Identification of Spatial-Temporal Switched ARX Systems

René Vidal
Center for Imaging Science, Johns Hopkins University, Baltimore MD, 21218, USA

***Abstract*— We consider the problem of identifying a model for data generated by a mixture of dynamical models, both in space and in time. We assume that the measurements at a particular time instant depend on a spatial variable, and that the dynamics of the data in different spatial regions can be modeled with different hybrid systems. We also assume that both the spatial regions as well as the discrete states of the hybrid systems are unknown. Furthermore, we allow the number of models to vary as a function of time. We call such a dynamical model a *spatial-temporal hybrid system*, and develop a recursive identification algorithm for the class of spatial-temporal switched ARX models. We demonstrate the applicability of our method to the segmentation of videos of dynamic textures, such as segmenting a bird floating on water.**

I. INTRODUCTION

The past few years have witnessed an increasing interest in the application of system-theoretic techniques to modeling visual dynamical processes. For instance, [1] models videos of dynamic textures, such as fire, steam, water, etc., as the output of an Auto-Regressive Moving Average (ARMA) model; [2] uses ARMA models to represent human gaits, such as walking, running, jumping, etc.; and [3], [4] use ARMA models to describe the appearance of moving faces. Given a video sequence, one can use standard system identification techniques, e.g., [5], to learn the parameters of the ARMA model. Given an ARMA model, one can use it to synthesize, classify, or recognize novel sequences.

An important assumption in all these applications is that the scene contains a single dynamic texture, so that the video can be modeled with a single ARMA model. This limitation has motivated recent work on modeling scenes consisting of multiple temporal events. [6], [7] model such videos as the output of a linear hybrid system, where each discrete state corresponds to a different event. Given a video sequence, the parameters of each dynamical model can be identified using existing techniques for switched ARX systems [8], [9], [10], [11] or piece-wise ARX systems [12], [13], [14].

Unfortunately, all existing algorithms for hybrid system identification assume that the measurements at time t are the output of a single linear model. This assumption is violated in several applications in computer vision, where different regions in an image may obey different dynamical models. Therefore, one needs to model the video as the output of several dynamical models, not only in time, but also in space.

Paper contributions. We propose a recursive algorithm for identifying the parameters of a spatial-temporal switched ARX (STSARX) system, whose output at time t, y_t, comes from $n_t \geq 1$ ARX models, without knowing which entries of y_t correspond to which model. As time proceeds, the number of models n_t could vary. Also, the ARX model associated with a particular entry of y_t could change due to the occurrence of an event, as in regular SARX models.

Our recursive identification algorithm exploits the fact that the output of an STSARX system at time t must satisfy a polynomial of degree n, where $n \geq n_t$ is the total number of ARX models. The coefficients of this polynomial are identified using recursive normalized least-squares. The parameters of each ARX model at each time and spatial location are identified from the derivatives of this polynomial at a measurement. The segmentation of the data according to the n_t ARX models is then obtained by clustering the ARX model parameters obtained at each spatial-temporal location.

We test our algorithm on the segmentation of a scene containing a bird floating on water. The algorithm not only segments the bird motion from the surrounding water motion, but also captures the periodic bird motion from the temporal evolution of the coefficients of the estimated polynomial. Our experiments also demonstrate the ability of the method to deal with appearing and disappearing motions in the scene.

Relation to prior work. Our work can be seen as a natural generalization of the work of [11] on recursive identification of SARX systems. The main differences is that we consider ARX models whose output has multiple entries, with each entry obeying a possibly different models depending on a spatial variable. Our work is also related to the work of [15], which proposes a recursive algorithm for clustering moving hyperplanes. The main difference is that we have an explicit (SARX) model for the temporal evolution of the data, while in [15] there is no model for the motion of the hyperplanes.

II. IDENTIFICATION OF STSARX SYSTEMS

A discrete-time Switched Auto-Regressive eXogenous (SARX) system is a system whose dynamics are given by

$$y_t = \sum_{j=1}^{n_a(q_t)} a_j(q_t) y_{t-j} + \sum_{j=1}^{n_c(q_t)} c_j(q_t) u_{t-j}, \qquad (1)$$

where $y_t \in \mathbb{R}$ is the *output*, $u_t \in \mathbb{R}$ is the *input*, $q : \mathbb{N} \to Q \doteq \{1, 2, \ldots, n\}$ is the *discrete state* or *mode*, which is assumed to be a deterministic but unknown sequence, $n_a(i)$ and $n_c(i)$ are the orders of the ith ARX model for $i = 1, \ldots, n$, and $\{a_j(i)\}_{j=1}^{n_a(i)}$ and $\{c_j(i)\}_{j=1}^{n_c(i)}$ are the model parameters.

In this paper, we will consider systems in which both the input and the output are actually functions of a spatial variable $\boldsymbol{x} \in \mathcal{X} \subset \mathbb{Z}^D$, i.e. $u_t : \mathcal{X} \to \mathbb{R}$ and $y_t : \mathcal{X} \to \mathbb{R}$. For example, when $D = 2$, $y_t(\boldsymbol{x})$ could be the intensity at

pixel $\boldsymbol{x}$ in an image, or the temperature reading of a mote at location $\boldsymbol{x}$ in a sensor network. Furthermore, we will assume that at time t the space $\mathcal{X}$ is partitioned into n_t regions

$$\mathcal{X} = \mathcal{X}_1(t) \cup \mathcal{X}_2(t) \cup \cdots \cup \mathcal{X}_{n_t}(t), \tag{2}$$

each one associated with a different SARX model. Although each one of these n_t SARX model has its own discrete state, we assume that their discrete-state spaces are all subsets of $Q = \{1, 2, \ldots, n\}$. This allows us to define a global discrete state $q_t : \mathcal{X} \to Q$ such that $q_t(\boldsymbol{x}) = i$ if $y_t(\boldsymbol{x})$ is generated by the ith ARX model. Therefore, we define a discrete-space discrete-time Spatial-Temporal SARX (STSARX) model as a system whose dynamics are given by

$$y_t(\boldsymbol{x}) = \sum_{j=1}^{n_a(q_t(\boldsymbol{x}))} a_j(q_t(\boldsymbol{x}))y_{t-j}(\boldsymbol{x}) + \sum_{j=1}^{n_c(q_t(\boldsymbol{x}))} c_j(q_t(\boldsymbol{x}))u_{t-j}(\boldsymbol{x}). \tag{3}$$

Remark 1: Note that an SARX system is a particular case of an STSARX system with $|\mathcal{X}| = 1$, and an ARX system is a particular case of an STSARX system with $|\mathcal{X}| = n = 1$.

Given input/output data $\{u_t, y_t\}_{t=0}^{\infty}$ generated by an STSARX system with n ARX models in space and time, we wish to identify the model orders $\{n_a(i), n_c(i)\}_{i=1}^{n}$ and the model parameters $\{a_j(i)\}_{j=1}^{n_a(i)}$ and $\{c_j(i)\}_{j=1}^{n_c(i)}$, without knowing the discrete state $\{q_t(\boldsymbol{x})\}$, the regions where each model is valid $\mathcal{X}_i(t)$, or the number of active models at time t, n_t. In order for this identification problem to be well posed, the parameters of the STSARX system must be uniquely defined, which requires an assumption of minimality. We will say that an STSARX model is *minimal* if its n individual ARX systems are minimal and different from each other.

A. Spatial-Temporal Hybrid Decoupling Polynomial

In [10] we showed that the input/output data generated by an SARX model with n discrete states lie in the zero set of a homogeneous polynomial of degree n. More specifically, if $\bar{n}_a \geq \max\{n_a(i)\}$ and $\bar{n}_c \geq \max\{n_c(i)\}$ are given upper bounds on the orders of the ARX models, $K \doteq \bar{n}_a + \bar{n}_c + 1$, and for all $i = 1, \ldots, n$ and $t \geq \max\{\bar{n}_a, \bar{n}_c\}$ we let

$$\boldsymbol{z}_t \doteq [u_{t-\bar{n}_c}, \ldots, u_{t-1}, y_{t-\bar{n}_a}, \ldots, y_{t-1}, -y_t]^\top \in \mathbb{R}^K \text{ and } \tag{4}$$

$$\boldsymbol{b}_i \doteq \left[\mathbf{0}_{\bar{n}_c - n_c}^\top, c_{n_c}(i), ..., c_1(i), \mathbf{0}_{\bar{n}_a - n_a}^\top, a_{n_a}(i), ..., a_1(i), 1\right]^\top, \tag{5}$$

then there exists a discrete state $q_t = i \in \{1, \ldots, n\}$ such that $\boldsymbol{b}_i^\top \boldsymbol{z}_t = 0$, hence the input/output data must satisfy the following *hybrid decoupling polynomial* [10]

$$q_n(\boldsymbol{z}_t) = \prod_{i=1}^{n} (\boldsymbol{b}_i^\top \boldsymbol{z}_t) = 0. \tag{6}$$

The coefficients of q_n can be estimated using a batch least-squares method [10] or a recursive normalized least-squares method [11]. Given an estimate of the coefficients of q_n, the vector of model parameters $\boldsymbol{b}_i$ associated with $\boldsymbol{z}_t$ can be computed from the derivatives of q_n at the regressor $\boldsymbol{z}_t$.

In the case of STSARX models, the situation is more complicated, because the regressor vector

$$\begin{aligned} \boldsymbol{z}_t(\boldsymbol{x}) \doteq [&u_{t-\bar{n}_c}(\boldsymbol{x}), \ldots, u_{t-1}(\boldsymbol{x}), \\ &y_{t-\bar{n}_a}(\boldsymbol{x}), \ldots, y_{t-1}(\boldsymbol{x}), -y_t(\boldsymbol{x})]^\top \end{aligned} \tag{7}$$

depends on the spatial variable $\boldsymbol{x}$, and the value of $\boldsymbol{x}$ determines which one of the n_t SARX models corresponds to the regressor vector $\boldsymbol{z}_t(\boldsymbol{x})$. Nevertheless, if we let $Q_1(\boldsymbol{x}) \subseteq Q$ be the set of discrete states associated with a spatial location $\boldsymbol{x} \in \mathcal{X}$, then for all $t \geq \max\{\bar{n}_a, \bar{n}_c\}$ the set of all regressors associated with $\boldsymbol{x}$ must satisfy the polynomial

$$\prod_{i \in Q_1(\boldsymbol{x})} (\boldsymbol{b}_i^\top \boldsymbol{z}_t(\boldsymbol{x})) = 0. \tag{8}$$

However, we cannot identify the coefficients this polynomial, because the n_t regions $\mathcal{X}_i(t)$ are unknown, thus so is $Q_1(\boldsymbol{x})$.

An alternative solution is to consider the set of n_t discrete states $Q_2(t) \subseteq Q$ that are active at time t. Then

$$\prod_{i \in Q_2(t)} (\boldsymbol{b}_i^\top \boldsymbol{z}_t(\boldsymbol{x})) = 0 \tag{9}$$

for all $\boldsymbol{x} \in \mathcal{X}$, no matter which region $\mathcal{X}_i(t)$ the vector $\boldsymbol{x}$ belongs to. However, the degree of this polynomial, hence the dimension of its vector of coefficients, changes with time. This rules out the possibility of using a recursive algorithm for identifying the polynomial coefficients.

As it turns out, there is a much simpler approach for defining a hybrid decoupling polynomial for STSARX systems that is valid for all $\boldsymbol{x}$ and all t, irrespective of the value of the discrete state $q_t(\boldsymbol{x})$ and of the regions $\mathcal{X}_i(t)$. The idea is to consider all n ARX models that define the STSARX model. At each time t, a subset of $n_t \leq n$ models with parameters $\{\boldsymbol{b}_i\}_{i \in Q_2(t)}$ is chosen. Then, for all $\boldsymbol{x} \in \mathcal{X}$ and for all $t \geq \max\{\bar{n}_a, \bar{n}_c\}$ there exists an $i \in \{1, 2, \ldots, n\}$ such that $\boldsymbol{b}_i^\top \boldsymbol{z}_t(\boldsymbol{x}) = 0$. Therefore, for all $\boldsymbol{x} \in \mathcal{X}$ and $t \geq \max\{\bar{n}_a, \bar{n}_c\}$, we have

$$p_n(\boldsymbol{z}_t(\boldsymbol{x})) = \prod_{i=1}^{n} (\boldsymbol{b}_i^\top \boldsymbol{z}_t(\boldsymbol{x})) = 0. \tag{10}$$

Notice that the degree n of the hybrid decoupling polynomial for STSARX models, p_n, does not need to coincide with the number of discrete states of any individual SARX model. Also n does not need to correspond with the number of models at time t, n_t. Instead, n is the total number of ARX models, both in space and in time. Therefore, the number of coefficients in p_n does not change with time, which allows us to develop a recursive algorithm for identifying its coefficients, as we will describe in the next subsections.

B. Spatial-Temporal Hybrid Model Parameters

Notice that $p_n(\boldsymbol{z}) \doteq \prod_{i=1}^{n} (\boldsymbol{b}_i^\top \boldsymbol{z})$ is a homogeneous polynomial of degree n in K variables. Hence, if we define the *Veronese map* of degree n, $\nu_n : \mathbb{R}^K \to \mathbb{R}^{M_n(K)}$, as [16]

$$\nu_n : [z_1, \ldots, z_K]^\top \mapsto [\ldots, \gamma_{n_1,\ldots,n_K} z_1^{n_1} \cdots z_K^{n_K} \ldots]^\top, \tag{11}$$

where $M_n(K) = \binom{n+K-1}{K-1} = \binom{n+K-1}{n}$, $0 \leq n_k \leq n$ for $k = 1, \ldots, K$, $\sum_{k=1}^{K} n_k = n$, and $\gamma_{n_1,\ldots,n_K} = \sqrt{\frac{n}{n_1! \cdots n_K!}}$, we can write p_n as a linear combination of the monomials $\gamma_{n_1,\ldots,n_K} z_1^{n_1} z_2^{n_2} \cdots z_K^{n_K}$ with coefficients $h_{n_1,\ldots,n_K} \in \mathbb{R}$ as

$$\sum h_{n_1,\ldots,n_K} \gamma_{n_1,\ldots,n_K} z_1^{n_1} z_2^{n_2} \cdots z_K^{n_K} = \boldsymbol{h}^\top \nu_n(\boldsymbol{z}). \tag{12}$$

The following lemma summarizes some properties of $\boldsymbol{h}$ and ν_n that will be used later in the paper (see [16]).

Lemma 1 (Properties of the Veronese map):

1) The vector $\boldsymbol{h} \in \mathbb{R}^{M_n(K)}$ is a vector representation of the symmetric part of the symmetric tensor product of the individual ARX model parameters $\{\boldsymbol{b}_i\}_{i=1}^n$, i.e.
$$\boldsymbol{h} \simeq \sum_{\sigma \in \mathfrak{S}_n} \boldsymbol{b}_{\sigma(1)} \otimes \boldsymbol{b}_{\sigma(2)} \otimes \cdots \otimes \boldsymbol{b}_{\sigma(n)}, \tag{13}$$
where $\mathfrak{S}_n$ is the permutation group of n elements, and $\otimes$ is the Kronecker product.
2) For all $\boldsymbol{z}_1, \boldsymbol{z}_2 \in \mathbb{R}^K$, the Veronese map of degree n, ν_n, is such that $\nu_n(\boldsymbol{z}_1)^\top \nu_n(\boldsymbol{z}_2) = (\boldsymbol{z}_1^\top \boldsymbol{z}_2)^n$. Hence,
$$\|\nu_n(\boldsymbol{z})\| = \|\boldsymbol{z}\|^n \text{ and } \|\nu_n(\boldsymbol{z})\|^i = \|\nu_i(\boldsymbol{z})\|^n. \tag{14}$$

Since the vector $\boldsymbol{h} \in \mathbb{R}^{M_n(K)}$ encodes the parameters $\{\boldsymbol{b}_i\}_{i=1}^n$ of the n ARX models, we will refer to $\boldsymbol{h}$ as the vector of *spatial-temporal hybrid model parameters* from now on. Notice that the last entry of $\boldsymbol{h}$ is always one, because the last entry of each $\boldsymbol{b}_i$ is also one. Therefore, there is a one-to-one correspondence between $\boldsymbol{h}$ and the ARX model parameters $\{\boldsymbol{b}_i\}_{i=1}^n$ modulo a permutation of the latter ones.

C. Recursive Identification of the Spatial-Temporal Hybrid Model Parameters

For the sake of simplicity, let us first revisit the problem of recursively identifying the parameters of an ARX model with known orders, i.e. $\bar{n}_a = n_a$ and $\bar{n}_c = n_c$. In this case, the *normalized gradient equation error identifier* [17] obtains an estimate $\hat{\boldsymbol{b}}_t$ of the parameter vector $\boldsymbol{b}$ from the data up to time t by following a normalized gradient of the prediction error $\sum_{\tau \geq n_a, n_c}^{t} (\boldsymbol{b}^\top \boldsymbol{z}_\tau)^2$. The update equation is
$$\hat{\boldsymbol{b}}_{t+1} = (I_K - \frac{\mu \Pi_K \boldsymbol{z}_t \boldsymbol{z}_t^\top}{1 + \mu \|\Pi_K \boldsymbol{z}_t\|^2}) \hat{\boldsymbol{b}}_t, \tag{15}$$
where $\mu > 0$ is a fixed parameter, I_K is the identity matrix in $\mathbb{R}^K$, $\Pi_K = \begin{bmatrix} I_{K-1} & \mathbf{0}_{K-1} \\ \mathbf{0}_{K-1}^\top & 0 \end{bmatrix} \in \mathbb{R}^{K \times K}$, and $\mathbf{0}_K \in \mathbb{R}^K$ is the zero vector.[1] It is well known (see e.g., [18]) that if the ARX model is *minimal* and the vectors $\{\Gamma_K \boldsymbol{z}_t\}$, where $\Gamma_K = [\, I_{K-1} \ \mathbf{0}_{K-1} \,] \in \mathbb{R}^{(K-1) \times K}$, are *persistently exciting*, i.e. there are $S \in \mathbb{N}$, $\rho_1, \rho_2 > 0$ such that for all $j \geq \max\{n_a, n_c\}$
$$\rho_1 I \prec \sum_{t=j}^{j+S} \Gamma_K \boldsymbol{z}_t \boldsymbol{z}_t^\top \Gamma_K^\top \prec \rho_2 I, \tag{16}$$
where $A \prec B$ means that $(B - A)$ is positive definite, then $\hat{\boldsymbol{b}}_t - \boldsymbol{b} \to \mathbf{0}$ exponentially.

When the model orders are over estimated, i.e. $\bar{n}_a > n_a$ and $\bar{n}_c > n_c$, then there are $d = \min\{\bar{n}_a - n_a, \bar{n}_c - n_c\}$ additional parameter vectors that fit the data. These additional parameter vectors are formed by shifting the locations of the zeros in the expression for the true ARX model parameters in (5). Therefore, $\hat{\boldsymbol{b}}_t$ no longer converges to $\boldsymbol{b}$ in (5). Instead, it converges to a point $\boldsymbol{b}^*$ orthogonal to the span of the regressors $\{\boldsymbol{z}_t\}$ that depends linearly on the initial condition $\hat{\boldsymbol{b}}_0$ [11]. In order for $\hat{\boldsymbol{b}}_t$ to converge to $\boldsymbol{b}$, we need to enforce that the leading entries of $\boldsymbol{b}$ be zero. If d were known, we could define $\Lambda_d = \begin{bmatrix} \mathbf{0}_{(K-d) \times d} & I_{K-d} \end{bmatrix}$ and replace $\boldsymbol{z}_t$ in (15) by $\Lambda_d \boldsymbol{z}_t$, i.e.
$$\hat{\boldsymbol{b}}_{t+1} = \left(I - \mu \frac{\Pi_{K-d} \Lambda_d \boldsymbol{z}_t \boldsymbol{z}_t^\top \Lambda_d^\top}{1 + \mu \|\Pi_{K-d} \Lambda_d \, \boldsymbol{z}_t\|^2} \right) \hat{\boldsymbol{b}}_t. \tag{17}$$
Then, $\Lambda_d^\top \hat{\boldsymbol{b}}_t \to \boldsymbol{b}$ if the vectors $\{\Gamma_{K-d} \Lambda_d \boldsymbol{z}_t\}$ are persistently exciting.

In reality, we do not necessarily know d, and we need to identify both $\boldsymbol{b}$ and d simultaneously. Thus we may start with $d = 0$ and run two identifiers of the form (17) in parallel starting from two different initial conditions. If the two identifiers converge to different vectors, then we know that the orders are over estimated. Thus, we may increase d by one and run the two identifiers again. By repeating this process with larger and larger values of d, the two identifiers will eventually converge to the same vector. Hence both the orders and the model parameters will be correctly identified.

In order to generalize the recursive identifier (17) to STSARX models such as (3), we exploit the properties of the hybrid decoupling polynomial. Rather than directly identifying the ARX model parameters $\{\boldsymbol{b}_i\}_{i=1}^n$, we propose to first identify the hybrid model parameters $\boldsymbol{h}$. The advantage of doing so is that $\boldsymbol{h}$ depends neither on the value of the discrete state $q_t(\boldsymbol{x})$, nor on the regions $\mathcal{X}_i(t)$. Furthermore, once $\boldsymbol{h}$ is known, one can easily identify the ARX model parameters $\{\boldsymbol{b}_i\}_{i=1}^n$ from the derivatives of the hybrid decoupling polynomial, as shown in [10], [11].

From the hybrid decoupling polynomial for STSARX models, we have that for all $t \geq t_0 = \max\{\bar{n}_a, \bar{n}_c\}$ and $\boldsymbol{x} \in \mathcal{X}$, $\boldsymbol{h}^\top \nu_n(\boldsymbol{z}_t(\boldsymbol{x})) = 0$. Therefore, given measurements up to time t, we can find an estimate $\hat{\boldsymbol{h}}_t$ of $\boldsymbol{h}$ by following the normalized gradient of the mean prediction error
$$f(\boldsymbol{h}) = \frac{1}{|\mathcal{X}|} \sum_{\tau=t_0}^{t} \sum_{\boldsymbol{x} \in \mathcal{X}} (\boldsymbol{h}^\top \nu_n(\boldsymbol{z}_t(\boldsymbol{x})))^2. \tag{18}$$
This leads to the following *hybrid equation error identifier* for STSARX systems:
$$\hat{\boldsymbol{h}}_{t+1} = (I - \frac{\mu}{|\mathcal{X}|} \frac{\sum_{\boldsymbol{x} \in \mathcal{X}} \Pi_{M_n(K)-d} \Lambda_d \nu_n(\boldsymbol{z}_t(\boldsymbol{x})) \nu_n(\boldsymbol{z}_t(\boldsymbol{x}))^\top \Lambda_d^\top}{1 + \frac{\mu}{|\mathcal{X}|} \sum_{\boldsymbol{x} \in \mathcal{X}} \|\Pi_{M_n(K)-d} \Lambda_d \nu_n(\boldsymbol{z}_t(\boldsymbol{x}))\|^2}) \hat{\boldsymbol{h}}_t. \tag{19}$$
Notice that (19) reduces to (17) when $|\mathcal{X}| = n = 1$. As in the case of ARX models, we set d entries of $\hat{\boldsymbol{h}}_t$ to zero by multiplying the embedded regressors by the matrix $\Lambda_d = \begin{bmatrix} \mathbf{0}_{(M_n(K)-d) \times d} & I_{M_n(K)-d} \end{bmatrix}$. In the case of STSARX models, d is the number of linearly independent polynomials that together with p_n fit the data. The value of d depends nontrivially on n, $\bar{n}_a$, $\bar{n}_c$, $\{n_a(i)\}$ and $\{n_c(i)\}$.

The natural question is whether $\Lambda_d^\top \hat{\boldsymbol{h}}_t$ converges to $\boldsymbol{h}$. As in the case of ARX models, the answer is *no* in general. This is because when we over estimate the model orders, there are two or more parameter vectors that fit the data coming from some of the ARX models, and so there is more than

[1] For notational convenience, we will drop the subindex K in I_K and $\mathbf{0}_K$ whenever understood.

one polynomial fitting the data coming from the STSARX model. In the case of STSARX models, the situation is even more complicated. For instance, even if the orders of the ARX models were known, it could be the case that different ARX models have different orders. Thus, even if we choose $\bar{n}_a = \max\{n_a(i)\}$ and $\bar{n}_c = \max\{n_c(i)\}$, we may still over estimate the orders for some of the ARX models. The following theorem shows that under a certain persistence of excitation condition, the sequence $\Lambda_d^\top \hat{\boldsymbol{h}}_t$ converges to $\boldsymbol{h}$ exponentially when d is known.

Theorem 1: Consider a minimal STSARX system of the form (3) and an identifier of the form (19). If there are $S \in \mathbb{N}$, $\rho_1, \rho_2 > 0$ such that for all $j \geq \max\{n_a(i), n_c(i)\}$

$$\rho_1 I \prec M \prec \rho_2 I, \tag{20}$$

where the matrix M is defined as

$$\sum_{t=j}^{j+S} \sum_{\boldsymbol{x} \in \mathcal{X}} \Gamma_{M_n(K)-d} \Lambda_d \nu_n(\boldsymbol{z}_t(\boldsymbol{x})) \nu_n(\boldsymbol{z}_t(\boldsymbol{x}))^\top \Lambda_d^\top \Gamma_{M_n(K)-d}^\top \tag{21}$$

then $\Lambda_d^\top \hat{\boldsymbol{h}}_t$ converges exponentially to $\boldsymbol{h}$.

Unfortunately, the theorem requires one to know the number of additional polynomials d that fit the data generated by the STSARX system. However, when the orders of the ARX models are unknown, the value of d is also unknown. Therefore, as in the case of ARX models, if we underestimate the value of d, then the recursive identifier will not converge to the correct hybrid model parameters. Instead, it will converge to a vector $\boldsymbol{h}^*$ that depends linearly on the initial condition $\hat{\boldsymbol{h}}_0$ and lies in the orthogonal complement of the span of the vectors $\nu_n(\boldsymbol{z}_t(\boldsymbol{x}))$. Therefore, we may start with $d = 0$ and run two hybrid identifiers in parallel starting from two different initial conditions. If the two identifiers converge to different vectors, then we know that d is under estimated. Thus, we may increase d by one, and run two identifiers again. By repeating this process with larger and larger values of d, the two identifiers will eventually converge to the same vector. Hence both d and $\boldsymbol{h}$ will be correctly identified.

D. Recursive Identification of the ARX Model Parameters

Given the vector of hybrid model parameters $\boldsymbol{h}$, it is well known (see e.g., [10], [11]) that one can obtain the individual ARX model parameters from the gradient of the hybrid decoupling polynomial

$$\nabla p_n(\boldsymbol{z}) = \frac{\partial p_n(\boldsymbol{z})}{\partial \boldsymbol{z}} = \sum_{i=1}^{n} \prod_{\ell \neq i} (\boldsymbol{b}_\ell^\top \boldsymbol{z}) \boldsymbol{b}_i \tag{22}$$

at $\boldsymbol{z} = \boldsymbol{z}_t(\boldsymbol{x})$. This is because if $\boldsymbol{z}_t(\boldsymbol{x})$ is generated by the ith ARX model, i.e. $q_t(\boldsymbol{x}) = i$ where $i = 1, \ldots, n$, then all the terms in the summation in (22) vanish, except for the ith, and so $\boldsymbol{b}_{q_t(\boldsymbol{x})} \sim \nabla p_n(\boldsymbol{z}_t(\boldsymbol{x}))$. Since in addition the Kth entry of $\boldsymbol{b}_{q_t(\boldsymbol{x})}$ is equal to one, we obtain

$$\boldsymbol{b}_{q_t(\boldsymbol{x})} = \frac{\nabla p_n(\boldsymbol{z}_t(\boldsymbol{x}))}{e_K^\top \nabla p_n(\boldsymbol{z}_t(\boldsymbol{x}))}, \tag{23}$$

where $e_K = [0, \cdots, 0, 1]^\top \in \mathbb{R}^K$.

In reality, however, we do not know the polynomial p_n exactly. Instead, we compute an estimate $\Lambda_d^\top \hat{\boldsymbol{h}}_t$ of its coefficients from the measurements of the input/output data collected up to time t. We can use this estimate to build the polynomial $\hat{p}_n(\boldsymbol{z}) = \hat{\boldsymbol{h}}_t^\top \Lambda_d \nu_n(\boldsymbol{z})$, from which we can identify the parameters of the ARX model at time t as

$$\hat{\boldsymbol{b}}_t(\boldsymbol{x}) = \frac{\nabla \nu_n^\top(\boldsymbol{z}_t(\boldsymbol{x})) \Lambda_d^\top \hat{\boldsymbol{h}}_t}{e_K^\top \nabla \nu_n^\top(\boldsymbol{z}_t(\boldsymbol{x})) \Lambda_d^\top \hat{\boldsymbol{h}}_t}, \tag{24}$$

The next theorem shows that the estimate given by this identifier converges exponentially to the true model parameters.

Theorem 2: Consider a minimal STARX system of the form (3) and assume that the hybrid recursive identifier (19) and (24) is used. If there exist $\rho_1, \rho_2 > 0$ and an integer S such that for all $j \geq \max\{n_a(i), n_c(i)\}$

$$\rho_1 I \prec M \prec \rho_2 I, \tag{25}$$

then $\hat{\boldsymbol{b}}_t(\boldsymbol{x}) - \boldsymbol{b}_{q_t(\boldsymbol{x})} \to \boldsymbol{0}_K$ exponentially.

E. Estimation of the Spatial Regions

Theorem 2 allows us to computing an estimate $\hat{\boldsymbol{b}}_t(\boldsymbol{x})$ of the parameters of the ARX model associated with the measurement $\boldsymbol{z}_t(\boldsymbol{x})$. Since all spatial locations in a region $\mathcal{X}_i(t)$ share the same ARX model parameters, we may find the regions $\{\mathcal{X}_i(t)\}$ by clustering the vectors $\{\hat{\boldsymbol{b}}_t(\boldsymbol{x})\}_{\boldsymbol{x} \in \mathcal{X}}$. We do so using the K-means algorithm [19]. More specifically, let us define the segmentation variables $w_{i\boldsymbol{x}}(t) \in \{0, 1\}$ as

$$w_{i\boldsymbol{x}}(t) = \begin{cases} 1 & \text{if } \boldsymbol{x} \in \mathcal{X}_i(t) \\ 0 & \text{else} \end{cases}. \tag{26}$$

At each t, the recursive K-means algorithms looks for n_t parameter vectors $\hat{\boldsymbol{b}}_i(t)$ and the segmentation of the data $w_{i\boldsymbol{x}}(t) \in \{0, 1\}$ that minimize

$$g(\{w_{i\boldsymbol{x}}(t)\}, \{\hat{\boldsymbol{b}}_i(t)\}) = \sum_{\boldsymbol{x} \in \mathcal{X}} \sum_{i=1}^{n} w_{i\boldsymbol{x}}(t) \|\hat{\boldsymbol{b}}_i(t) - \hat{\boldsymbol{b}}_t(\boldsymbol{x})\|^2. \tag{27}$$

The minimization of this cost function is carried out using the following coordinate descent algorithm:

1) Given a current estimate for $\hat{\boldsymbol{b}}_i(t)$, compute the segmentation of the data as

$$w_{i\boldsymbol{x}}(t) = \begin{cases} 1 & \text{if } i = \underset{k=1,\ldots,n}{\arg\min} \|\hat{\boldsymbol{b}}_k(t) - \hat{\boldsymbol{b}}_t(\boldsymbol{x})\|^2 \\ 0 & \text{else} \end{cases}. \tag{28}$$

2) Given a current estimate of $w_{i\boldsymbol{x}}(t)$, estimate the model parameters as

$$\boldsymbol{b}_i(t) = \frac{\sum_{\boldsymbol{x} \in \mathcal{X}} w_{i\boldsymbol{x}}(t) \hat{\boldsymbol{b}}_t(\boldsymbol{x})}{\sum_{\boldsymbol{x} \in \mathcal{X}} w_{i\boldsymbol{x}}(t)}, \tag{29}$$

and the spatial regions at time t as

$$\mathcal{X}_i(t) = \{\boldsymbol{x} \in \mathcal{X} : w_{i\boldsymbol{x}}(t) = 1\}. \tag{30}$$

3) Iterate until the memberships do not change.

This method gives an estimate of the ARX model parameters and the spatial regions at each time t. One may use the solution at time t to initialize the method at time $t + 1$.

The batch K-means algorithm is also used as part of existing hybrid system identification algorithms, such as [13].

III. EXPERIMENTS

In this section, we apply our identification algorithm to the problem of segmenting video sequences of dynamic textures, such as fire, steam, water, or fire. As proposed in [1], a video sequence of a single dynamic texture can be modeled as the output of a linear dynamical system. Here we propose to model video sequences containing several dynamic textures in space and time using an STSARX model.

A. Segmentation of a Periodic Motion

We first apply our algorithm for segmenting a sequence ($110 \times 192, 130$ frames) with a bird floating on water while rotating around a fixed point. The task is to segment the bird's rigid motion from the water's dynamic texture. This sequence was segmented in [15] using a recursive hyperplane clustering technique. Here we segment it using the proposed recursive identification algorithm for STARX models. After using the same projection steps as in [15], we model the projected data as the output of two AR models of order $n_c = 4$, one representing the bird's motion, and the other representing the water motion. We initialize our recursive identification algorithm by applying Generalized Principal Component Analysis (GPCA) [7] to the projected data in the first 5 frames of the video sequence. GPCA is a batch method for clustering data living in multiple hyperplanes. Since the equation of an ARX model defines a hyperplane in the space of regressors, identifying multiple ARX models can be seen as a hyperplane clustering problem.

Figure 1 shows the segmentation results. Notice that the bird is relatively well segmented from the water. However, there are some portions of the water that are assigned to the bird. This is because our method does not constraint the regions $\mathcal{X}_i(t)$ to be connected, thus the final segmentation need not be spatially coherent. Also, although the sequence is short for the vector of hybrid parameters $\boldsymbol{h}$ to converge, it is clear from the last row of Figure 1 that $\boldsymbol{h}$ already captures the periodicity of the motion. For instance, notice that when the bird is facing to the right, $\boldsymbol{h}_8$ achieves a local maximum. On the contrary, if the bird is oriented to the left, $\boldsymbol{h}_8$ achieves a local minimum. Also, some irregularities seem to appear at the local minima of this coefficient: they actually correspond to a rapid motion of the bird. This example shows that the coefficients of the estimated hybrid decoupling polynomial give useful information about the scene motion.

B. Segmentation of Appearing and Disappearing Motions

To test the performance of our algorithm on a video sequence with a variable number of motions, we extracted a sub-clip of the bird sequence (55×192, 130 frames) in which the camera moves up at 1 pixel/frame until the bird disappears at $t = 51$. Then, the camera stays stationary from $t = 56$ to $t = 66$. Finally, the camera moves down at 1 pixel/frame, and the bird reappears at $t = 76$. We compare our recursive identification algorithm against GPCA applied to a moving window of 5 consecutive frames. We initialize our recursive identification algorithm with the solution of GPCA applied to the first 5 frames.

Figure 2 shows the segmentation results. Notice that both methods give excellent results during the first few frames where both the bird and the water are present. This is expected, as our method is initialized with GPCA. Nevertheless, notice that the performance of GPCA deteriorates dramatically when the bird disappears, because GPCA overestimates the number of models, whereas our method is robust to this change and keeps segmenting the scene correctly, i.e. assigning most of the pixels to the background. When the bird reappears, our method detects the bird correctly from the first frame whereas GPCA produces a wrong segmentation for the first few frames after the bird reappears. Toward the end of the sequence, the segmentation given by both algorithms is good, though our algorithm performs better, because it incorporates temporal coherence. This demonstrates how our method has the ability to deal with a variable number of motions, while GPCA does not.

Acknowledgements: We thank Camille Izard for performing the experiments. Work supported by grants NSF-EHS 05-09101, NSF-CAREER 04-47739, and ONR N00014-05-1083.

IV. CONCLUSIONS AND FUTURE WORK

We have presented a recursive identification algorithm for spatial-temporal switched ARX systems. The algorithm is based on recursively updating the coefficients of a polynomial that is satisfied by all measurements, regardless of their spatial-temporal location. The parameters of each ARX model are identified from the derivatives of the polynomial. The regions associated with each model at each time instant are obtained by clustering the ARX model parameters.

Future work involves exploiting the spatial structure of STSARX systems to obtain more efficient solutions. For example, [21] shows how least-squares problems with spatially decaying operators can be solved more efficiently.

REFERENCES

[1] G. Doretto, A. Chiuso, Y. Wu, and S. Soatto. Dynamic textures. *International Journal of Computer Vision*, 51(2):91–109, 2003.

[2] P. Saisan, G. Doretto, Y. N. Wu, and S. Soatto. Dynamic texture recognition. In *Conference on Computer Vision and Pattern Recognition*, volume II, pages 58–63, 2001.

[3] G. Aggarwal, A. Roy-Chowdhury, and R. Chellappa. A system identification approach for video-based face recognition. In *IEEE International Conference on Pattern Recognition*, pages 23–26, 2004.

[4] P. Saisan, A. Bissacco, A. Chiuso, and S. Soatto. Modeling and synthesis of facial motion driven by speech. In *European Conference on Computer Vision*, volume 3, pages 456–467, 2004.

[5] P. Van Overschee and B. De Moor. Subspace algorithms for the stochastic identification problem. *Automatica*, 29(3):649–660, 1993.

[6] K. Huang, A. Wagner, and Y. Ma. Identification of hybrid linear time-invariant systems via subspace embedding and segmentation. In *Conference on Decision and Control*, 2004.

[7] R. Vidal, Y. Ma, and S. Sastry. Generalized Principal Component Analysis (GPCA). *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 27(12):1–15, 2005.

[8] R. Vidal, S. Soatto, Y. Ma, and S. Sastry. An algebraic geometric approach to the identification of a class of linear hybrid systems. In *Conference on Decision and Control*, pages 167–172, 2003.

[9] R. Vidal and B.D.O. Anderson. Recursive identification of switched ARX hybrid models: Exponential convergence and persistence of excitation. In *Conf. on Decision and Control*, pages 32–37, 2004.

[10] Y. Ma and R. Vidal. Identification of deterministic switched ARX systems via identification of algebraic varieties. In *Hybrid Systems: Computation and Control*, pages 449–465. Springer Verlag, 2005.

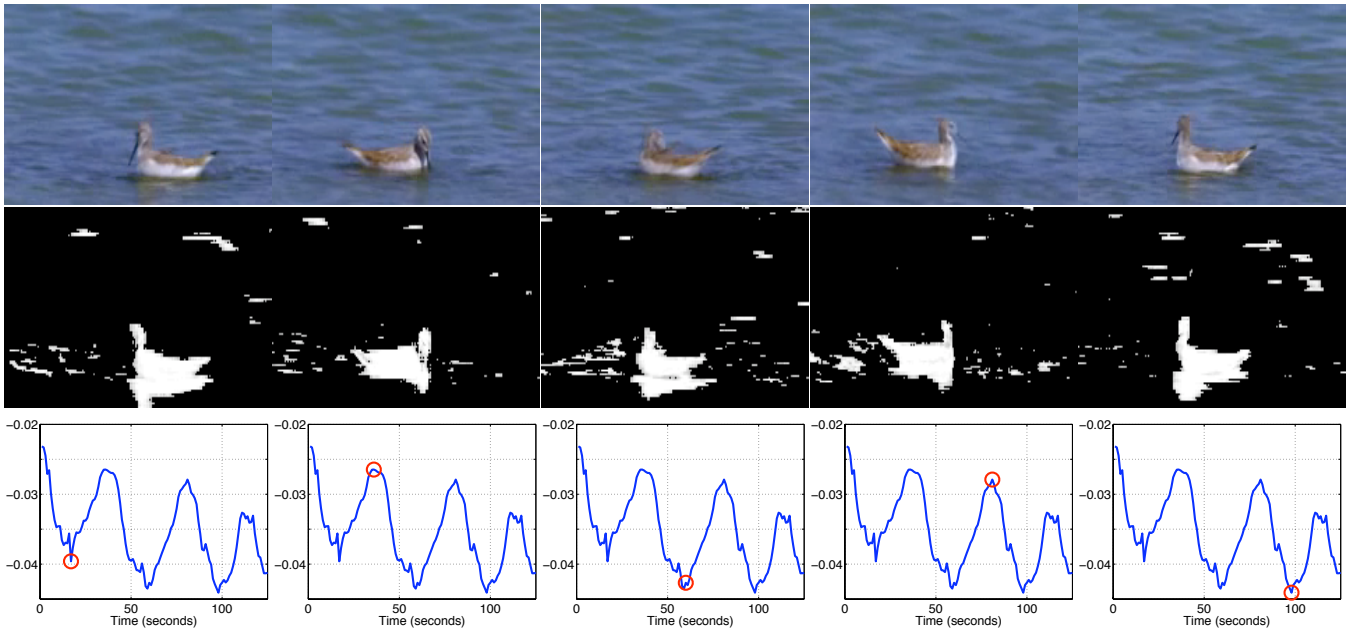

Fig. 1. Using hybrid system identification to segment a bird floating on water. The first row shows frames 17, 36, 60, 81, and 98 of the video sequence. The middle row shows the segmentation of the bird (white) from the water (black) obtained using our recursive identification algorithm. The bottom row shows the temporal evolution of the coefficient $\boldsymbol{h}_8$ of the hybrid decoupling polynomial. The red dot indicates the location of the corresponding frame.

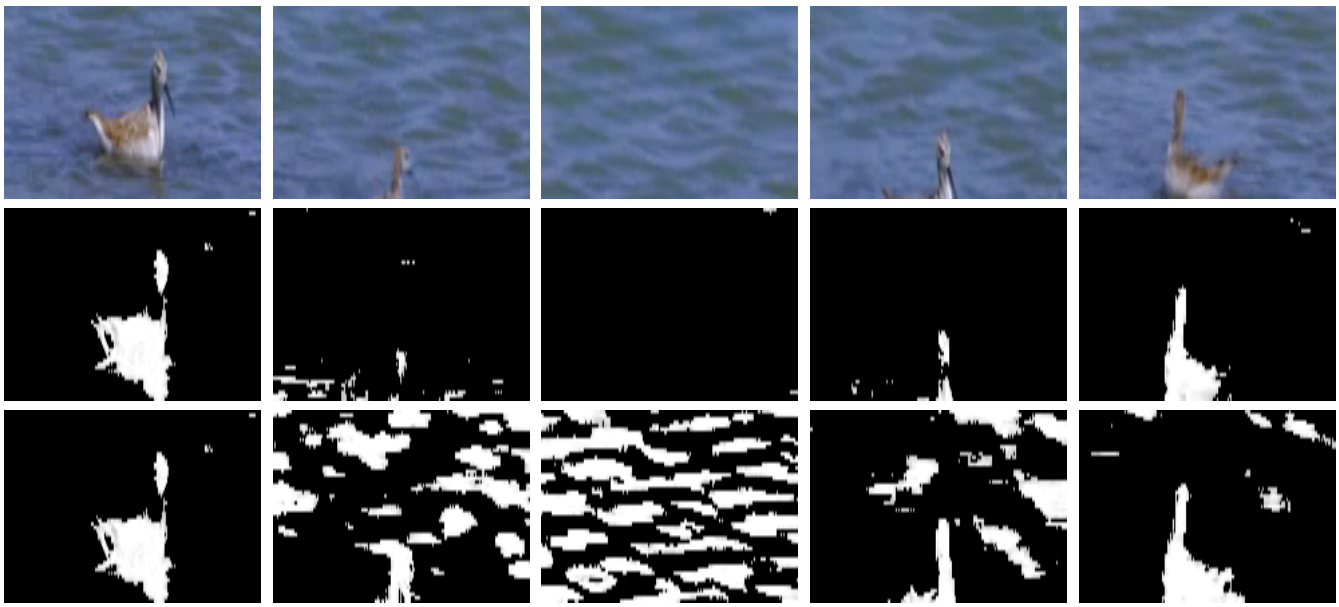

Fig. 2. Using hybrid system identification to segment a video sequence containing a variable number of motions. The first row shows frames 1, 29, 65, 77, and 101 of a video sequence containing a bird floating on water that disappears (third column) and reappears (fourth column) in the scene. The middle row shows the segmentation of the bird (white) from the water (black) using our method. The bottom row shows the segmentation using GPCA.

[11] Y. Hashambhoy and R. Vidal. Recursive identification of switched ARX models with unknown number of models and unknown orders. In *Conference on Decision and Control*, pages 6115–6121, 2005.

[12] A. Bemporad, A. Garulli, S. Paoletti, and A. Vicino. A bounded-error approach to piecewise affine system identification. *IEEE Transactions on Automatic Control*, 50(10):1567–1580, 2005.

[13] G. Ferrari-Trecate, M. Muselli, D. Liberati, and M. Morari. A clustering technique for the identification of piecewise affine systems. *Automatica*, 39(2):205–217, 2003.

[14] A. Juloski, S. Weiland, and M. Heemels. A Bayesian approach to identification of hybrid systems. *IEEE Transactions on Automatic Control*, 50(10):1520–1533, 2005.

[15] R. Vidal. Online clustering of moving hyperplanes. In *Neural Information Processing Systems, NIPS*, 2006.

[16] J. Harris. *Algebraic Geometry: A First Course*. Springer-Verlag, 1992.

[17] L. Ljung. *System Identification: theory for the user*. Prentice Hall, 1987.

[18] B.D.O. Anderson, R.R. Bitmead, C.R. Johnson Jr., P.V. Kokotovic, R.L. Ikosut, I.M.Y. Mareels, L. Praly, and B.D. Riedle. *Stability of Adaptive Systems*. MIT Press, 1986.

[19] R. Duda, P. Hart, and D. Stork. *Pattern Classification*. Wiley, New York, 2nd edition, 2000.

[20] R. Vidal and A. Ravichandran. Optical flow estimation and segmentation of multiple moving dynamic textures. In *Conference on Computer Vision and Pattern Recognition*, volume II, pages 516–521, 2005.

[21] N. Motee and A. Jadbabaie. Distributed quadratic programming over arbitrary graphs. *IEEE Transactions on Automatic Control*, 2007.