Identification method for time-varying ARX models

Quentin Rentmeesters, P.-A. Absil, and Paul Van Dooren

Département d'ingénierie mathématique
Université catholique de Louvain
B-1348 Louvain-la-Neuve, Belgium
{Quentin.Rentmeesters,PA.Absil,Paul.Vandooren}@uclouvain.be

Summary. This paper presents a new approach to identify time-varying ARX models by imposing a penalty on the coefficient variation. Two different coefficient normalizations are compared and a method to solve the two corresponding optimization problems is proposed.

1 Introduction

Time-varying processes appear in many applications such as speech processing, time-varying behavior detection (fault detection or wear detection) or more generally when some parameters of a linear system vary over time. In this paper, we are interested in time-varying systems identification using an ARX model of order $N-1$:

$$\sum_{i=0}^{N-1} y(t-i)\alpha_i(t) = \sum_{i=0}^{N-1} u(t-i)\beta_i(t) \tag{1}$$

where y is the output of the time-varying system, u is the input and $\alpha_i(t)$ and $\beta_i(t)$ are the coefficients of the model at time t.

Several approaches have been adopted to deal with time-varying modeling problems. One of the most popular ones is to use an adaptive algorithm that computes iteratively the coefficients of the model; see, e.g., [1]. This approach works quite well under the assumption that the time variations are slow.

Another approach is to expand the coefficients of the model in a finite set of basis functions [2]. The problem then becomes time-invariant with respect to the parameters in the expansion and is hence reduced to a least squares problem. The two main issues which are encountered when this approach is applied to general time-varying systems, are how to choose a family of basis functions, and how to select finitely many significant ones.

Here, we consider a method which identifies the time-varying coefficients in a fixed time window. This method is not recursive and does not assume

strong hypotheses on the evolution of the coefficients. Moreover, at each time step, a value for the coefficients of the model is identified. Thus, it is not necessary to find a basis to expand the coefficients which is an important practical advantage. It will still be possible to choose a basis of functions to expand the coefficients after the identification to reduce the space complexity of the identified model. Our approach is based on a trade-off between the minimization of the prediction error and the minimization of the variation of the coefficients. The penalization of the variation of the coefficients enables the reduction of high frequency noises and the use of classical techniques to find the order of the model.

The paper is organized as follows. Section 2 introduces our approach and describes a method to solve efficiently the least squares problem that arises. Section 3 presents another normalization of the cost function introduced in section 2 that leads to an optimization problem on the Cartesian product of spheres. Numerical experiments and some ways to find the parameters of the method are presented in section 4.

2 Our approach

On the one hand, the coefficients must be allowed to vary sufficiently to deal with possibly large coefficient variations and to fit the data points. But, on the other hand, the variation of the coefficients must be penalized to reduce the influence of high frequency noises or outliers. To achieve this trade-off, the following cost function is considered:

$$\min_{X(0),\dots,X(T-1)} \sum_{t=1}^{T-1} \|X(t) - X(t-1)\|_2^2 + \mu \sum_{t=0}^{T-1} \|\phi^\top(t) X(t)\|_2^2, \qquad (2)$$

where T is the size of the time window where the identification is performed, $X(t) = \big[\alpha_0(t), \beta_0(t), \dots, \alpha_{N-1}(t), \beta_{N-1}(t)\big]^\top$ is the coefficient vector and $\phi(t) = \big[y(t), -u(t), \dots, y(t-N+1), -u(t-N+1)\big]^\top$ is the data vector. It is also possible to identify the model structure (1) where some of the coefficients are set to zero: it suffices to delete the coefficients in $X(t)$ and the corresponding inputs or outputs in $\phi(t)$.

The first term imposes that the coefficients do not vary too fast and the second term corresponds to the square of prediction error. The parameter $\mu > 0$ can be chosen to find a compromise between fitting the data and preventing the coefficients from varying too quickly.

This problem admits the trivial solution: $X(t) = 0$ for all t. Consequently, we must normalize the coefficient vector. Two kinds of normalizations are considered: fixing one coefficient at 1 for all t, and imposing $\|X(t)\| = 1$ for all t. The first one yields a least squares problem. The second one yields an optimization problem on the Cartesian product of spheres and is the subject of the next section.

The rest of this section explains how to solve the problem efficiently when the normalization: $\alpha_0(t) = 1$ for all t is chosen. In this case, the problem (2) can be rewritten as the following least squares problem:

$$\min_{X_2} \left\| \underbrace{\begin{bmatrix} I_{2N-1} & -I_{2N-1} & & & & & \\ & I_{2N-1} & -I_{2N-1} & & & & \\ & & \ddots & \ddots & & & \\ & & & \ddots & -I_{2N-1} & & \\ & & & & I_{2N-1} & -I_{2N-1} & \\ \sqrt{\mu}\phi_2^\top(0) & & & & & & \\ & \sqrt{\mu}\phi_2^\top(1) & & & & & \\ & & \ddots & & & & \\ & & & \ddots & & & \\ & & & & \sqrt{\mu}\phi_2^\top(T-2) & & \\ & & & & & \sqrt{\mu}\phi_2^\top(T-1) \end{bmatrix}}_{A_2} \underbrace{\begin{bmatrix} X_2(0) \\ \vdots \\ X_2(T-1) \end{bmatrix}}_{X_2} - \underbrace{\begin{bmatrix} 0 \\ \vdots \\ 0 \\ -\sqrt{\mu}y(0) \\ -\sqrt{\mu}y(1) \\ \vdots \\ -\sqrt{\mu}y(T-2) \\ -\sqrt{\mu}y(T-1) \end{bmatrix}}_{b} \right\|_2^2$$

where $X_2(t) = [\beta_0(t), \ldots, \alpha_{N-1}(t), \beta_{N-1}(t)]^\top$ and $\phi_2(t) = [-u(t), y(t-1), \ldots]^\top$. To preserve the structure, a method based on the normal equations (${A_2}^\top A_2 X_2 = {A_2}^\top b$) is proposed to solve the problem. The ${A_2}^\top A_2$ matrix is:

$$\underbrace{\begin{bmatrix} I & -I & & & & \\ -I & 2I & -I & & & \\ & \ddots & \ddots & \ddots & & \\ & & \ddots & \ddots & -I & \\ & & & \ddots & 2I & -I \\ & & & & -I & I \end{bmatrix}}_{M} + \mu \underbrace{\begin{bmatrix} \phi_2(0)\phi_2^\top(0) & & & \\ & \phi_2(1)\phi_2^\top(1) & & \\ & & \ddots & \\ & & & \ddots \\ & & & & \phi_2(T-1)\phi_2^\top(T-1) \end{bmatrix}}_{\Phi} \tag{3}$$

where I is the identity matrix of size $2N-1$.

The matrix ${A_2}^\top A_2$ is block tri-diagonal and is the sum of two positive semi-definite matrices M and Φ. Hence, ${A_2}^\top A_2$ is invertible if the kernel of M has no intersection with the kernel of Φ. The eigenvalues λ_k and the corresponding eigenspaces v_k of M are (see [3]):

$$v_k = \left[\cos((0+\tfrac{1}{2})\tfrac{k\pi}{T})I \cdots \cos((j+\tfrac{1}{2})\tfrac{k\pi}{T})I \cdots \cos(((T-1)+\tfrac{1}{2})\tfrac{k\pi}{T})I\right]$$

$$\lambda_k = 2 - 2\cos(\frac{k\pi}{T}) \qquad 0 \le k \le T-1$$

The eigenspace relative to $\lambda_0 = 0$ is: $v_0 = \left[I \ldots I\right]^\top$. Consequently, in order to get a unique solution, the following condition is required:

$$v_0^\top {A_2}^\top A_2 v_0 = \mu v_0^\top \Phi v_0 = \mu \sum_{i=0}^{T-1} \phi_2(i)\phi_2(i)^\top \succ 0.$$

This is true if $\lambda_{\min}\left(\sum_{i=0}^{T-1}\phi_2(i)\phi_2(i)^\top\right) > 0$ which means that the data vector $\phi_2(t)$ must span a space of dimension $2N-1$ on the whole time horizon of size T. This condition will be easily satisfied if the input is sufficiently exciting and if the order of the model is not overestimated. Notice that this tells no information about the reliability of the identified coefficients. To be able to recover the true coefficients of a model, the data should be unperturbed and as exciting as possible. If $\lambda_{\min}\left(\sum_{i=k}^{k+2N-2}\phi_2(i)\phi_2(i)^\top\right) > 0 \quad \forall k$, the data are very informative, and this will provide a more reliable approximation of the coefficients.

The system of normal equations can be efficiently solved by performing a block tri-diagonal LU factorization of the ${A_2}^\top A_2$ matrix (3), see [4, 4.5] for more details. This decomposition has a complexity of $\mathrm{O}((T-1)(2N-1)^3)$ operations which is linear in T.

Using the same technique, it is also possible to normalize another coefficient than α_0 and to take into account already known coefficients by fixing them at their value. Unfortunately, the solution of the problem will depend on the coefficient which is normalized, that is why another normalization is proposed in the next section.

3 Normalization of the coefficient vector

In this section, we explain why it can be interesting to normalize the coefficient vector, i.e., fixing $\|X(t)\| = 1$ for all t and we describe the method used to solve the corresponding optimization problem.

The main idea behind this normalization is the following. The ARX relation (1) can be rewritten as:

$$X(t)^\top \phi(t) = 0$$

and is unchanged if it is multiplied by a scalar $\gamma(t) \neq 0$ which means that $\gamma(t)X(t)$ corresponds to the same ARX model as $X(t)$. Consequently, an ARX model at time t is not represented by a particular coefficient vector but by a direction in $\mathbb{R}^{2N}$. Hence, a good notion of distance between two ARX models is the relative angle. In fact, this notion of distance does not depend on the particular choice of vector in $\mathbb{R}^{2N}$ used to represent an ARX model. When $\|X(t)\| = 1$ for all t, the first term of (2) becomes:

$$\sum_{t=1}^{T-1} 4\sin^2\left(\frac{\angle X(t)X(t-1)}{2}\right)$$

and only depends on the angle $\angle X(t)X(t-1)$ between two coefficient vectors representing two ARX models at consecutive time steps.

This is also a more neutral normalization because the cost on the variation of the coefficients is uniformly distributed over all coefficients, as opposed to the normalization of the α_0 coefficient. In fact, when the α_0 coefficient is normalized, the distance between two ARX models represented by $\|\frac{X(t)}{\alpha_0(t)} - \frac{X(t-1)}{\alpha_0(t-1)}\|_2^2$ will be larger if the model at time t is well represented by a model whose α_0 coefficient gets close to 0 and lower if the model at time t is well represented by a model whose α_0 coefficient is large. This is shown in the following example. At time $t = 150$, the α_0 coefficient of the following system:

$$\begin{aligned}
\alpha_0(t) &= 0.5 + 0.45 \sin\left(\tfrac{t2\pi}{200}\right) \quad 1 \leq t \leq 200 \\
\beta_0(t) &= 5 \\
\alpha_1(t) &= 0.01 \\
\beta_1(t) &= -4
\end{aligned}$$

gets close to zero. Fig. 1. shows the identified β_0 coefficient using the two normalizations. If the coefficient α_0 is normalized, the true coefficient is not recovered in the neighborhood of $t = 150$ because a coefficient variation is highly penalized in this neighborhood. This is avoided when the coefficient vector is normalized since the cost on the variation of the coefficients depends only on the angle.

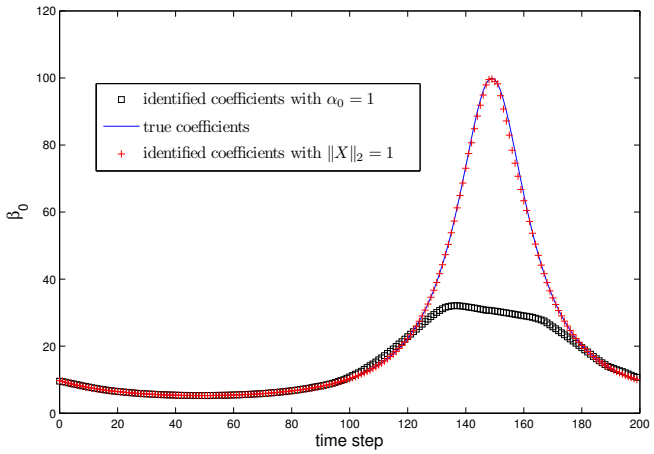

Fig. 1. true and identified coefficient β_0 when $\|X(t)\|_2 = 1$ for all t

With this constraint, the optimization problem (2) is no longer a least squares problem and an optimization technique on manifolds is proposed. We will only describe the main points of this method. For more details, see [5].

By introducing the following notation:

$$Y = \begin{bmatrix} X(0) \\ \vdots \\ X(T-1) \end{bmatrix} = \begin{bmatrix} X_0 \\ \vdots \\ X_{T-1} \end{bmatrix} \in \mathbb{R}^{2NT},$$

the constraint $\|X(t)\| = 1$ for all t can be also rewritten as: $Y \in (\mathbb{S}^{2N-1})^T$ where $(\mathbb{S}^{2N-1})^T$ stands for the Cartesian product of T unit spheres in $\mathbb{R}^{2N}$:

$$(\mathbb{S}^{2N-1})^T = \underbrace{S^{2N-1} \times \cdots \times S^{2N-1}}_{T} \subset \mathbb{R}^{2NT}$$

where $S^{2N-1} = \{x \in \mathbb{R}^{2N} | x^\top x = 1\}$ is the unit sphere in $\mathbb{R}^{2N}$. This is a submanifold of $\mathbb{R}^{2NT}$ and its tangent space at Y is:

$$T_Y(\mathbb{S}^{2N-1})^T = \{Z = \begin{bmatrix} Z_0 \ \dots \ Z_{T-1} \end{bmatrix}^\top \in \mathbb{R}^{2NT} | X_i^\top Z_i = 0 \quad 0 \leq i \leq T-1\}.$$

The orthogonal projection on this tangent space at the point Y is given by:

$$P_Y(Z) = \begin{bmatrix} P_{X_0}(Z_0) \\ \vdots \\ P_{X_{T-1}}(Z_{T-1}) \end{bmatrix} = \begin{bmatrix} (I_{2N} - X_0 X_0^\top) Z_0 \\ \vdots \\ (I_{2N} - X_{T-1} X_{T-1}^\top) Z_{T-1} \end{bmatrix}.$$

Then, the problem (2) becomes the following optimization problem on $(\mathbb{S}^{2N-1})^T$:

$$\boxed{\begin{array}{c} \min\limits_{Y} \quad \overline{f} : \mathbb{R}^{2NT} \longrightarrow \mathbb{R}, Y \longrightarrow Y^\top A^\top A Y \\ \text{s.t.} \quad Y \in (\mathbb{S}^{2N-1})^T \end{array}}$$

where the $A^\top A$ matrix is given by:

$$\begin{bmatrix} I + \mu\Phi(0) & -I & & & & & \\ -I & 2I + \mu\Phi(1) & -I & & & & \\ & \ddots & \ddots & \ddots & & & \\ & & \ddots & \ddots & -I & & \\ & & & \ddots & 2I + \mu\Phi(T-2) & -I & \\ & & & & -I & I + \mu\Phi(T-1) \end{bmatrix} \tag{4}$$

with $\Phi(t) = \phi(t)\phi^\top(t)$ and I is the identity matrix of size $2N$. The restriction of $\overline{f}$ to $(\mathbb{S}^{2N-1})^T$ is denoted by f.

A Newton method on the Cartesian product of spheres has been chosen to solve this problem because our numerical experiments have shown that the solution of the least squares problem (when α_0 is normalized) belongs to the attraction basin of the Newton method. So, the solution of the least squares problem can be used as a starting value for the local optimization problem on the Cartesian product of spheres. The Newton equation is given by:

$$\nabla_Z \text{grad}\, f = -\text{grad}\, f(Y), \quad Z \in T_Y(\mathbb{S}^{2N-1})^T \tag{5}$$

where grad $f(Y)$ represents the gradient at the current iterate Y and $\nabla_Z \text{grad}\, f$ stands for the Riemannian covariant derivative of the vector field grad $f(Y)$ in the direction Z where Z will be the next Newton direction.

To implement this method, an expression for the gradient and for the Riemannian connection ∇ is required. The gradient with respect to the induced metric is the unique element grad $f(Y)$ of $T_Y(\mathbb{S}^{2N-1})^T$ which satisfies:

$$\text{grad}\, f(X)^\top Z = DF(Y)[Z] \quad \forall Z \in T_Y(\mathbb{S}^{2N-1})^T$$

where $DF(Y)[Z]$ stands for the differential at Y in the direction Z. In our case, this gives:

$$\text{grad}\, f(Y) = P_Y(2A^\top AY).$$

Since $(\mathbb{S}^{2N-1})^T$ is an $\mathbb{R}^n$ submanifold of the Euclidean space $\mathbb{R}^{2NT}$, the $\mathbb{R}^n$ connection is equivalent to the classical directional derivative in $\mathbb{R}^{2NT}$ followed by a projection on the tangent space at Y: $\nabla_Z \text{grad}\, f = P_Y(D\text{grad}\, f(Y)[Z])$. Since

$$(D\text{grad}\, f(Y)[Z])_i = 2((-X_i Z_i^\top - Z_i X_i^\top)B_i Y + P_{X_i}(B_i Z)),$$

the Newton equation (5) becomes:

$$2\begin{bmatrix} P_{X_0}(B_0 Z) - Z_0 X_0^\top B_0 Y \\ \vdots \\ P_{X_{T-1}}(B_{T-1} Z) - Z_{T-1} X_{T-1}^\top B_{T-1} Y \end{bmatrix} = -\text{grad}\, f(Y) \tag{6}$$

$$Z \in T_Y(\mathbb{S}^{2N-1})^T \tag{7}$$

where B_i is the block matrix composed of the rows $i2N+1$ up to $(i+1)2N$ and all the columns of $A^\top A$ in (4). By introducing the following change of variables,

$$\omega_i = {X_i^\perp}^\top Z_i \text{ where } [X_i|X_i^\perp]^\top [X_i|X_i^\perp] = I_{2N}$$

the condition (7) is trivially satisfied and (6) becomes:

$$\begin{gathered} K_0\omega_0 - {X_0^\perp}^\top X_1^\perp \omega_1 = -{X_0^\perp}^\top B_0 Y \\ -{X_i^\perp}^\top X_{i-1}^\perp \omega_{i-1} + K_i\omega_i - {X_i^\perp}^\top X_{i+1}^\perp \omega_{i+1} = -{X_i^\perp}^\top B_i Y \text{ for } 1 \le i \le T-2 \\ -{X_{T-1}^\perp}^\top X_{T-2}^\perp \omega_{T-2} + K_{T-1}\omega_{T-1} = -{X_{T-1}^\perp}^\top B_{T-1} Y \end{gathered}$$

where $K_i = {X_i^\perp}^\top \mu\Phi(i) X_i^\perp - I X_i^\top B_i Y$. This system is block tri-diagonal and can be easily solved using a block LU factorization which requires $O((T-1)(2N-1)^3)$ operations. Consequently from a computational complexity point of view, one iteration of this Newton method is equivalent to the least squares method presented in the previous section. Once the Newton step Z has been computed, the following retraction:

$$R_Y(Z) = \begin{bmatrix} \frac{X_0+Z_0}{\|X_0+Z_0\|} \\ \vdots \\ \frac{X_{T-1}+Z_{T-1}}{\|X_{T-1}+Z_{T-1}\|} \end{bmatrix}$$

can be used to compute the update $Y_+ = R_Y(Z)$.

4 Choice of μ and the order

In this section, some numerical experiments and methods to select or gain some insight in the μ parameter value and the order of the system are presented. Let us consider the system defined by the following coefficient vector:

$$X(t) = \begin{bmatrix} \alpha_0(t) \\ \beta_0(t) \\ \alpha_1(t) \\ \beta_1(t) \\ \alpha_2(t) \\ \beta_2(t) \end{bmatrix} = \begin{bmatrix} 1 \\ 1 - 0.2e^{-\left(\frac{t-100}{50}\right)^2} \\ -0.8 \\ -0.2 \\ 0.6 \\ -0.6 \end{bmatrix}.$$

This system was simulated with a white noise of unitary variance as input. The output was perturbed in the following way: $y(t) \leftarrow y(t) + \Delta|y(t)|U(t)$ where $U(t)$ is a random variable distributed uniformly on $[-1, 1]$. Fig. 2. shows the error on the coefficients in function of μ for different levels of perturbation. For an unperturbed model ($\Delta = 0$), the error on the coefficients is smaller for a large value of μ because the bias introduced by the first term of our cost function is reduced. For a perturbed system, it is not optimal to trust too much the data, and there exists an optimal value of μ that minimizes the error on the coefficients. To get an insight of this optimal value of μ in practice, we can look at the identified coefficient β_0 shown in Fig. 3. For a small value of μ, we get an almost constant coefficient and for a large value of μ we identify a coefficient that oscillates around the true coefficient. This means that we are identifying the noise. So it is possible to get an idea of the best value of μ that makes a desired trade-off between the slow coefficient variation or equivalently the risk of bias and the rejection of the perturbations.

The notion of order for a time invariant system somehow represents the complexity of the model. If this complexity is increased, the model will better fit the data. So, a common criterion to find the order of a time-invariant system consists in measuring the fitting error (the prediction error in our case) and selecting the order that corresponds to a drop on the fit level. This idea does not directly extend to time-varying models. In fact, even with a time-varying model of order 0, it is easy to make the fitting error go to 0. But by imposing a cost on the variation of the coefficients, the same idea can be applied as shown in the following experiment. A time-varing ARX system of order 4 was identified using different models (different values of the order) and different

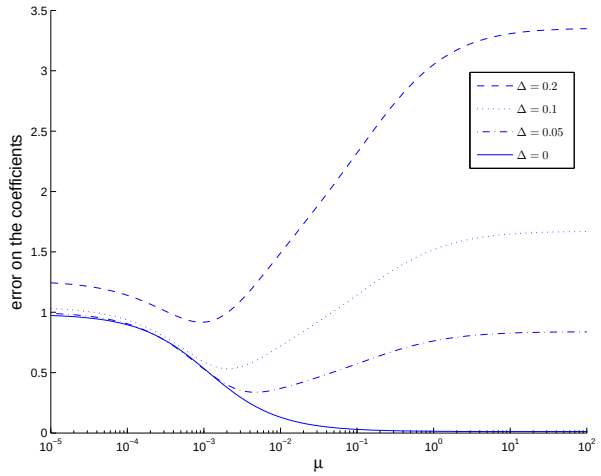

Fig. 2. difference between the true X_2 and the identified coefficients $\tilde{X}_2$: $\|X_2 - \tilde{X}_2\|_2$ in function of μ for different levels of perturbation Δ

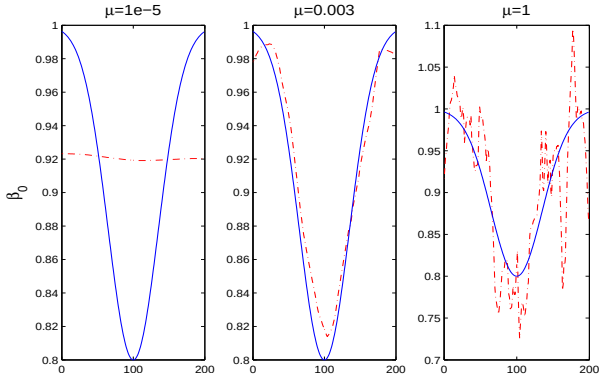

Fig. 3. identified (.-) and true (-) coefficients β_0 for different values of μ when $\Delta = 0.1$

values of μ, see Fig. 4. When we go from a model of order 3 to a model of order 4, the error drops and remains rather constant if the order is further increased. This drop indicates that the model order is probably 4 and it is interesting to notice that this conclusion does not depend on the value of μ.

5 Conclusions

We have presented a method to identify a time-varying ARX model by penalizing the variation of the coefficients. By doing so, we can choose the order

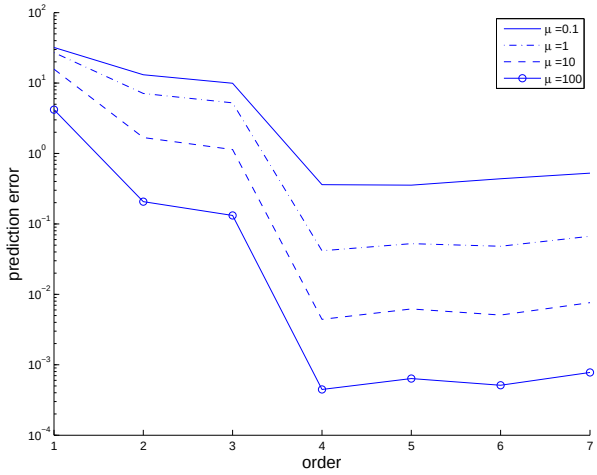

Fig. 4. prediction error ($\sum_{t=0}^{T-1} \|\phi^\top(t)\tilde{X}(t)\|_2$ where $\tilde{X}(t)$ stands for the identified coefficient vector) as a function of the order, for different values of μ

using classical techniques and the influence of the perturbations can be reduced. A more neutral normalization of the coefficient vector has also been proposed. This normalization leads to better results on models whose α_0 coefficient gets close to 0. In later work, we will extend these methods to MIMO systems. When the coefficient matrix is normalized, this yields an optimization problem on the Cartesian product of Grassmann manifolds.

References

1. L. Guo and L. Ljung, Performance analysis of general tracking algorithms, IEEE Trans. Automat. Control, volume 40, pages 1388–1402, 1995.
2. H.L. Wei and S.A. Billings, Identification of time-varying systems using multiresolution wavelet models, International Journal of Systems Science, 2002.
3. G. Strang, The discrete cosine transform, SIAM Review, volume 41, pages 135–147 (electronic), 1999.
4. G.H. Golub, and C.F. Van Loan, Matrix Computations second edition, Johns Hopkins University Press, 1989.
5. P.-A. Absil, R. Mahony and R. Sepulchre, Optimization algorithms on matrix manifolds, Princeton University Press, 2008.

Acknowledgement. This paper presents research results of the Belgian Network DYSCO (Dynamical Systems, Control, and Optimization), funded by the Interuniversity Attraction Poles Programme, initiated by the Belgian State, Science Policy Office. The scientific responsibility rests with its author(s). This work was also supported by "Communauté française de Belgique - Actions de Recherche Concertées". The authors are grateful to Rik Pintelon and John Lataire for fruitful discussions.