Recursive Bayesian Identification of Nonlinear Autonomous Systems

Tiago Simão, Miguel Barão, Jorge S. Marques

***Abstract*— This paper concerns the recursive identification of nonlinear discrete-time systems for which the original equations of motion are not known. Since the true model structure is not available, we replace it with a generic nonlinear model. This generic model discretizes the state space into a finite grid and associates a set of velocity vectors to the nodes of the grid. The velocity vectors are then interpolated to define a vector field on the complete state space. The proposed method follows a Bayesian framework where the identified velocity vectors are selected by the maximum a posteriori (MAP) criterion. The resulting algorithms allow a recursive update of the velocity vectors as new data is obtained. Simulation examples using the recursive algorithm are presented.**

I. Introduction

The motion of objects has received a special attention in several research areas. In the center of that research is human motion analysis in the surveillance area [1], [2], [3]. With the purpose of identifying and studying actions of living beings or objects with motion, a trajectory can be obtained by resorting to image processing techniques [2], [4], [5], [6]. In this paper, we assume that video sequences are acquired and processed to generate sample of trajectories. This trajectories are then used to identify a vector field in the image space that best describes it [5], [6], [7].

We assume that the original system generating the observed trajectory is unknown. Therefore, the vector field is obtained through the interpolation of a set of nodes in a uniform grid, where both norm and direction are needed to reproduce the trajectory. We can assume the vector field models as a nonlinear system, where we only have access to the output measurements defined as the trajectory positions. Previous works related with the identification of nonlinear systems already encompass a variety of approaches [8], [9], [10], [11], [12], [13]. The algorithms used to estimate parameters include expectation-maximization [14] and particle filters [15], [16].

Trajectory analysis based on vector fields has already been performed by [5], [6], which is similar to the Gaussian process approaches [17] . However, parameter estimation is done off-line and the update of a new step requires the computation of the complete trajectory over again. Here, we first present the off-line approach as the standard solution and then describe the recursive parameter estimation solution, which can be obtained from the former. A problem arises when a new position is added to the trajectory. The previous trajectory information must be stored to repeat the estimation of the vector field including the new position, but without computing all the trajectory positions already processed.

The main contribution of this work is the recursive version of the Bayesian algorithm developed in [5]. Our algorithm achieves the same results by updating a sufficient statistic of the past data. This update is done in constant time and memory complexity and avoids some matrix inversions that appear in the original problem.

In section II we present an approach method to the object motion problem including the space discretization. Section III introduces the vector field prior we used for the MAP solution and sections IV and V describe the off-line and on-line vector field estimation. Section VI shows some synthetic examples using the recursive vector field estimation and in section VII we present the conclusions and future work.

II. Problem formulation

Our objective is to estimate a vector field that best describes an observed trajectory. We consider that each trajectory position is given by

$$x_{t+1} = x_t + T(x_t) + w_t, \tag{1}$$

where $x_t \in \mathbb{R}^{D\times 1}$ is the space position with dimension D at instant t and $T(x_t) \in \mathbb{R}^{D\times 1}$ is the velocity vector (or step) at position x_t. We assume that $w(t) \sim \mathcal{N}_D(0, \Sigma_t)$ is a zero-mean multivariable Gaussian error with covariance matrix Σ_t.

We can represent the next position in the trajectory, given the current position, as the conditional probability

$$\begin{aligned} p(x_{t+1}|x_t) &= \mathcal{N}_D(x_{t+1}|\, x_t + T(x_t), \Sigma_t) \\ &= \tfrac{1}{\sqrt{(2\pi)^D|\Sigma_t|}} e^{-\frac{1}{2}\|x_{t+1}-x_t-T(x_t)\|^2_{\Sigma_t^{-1}}}, \end{aligned} \tag{2}$$

where $|\Sigma_t|$ is the determinant of the matrix Σ_t and where we define $\|X\|_A^2 \triangleq \operatorname{Tr}\left(X^\mathsf{T} A X\right)$ with $X \in \mathbb{R}^{m\times n}$ and $A \in \mathbb{R}^{m\times m}$ for any positive integer m, n. The $\operatorname{Tr}(B)$ form represents the trace of the square matrix B. Equation (2) can also be referred to as the *step probability distribution*.

Consider that $X_L = [x_1, x_2, \ldots, x_{L+1}] \in \mathbb{R}^{D\times(L+1)}$ contains all the $L+1$ positions of a single trajectory. The

The work reported in this paper was made in the framework of project *ARGUS - Activity recognition and object tracking based on multiple models*, financed by FCT Portugal under contract PTDC/EEA-CRO/098550/2008 and by INESC-ID multiannual funding through the PIDDAC Program Funds.

T. Simão, M. Barão are with INESC-ID, Av. Rovisco Pais, 1, 1049-001, Lisboa, Portugal, tiago.simao@ist.utl.pt, mjsb@ramses.inesc-id.pt

J. S. Marques is with the Instituto de Sistemas e Robótica, Instituto Superior Técnico, Av. Rovisco Pais, 1, 1049-001, Lisboa, Portugal, jsm@isr.ist.utl.pt

joint probability of the complete trajectory is

$$\begin{aligned} p(X_L) &= p(x_1, \ldots, x_{L+1}) \\ &= p(x_1)\, p(x_2|\, x_1) \ldots\, p(x_{L+1}|\, x_L) \\ &= p(x_1) \prod_{t=1}^{L} p(x_{t+1}|\, x_t), \end{aligned} \tag{3}$$

where $p(x_1)$ is the probability distribution of the initial position.

As we cannot compute the velocity vector $T(x_t)$ for an infinite number of positions in the D-dimensional space, we need to perform a space discretization. Therefore, the space is discretized into a finite grid and a set of velocity vectors t_n are associated to the nodes of the grid. We then assume that the velocity vector $T(x)$ is obtained by interpolation

$$T(x_t) \triangleq \sum_{n=1}^{N} t_n \phi_n(x_t), \tag{4}$$

where N is the total number of nodes, $t_n \in \mathbb{R}^{D\times 1}$ is the velocity vector at node n and $\phi_n(x_t) \in \mathbb{R}$ is a normalized weight function [18]

$$\phi_n(x_t) \triangleq \frac{w_n(x_t)}{\sum_{j=1}^{N} w_j(x_t)}, \tag{5}$$

where the weights

$$w_n(x_t) \triangleq \big(d(x_t, g_n) + d_0\big)^{-p}, \tag{6}$$

represent the inverse of the distance $d(x_t, g_n)$ from position x_t to node g_n. The parameter $p > 0$ adjusts the smoothness and $d_0 > 0$ is a small additive constant to avoid an infinite weight w_n when $x_t = g_n$.

Equation (4) can be rewritten as

$$T(x_t) = \mathcal{T}\, \Phi_{x_t}, \tag{7}$$

where matrices $\mathcal{T} \in \mathbb{R}^{D\times N}$ and $\Phi_{x_t} \in \mathbb{R}^{N\times 1}$ are defined by

$$\mathcal{T} \triangleq [t_1,\, t_2,\, \ldots,\, t_N]\,, \quad \Phi_{x_t} \triangleq \begin{bmatrix} \phi_1(x_t) \\ \phi_2(x_t) \\ \vdots \\ \phi_N(x_t) \end{bmatrix}. \tag{8}$$

We define the velocity at position x_t as $v_{x_t} \triangleq x_{t+1} - x_t$ and use equation (7) to represent the multivariable Gaussian distribution of equation (2) as

$$p(x_{t+1}|\, x_t, \mathcal{T}) = \frac{1}{\sqrt{(2\pi)^D\, |\Sigma_t|}}\, e^{-\frac{1}{2}\|v_{x_t} - \mathcal{T}\Phi_{x_t}\|^2_{\Sigma_t^{-1}}}, \tag{9}$$

where $\mathcal{T}$ parameterizes the vector field we want to estimate.

The estimation of matrix $\mathcal{T}$ is done within a Bayesian framework. In this framework an observation model and a prior distribution are required to produce a posterior distribution of $\mathcal{T}$ given the observed trajectory. We use equation (9) as the observation model, and then define a prior distribution $p(\mathcal{T})$ for the vector field.

III. Vector field prior

Since there is no prior information on particular directions of the vector field, the prior distribution does not convey that type of information. However, we can state that the vector field has some degree of smoothness so that neighboring nodes follow similar directions.

We represent the vector field prior by $p(\mathcal{T})$. Let $\mathcal{I}$ denote the set of pairs of indices (i, j) containing neighboring nodes, *i.e.*,

$$\mathcal{I} = \{(i, j)|\, i \text{ and } j \text{ are neighbors, and } i \neq j\}.$$

In a 2-dimensional space, we choose the neighbors of a node as the closest ones in all vertical, horizontal and diagonal directions. Therefore, a node in the middle of a 2-dimensional grid has 8 neighboring nodes, in the border it has 5 neighboring nodes and in a corner it has 3 neighboring nodes.

Assuming that neighboring nodes have similar directions, the prior is defined as a multivariable Gaussian

$$p(\mathcal{T}) \propto e^{-\frac{1}{2\sigma}\sum_{(i,j)\in\mathcal{I}} \|t_i - t_j\|^2}. \tag{10}$$

Letting $\Delta \in \{-1, 0, 1\}^{N\times(\#\mathcal{I})}$ denote the matrix that operates the differences between neighbors, we obtain

$$\begin{aligned} \mathcal{T}\Delta &= [t_1,\, t_2,\, \ldots,\, t_N] \begin{bmatrix} 1 & \cdots & 0 & \cdots & 0 \\ -1 & \cdots & 1 & \cdots & 0 \\ 0 & \cdots & 0 & \cdots & 0 \\ 0 & \cdots & -1 & \cdots & 0 \\ \vdots & & \vdots & \ddots & \vdots \\ 0 & \cdots & 0 & \cdots & 1 \\ 0 & \cdots & 0 & \cdots & -1 \end{bmatrix} \\ &= [t_1 - t_2,\, \ldots,\, t_2 - t_4,\, \ldots,\, t_{N-1} - t_N]\,, \end{aligned} \tag{11}$$

where each element $t_i - t_j$ measures the change of velocity between two neighboring nodes $(i, j) \in \mathcal{I}$.

The exponent in equation (10) can be written as a Frobenius norm on $\mathcal{T}$ as[1]

$$\begin{aligned} \sum_{(i,j)\in\mathcal{I}} \|t_i - t_j\|^2 &= \mathrm{Tr}\Big((\mathcal{T}\Delta)^{\mathsf{T}}(\mathcal{T}\Delta)\Big) = \mathrm{Tr}\Big((\mathcal{T}\Delta)(\mathcal{T}\Delta)^{\mathsf{T}}\Big) \\ &= \mathrm{Tr}\Big(\mathcal{T}\underbrace{\Delta\Delta^{\mathsf{T}}}_{\Lambda}\mathcal{T}^{\mathsf{T}}\Big) = \|\mathcal{T}^{\mathsf{T}}\|^2_{\Lambda}. \end{aligned} \tag{12}$$

Therefore, the prior multivariable Gaussian can be represented by

$$p(\mathcal{T}) \propto e^{-\frac{1}{2\sigma}\|\mathcal{T}^{\mathsf{T}}\|^2_{\Lambda}}. \tag{13}$$

The parameter σ is adjusted manually, according to the complexity of a trajectory, and it tunes the dependency of neighboring nodes, thereby controlling the smoothness of the vector field.

[1] Equation (12) is achieved based on trace properties and the definition of $\|X\|^2_A$ in section II.

IV. Maximum a Posteriori Solution / MAP

Our objective is to estimate the vector field parameterized by matrix $\mathcal{T}$, given a set of positions of a trajectory X. Applying the Bayes' law, we get to the posterior distribution

$$p(\mathcal{T}|\,X) = C\,p(X|\,\mathcal{T})p(\mathcal{T}), \tag{14}$$

where $C = \frac{1}{p(X)}$ is a normalization factor, $p(X|\,\mathcal{T})$ is the trajectory joint probability model obtained in (3) and $p(\mathcal{T})$ is the prior defined in (13).

For estimation purposes, we use the *maximum a posteriori* (MAP) criterion, i.e., we must find the parameters $\mathcal{T}$ that maximize $p(\mathcal{T}|\,X)$

$$\begin{aligned} \hat{\mathcal{T}} &= \arg\max_{\mathcal{T}}\, p(\mathcal{T}|\,X) = \arg\max_{\mathcal{T}}\, \log p(\mathcal{T}|\,X) \\ &= \arg\max_{\mathcal{T}} \Big(\log C + \log p(X|\,\mathcal{T}) + \log p(\mathcal{T})\Big). \end{aligned} \tag{15}$$

To solve equation (15) we find the stationarity points with

$$\left.\frac{\partial \log p(\mathcal{T}|\,X)}{\partial \mathcal{T}}\right|_{\mathcal{T}=\hat{\mathcal{T}}} = 0. \tag{16}$$

Since $\log p(\mathcal{T}|\,X)$ is concave, the unique stationarity point found is the global maximum.

Separately considering each term in (15), we can discard the first, $\log C$, since it does not depend on $\mathcal{T}$.

The second term can be obtained with the logarithm of equation (3) and using equation (9), resulting in

$$\begin{aligned} \log p(X|\,\mathcal{T}) &= \log p(x_1) + \sum_{t=1}^{L} \log p(x_{t+1}|\,x_t) \\ &= \log p(x_1) - \frac{1}{2}\sum_{t=1}^{L} \log (2\pi)^D\,|\Sigma| \\ &\quad - \frac{1}{2}\sum_{t=1}^{L} \|v_{x_t} - \mathcal{T}\Phi_{x_t}\|^2_{\Sigma^{-1}}. \end{aligned} \tag{17}$$

We assume a constant disturbance covariance matrix $\Sigma = \Sigma_t$ and define $\Phi_X \in \mathbb{R}^{N\times L}$ and $v_X \in \mathbb{R}^{D\times L}$ by

$$\begin{aligned} \Phi_X &\triangleq [\Phi_{x_1},\, \Phi_{x_2},\, \cdots,\, \Phi_{x_L}] \\ v_X &\triangleq [v_{x_1},\, v_{x_2},\, \cdots,\, v_{x_L}]\,. \end{aligned} \tag{18}$$

Performing some calculations based on matrix trace properties, we get to

$$\begin{aligned} \log p(X|\,\mathcal{T}) &= \log p(x_1) - \frac{L}{2}\log (2\pi)^D\,|\Sigma| \\ &\quad - \frac{1}{2}\|v_X - \mathcal{T}\Phi_X\|^2_{\Sigma^{-1}}, \end{aligned} \tag{19}$$

which resembles the least squares form. The prior term from equation (15) can be obtained with the logarithm of equation (13)

$$\begin{aligned} \log p(\mathcal{T}) &= \xi - \tfrac{1}{2\sigma}\|\mathcal{T}^{\mathsf{T}}\|^2_{\Lambda} \\ &= \xi - \tfrac{1}{2\sigma}\mathrm{Tr}\big(\mathcal{T}\Lambda\,\mathcal{T}^{\mathsf{T}}\big), \end{aligned} \tag{20}$$

where ξ is a constant independent from $\mathcal{T}$.

Computing the derivatives separately yields

$$\begin{aligned} \frac{\partial}{\partial \mathcal{T}} \log p(X|\,\mathcal{T}) &= \frac{\partial}{\partial \mathcal{T}}\left(-\frac{1}{2}\|v_X - \mathcal{T}\Phi_X\|^2_{\Sigma^{-1}}\right) \\ &= \Phi_X\big(v_X - \mathcal{T}\,\Phi_X\big)^{\mathsf{T}}\Sigma^{-1} \\ &= \Phi_X v_X^{\mathsf{T}}\Sigma^{-1} - \Phi_X\Phi_X^{\mathsf{T}}\mathcal{T}^{\mathsf{T}}\Sigma^{-1}, \end{aligned} \tag{21}$$

and

$$\frac{\partial}{\partial \mathcal{T}} \log p(\mathcal{T}) = -\tfrac{1}{\sigma}\Lambda\,\mathcal{T}^{\mathsf{T}}. \tag{22}$$

Therefore, the result is

$$\begin{aligned} \frac{\partial}{\partial \mathcal{T}} \log p(\mathcal{T}|\,X) &= \frac{\partial}{\partial \mathcal{T}} \log p(X|\,\mathcal{T}) + \frac{\partial}{\partial \mathcal{T}} \log p(\mathcal{T}) \\ &= \Phi_X v_X^{\mathsf{T}}\Sigma^{-1} - \Phi_X\Phi_X^{\mathsf{T}}\mathcal{T}^{\mathsf{T}}\Sigma^{-1} - \tfrac{1}{\sigma}\Lambda\,\mathcal{T}^{\mathsf{T}}. \end{aligned} \tag{23}$$

Replacing it into equation (16) and transposing yields

$$-\Sigma^{-\mathsf{T}}\mathcal{T}\Phi_X\Phi_X^{\mathsf{T}} - \tfrac{1}{\sigma}\mathcal{T}\Lambda + \Sigma^{-\mathsf{T}}v_X\Phi_X^{\mathsf{T}} = 0, \tag{24}$$

which is a linear equation in $\mathcal{T}$ of the Sylvester type.

If we further assume that $\Sigma = \kappa I_D$ where $I_D \in \mathbb{R}^{D\times D}$ is the identity matrix, we achieve

$$\begin{aligned} &-\tfrac{1}{\kappa}\mathcal{T}\Phi_X\Phi_X^{\mathsf{T}} - \tfrac{1}{\sigma}\mathcal{T}\Lambda + \tfrac{1}{\kappa}v_X\Phi_X^{\mathsf{T}} = 0 \\ \Leftrightarrow\ &-\mathcal{T}\left(\tfrac{1}{\kappa}\Phi_X\Phi_X^{\mathsf{T}} + \tfrac{1}{\sigma}\Lambda\right) + \tfrac{1}{\kappa}v_X\Phi_X^{\mathsf{T}} = 0 \\ \Leftrightarrow\ &\quad \mathcal{T} = v_X\Phi_X^{\mathsf{T}}\big(\Phi_X\Phi_X^{\mathsf{T}} + \tfrac{\kappa}{\sigma}\Lambda\big)^{-1}. \end{aligned} \tag{25}$$

Equation (25) defines the standard batch vector field estimate for a trajectory of length L. As the trajectory length increases, so do the dimensions of matrices v_X and Φ_X increase, along with the time complexity to estimate $\mathcal{T}$. Therefore, this solution can only be used in constrained problems.

V. Recursive implementation

In this section, we propose a recursive version of the algorithm that does not suffer from the time and memory complexity of the batch mode version. Specifically, we seek an algorithm that has constant memory and computational complexity. These objectives can be achieved by using a sufficient statistic of the data that is updated at each step as new data is acquired.

We start by rewriting equation (3) as

$$p(X_t) = \underbrace{p(x_1)\prod_{i=1}^{t-1} p(x_{i+1}|\,x_i)}_{p(X_{t-1})}\,p(x_{t+1}|\,x_t), \tag{26}$$

where $p(X_{t-1})$ is a known joint probability of the complete trajectory on the previous time instant $t-1$. The probability for the new step $p(x_{t+1}|\,x_t, \mathcal{T}_t)$ follows equation (9). Therefore, the Bayes' law can be represented as

$$\begin{aligned} p(\mathcal{T}_t|\,X_t) &= C_t\,p(X_t|\,\mathcal{T}_t)p(\mathcal{T}_t) \\ &= C_t\,p(x_{t+1}|\,x_t, \mathcal{T}_t)\overbrace{p(X_{t-1}|\,\mathcal{T}_t)p(\mathcal{T}_t)}^{Prior}, \end{aligned} \tag{27}$$

where now the prior includes all the previous information.

The stationary condition (16) can now be rewritten as

$$\begin{aligned}\frac{\partial}{\partial \mathcal{T}_t}\log p(\mathcal{T}_t|\, X_t) &= \frac{\partial}{\partial \mathcal{T}_t}\log p(x_{t+1}|\, x_t, \mathcal{T}_t)\\ &+\frac{\partial}{\partial \mathcal{T}_t}\log\big(p(X_{t-1}|\, \mathcal{T}_t)p(\mathcal{T}_t)\big)\\ &= \Phi_{x_t}v_{x_t}^{\mathsf{T}}\Sigma_t^{-1} - \Phi_{x_t}\Phi_{x_t}^{\mathsf{T}}\mathcal{T}_t^{\mathsf{T}}\Sigma_t^{-1}\\ &+\Phi_X v_X^{\mathsf{T}}\Sigma^{-1} - \Phi_X\Phi_X^{\mathsf{T}}\mathcal{T}_t^{\mathsf{T}}\Sigma^{-1} - \tfrac{1}{\sigma}\Lambda\mathcal{T}_t^{\mathsf{T}},\end{aligned} \tag{28}$$

where now Φ_X and v_X correspond to data obtained before time t.

Again, assuming $\Sigma_t = \Sigma = \kappa I_D$ we get

$$\tfrac{1}{\kappa}v_{x_t}\Phi_{x_t}^{\mathsf{T}} + \tfrac{1}{\kappa}v_X\Phi_X^{\mathsf{T}} = \mathcal{T}_t\left(\tfrac{1}{\kappa}\Phi_{x_t}\Phi_{x_t}^{\mathsf{T}} + \tfrac{1}{\kappa}\Phi_X\Phi_X^{\mathsf{T}} + \tfrac{1}{\sigma}\Lambda\right), \tag{29}$$

leading to the solution

$$\mathcal{T}_t = \left(v_{x_t}\Phi_{x_t}^{\mathsf{T}} + v_X\Phi_X^{\mathsf{T}}\right)\left(\Phi_{x_t}\Phi_{x_t}^{\mathsf{T}} + \Phi_X\Phi_X^{\mathsf{T}} + \tfrac{\kappa}{\sigma}\Lambda\right)^{-1}. \tag{30}$$

Equation (30) can be recursively computed by using

$$\begin{aligned}&\alpha(t) = v_{x_t}\Phi_{x_t}^{\mathsf{T}} + \alpha(t-1), \quad \alpha(t-1) = v_X\Phi_X^{\mathsf{T}},\\ &\beta(t) = \Phi_{x_t}\Phi_{x_t}^{\mathsf{T}} + \beta(t-1), \quad \beta(t-1) = \Phi_X\Phi_X^{\mathsf{T}},\end{aligned} \tag{31}$$

where $\alpha, \beta \in \mathbb{R}^{N\times N}$ are two matrices that store past information needed in the next iteration. The estimated vector field at instant t is given by

$$\mathcal{T}(t) = \alpha(t)\left(\beta(t) + \tfrac{\kappa}{\sigma}\Lambda\right)^{-1}, \tag{32}$$

with $\alpha(0) = 0$ and $\beta(0) = 0$. Moreover, since the denominator $\beta(t) + \frac{\kappa}{\sigma}\Lambda$ appears in every iteration, we can simplify the algorithm to

$$\mathcal{T}(t) = \alpha(t)\beta(t)^{-1}, \tag{33}$$

using the initialization $\beta(0) = \frac{\kappa}{\sigma}\Lambda$.

The algorithm can be further improved by avoiding the explicit inversion of $\beta(t)$. For this, we rewrite equation (33) as a recursive least squares algorithm

$$\begin{aligned}K_T(t) &= \frac{\Phi_{x_t}^{\mathsf{T}}P(t-1)}{1 + \Phi_{x_t}^{\mathsf{T}}P(t-1)\Phi_{x_t}}\\ \mathcal{T}(t) &= \mathcal{T}(t-1) + \big[v_{x_t} - \mathcal{T}(t-1)\Phi_{x_t}\big]K_T(t)\\ P(t) &= \big[P(t-1) - P(t-1)\Phi_{x_t}K_T(t)\big],\end{aligned} \tag{34}$$

where K_T is known as the Kalman gain and P is the covariance matrix of the estimation error [19], [20]. From matrix P, we can obtain the estimated variance of each node with $\mathbf{diag}(P) = [\sigma_1^2, \ldots, \sigma_N^2]$. In the algorithm (34), the information needed in the next iteration is stored in $\mathcal{T}(t-1)$ and $P(t-1)$, meeting the sufficient statistics condition. However, note that the first iteration must be computed with equation (32), since it is not possible to calculate $P(0) = \beta(0)^{-1} \propto \Lambda^{-1}$ because Λ is singular. Therefore, for the second iteration we compute $\mathcal{T}(2)$ with equation (34) and with $P(1) = \beta(1)^{-1}$. It is possible to verify that the resulting estimated vector field from equation (34) is the same as the one obtained in equation (25).

VI. Simulation results

To illustrate the proposed algorithm, three simulations were performed. In the first one, *A*, we estimate the vector field for different instants of a trajectory. In the second, *B*, we compare the estimated vector field using a single or two consecutive distinct trajectories. At last, in the third one, *C*, we compare the performance of using a single or both trajectories to predict the future trajectory steps. The standard parameters we choose for all examples are $p = 4$, $\kappa = 1$, $N = 15\times 15$ and $\sigma = 100$.

We generate synthetic trajectories using the Van der Pol oscillator

$$\begin{aligned}&x_1(t+1) = x_2(t)dt + x_1(t)\\ &x_2(t+1) = \Big(\mu\big(1 - x_1(t)^2\big)x_2(t) - x_1(t)\Big)dt + x_2(t),\end{aligned} \tag{35}$$

where we assume $v_{x_t} = [x_1(t{+}1){-}x_1(t), x_2(t{+}1){-}x_2(t)]^{\mathsf{T}}$, $\mu = 0.01$ and $dt = 0.1$. The system chosen has a limit cycle to which every trajectory converges. A trajectory outside that limit cycle has the behavior of a stable system, and a trajectory inside that limit cycle has the behavior of an unstable system. Every figure presented has a black-white squared background that represents the variance of each node, obtained with matrix $P(t)$ from equation (34). The less variant nodes are white, and the more variant ones are black.

A. Recursive vector field estimation

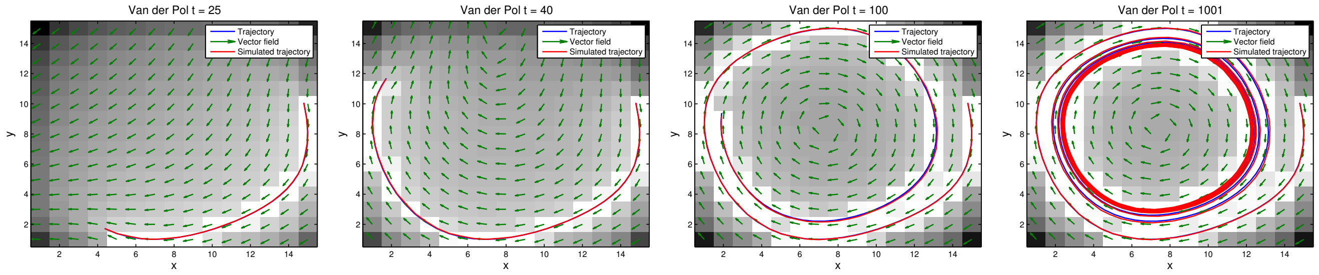
Experiment *A* consists in the estimation of a vector field from the same trajectory at different time instants. In figure 1, we present the evolution of the vector field obtained from a trajectory outside the limit cycle for the instants $t_1 = 25$, $t_2 = 40$, $t_3 = 100$ and $t_4 = 1001$, which are shown in plots 1a, 1b, 1c and 1d, respectively.

Each plot shows the trajectory generated up to the specified time (blue). It also shows the estimated vector field (green) and a simulated trajectory (red) computed with the estimated vector field and the first position of the generated trajectory. Every plot in figure 1 can also be obtained in a batch mode using equation (25).

B. Consecutive vector field estimation

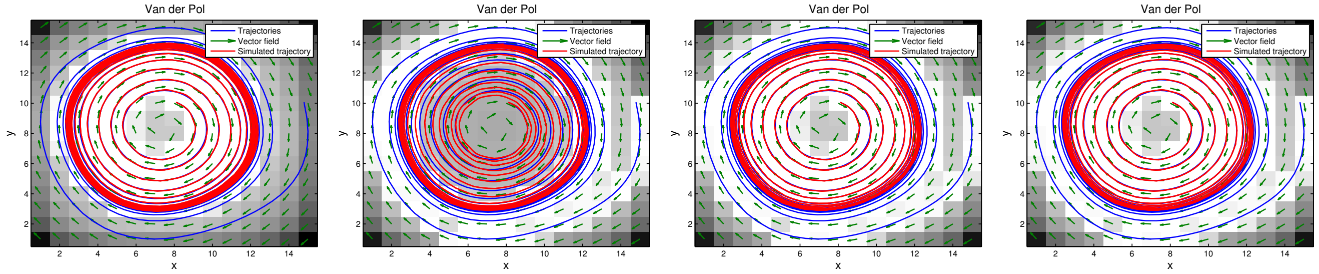
Experiment *B* compares the vector field estimation using a single trajectory with a pair of consecutive trajectories, each of them in different sides of the limit cycle. For that purpose, we estimate a vector field from the first trajectory and then use it as prior for the estimation of an improved vector field. We show in figure 2 the estimated vector field and the inside simulated trajectory for four possible cases, which are shown in plots 2a, 2b, 2c and 2d. The plots 2a and 2b present the vector field estimation using the trajectories inside and outside the limit cycle, respectively. Plot 2c presents the vector field estimation using the inside trajectory followed by the outside one, and 2d presents the same as 2c with the trajectories order switched.

We can verify that the simulated trajectories in 2a, 2c and 2d fit acceptably the generated trajectories. Unlike the other



(a) (b) (c) (d)

Fig. 1: Recursive vector field estimation for different time instants $t_1 = 25$, $t_2 = 40$, $t_3 = 100$ and $t_4 = 1001$ of Van der Pol outside trajectory. Each plot has a black-white squared background that represents the variance of each node.



(a) (b) (c) (d)

Fig. 2: Vector field estimation using trajectories with length of 1001 steps. Plots 2a and 2b use an inside and outside trajectory respectively. Plot 2c presents the vector field estimation using the inside trajectory followed by the outside one, and in 2d presents the same as 2c with the trajectories order switched.

plots, the simulated trajectory of plot 2b (estimation using the outside trajectory) does not fit the generated trajectory. The results mentioned can be explained by the lack of information of the inside of the limit cycle in 2b.

C. *Trajectory prediction using vector field estimation*

In experiment *C*, we concern the performance of the estimated vector field to predict the future steps of a trajectory, for two estimation examples. The vector field estimation follows the same idea as in section VI-B, but here we only use the first 200 steps of a trajectory. Then, we present in figure 3 the generated and simulated trajectories, both with 1001 steps. Four possible cases are shown in plots 3a, 3b, 3c and 3d.

The vector field estimation using both inside and outside trajectories achieves better results than using just a single sided trajectory. These results meet the conclusions in VI-B, where the use of two distinct trajectories yields acceptable model identification.

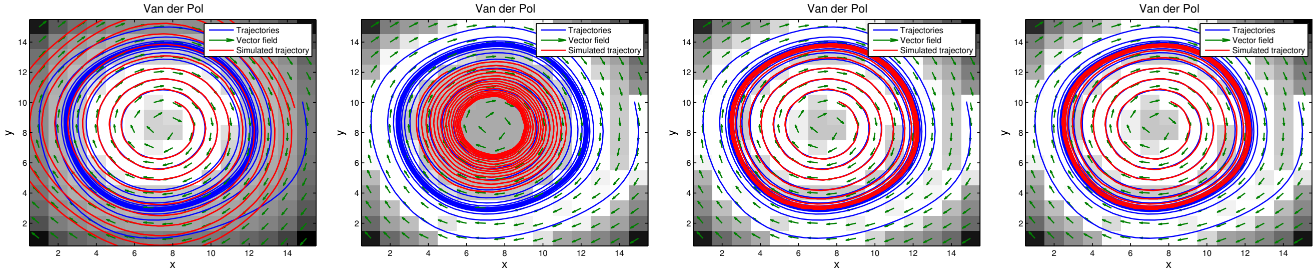
To compare the vector field model in each figure, we present in table I the squared mean error (MSE) of w_t in equation (1). The first line of table I corresponds to the experiment of section VI-A and each column to each plot (a) to (d) of figure 1. The second line of table I corresponds to the experiment of section VI-C and each column to each plot (a) to (d) of figure 3. The MSE values are obtained from the difference between the steps of the generated trajectory v_{x_t} and the estimated velocity vectors $T(x_t)$, computed with equation (7). Note that in table I, we only compare the steps, i.e. the vectors between the generated trajectory and the vectors simulated using the vector field. We do not use the simulated model (in red in the figures) for the MSE computation.

TABLE I: Mean squared error (MSE) for experiences *A* and *C* corresponding to sections VI-A and VI-C. Each table line corresponds to the figures 1 and 3, or experiments A and C, where each column corresponds to each figure plot (a) to (d).

MSE	(a)	(b)	(c)	(d)
Exp. *A*	6.33E−7	7.38E−7	6.29E−7	7.74E−7
Exp. *C*	4.86E−5	1.94E−6	1.73E−6	1.73E−6

The results for experiment *A* show that the MSE does not increase with the trajectory length as expected. Instead, there is a tradeoff between the length and the space distribution of a trajectory, in order to obtain better model accuracy.

Moreover, the trajectory prediction (experiment *C*) using two trajectories of each side of the limit cycle presents a lower MSE than using two trajectories of the same side. In addition, we find that the results of (c) and (d) of experiment *C* are the same. From this, we can conclude that the order in which each trajectory is computed is not relevant. On the other hand, we can verify that the model identification, for



(a) (b) (c) (d)

Fig. 3: Vector field estimation using trajectories with length of 200 steps and simulation of a trajectory with length of1001 steps. Plots 3a and 3b use an inside and outside trajectory respectively. Plot 3c presents the vector field estimation using the inside trajectory followed by the outside one, and in 3d presents the same as 3c with the trajectories order switched.

the present system, shows promising results for the prediction of future trajectory steps.

VII. Conclusions

This paper presents the identification of an object motion, consisting in a trajectory constructed with interpolation of a vector field. The vector field has nodes distributed in a uniform grid that can smoothly depend on neighboring nodes. We present two estimation algorithms — a batch algorithm and a recursive one — to estimate the vector field. We take advantage of the recursive algorithm properties to perform some experiments computing consecutive trajectories. The proposed algorithm improves the computation of vector fields by lowering time and memory complexity, removing the trajectory length dependency and enabling the use of previous information as prior.

The results presented show a reasonable identification of a nonlinear system, which suggests that we could use the algorithm for system modeling.

For future work, we intend to approach the same problem for multiple trajectories and multiple vector fields using a recursive algorithm. In addition, the estimation of some parameters of the algorithm can also improve its adaptation to different situations and increase its independence of human intervention.

References

[1] O. Masoud and N. Papanikolopoulos, “A method for human action recognition,” *Image and Vision Computing*, vol. 21, no. 8, pp. 729 – 743, 2003.

[2] M. P. A. Mecocci, “A completely autonomous system that learns anomalous movements in advanced videosurveillance applications,” *IEEE International Conference on Image Processing*, vol. 2, pp. II–586–9, 2005.

[3] C. R. Wren, A. Azarbayejani, T. Darrell, and A. P. Pentland, “Pfinder: Real-time tracking of the human body,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 19, no. 7, pp. 780–785, 1997.

[4] Z. Fu, W. Hu, and T. Tan, “Similarity based vehicle trajectory clustering and anomaly detection,” *IEEE International Conference on Image Processing*, vol. 2, pp. II–602–5, 2005.

[5] J. C. Nascimento, M. A. T. Figueiredo, and J. S. Marques, “Trajectory analysis in natural images using mixtures of vector fields,” *Proc. 16th IEEE International Conference on Image Processing*, pp. 4353–4356, 2009.

[6] J. C. Nascimento, M. A. T. Figueiredo, and J. S. Marques, “Recognition of human activities using space dependent switched dynamical models,” *IEEE International Conference on Image Processing*, vol. 3, pp. III–852–5, 2005.

[7] J. C. Nascimento, M. A. T. Figueiredo, and J. S. Marques, “Trajectory classification using switched dynamical hidden Markov models,” *IEEE Transactions on Image Processing*, vol. 19, no. 5, pp. 1338–1348, 2010.

[8] J. Mendel, “Multistate least-squares parameter estimators,” *IEEE Transactions on Automatic Control*, vol. 20, no. 6, pp. 775–782, 1975.

[9] M. Tornio and T. Raiko, “Variational Bayesian approach for nonlinear identification and control,” *In Proc. of the IFAC Workshop on Nonlinear Model Predictive Control for Fast Systems*, pp. 41–46, 2006.

[10] R. van der Merwe and E. Wan, “Gaussian mixture sigma-point particle filters for sequential probabilistic inference in dynamic state-space models,” *IEEE International Conference on Acoustics, Speech, and Signal Processing*, vol. 6, pp. VI–701–4 vol.6, 2003.

[11] Z. Xionghu, S. Shubiao, and P. Chengming, “Time-varying parameters estimation based on Kalman particle filter with forgetting factors,” *The International Conference on Computer as a Tool, EUROCON*, vol. 2, pp. 1558–1561, 2005.

[12] B. L. Pence, H. K. Fathy, and J. L. Stein, “A maximum likelihood approach to recursive polynomial chaos parameter estimation,” *American Control Conference*, pp. 2144–2151, 2010.

[13] L. Ljung, *System Identification: Theory for the User*. Prentice Hall, 2 ed., Jan. 1999.

[14] A. Wills, T. B. Schön, and B. Ninness, “Parameter estimation for discrete-time nonlinear systems using EM,” *Proc. 17th IFAC World Congress*, pp. 4012–4017, 2008.

[15] V. Kadirkamanathan, M. H. Jaward, S. G. Fabri, and M. Kadirkamanathan, “Particle filters for recursive model selection in linear and nonlinear system identification,” *Proc. 39th IEEE Conference on Decision and Control*, vol. 3, pp. 2391–2396, 2000.

[16] X. Q. Wei, X. J. Zhang, H. Yu, and S. M. Song, “The ensemble unscented particle filter,” *2nd International Conference on Intelligent Control and Information Processing*, vol. 2, pp. 844–848, 2011.

[17] C. E. Rasmussen and C. Williams, *Gaussian Processes for Machine Learning*. MIT Press, 2006.

[18] D. Shepard, “A two-dimensional interpolation function for irregularly-spaced data,” *Proc. 23rd ACM national conference*, pp. 517–524, 1968.

[19] D. Simon, “Kalman filtering with state constraints: A survey of linear and nonlinear algorithms,” *IET Control Theory & Applications*, vol. 4, no. 8, pp. 1303–1318, 2010.

[20] B. M. Bell and F. W. Cathey, “The iterated Kalman filter update as a Gauss-Newton method,” *IEEE Transactions on Automatic Control*, vol. 38, no. 2, pp. 294–297, 1993.