \gtreqless \pm c_2 \;\; \text{according to:} \; y_l^2 = \pm 1 \;. \end{cases} \tag{27}$$

It is also assumed that these two separating planes are associated with time periods T and $T + \varDelta T$, $\varDelta T$ known. It is easily seen that the margin for the separating plane $\mathcal{H}_1$ is $\frac{c_1}{\|\mathbf{w}\|}$, while for the plane $\mathcal{H}_2$ it is $\frac{c_2}{\|\mathbf{w}\|}$. Thus, the problem we have to solve reads:

$$\left| \begin{array}{l} \min\limits_{\mathbf{w},c_1,c_2,b_1,b_2} \quad \left[\max\limits_{1,2} \left(\dfrac{\|\mathbf{w}\|^2}{c_1^2}, \dfrac{\|\mathbf{w}\|^2}{c_2^2} \right) \right] , \\ \text{s.t.: } \; y_l^1 \; \left(\langle \mathbf{w}, x_l^1 \rangle + b_1 \right) \geq c_1 \; , \; y_l^2 \; \left(\langle \mathbf{w}, x_l^2 \rangle + b_2 \right) \geq c_2 \; \forall l. \end{array} \right. \tag{28}$$

At a first glance, this problem appears as very complicated. But, without restricting generality, we can assume that $c_1 < c_2$. This means that $\max\limits_{1,2} \left(\dfrac{\|\mathbf{w}\|^2}{c_1^2}, \dfrac{\|\mathbf{w}\|^2}{c_2^2} \right) = \dfrac{\|\mathbf{w}\|^2}{c_1^2}$. Making the changes $\frac{1}{c_1}\,\mathbf{w} \rightarrow \mathbf{w}'$ and $\frac{b_1}{c_1} \rightarrow b_1'$ then leads to consider the classical problem:

$$\left| \begin{array}{l} \min\limits_{\mathbf{w}',b_1',b_2'} \; \|\mathbf{w}\|^2 \\ \text{s.t. : } \; y_l^1 \; \left(\langle \mathbf{w}', x_l^1 \rangle + b_1' \right) \geq 1 \; , \; y_l^2 \; \left(\langle \mathbf{w}', x_l^2 \rangle + b_2' \right) \geq 1 \; \forall l. \end{array} \right. \tag{29}$$

Let $\mathbf{w}^*$ be the (unique) solution of eq. 29, then a straightforward calculation yields the distance $d(\mathcal{H}_1^*, \mathcal{H}_2^*)$ between the two separating planes, i.e.:

$$d(\mathcal{H}_1^*, \mathcal{H}_2^*) = \frac{|b_1^* - b_2^*|}{\|\mathbf{w}^*\|} \; .$$

Finally, we deduce that the estimated velocity vector $\hat{\mathbf{v}}$ is given by:

$$\hat{\mathbf{v}} = \alpha \; \mathbf{w}^* \;\; \text{and: } \; \hat{v} = \frac{1}{\varDelta T} \; d(\mathcal{H}_1^*, \mathcal{H}_2^*) \; . \tag{30}$$

The previous analysis can be easily extended to an arbitrary number of periods, as long as the target trajectory remains rectilinear. Another definite advantage is that it can be easily extended to multitarget tracking.

3D-SVM We can also mix the SVM ideas with that of section 3. Indeed, instead of focusing on a 2-D dataset, we can consider a 3-dimensional dataset (sensor coordinates and values of the sensor counters). The second 3-D dataset is the same, but the value of the counter is increased with *unity*. So, the separation plane is 2-D, and will be as closed to the velocity plane as the sensor number can allow. See fig. 4 for a more explicit understanding. The results of the SVM estimation of the velocity plane are discussed in the Simulation Results section.

4.2 Projection Pursuit Regression

The projection pursuit methods have first been introduced by Friedman and Tuckey [3]. Then, they have been developed for regression with the projection

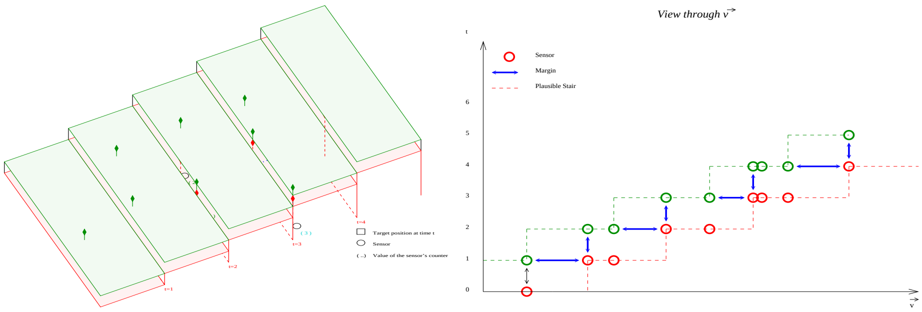

Fig. 4. *The theoretical stairway of the trajectory.*

pursuit regression (PPR) by Friedman and Stuetzle [4]. PPR is mainly a non-parametric method to estimate a regression, with however a certain particularity. Indeed, instead of estimating a function f such as $Y_i = f(X_i) + \varepsilon_i$, where X_i and Y_i are known, and ε_i assuming to follow a certain law, PPR estimates g such as $Y_i = g(X_i\ \theta) + \varepsilon_i$. The first step of the algorithm is to estimate the direction θ, and then $\hat{g}$. In our specific case, θ will represent the direction of the target, and $\hat{g}$ will give us the value of the velocity.

Modeling Let Y_i be the value of the i-th sensor counter. X_i are the sensor coordinates. If $n(X_i\ \theta)$ is the value of the counter i at the end of the track, and p the probability to have the right $\{+, -\}$ decision, we then have ($\mathcal{B}$: binomial):

$$\mathcal{L}(Y_i | X_i\ \theta) = \mathcal{B}(n(X_i\ \theta), p) \tag{31}$$

Assuming in a first time that $p = 1$, the two parameters we would like to estimate are the θ parameter and the $n(.)$ function.

The PPR method in the network context We have some additional constraints on $n(.)$. First of all, it only takes integer values. Then, it is an increasing function (because $p = 1$). The optimization problem we have to solve is the following:

$$\hat{\theta} = \arg\min_{\theta} \sum \left(\hat{n}(X_i\ \theta) - Y_i\right)^2 , \tag{32}$$

where $\hat{n}$ is calculated in a quite special way. First, we define a non parametric estimation of a function f, via:

$$\hat{f}(u) = \frac{\sum Y_i K_h(X_i\ \theta - u)}{\sum K_h(X_i\ \theta - u)} . \tag{33}$$

Then, we sort $(X\ \theta)_i$ into a vector $(X\theta)_{(i)}$ from the smallest to the biggest. After which we define $\hat{n}(.)$ via:

$$\begin{cases} \hat{n}(X\ \theta_{(i)}) = \hat{f}(X\ \theta_{(i)}) & \text{if} \quad \hat{f}(X\ \theta_{(i)}) \geq \hat{f}(X\ \theta_{(i-1)}) , \\ \hat{n}(X\ \theta_{(i)}) = \hat{f}(X\ \theta_{(i-1)}) & \text{otherwise} . \end{cases} \tag{34}$$

Sometimes, due to the integer value of the estimated $n(.)$ function, we have to deal with many possible values of $\widehat{\theta}$. Then, in this case, we choose the mean value of θ. Due to the specific behavior of our target and our modeling, we know in addition that the general form of n (say $\tilde{n}$) is given by:

$$\tilde{n}(u) = \sum i\mathbb{I}_{[(X\theta)^{\perp}+(i-1)v,(X\theta)^{\perp}+iv]}(u) \ . \tag{35}$$

The next step is then to estimate v. Such an estimation is given by the following optimization program:

$$\hat{v} = \arg\min_{v} \sum (\hat{\tilde{n}}(X\widehat{\theta}_i) - Y_i)^2 \tag{36}$$

Convergence We will study if the estimation is good with an infinite number N of sensors. Assuming we have an infinite number of sensors in a closed space, this means that each point of the space gives us an information $\{+,-\}$. We then will have the exact parameters of the stairwise functional. To that aim, we will show in the following paragraph that the probability of having a sensor arbitrary close to the limits of each stair steps is 1. We assume that the sensor positions are randomly distributed, following an uniform law. Then, y being fixed:

$$\mathcal{L}(X|Z) = \mathcal{U}_{[B_{inf};B_{sup}]} \tag{37}$$

If the velocity vector $\mathbf{v}$ is denoted with $[a;b]$, then:

$$B_{\text{inf}} = -\frac{b}{a}y - \frac{c_{\text{inf}}}{a} \ , \qquad B_{\text{sup}} = -\frac{b}{a}y - \frac{c_{\text{sup}}}{a} \ , \tag{38}$$

where $(c_{\text{inf}}, c_{\text{sup}})$ only depends on $\mathbf{v}$ and $\mathbf{x}_0$, which means that they are deterministic, and independent from X. It is quite obvious that B_{inf} represents the smaller x-limit of a step, when B_{sup} represents its higher x-limit. Then, considering the velocity plane, B_{inf} and B_{sup} both belong to the plane. Denote $u = \inf_i(X_i)$, then:

$$\begin{aligned} \forall\varepsilon > 0 \qquad P(|u - B_{\text{inf}}| < \varepsilon) &= P(u - B_{\text{inf}} < \varepsilon) \ , \\ &= P(u < \varepsilon + B_{\text{inf}}) \ . \end{aligned} \tag{39}$$

We know that:

$$\mathcal{L}(X|Z) = \mathcal{U}_{[B_{\text{inf}};B_{\text{sup}}]} \Rightarrow P(\inf X_i \leq t) = \begin{cases} 0 & \text{if } t \leq B_{\text{inf}} \ , \\ 1 - (\frac{B_{\text{sup}}-t}{B_{\text{sup}}-B_{\text{inf}}})^N & \text{if } t \in [B_{\text{inf}}; B_{\text{sup}}] \ , \\ 1 & \text{if } t > B_{\text{sup}} \ . \end{cases} \tag{40}$$

Then, we have the following probability calculations:

$$\begin{aligned} \forall\varepsilon > 0 \qquad P(|u - B_{\text{inf}}| < \varepsilon) &= P(u < \varepsilon + B_{\text{inf}}) \ , \\ &= 1 - (\frac{B_{\text{sup}} - (\epsilon + B_{\text{inf}})}{B_{\text{sup}} - B_{\text{inf}}})^N 1_{[B_{\text{inf}};B_{\text{sup}}]}((\varepsilon + B_{\text{inf}})) \ , \\ &= \begin{cases} 0 & \text{if } \varepsilon \leq 0 \\ 1 - (1 - \frac{\varepsilon}{B_{\text{sup}}-B_{\text{inf}}})^N & \text{if } \epsilon \in]0; B_{\text{sup}} - B_{\text{inf}}] \\ 1 & \text{if } \varepsilon > B_{\text{sup}} - B_{\text{inf}} \ . \end{cases} \end{aligned} \tag{41}$$

Given the above equation, $1 - \frac{\varepsilon}{B_{\text{sup}} - B_{\text{inf}}}$ is smaller than one, which means that $(1 - \frac{\epsilon}{B_{sup} - B_{inf}})^N$ converges to 0 as N increases to infinity. Thus, we have finally:

$$\forall \varepsilon > 0 \qquad \lim_{N \to \infty} \qquad P(|u - B_{\text{inf}}| < \varepsilon) = 1 \; . \tag{42}$$

ending the proof.

5 Simulation Results

We shall now investigate the previous developments via simulations. The first figure (fig.5) will show the stair built by the previously explained method (N: fixed). The position of the sensors are considered random, following a uniform law on the surveillance set. To evaluate the performance of our methods, we

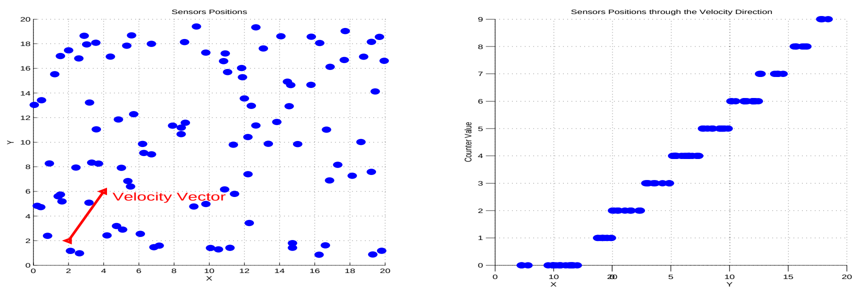

Fig. 5. *2D sensor position and velocity's direction projection. N=100 sensors, Velocity Vector is [1,2]*

decided to calculate the mean square error of the two estimated parameters, which are the velocity value and the velocity direction. Fig. 6 shows the two MSEs values for both direction and velocity values, assuming the sensor number is growing from 10 to 100, and the velocity vector is the $[1, 2]$ vector (m/s).

Providing 2000 simulations, the MSEs seems to be unstable. However, the two parameter estimation methods leads to a very different conclusion. In the case of the direction estimation, the PPR method works highly better than the SVM method, and seems quite stable as the sensor number N grows. On the other side, the SVM method is more erratic.One possible explanation is that the PPR method has been first developed for the particular case of direction estimation, while the SVM method is more focused on the margins maximization, which means in our case a simultaneous estimation of both parameters. The conclusions we can make on the velocity value estimation are rather opposite. The MSE becomes reasonable only for the SVM method, and for a number of sensors up to 60. Indeed, we have a 0.05 m/s error on a velocity value estimation for a theoretical value of $\sqrt{5}$. As erratic as the SVM's MSE was in the direction estimation, it was however less erratic than the result we have for the PPR value. One answer to the MSE erratic value for the PPR could be to find a best way

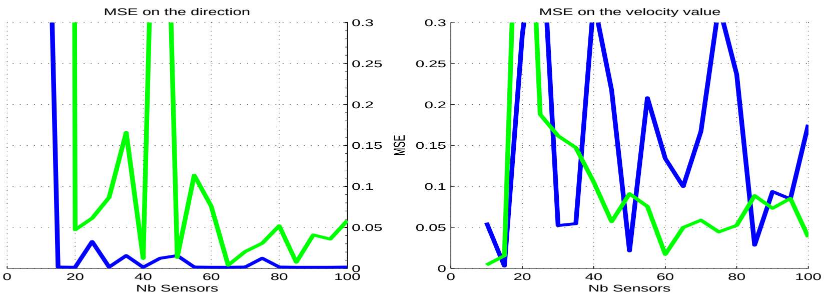

Fig. 6. *Mean Square Error of the Velocity Estimators. Green for the SVM, Blue for the PPR.*

to estimate the velocity value. Indeed, in our case, we choose for estimating functional a sum of indicators functions. However, it is not clear that this optimization gives a single minimum solution. There could be a finest functional that could lead to a most robust optimization solution, and this would be the subject of future works.

6 Conclusion

In this paper, we chose to focus on the use of the $\{-,+\}$ at the level of information processing for a sensor network. Though this information is rather poor, it has been shown that it can provide very interesting results about the target velocity estimation. The theoretical aspects of our methods have been thoroughly investigated, and it has been shown that the PPR method leads to the right velocity plane if the number of sensors increase to infinity. The feasibility of the new concept ("velocity plane") for estimating the target trajectory parameters has been put in evidence. The proposed methods seem to be sufficiently general and versatile to explore numerous extensions like: target tracking and dealing with multiple targets within the same binary context.

References

1. J. Aslam, Z. Butler, F. Constantin, V. Crespi, G. Cybenko, D. Rus, Tracking a moving object with a binary sensor network. *Proc. of the 1st international Conference on Embedded Networked Sensor Systems*, Nov 2005, pp. 150–161.
2. C. Cortes, V. Vapnik, Support-Vector Networks. Machine Learning, **20**, 1995, pp. 273–297.
3. J.H. Friedman and J. H. Tuckey, A Projection Pursuit Algorithm for Exploratory Data Analysis. IEEE Trans. Comput. **23**, 1974, pp. 881–889.
4. J. H. Friedman and W. Stuetzle, Projection Pursuit Regression. J. Amer. Stat. Soc., **76**, 1981, pp. 817–823.
5. L. Lazos, R. Poovendran and J.A. Ritcey , Probabilistic Detection of Mobile Targets in Heterogeneous Sensor Networks. *Proc. of the 6-th IPSN*, Apr. 2007.
6. X. Wang and B. Moran , Multitarget Tracking Using Virtual Measurements of Binary Sensor Networks. *Proc. of the 9-th Int. Conf. on Information Fusion*, Jul. 2006.