
Recursive Filtering using Quasi-Realizations

Bart Vanluyten^{1, 2}, Jan C. Willems¹, and Bart De Moor¹

¹ Department of Electrical Engineering (ESAT), Katholieke Universiteit Leuven, 3001 Heverlee (Leuven), Belgium. `bart.vanluyten@esat.kuleuven.be`

² Bart Vanluyten is a Research Assistant with the fund for Scientific Research-Flanders (FWO-Vlaanderen).

Summary. In this paper we consider a finite state Markov chain with two outputs, an observed output and a to-be-estimated output, and derive a recursive estimator for the to-be-estimated output from an observed output string. The main point of this article is to illustrate that for this kind of filtering problem, it is not needed to have a positive hidden Markov realization of the joint process, but it suffices to have a quasi-realization. We also present an approximate quasi-realization algorithm. We perform a simulation comparing the behavior of the exact, experimental and approximate quasi-realizations and checking the performance of the estimator.

1 Filtering problems for finitary processes

Consider a stochastic process $[y z]^\top$ defined on the time axis \mathbb{N} , where both outputs y and z take values from finite sets. The main problem is to derive a recursive estimator for the to-be-estimated output z from an observed output string y . As data for the problem we assume the string probabilities of all possible strings $\left([y(1) z(1)]^\top [y(2) z(2)]^\top, \dots, [y(T) z(T)]^\top \right)$ for all $T \in \mathbb{N}$.

Our approach is to split up the problem in two steps. In the first step we model the given output probabilities by a joint quasi-hidden Markov model. The second step is the filtering step where we calculate the estimate for z based on the joint quasi-hidden Markov model and the observed string y . We will show that for this filtering application, it suffices to have a quasi-realization of the string probabilities rather than a true realization. The advantages of this last fact are twofold: first of all, there is no need to calculate a true realization, which is usually much more complicated to obtain than a quasi-realization [1, 5]. In fact, there exist no general algorithms for computing a true stochastic realization. Moreover, there are processes for which a true realization does not exist while a quasi-realization does exist. Second, the dynamic order of a quasi-realization is often smaller than the order of a true realization which makes the filtering computation less expensive.

In Section 2 we give an overview of the quasi-realization problem and recall an algorithm to solve it. In Section 3 we derive the formulas for the recursive filter. In Section 4 we present an approximate quasi-realization algorithm which obtains a

balanced reduced quasi-realization of the string probabilities and in Section 5 finally, we give a simulation example showing the effectiveness of the proposed estimator.

The following notation is used throughout the paper. If X is a matrix, then $X_{i:k,j:l}$ denotes the submatrix of X formed by the i -th to the k -th row and by the j -th to the l -th column of X . With $X_{i,j}$, we mean the i, j -th element of X .

2 Quasi-realizations of finitary processes

Consider a stochastic process y defined on the time axis \mathbb{N} taking values from a finite set \mathbb{Y} , called the output alphabet, with $|\mathbb{Y}|$ the cardinality of \mathbb{Y} . Denote by \mathbb{Y}^* the set of all finite strings with symbols from the set \mathbb{Y} (including the empty string) and by $y = y_1 y_2 \dots y_{|y|}$ an output sequence from \mathbb{Y}^* , where $|y|$ denotes the length of y . Let $\mathcal{P} : \mathbb{Y}^* \mapsto [0, 1]$ be string probabilities, defined as $\mathcal{P}(y) := P(y(1) = y_1, y(2) = y_2, \dots, y(|y|) = y_{|y|})$. Of course, the string probabilities satisfy $\mathcal{P}(\phi) = 1$ and $\sum_{y \in \mathbb{Y}} \mathcal{P}(y) = \mathcal{P}(\phi)$.

A *quasi-hidden Markov model* is defined as $(\mathbb{X}_q, \mathbb{Y}, \Pi_q, \pi_q, e_q)$, where

- \mathbb{X}_q with $|\mathbb{X}_q| < \infty$ is the quasi-state alphabet, and \mathbb{Y} is the output alphabet;
- e_q is a column vector in $\mathbb{R}^{|\mathbb{X}_q|}$, and π_q is a row vector in $\mathbb{R}^{|\mathbb{X}_q|}$ with $\pi_q e_q = 1$.
- Π_q is a mapping from \mathbb{Y} to $\mathbb{R}^{|\mathbb{X}_q| \times |\mathbb{X}_q|}$ with the matrix $\Pi_{\mathbb{X}_q} := \sum_{y \in \mathbb{Y}} \Pi_q(y)$ such that $\Pi_{\mathbb{X}_q} e_q = e_q$.

In the *quasi-realization* problem, we are given the output string probabilities \mathcal{P} and the problem is to find a quasi-HMM that realizes \mathcal{P} , which means that for all $y = y_1 y_2 \dots y_{|y|} \in \mathbb{Y}^*$, it holds that $\mathcal{P}(y) = \pi_q \Pi_q(y) e_q$, where $\Pi_q(y) = \Pi_q(y_1) \Pi_q(y_2) \dots \Pi_q(y_{|y|})$.

A quasi-realization $(\mathbb{X}'_q, \mathbb{Y}, \Pi'_q, \pi'_q, e'_q)$ of \mathcal{P} is called *minimal* if for any other realization $(\mathbb{X}''_q, \mathbb{Y}, \Pi''_q, \pi''_q, e''_q)$ of \mathcal{P} , it holds that $|\mathbb{X}'_q| \leq |\mathbb{X}''_q|$.

The (*generalized*) *Hankel matrix* of \mathcal{P} plays a central role in the quasi-realization problem [4]. To build the Hankel matrix, we need two arbitrary orderings $u := (u_i, i = 1, 2, \dots)$ and $v := (v_j, j = 1, 2, \dots)$ of the strings of \mathbb{Y}^* . The generalized Hankel matrix \mathfrak{H} of \mathcal{P} is now defined as the doubly infinite matrix with i, j -th element $\mathcal{P}(u_i v_j)$, where u_i and v_j are the i -th and j -th elements of u and v , and where $u_i v_j$ denotes the concatenation of the strings u_i and v_j . Typically, in the first ordering the strings are ordered lexicographically from right to left, which gives $(\phi, 0, 1, 00, 10, 01, 11, 000, 100, \dots)$ for $\mathbb{Y} = \{0, 1\}$. In the second ordering the strings are ordered lexicographically from left to right, which means $(\phi, 0, 1, 00, 01, 10, 11, 000, 001, \dots)$ for $\mathbb{Y} = \{0, 1\}$. The top left corner of the Hankel matrix \mathfrak{H} for the case where $\mathbb{Y} = \{0, 1\}$ then looks like

$$\begin{bmatrix} \mathbf{1} & \mathcal{P}(0) & \mathcal{P}(1) & \mathcal{P}(00) & \mathcal{P}(01) & \mathcal{P}(10) & \mathcal{P}(11) \\ \mathcal{P}(0) & \mathcal{P}(00) & \mathcal{P}(01) & \mathcal{P}(000) & \mathcal{P}(001) & \mathcal{P}(010) & \mathcal{P}(011) \\ \mathcal{P}(1) & \mathcal{P}(10) & \mathcal{P}(11) & \mathcal{P}(100) & \mathcal{P}(101) & \mathcal{P}(110) & \mathcal{P}(111) \\ \mathcal{P}(00) & \mathcal{P}(000) & \mathcal{P}(001) & \mathcal{P}(0000) & \mathcal{P}(0001) & \mathcal{P}(0010) & \mathcal{P}(0011) \\ \mathcal{P}(10) & \mathcal{P}(100) & \mathcal{P}(101) & \mathcal{P}(1000) & \mathcal{P}(1001) & \mathcal{P}(1010) & \mathcal{P}(1011) \\ \mathcal{P}(01) & \mathcal{P}(010) & \mathcal{P}(011) & \mathcal{P}(0100) & \mathcal{P}(0101) & \mathcal{P}(0110) & \mathcal{P}(0111) \\ \mathcal{P}(11) & \mathcal{P}(110) & \mathcal{P}(111) & \mathcal{P}(1100) & \mathcal{P}(1101) & \mathcal{P}(1110) & \mathcal{P}(1111) \end{bmatrix}.$$

The Hankel matrix of the string probabilities \mathcal{P} of the output process of a minimal quasi-HMM with $|\mathbb{X}_q|$ finite can be decomposed as $\mathfrak{H} = \mathcal{O}_q \mathcal{C}_q$, with

$$\begin{aligned}\mathcal{O}_q &= \text{col}(\pi_q, \pi_q \Pi_q(0), \pi_q \Pi_q(1), \pi_q \Pi_q(00), \pi_q \Pi_q(10), \dots), \\ \mathcal{C}_q &= \text{row}(e_q, \Pi_q(0)e_q, \Pi_q(1)e_q, \Pi_q(00)e_q, \Pi_q(01)e_q, \dots),\end{aligned}$$

where \mathcal{O}_q is injective and \mathcal{C}_q is surjective. Now, the following theorem is well-known [2]:

Theorem 1 *Let \mathcal{P} be the string probabilities of a process with values from a finite set \mathbb{Y} . Then:*

1. *There exists a quasi-realization $(\mathbb{X}_q, \mathbb{Y}, \Pi_q, \pi_q, e_q)$ of \mathcal{P} if and only if the rank of the (infinite) generalized Hankel matrix \mathfrak{H} of \mathcal{P} , is finite.*
2. *The minimal order of a quasi-realization $|\mathbb{X}_q|_{\min}$ is equal to the rank of the Hankel matrix \mathfrak{H} .*
3. *If $(\mathbb{X}_q, \mathbb{Y}, \Pi_q, \pi_q, e_q)$ and $(\mathbb{X}_q, \mathbb{Y}, \Pi'_q, \pi'_q, e'_q)$ are two minimal quasi-realizations, then there exists a nonsingular matrix T such that*

$$\pi_q = \pi'_q T, \quad \Pi(y) = T^{-1} \Pi'(y) T, \quad e_q = T^{-1} e'_q.$$

We now present a general algorithm to find a minimal quasi-realization given the generalized Hankel matrix \mathfrak{H} associated with the output string probabilities.

Step 1: Find a sub-matrix $M \in \mathbb{R}^{n' \times n''}$ of \mathfrak{H} with $\text{rank}(M) = \text{rank}(\mathfrak{H})$. Assume that M is formed by the elements in the rows indexed by the strings $u_{r_1}, u_{r_2}, \dots, u_{r_{n'}}$ and columns $v_{c_1}, v_{c_2}, \dots, v_{c_{n''}}$.

Step 2: Let $R \in \mathbb{R}^{1 \times n''}$ be the sub-matrix of \mathfrak{H} formed by the elements in the first row and columns indexed by the strings $v_{c_1}, v_{c_2}, \dots, v_{c_{n''}}$ and analogously $K \in \mathbb{R}^{n' \times 1}$ be the sub-matrix of \mathfrak{H} formed by the elements in the rows indexed by the strings $u_{r_1}, u_{r_2}, \dots, u_{r_{n'}}$ and the first column. For each $y \in \mathbb{Y}$ define $\sigma_y M \in \mathbb{R}^{n' \times n''}$ as the submatrix of \mathfrak{H} formed by the elements in the rows indexed by the strings $u_{r_1} y, u_{r_2} y, \dots, u_{r_{n'}} y$ and in the columns indexed by the strings $v_{c_1}, v_{c_2}, \dots, v_{c_{n''}}$, where $u_{r_i} y$ denotes the concatenation of the string u_{r_i} and the symbol y .

Step 3: Find $P \in \mathbb{R}^{|\mathbb{X}_q|_{\min} \times n'}$ and $Q \in \mathbb{R}^{n'' \times |\mathbb{X}_q|_{\min}}$ such that $PMQ = I_{|\mathbb{X}_q|_{\min}}$.

Step 4: A minimal quasi-realization $(\mathbb{X}_q, \mathbb{Y}, \Pi_q, \pi_q, e_q)$ is now obtained as follows:

$$\begin{aligned}\Pi_q(y) &= P \sigma_y M Q \quad \forall y \in \mathbb{Y}, \\ \pi_q &= RQ, \\ e_q &= PK.\end{aligned}$$

A *hidden Markov model* (HMM) $(\mathbb{X}, \mathbb{Y}, \Pi, \pi, e)$ is a special case of a quasi-hidden Markov model where the elements of $\pi, \Pi(y), y \in \mathbb{Y}$ are nonnegative and $e := [1 \ 1 \ \dots \ 1]^\top$.

In the case of a hidden Markov model, the system matrices have a probabilistic interpretation. There is an underlying state process x which generates the output process y . The process x takes values from the finite set \mathbb{X} with cardinality $|\mathbb{X}|$.

Without loss of generality, we take $\mathbb{X} = \{1, 2, \dots, |\mathbb{X}|\}$. The element $\Pi(\mathbf{y})_{i,j}$ is equal to $P(x(t+1) = j, y(t) = \mathbf{y} | x(t) = i)$, the probability of going from state i to state j while producing the output symbol \mathbf{y} . The element π_i is equal to $P(x(1) = i)$, the initial distribution of the underlying state process.

The hidden Markov model $(\mathbb{X}, \mathbb{Y}, \Pi, \pi, e)$ is said to be a *stochastic realization* of \mathcal{P} if, for all $y_1 y_2 \dots y_{|\mathbb{Y}|} \in \mathbb{Y}^*$, it holds that $\mathcal{P}(\mathbf{y}) = \pi \Pi(y_1) \Pi(y_2) \dots \Pi(y_{|\mathbb{Y}|}) e$.

It is immediately clear that the minimal order of a true stochastic realization of an output process \mathcal{P} is at least as large than the minimal order of a quasi-realization of \mathcal{P} . In Section 5, we give an example where the minimal order is equal to 4, while the minimal quasi-order is equal to 3.

3 Recursive filtering

We consider a quasi-hidden Markov model $(\mathbb{X}_q, \mathbb{Y} \times \mathbb{Z}, \Pi_q, \pi_q, e_q)$ with two output processes, an observed output process y and a to-be-estimated output process z . The output alphabets are \mathbb{Y} and \mathbb{Z} respectively. The aim is to find a mapping \hat{z} from $\mathbb{Z} \times \mathbb{Y}^*$ to \mathbb{R}_+ such that

$$\hat{z}(\mathbf{z}; y_1 \dots y_{t-1}) = P(z(t) = \mathbf{z} | y(1) = y_1, \dots, y(t-1) = y_{t-1}).$$

Define $B_q^{(z)}$ as a mapping from \mathbb{Z} to $\mathbb{R}^{|\mathbb{X}_q|}$ where $B_q^{(z)}(\mathbf{z}) = \sum_{y \in \mathbb{Y}} \Pi_q(\mathbf{y}, \mathbf{z}) e_q$ and $\Pi_q^{(y)}$ as a mapping from \mathbb{Y} to $\mathbb{R}^{|\mathbb{X}_q| \times |\mathbb{X}_q|}$ where $\Pi_q^{(y)}(\mathbf{y}) := \sum_{z \in \mathbb{Z}} \Pi_q(\mathbf{y}, \mathbf{z})$.

Proposition 1 *The following equations define a recursive algorithm that computes \hat{z} from the past of y :*

$$\begin{aligned} \tilde{\pi}_1 &= \pi_q, \\ \tilde{\pi}_{t+1} &= \tilde{\pi}_t \Pi_q^{(y)}(y_t), \\ \hat{\pi}_t &= \frac{\tilde{\pi}_t}{\tilde{\pi}_t e_q}, \\ \hat{z}(\mathbf{z}; y_1 \dots y_{t-1}) &= \hat{\pi}_t B_q^{(z)}(\mathbf{z}), \quad \forall \mathbf{z} \in \mathbb{Z}. \end{aligned}$$

This can be seen from

$$\begin{aligned} \hat{z}(\mathbf{z}; y_1 \dots y_{t-1}) &= \frac{P(y(1) = y_1, \dots, y(t-1) = y_{t-1}, z(t) = \mathbf{z})}{\mathcal{P}(y_1 y_2 \dots y_{t-1})} \\ &= \frac{\tilde{\pi}_t B_q^{(z)}(\mathbf{z})}{\tilde{\pi}_t e_q}. \end{aligned}$$

As a true realization is a special case of a quasi-realization, the formulas are also valid for a true realization. In that case, the intermediate variable $\hat{\pi}_t$ has a probabilistic interpretation. One can show that $\hat{\pi}_t = P(x(t) = j | y(1) = y_1, \dots, y(t-1) = y_{t-1})$.

So we derived a recursive filter which can be calculated from a quasi-realization without the need for calculating a true stochastic realization. As already mentioned, the advantage of this approach is twofold. First of all, there is no need to calculate a true stochastic realization, which is typically more expensive than calculating a quasi-realization. In fact, there exist no general algorithms for computing a true stochastic realization. Second, a quasi-realization typically has lower order than a true realization, which makes that the filter itself becomes less complex.

4 Approximate quasi-realization

In this section, we extend the idea of balanced realizations for linear time-invariant systems to quasi-realizations of hidden Markov models. We will also show that balanced realizations can be used for model reduction.

First define matrices W_q and M_q , which are the analogue of the controllability and observability Gramians in system theory, as:

$$W_q := \sum_{y \in \mathbb{Y}^*} \Pi_q(y) e_q e_q^\top \Pi_q(y)^\top = \mathcal{C}_q \mathcal{C}_q^\top,$$

$$M_q := \sum_{y \in \mathbb{Y}^*} \Pi_q(y)^\top \pi_q^\top \pi_q \Pi_q(y) = \mathcal{O}_q^\top \mathcal{O}_q.$$

Obviously, $W_q = W_q^\top \geq 0$ and $M_q = M_q^\top \geq 0$. Moreover, if \mathcal{C}_q is surjective and \mathcal{O}_q is injective (as is the case for minimal quasi-realizations), then the strict inequality holds.

We assume that the infinite sums in the definitions above, are finite. It is a topic of our current research to check under which conditions on $\Pi_q(y), y \in \mathbb{Y}$ this assumption is fulfilled.

If the matrices W_q and M_q are finite, then it is easy to verify that they are solutions to the Lyapunov equations:

$$\sum_{y \in \mathbb{Y}} \Pi_q(y) W_q \Pi_q(y)^\top - W_q = -e_q e_q^\top, \quad (1)$$

$$\sum_{y \in \mathbb{Y}} \Pi_q(y)^\top M_q \Pi_q(y) - M_q = -\pi_q^\top \pi_q. \quad (2)$$

A realization is called *balanced* if the matrices W_q and M_q are diagonal and equal to each other. It can be shown that for every quasi-realization, there exists an equivalent balanced quasi-realization. We now show that the algorithm of Section 2 can be modified such that it gives immediately a balanced quasi-realization.

The sub-matrix M of *Step 1* of the algorithm is taken equal to the complete Hankel matrix $\mathfrak{H} \in \mathbb{R}^{\infty \times \infty}$. The matrices K and R of *Step 2* become $\mathfrak{H}_{1:\infty, 1}$ and $\mathfrak{H}_{1, 1:\infty}$ respectively. The decomposition of *Step 3* is performed using the singular value decomposition (SVD) of the Hankel matrix $\mathfrak{H} = U \Sigma V^\top = U \sqrt{\Sigma} \sqrt{\Sigma} V^\top$, with $\Sigma = \text{diag}(\sigma_1, \sigma_2, \dots, \sigma_{|\mathbb{X}_q|_{\min}})$. Then one can show that the realization of *Step 4* is balanced and is given by

$$\begin{aligned} \Pi_q^b(y) &= \sqrt{\Sigma^{-1}} U^\top \sigma_y \mathfrak{H} V \sqrt{\Sigma^{-1}} \quad \forall y \in \mathbb{Y}, \\ \pi_q^b &= \mathfrak{H}_{1,1:\infty} V \sqrt{\Sigma^{-1}}, \\ e_q^b &= \sqrt{\Sigma^{-1}} U^\top \mathfrak{H}_{1:\infty,1}. \end{aligned}$$

If the quasi-realization is in balanced form, a reduced model of order $|\mathbb{X}_q^r|$ can be obtained by *truncating* the model such that only the first $|\mathbb{X}_q^r|$ states are retained. We are presently working on obtaining error bounds for this balanced reduced quasi-realization.

5 Simulation example

We now apply the ideas of filtering and SVD-based approximate realization in a simulation. Suppose we are given two corresponding strings (of length 5000) $y^{(1)}$ and $z^{(1)}$ of an unknown hidden Markov model with two outputs, an observed output y and a to-be-estimated output z . We are also given another string $y^{(2)}$ of length 100 of the observed output y and the problem is to find an estimate of the corresponding string $z^{(2)}$ of the to-be estimated output.

The strings $y^{(1)}$ and $z^{(1)}$ were generated using an HMM $\mathcal{S} = (\mathbb{X}, \mathbb{Y}, \Pi, \pi, e)$

$$\begin{aligned} \Pi^{(a,0)} &= \begin{bmatrix} 0 & .09 & 0 & 0 \\ 0 & .09 & 0 & 0 \\ 0 & 0 & .01 & 0 \\ 0 & 0 & 0 & .01 \end{bmatrix}, \quad \Pi^{(a,1)} = \begin{bmatrix} 0 & .01 & 0 & 0 \\ 0 & .01 & 0 & .81 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & .81 \end{bmatrix}, \quad \Pi^{(b,0)} = \begin{bmatrix} .81 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0.9 \\ .81 & 0 & 0 & 0 \\ 0 & 0 & 0 & .09 \end{bmatrix}, \quad \Pi^{(b,1)} = \begin{bmatrix} .09 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ .09 & 0 & 0.09 & 0 \\ 0 & 0 & .09 & 0 \end{bmatrix}, \\ \pi &= [.45 \ .05 \ .05 \ .45], \end{aligned}$$

where Π is a mapping from $\mathbb{Y} \times \mathbb{Z}$ to $\mathbb{R}_+^{|\mathbb{X}| \times |\mathbb{X}|}$ with $|\mathbb{X}| = 4$, $\mathbb{Y} = \{a, b\}$ and $\mathbb{Z} = \{0, 1\}$. This model is unknown, but is given here to check the quality of the results.

It can be shown, using the method on page 26 of [3], that this fourth order true realization is minimal. However, the rank of the Hankel matrix is equal to 3, which means that a minimal quasi-realization has order 3. For that reason, our filtering algorithm will have an extra advantage. Not only, there is no need to compute a true instead of a quasi-realization, but furthermore, the minimal quasi-realization has a lower order than the minimal realization such that the filtering computations become less expensive.

We start with the modeling step. A hidden Markov with two output processes y and z with output alphabets \mathbb{Y} and \mathbb{Z} is equivalent with a hidden Markov model with one output process w with alphabet $\mathbb{W} := \mathbb{Y} \times \mathbb{Z}$, in this example, $\mathbb{Y} \times \mathbb{Z} = \{\alpha := a0, \beta := a1, \gamma := b0, \delta := b1\}$.

From the output string $w^{(1)}$ (which is the equivalent of the strings $y^{(1)}$ and $z^{(1)}$) of length 5000, the probabilities of strings up to length 4 are estimated. As expected, the (21×21) Hankel matrix associated with these estimated string probabilities has

full rank. However, there are 3 dominant singular values (the singular values ordered from high to low are: 1.6105; 0.5240; 0.0171; 0.0081; 0.0054; 0.0030; 0.0018; 0.0015; ...). We now use the algorithm of Section 4 to find an approximate quasi-realization $\mathcal{S}^r = (\mathbb{X}_q^r, \mathbb{Y}, \Pi_q^r, \pi_q^r, e_q^r)$ of order 3.

Table 1. String probabilities for strings of length 2 and length 6.

Sequence	Exact	Experimental	Approximate	Sequence	Exact	Experimental	Approximate
$\alpha\alpha$	0.0041	0.0052	0.0052	$\gamma\gamma$	0.3321	0.3225	0.3199
$\alpha\beta$	0.0369	0.0386	0.0386	$\gamma\delta$	0.0405	0.0382	0.0408
$\alpha\gamma$	0.0081	0.0072	0.0072	$\delta\alpha$	0.0045	0.0046	0.0046
$\beta\delta$	0.0009	0.0012	0.0012	$\delta\beta$	0.0004	0.0004	0.0004
$\beta\alpha$	0.0045	0.0028	0.0028	$\delta\gamma$	0.0729	0.0732	0.0758
$\beta\beta$	0.3322	0.3377	0.3377	$\delta\delta$	0.0121	0.0152	0.0126
$\beta\gamma$	0.0369	0.0362	0.0362	$\beta\beta\beta\beta\alpha$	0.0018	-	0.0012
$\gamma\delta$	0.0365	0.0388	0.0388	$\beta\beta\beta\beta\beta$	0.1430	-	0.1453
$\gamma\alpha$	0.0369	0.0396	0.0396	$\beta\beta\beta\beta\gamma$	0.0159	-	0.0161
$\gamma\beta$	0.0405	0.0386	0.0386	$\beta\beta\beta\beta\delta$	0.0159	-	0.0169

In the first part of Table 1, we show the exact, experimental and approximated string probabilities for strings of length 2. To check the performance of the approximation, the string probabilities of strings longer than 4 symbols, have to be examined. In the second part of Table 1, we show the exact and approximated string probabilities for a selection (due to space limitations) of strings of length 6. We conclude that the approximate quasi-realization algorithm performs quite well.

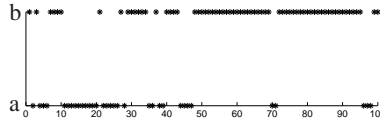


Fig. 1. True first output, $y^{(2)}$.

In the second step of our simulation, we are given a string $y^{(2)}$ and are asked to find an estimate for the corresponding string $z^{(2)}$. In Figure 1, we show the string $y^{(2)}$.

In Figure 2(a), we show the true second output string $z^{(2)}$ with '*', and the estimated probability of observing the symbol 0, based on the approximate quasi-realization, with '•'. One easily sees that, in general, the probability to observe a 0 is high, when the true output is equal to 0, and vice-versa, from which we conclude that the estimator works quite well. In Figure 2(b), we show the difference between the probability of observing the symbol 0 based on the approximate quasi-realization and the same probability based on an exact quasi-realization. We notice that the differences are minor. From these facts, we conclude that for the filtering problem, the approximate quasi-realization of order 3 performs well, and there is no need to calculate a quasi-realization of higher order or a true (nonnegative) realization.

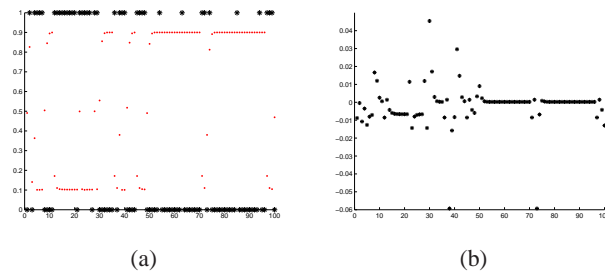


Fig. 2. (a) True second output, $z^{(2)}$ (••) and estimated probability of observing a 0 (••). (b) Error on probability of observing a 0.

In this simulation example, there is no need to calculate a true realization, which is more complicated than obtaining a quasi-realization. In addition, the order of a quasi-realization is smaller than the order of a true realization, which makes that the filtering computations become much less expensive.

6 Conclusions

In this paper, we considered HMMs with two outputs, an observed and a to-be-estimated output. We derived filter equations for the second output based on the past of the first output. It turned out that a quasi-realization suffices to obtain recursive filter equations. By combining an exact quasi-realization algorithm with an SVD-based approach, we proposed an approximate quasi-realization algorithm.

Acknowledgements

The SISTA program is supported by: Research Council KUL: GOA AMBioRICS, several PhD/postdoc and fellow grants; Flemish Government: FWO: PhD/postdoc grants, projects, G.0407.02 (support vector machines), G.0197.02 (power islands), G.0141.03 (Identification and cryptography), G.0491.03 (control for intensive care glycaemia), G.0120.03 (QIT), G.0452.04 (new quantum algorithms), G.0499.04 (Statistics), G.0211.05 (Nonlinear), research communities (ICCoS, ANMMM, MLDM); IWT: PhD Grants.GBOU (McKnow); Belgian Federal Science Policy Office: IUAP P5/22 ('Dynamical Systems and Control: Computation, Identification and Modeling', 2002-2006); PODO-II (CP/40: TMS and Sustainability); EU: FP5-Quprodis; ERNSI; Contract Research/agreements: ISMC/IPCOS, Data4s, TML, Elia, LMS, Mastercard.

References

1. B.D.O. Anderson. The realization problem for hidden markov models. *Mathematics of Control, Signals, and Systems*, 12(1):80–120, 1999.
2. E.J. Gilbert. The identifiability problem for functions of markov chains. *Annals of Mathematical Statistics*, 39:938–946, 1959.
3. A. Paz. *Introduction to probabilistic automata*. Academic Press, New York, 1971.
4. G. Picci. On the internal structure of finite-state stochastic processes. In R.R. Mohler and A. Roberti, editors, *Lecture notes in Economics and Mathematical Systems*, volume 162, pages 288–304. Springer-Verlag, Berlin, 1978.
5. M. Vidyasagar. A realization theory for hidden markov models: the complete realization problem. 2005.