An Improved EM-method for the Estimation of Transition Probabilities in Multiple Model Switching Systems

Miguel Barão * **Jorge S. Marques** ** **João Miranda Lemos** ***

* *INESC-ID/U. Évora, Portugal*
** *ISR/IST, Portugal*
*** *INESC-ID/IST, Portugal*

Abstract: This paper concerns the joint multiple model system identification and its switching model. The problem is formulated in a probabilistic framework, where multiple vector fields are estimated from data, and a Markov switching model is identified. An Expectation-Maximization method is employed for the identification task. The present paper focus mainly the Markov identification and more specifically the M-step of the EM method. For this purpose a natural gradient algorithm is employed.

1. INTRODUCTION

Identification of dynamical systems is a central and recurrent problem in control. The most commonly used techniques are based on least squares or some variant thereof, where a linear system is fitted to the data so as to minimize a quadratic criteria. This kind of techniques fail when the system under consideration is far from being linear. One such example is the well known Lorenz strange attractor, a deterministic nonlinear system capable of showing chaotic behavior. These are the kind of situations intended for the algorithms presented in this paper.

Coincidentally, the image processing community has already dealt with problems where linearity assumptions are not made. One such problem, Nascimento et al. [2009], is the observation of a set of trajectories of objects in a video sequence, and then identify vector fields that "best" describe the observations. Related works are *e.g.* Pavlovic et al. [1999], where an hybrid state is estimated from data. While the former approach deals with the estimation of the underlying model, the later concerns a filtering problem given a Markov switched linear model. The present paper is heavily based on the first formulation for the identification of nonlinear dynamical systems.

Our purpose is to identify vector fields that best describe the observed trajectories of the state. The system is allowed to switch among a number of models according to a probabilistic mechanism whose parameters can change along the state space. The estimation is performed with a maximum *a posteriori* criteria and the EM algorithm (Dempster et al. [1977]) is used for this purpose.

The main contributions of the paper are the extension of the trajectory estimation to an arbitrarily dimensional space, the optional use of an irregular grid, and an improved algorithm for the estimation of the switching probabilities based on the natural gradient.

The paper is organized as follows. Section 2 formulates the problem more precisely; Section 3 presents the EM-algorithm used, along with the formulas for vector field estimation; Section 4 focus on the optimization of switching probabilities; Section 5 presents an example showing the Lorenz strange attractor and how it can be described by two switching vector fields; Finally section 6 draws conclusions.

2. PROBLEM STATEMENT

The problem under consideration is the one of identification of a set of vector fields $T_k(x)$, $k \in \{1, \ldots, K\}$, and associated transition probabilities $b_{ij}(x)$ that best describe a multiple model switched nonlinear system. The switching mechanism is governed by a state dependent hidden Markov model.

The identification procedure is performed offline using a collection of S recorded trajectories. Each trajectory x^s, $s \in \{1, \ldots, S\}$, is an ordered set of points $(x_1^s, x_2^s, \ldots, x_{L_s}^s)$, where $x_t^s \in \mathbb{R}^D$.

It shall be assumed that, at each time step, the system state is described by an hybrid state (x_t, k_t) where k_t indicates the active model at time t. Each model is described by a different vector field $T_k(x)$. The active model k_t can change according to a space dependent Markov chain with transition probabilities $b_{ij}(x)$, the dependency in x meaning that it is more likely to change the active model in some points than in others. A system model can then be written as

$$\Pr\{k_t = j | k_{t-1} = i, x_{t-1}\} = b_{ij}(x_{t-1}) \quad (1)$$

$$x_t = x_{t-1} + T_{k_t}(x_{t-1}) + w_t \quad (2)$$

where $w_t \sim \mathcal{N}(\mathbf{0}, \Sigma_{k_t})$ is a state disturbance and $b_{ij}(x)$ is the transition probability from the vector field T_i to T_j at position x. Equation (1) computes the active field T_{k_t} used in equation (2) to find the updated state x_t. Figure 1

⋆ This work was supported by FCT (INESC-ID multiannual funding) through the PIDDAC Program funds and project "ARGUS - Activity Recognition and Object Tracking Based on Multiple Models", PTDC/EEA-CRO/098550/2008.

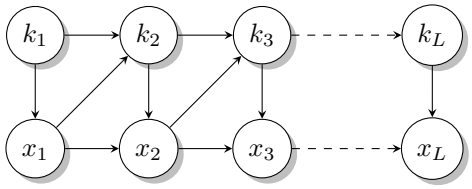

Fig. 1. Markov diagram showing the state variable x_t and active model k_t updates.

shows the dependencies among the variables during the update procedure.

The solution to the identification problem is sought in a probabilistic framework assuming a Bayesian viewpoint and follows closely the framework developed in Nascimento et al. [2009].

2.1 Space discretization

First, a region of interest in $\mathbb{R}^D$ is selected and N nodes distributed over it generating a (not necessarily regular) grid $g_n \in \mathbb{R}^D$, $n \in \{1, \ldots, N\}$. Then, for each node n, K vectors T_k^n are estimated corresponding to the K desired models. The transition matrix b_{ij}^n associated with that node is also estimated. After having estimated the vectors for every node of the grid, the vector field in $\mathbb{R}^D$ is obtained by interpolation:

$$T_k(x) = \sum_{n=1}^{N} T_k^n \phi_n(x), \tag{3}$$

where $\phi_n(x)$ is a previously defined interpolation function satisfying convexity constraints

$$\phi_n(x) > 0, \quad \sum_{n=1}^{N} \phi_n(x) = 1, \quad \forall x \in \mathbb{R}^D. \tag{4}$$

Transition probabilities are also interpolated from those estimated at the nodes as follows:

$$b_{ij}(x) = \sum_{n=1}^{N} b_{ij}^n \phi_n(x). \tag{5}$$

This is a convex combination of stochastic matrices that ensures the interpolated one to be a valid stochastic matrix.

Although other interpolation functions could be used, the following one is considered:

$$\phi_n(x) \propto e^{-\frac{\alpha}{2}\|x - g_n\|^2}. \tag{6}$$

This function gives higher interpolation weight to nodes g_n closer to x. It has the property that $\phi_n(x) < 1$, even when $x = g_n$. As a consequence, $T_k(g_n) \neq T_k^n$, meaning that T_k^n should only be seen as a parameter vector and not an element of the vector field.

2.2 Parameter estimation

Parameter estimation aims to find the vector fields and switching probabilities at each node of the grid from a set of sampled trajectories in $\mathbb{R}^D$.

The model depends on unknown parameters $\theta = (\mathcal{T}, \mathcal{B})$, that include the set of vectors $\mathcal{T} = \{T_k^n\}$ for all nodes and trajectories, and the set of transition matrices $\mathcal{B} = \{B^1, \ldots, B^N\}$ for the nodes.

Assuming independence, the prior $p(\theta)$ becomes

$$p(\theta) = p(\mathcal{T})p(\mathcal{B}). \tag{7}$$

The following additional assumptions are made:

- $p(\mathcal{T})$ is built from independent distributions $p(T_k^n)$ defined over the set of trajectory models k. For each individual model k, parameters T_k^n are assumed to be dependent across neighbor nodes according to a multivariable Gaussian distribution with covariance matrix $\mathbf{\Lambda}$. This dependence produces a regularization effect: in the absence of data, far nodes gather information from their neighbors, thereby introducing smoothness into the estimated vector field.

  Pairs of neighbor nodes are collected in the set

  $$\mathcal{I} = \big\{(i,j) \;\big|\; \|g_i - g_j\| < d_{\max},\ i \neq j\big\}. \tag{8}$$

  This set is used to build a sparse matrix $\mathbf{\Delta}$ where each column contains a 1 and a -1 marking the pairs of neighbor nodes $(i,j) \in \mathcal{I}$. Defining $\mathbf{T}_k = \begin{bmatrix} T_k^1 & \cdots & T_k^N \end{bmatrix}$ and computing the product $\mathbf{\Delta}^T \mathbf{T}_k^T$, a matrix is obtained with the vector differences between all neighbor vectors T_k^i and T_k^j. The regularization then amounts to attribute higher probability to smaller differences:

  $$p(\mathbf{T}_k) \propto e^{-\alpha \operatorname{Tr}(\mathbf{T}_k \mathbf{\Lambda}^{-1} \mathbf{T}_k^T)}, \tag{9}$$

  where

  $$\mathbf{\Lambda}^{-1} = \epsilon \mathbf{I} + \mathbf{\Delta}\mathbf{\Delta}^T. \tag{10}$$

  The small additional term $\epsilon \mathbf{I}$ ensures positive definiteness of $\mathbf{\Lambda}^{-1}$.
- $p(\mathcal{B})$ is set to a constant density in the $K-1$ simplex defined by its parameters (although it is not a noninformative Jeffreys prior, it will be easier to deal with in the optimization part of the algorithm).

Given these priors and a set of recorded trajectories $\mathcal{X} = \{x^1, \ldots, x^S\}$ with lengths L_s, the maximum *a posteriori* estimate $\hat{\theta}$ of the parameters θ is defined as

$$\hat{\theta} = \arg\max_{\theta} p(\theta|\mathcal{X}) = \arg\max_{\theta} p(\theta) \sum_{\mathcal{K}} p(\mathcal{X}, \mathcal{K}|\theta). \tag{11}$$

Unfortunately, the summation in $\mathcal{K}$ is unfeasible since it requires in the order of K^{SL} additions. To solve this problem the Expectation-Maximization (EM) algorithm (Dempster et al. [1977]) is employed on the complete joint probability $p(\mathcal{X}, \mathcal{K}, \theta)$.

The optimal estimate can be found by the following maximization problem:

$$\hat{\theta} = \arg\max_{\theta} \big(U(\theta, \hat{\theta}) - V(\theta, \hat{\theta})\big),$$

where

$$V(\theta, \hat{\theta}) \stackrel{\text{def}}{=} E\big[\log p(\mathcal{K}|\mathcal{X}, \theta) \big| \mathcal{X}, \hat{\theta}\big] \leq V(\hat{\theta}, \hat{\theta}), \quad \forall \theta, \tag{12}$$

and

$$U(\theta, \hat{\theta}) \stackrel{\text{def}}{=} E\big[\log p(\mathcal{X}, \mathcal{K}, \theta) \big| \mathcal{X}, \hat{\theta}\big] \tag{13}$$

is what is actually used in the EM-algorithm to estimate $\hat{\theta}$. The EM method encompasses two steps: first, in the E-step, given an initial estimate $\hat{\theta}$, $U(\theta, \hat{\theta})$ is computed; then, in the M-step, the previously computed function $U(\theta, \hat{\theta})$ is maximized with respect to θ while keeping $\hat{\theta}$ fixed. The two steps are then iterated until convergence to a local maxima is attained.

3. EXPECTATION-MAXIMIZATION ALGORITHM

3.1 E-step

In the E-step part of the algorithm, the function $U(\theta,\hat{\theta})$ is found to be given by

$$\begin{aligned} U(\theta,\hat{\theta}) = C - \frac{1}{2}\sum_{k=1}^{K}\mathrm{Tr}(\mathbf{T}_k\mathbf{\Lambda}^{-1}\mathbf{T}_k^T) + \\ - \frac{1}{2}\sum_{s,t,k} w_k^s(t)\Big\|x_t^s - x_{t-1}^s - \sum_{n=1}^{N}\phi_n(x_{t-1}^s)T_k^n\Big\|_{\Sigma_k^{-1}}^2 + \\ + \sum_{s,t}\sum_{i,j=1}^{K} w_{ij}^s(t)\log\Big(\sum_{n=1}^{N} b_{ij}^n\phi_n(x_{t-1}^s)\Big). \end{aligned} \tag{14}$$

where the symbols w_i and w_{ij} are defined by

$$w_j^s(t) \stackrel{\text{def}}{=} \Pr\big\{k_t^s = j|\mathcal{X},\hat{\theta}\big\}, \tag{15}$$

$$w_{ij}^s(t) \stackrel{\text{def}}{=} \Pr\big\{k_{t-1}^s = i, k_t^s = j|\bar{x}^s,\hat{\theta}\big\}, \tag{16}$$

and calculated using the forward-backward algorithm (see Rabiner [1990]).

3.2 M-step

In the M-step part of the algorithm, equation (14) is maximized with respect to the vector field T_k^n and transition probabilities b_{ij}^n. The vector field maximization can be done explicitly. The stationarity points are the solutions of

$$\frac{\partial U(\theta,\hat{\theta})}{\partial \mathbf{T}_\alpha} = -\mathbf{\Lambda}^{-1}\mathbf{T}_\alpha^T - \mathsf{A}_\alpha\mathbf{T}_\alpha^T\Sigma_\alpha^{-1} + \mathsf{B}_\alpha = \mathbf{0}, \tag{17}$$

where

$$\mathsf{A}_\alpha \stackrel{\text{def}}{=} \sum_{s=1}^{S}\sum_{t=1}^{L_s} w_\alpha^s(t)\Phi(x_{t-1}^s)\Phi(x_{t-1}^s)^T, \tag{18}$$

$$\mathsf{B}_\alpha \stackrel{\text{def}}{=} \sum_{s=1}^{S}\sum_{t=1}^{L_s} w_\alpha^s(t)\Phi(x_{t-1}^s)(x_t^s - x_{t-1}^s)^T\Sigma_\alpha^{-1}. \tag{19}$$

Premultiplying (17) by the positive definite matrix $\mathbf{\Lambda}$ yields the Sylvester equation

$$(-\mathbf{\Lambda}\mathsf{A}_\alpha)\mathbf{T}_\alpha^T\Sigma_\alpha^{-1} - \mathbf{T}_\alpha^T + \mathbf{\Lambda}\mathsf{B}_\alpha = \mathbf{0}, \tag{20}$$

which can be efficiently solved for $\mathbf{T}_\alpha$ using standard numerical packages.

Regarding the optimization of the transition probabilities, a natural gradient method is employed. The next section presents in more detail the algorithm developed.

4. SWITCHING PROBABILITIES

Here, a differential-geometric point of view is applied to the probability manifold $\mathcal{B}$. General differential-geometric concepts like the differential and gradient can be consulted *e.g.* in Boothby [1986]. The development of these concepts specifically in the probabilistic framework is known as information geometry, a term coined by Amari. Standard references are Amari [1985], Amari and Nagaoka [2000].

Differentiating $U(\theta,\hat{\theta})$ with respect to the transition probabilities b_{ij}^n at every node yields

$$\frac{\partial U(\theta,\hat{\theta})}{\partial b_{\alpha\beta}^\gamma} = \sum_{s=1}^{S}\sum_{t=1}^{L_s} w_{\alpha\beta}^s(t)\frac{1}{\sum_{n=1}^{N} b_{\alpha\beta}^n\phi_n(x_{t-1}^s)}\phi_\gamma(x_{t-1}^s). \tag{21}$$

Unfortunately, the stationarity condition $\partial U(\theta,\hat{\theta})/\partial b_{\alpha\beta}^\gamma = 0$ doesn't appear to have an explicit solution. Thus, a natural gradient based iterative method will be used instead.

For the optimization, a mixture parametrization of the transition probabilities is considered where parameters are the probabilities themselves. Since $\sum_j b_{ij}^n = 1$ for any i and n, only $K-1$ parameters are in fact independent. Therefore, the K-th probability for each (i,n) is obtained from the remaining ones by

$$b_{iK}^n = 1 - \sum_{j=1}^{K-1} b_{ij}^n. \tag{22}$$

For each (i,n), the $K-1$ probabilities bi-univocally define a point in the probability manifold $\mathcal{B}$, and the $K-1$ partial derivatives in equation (21) define the differential $\mathrm{d}U$ of the function U.

For the iterative procedure, the coordinates ($K-1$ probabilities) are updated in the direction for which the function U is more steep, *i.e.*, the direction $v_i^n \in \mathbb{R}^{K-1}$ given by

$$v_i^n = \arg\max_{v_i^n} \mathrm{d}U(v_i^n) = \arg\max_{v_i^n}\sum_{j=1}^{K-1}\frac{\partial U}{\partial b_{ij}^n}v_{ij}^n. \tag{23}$$

This problem is ill-posed since the vector v_i^n can have an infinite length. Therefore, it is necessary to include a constraint in its norm such as $\|v_i^n\| = 1$. The use of a norm requires some form of metric structure in the space $\mathcal{B}$. The direction of greatest ascent can now be found using the Lagrange multipliers method. For that sake, define the Lagrange function L with Lagrange multiplier λ:

$$L = U + \tfrac{1}{2}\lambda\big(1 - \|v_i^n\|^2\big) = U + \tfrac{1}{2}\lambda\big(1 - (v_i^n)^T G v_i^n\big). \tag{24}$$

The stationarity points can be found solving

$$\mathrm{d}L = \mathrm{d}U - \lambda(v_i^n)^T G = 0, \tag{25}$$

and thus

$$(\lambda v_i^n)^T = G^{-1}(\,\mathrm{d}U)^T. \tag{26}$$

The resulting scaled vector λv_i^n is the natural gradient ∇U defined by the equality

$$\langle \nabla U, w\rangle = \mathrm{d}U(w), \quad \forall w. \tag{27}$$

This means that the gradient vector depends on the selected metric. If an Euclidean metric is used, then G is the identity matrix, and the natural gradient is simply a vector whose components are the partial derivatives of U. In general, however, that is not the case.

In the present problem, the chosen metric is the Fisher metric as it is invariant with respect to reparametrization. The Fisher metric is defined by the inner product

$$\langle v, w\rangle = \sum_{\alpha,\beta} v^\alpha w^\beta g_{\alpha\beta}, \tag{28}$$

where $g_{\alpha\beta}$ are the components of the Fisher information matrix. Using the $K-1$ probabilities as coordinates, the

Fisher information matrix is a $(K-1)\times(K-1)$ matrix whose components are given by

$$g_{\alpha\beta} = \frac{\delta^{\beta}_{\alpha}}{b^n_{i\alpha}} + \frac{1}{1-\sum_{k=1}^{K-1} b^n_{ik}}, \tag{29}$$

and where δ^{β}_{α} is the Kronecker delta. Its inverse $G^{-1} = [g_{\alpha\beta}]^{-1} = [g^{\alpha\beta}]$ can be computed in component notation [1] yielding

$$g^{\alpha\beta} = b^n_{i\alpha}\delta^{\alpha}_{\beta} - b^n_{i\alpha}b^n_{i\beta}, \tag{30}$$

or in matrix notation

$$G^{-1} = \begin{bmatrix} b^n_{i1} & & 0 \\ & \ddots & \\ 0 & & b^n_{i(K-1)} \end{bmatrix} - \begin{bmatrix} b^n_{i1} \\ \vdots \\ b^n_{i(K-1)} \end{bmatrix} \begin{bmatrix} b^n_{i1} & \cdots & b^n_{i(K-1)} \end{bmatrix}. \tag{31}$$

The product of G^{-1} and $(\mathrm{d}U)^T$ present in equation (26) can be computed alternatively without explicitly building the matrix G^{-1}. This can be done with the formula

$$\nabla U = b^n_{i:} \circ (\mathrm{d}U)^T - b^n_{i:}\big(b^n_{i:} \cdot (\mathrm{d}U)^T\big), \tag{32}$$

where $b^n_{i:}$ denotes the column of $K-1$ probabilities, $\circ$ is the element-wise product (also known as Hadamard or Schur product) and $\cdot$ is the usual dot product.

The gradient method then uses the direction ∇U to update the probabilities $b^n_{i:}$:

$$b^n_{i:} \leftarrow b^n_{i:} + \eta \nabla U \tag{33}$$

The remaining probability b^n_{iK} is then computed as shown in (22). The iterative process is repeated for each pair of models and nodes (i, n).

The natural gradient has remarkable properties Barão [2009b,a], Barão and Lemos [2008] that makes it a good choice for these kind of optimization tasks:

- It automatically satisfies the probability non-negativity constraints. Optimization becomes unconstrained.
- Achieves faster convergence rates for many interesting problems, since the Fisher information matrix is also the Hessian of several information measures, turning the iterations into an asymptotically quasi-Newton method.

Although not proved to have quadratic convergence in the problem at hand, the simulations done so far show it to be much faster than the standard Euclidean gradient (for which the probability constraints have to be explicitly taken care of).

5. SIMULATION RESULTS

As an illustrative example, the algorithm was applied to a Lorenz strange attractor described by the following differential equations:

$$\begin{aligned} \dot{x}_1 &= 10(x_2 - x_1) \\ \dot{x}_2 &= x_1(28 - x_3) - x_2 \\ \dot{x}_3 &= x_1 x_2 - \tfrac{8}{3} x_3 \end{aligned}$$

This is a 3-dimensional autonomous system that exhibits chaotic behavior with trajectories resembling a figure eight in space. Figure 2 shows a few sample trajectories (x_1, x_3) generated from several different initial conditions. Variables are rescaled to the unit interval. The generated trajectories can loosely be described as rotating a few turns in one plane, then moving on to rotate in the other plane, then back again and so on. We will estimate two vector fields and switching probabilities for the (x_1, x_3)-trajectories. Note that, since a projection $\mathbb{R}^3 \to \mathbb{R}^2$ is being applied, a trajectory can intersect itself when observed from the lower dimensional space.

[1] Einstein summation convention is not used here.

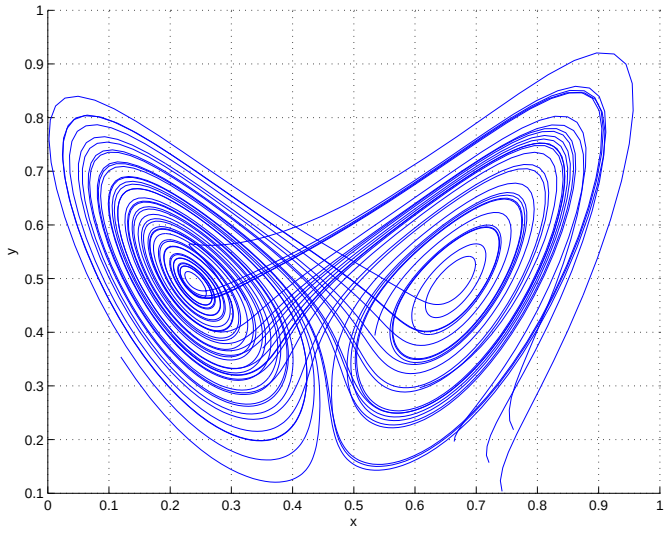

Fig. 2. Lorenz strange attractor.

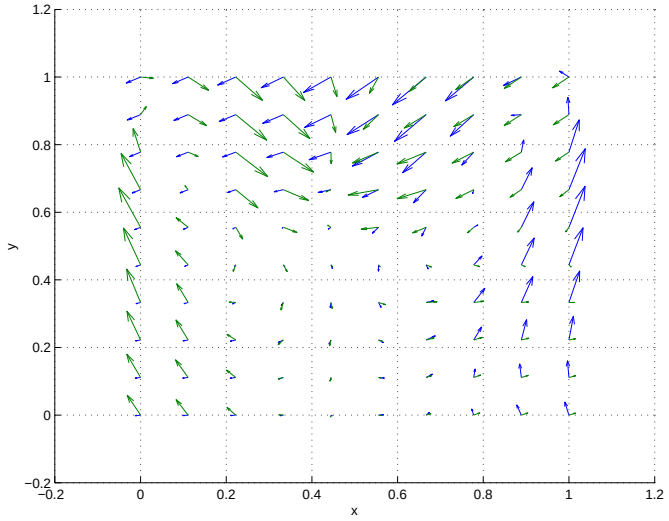

Fig. 3. Estimated vector fields.

For the estimation procedure, a regularly spaced 10×10 grid was used. Two vector fields and corresponding transition probabilities were estimated. Figures 3 and 4 show the results obtained for the vector fields and corresponding switching probabilities after 100 iterations of the EM algorithm. A stochastic simulation based on the estimated vector fields and switching probabilities is shown in figure 5. The simulation is performed using equations (1) and (2) starting from 30 different random initial conditions. It should be emphasized that although the original systems evolves in $\mathbb{R}^3$, the simulated system lives in $\mathbb{R}^2$. Still, it can capture reasonably well the original behavior.

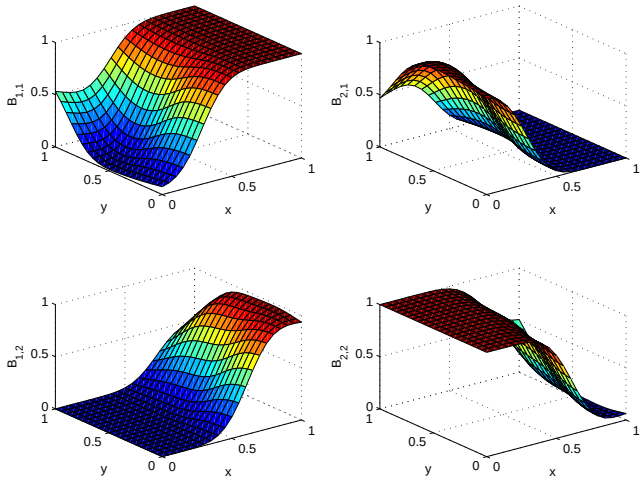

Fig. 4. Estimated transition probabilities. Each plot represents one component of the transition matrix $b_{ij}(x)$. Each surface shows how the (i, j)-component changes in space x.

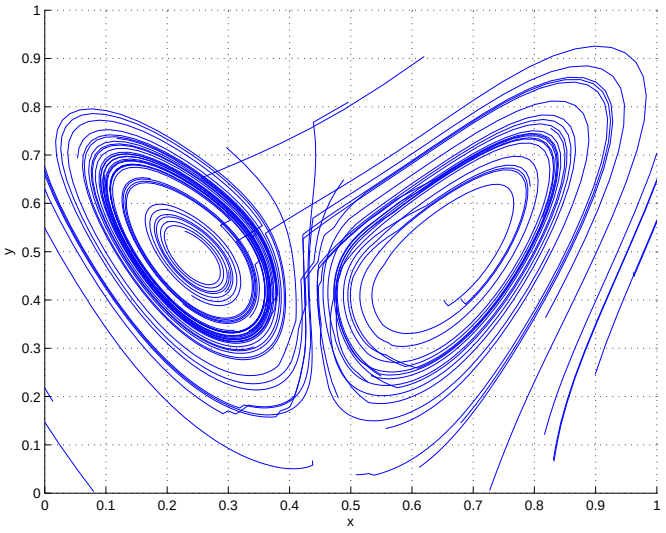

Fig. 5. Stochastic simulation based on the estimated vector fields and estimated switching probabilities.

6. CONCLUSIONS

In this paper an attempt was done to identify a nonlinear system using switched multiple nonlinear models. Each nonlinear model tries to capture the nonlinear behavior of sampled trajectories used for the identification task. This is accomplished by directly estimating the vector fields which best explain the observed data using a Bayesian approach. Simultaneously to the estimation of the vector fields, a space varying Markov model is identified. The Markov model generates hidden variables indicating the active field. The EM-algorithm was used. In the maximization step of the EM, the transition probabilities of the Markov model are obtained using a natural gradient method, which have shown better convergence and speed properties than standard gradient based methods for optimization of probability distributions.

REFERENCES

Shun-Ichi Amari. *Differential-Geometrical Methods in Statistics*. Lecture Notes in Statistics. Springer-Verlag, 1985.

Shun-Ichi Amari and Hiroshi Nagaoka. *Methods of Information Geometry*. AMS, 2000.

M Barão. Optimization on discrete probability spaces and applications to probabilistic control design. *Proceedings of the European Control Conference, Budapest, Hungary*, pages 1–5, Jun 2009a.

M Barão. Probabilistic control design using an information geometric framework. *1st IFAC Workshop on Estimation and Control of Networked Systems (NecSys'09)*, pages 1–6, Jul 2009b.

M Barão and JM Lemos. An efficient kullback-leibler optimization algorithm for probabilistic control design. *Control and Automation, 2008 16th Mediterranean Conference on*, pages 198–203, 2008.

William M. Boothby. *Differentiable Manifolds and Riemannian Geometry*. Academic Press, 1986.

AP Dempster, NM Laird, and DB Rubin. Maximum likelihood from incomplete data via the em algorithm. *Journal of the Royal Statistical Society. Series B (Methodological)*, 39(1):1–38, 1977.

Jacinto C Nascimento, Mário A T Figueiredo, and Jorge S Marques. Trajectory analysis in natural images using mixtures of vector fields. *IEEE International Conference on Image Processing*, pages 4353—4356, 2009.

V Pavlovic, J.M Rehg, Tat-Jen Cham, and K.P Murphy. A dynamic bayesian network approach to figure tracking using learned dynamic models. *Proceedings of the Seventh IEEE International Conference on Computer Vision, 1999.*, 1:94–101 vol.1, 1999. URL `10.1109/ICCV.1999.791203`.

LR Rabiner. A tutorial on hidden markov models and selected applications in speech recognition. *Readings in speech recognition*, 53(3):267–296, 1990.