PERGAMON

Automatica 37 (2001) 2057–2067

automatica

www.elsevier.com/locate/automatica

Brief Paper

Misfit versus latency[☆]

Philippe Lemmerling*, Bart De Moor

Department of Electrical Engineering, ESAT-SISTA, Katholieke Universiteit Leuven, Kasteelpark Arenberg 10, B-3001 Leuven-Heverlee, Belgium

Received 30 August 2000; revised 6 March 2001; received in final form 28 June 2001

Abstract

In this paper we present a framework that combines some ideas of the behavioral modeling approach and the prediction error modeling approach. It is shown that the proposed model selection procedure can be rephrased as an optimization problem that only depends on the model parameters. Experiments illustrate the potential of the so-called misfit versus latency framework.

Keywords: Prediction error methods; Behavioral modeling; Dynamic linear time-invariant systems

1. Introduction

Mathematical models are the heart of many applications in areas such as control system design, prediction, simulation, fault detection, etc. In this paper a new framework for modeling dynamic linear time-invariant (LTI) systems is developed. It combines some ideas of the prediction error modeling (PEM) approach (Ljung, 1987; Söderström & Stoica, 1989) and the behavioral modeling approach (Willems, 1986; Roorda, 1995; Roorda & Heij, 1995; De Moor & Roorda, 1994; De Moor, 1993, 1994, 1997). Both approaches—the PEM approach and the behavioral modeling approach—aim to fit a model of a proposed model set to observed data. The PEM method minimizes a cost function involving the so-called prediction errors, which are computed using the model and the observed data. These prediction errors can be thought of as "unobserved" signals, with certain pre-assumed properties, such as for instance a typical stationary zero mean white Gaussian noise assumption. Such signals will be labeled latent. Summarizing we can say that in the PEM approach the observed data can be explained by a model belonging to the proposed model set owing to the introduction of latent variables.

The behavioral modeling framework approaches the modeling problem from a set theoretic point of view: the set of all conceivable input/output time series compatible with the model, is called the behavior of the system. A model is then found by selecting—according to some user-defined distance measure—the time series of the behavior closest to the observed data. The latter distance is termed "misfit". Summarizing, we can say that in the behavioral modeling approach the observed data are explained by a model of the proposed model set, through the modification of the observed data in such a way that they become compatible with a model of the model set. The latter modification will be referred to as "misfit". In this paper, we develop the idea of obtaining mathematical models that "explain" observed data, by modifying the data ("misfit") or introducing new "unobserved" additional data ("latency") or by applying a combination of misfit and latency (De Moor & Lemmerling, 1998). The difference between "misfit" and "latency" can also be explained in a more pragmatic way. "Misfit" accounts for the measurement errors on the data and "latency" accounts for some unobserved input. A priori, there is no reason to assume that the measurement noise and the unobserved inputs have similar properties and therefore it makes sense to combine both while assigning a different weight to each of them.

The paper is structured as follows: the next section introduces the idea of misfit versus latency for static LTI models (linear regression models). Section 3 extends this

[☆]This paper was presented at the SYSID2000 meeting. This paper was recommended for publication in revised form by associate Editor John Schoukens under the direction of editor Torsten Söderström.

* Corresponding author. Tel.: + 32-(0)16-321709; fax: + 32-(0)16-321970.

E-mail address: philippe.lemmerling@esat.kuleuven.ac.be (P. Lemmerling).

0005-1098/01/$ - see front matter
PII: S0005-1098(01)00180-7

idea to a general set of dynamic LTI models. We motivate the use of this framework in system identification and rewrite the associated model selection procedure in a matrix framework. By using Lagrange multipliers the new framework can be recast into an optimization framework with only the model parameters as variables. Not only does this allow us to perform some experiments (see Section 5), it also provides additional insight into the model selection procedure for the new framework. Furthermore, we discuss the importance of using the correct constraints that have to be imposed on the model parameters. Section 4 derives some properties of the new framework. The potential of the misfit versus latency framework is illustrated with some simulation examples in Section 5.

In the remainder of the paper, we will adopt a Matlab like notation for vectors and matrices:

- $A(i, j)$: the entry in the jth column of the ith row of A.
- $A(i,:)$: the ith row of A.
- $A(:, j)$: the jth column of A.
- $A(p:q, r:s)$: the $(q-p+1)\times(s-r+1)$ submatrix of A containing the entries that belong to rows p till q *and* to columns r till s.
- $b(i)$: the entry on the ith row of column vector b.
- $b(p:q)$: the $(q-p+1)\times 1$ subvector of b containing the entries of row p till row q.
- $b(q:-1:p)$: this vector is equal to the previous one but with the elements in reversed order.

2. Static case

In this section we address static LTI models of the form

$$Ax = b \tag{1}$$

with $A\in\mathbb{R}^{m\times n}$, $m\geqslant n$, $b\in\mathbb{R}^{m\times 1}$ the data and $x\in\mathbb{R}^{n\times 1}$ the parameter vector. In general, Eq. (1) is not satisfied for measured data as it implies the rank deficiency of the matrix $[A\ \ b]\in\mathbb{R}^{m\times(n+1)}$. In those cases, the rank of $[A\ \ b]$ equals $n+1$ instead of n and therefore we will have to determine *approximate* models. We will now describe three different approximate models and associated model selection procedures.

Case 1: The first method consists of introducing a so called vector of "latent variables" $e\in\mathbb{R}^{m\times 1}$, such that $Ax = b + e$, and the appropriate model is selected by the following optimization problem:

$$\min_{e,x} \|e\|_2^2 \quad \text{such that } Ax = b + e, \tag{2}$$

where $\|.\|_2$ indicates the vector 2-norm. The latter approach is called the "latency" approach and corresponds to the well-known LS solution of an overdetermined system of equations. As can be seen from (2), the inconsistency of the equations (revealed by $\text{rank}([A\quad b]) = \text{rank}(A)+1$) is remedied by the introduction of a new vector of unknowns e. The minimization of $\|e\|_2$ can heuristically be explained by the desire to find the "minimum energy" e so that $Ax = b + e$. The latent variable can be thought of as an unobserved input.

Case 2: A second method does not introduce new variables, but it assumes a misfit $[\Delta A\quad \Delta b]$ between the observed data $[A\quad b]$ and the true data, such that $(A+\Delta A)x = b + \Delta b$. The model selection procedure can be formulated as follows:

$$\min_{\Delta A,\Delta b,x} \|[\Delta A\quad \Delta b]\|_{\mathrm{F}}^2, \quad \text{such that } (A+\Delta A)x = b+\Delta b, \tag{3}$$

where $\|.\|_{\mathrm{F}}$ stands for the Frobenius norm. Formulation (3) obviously corresponds to the classical errors-in-variables (EIV) approach, which solves the overdetermined system of equations $Ax\approx b$ in a total least-squares (TLS) sense. It is well known (Golub & Van Loan, 1996; Van Huffel & Vandewalle, 1991) that the solution to (3) follows from the Singular Value Decomposition (SVD) of $[A\quad b]$. The correction $[\Delta A\quad \Delta b]$ is a rank one matrix obtained from the SVD of $[A\quad b]$. Further on, this type of approach will be referred to as the misfit approach. The misfit can be thought of as measurement error on the data.

Case 3: The third way is to combine both the latency and the misfit approach, so that we obtain the following general model: $(A+\Delta A)x = b+\Delta b + e$. To select the model x in this case, we propose to solve the following optimization problem:

$$\min_{\Delta A,\Delta b,e,x} \alpha\|[\Delta A\quad \Delta b]\|_{\mathrm{F}}^2 + \beta\|e\|_2^2,$$

$$\text{such that } (A+\Delta A)x = b+\Delta b+e, \tag{4}$$

with α and β user-specified positive scalar weights that are inversely proportional to the importance assigned to the misfit respectively the latency contributions. The introduction of weights is a logical consequence if one starts from the assumption that the properties of the unobserved input (latency) and the measurement errors (misfit) are different. Use of the method of Lagrange multipliers shows that the solution of the latter optimization problem is the solution of a weighted SVD problem: $x = -V(1:n, n+1)/V(n+1, n+1)$ with $[A\quad b]W^{-1/2} = U\Sigma Z^{\mathrm{T}}$, the SVD of $[A\quad b]W^{-1/2}$, $V = W^{-1/2}Z$,

$$W = \begin{bmatrix} \frac{1}{\alpha} I_q & 0 \\ 0 & \frac{1}{\alpha}+\frac{1}{\beta} \end{bmatrix}$$

under the assumption that $V(n+1, n+1)\neq 0$. We see that the approach (4) is a generalization of the approaches described in (2) and (3). By taking e.g. $\alpha = \infty$

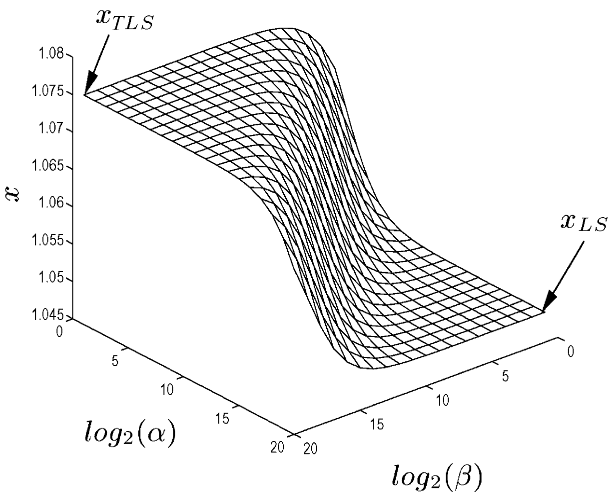

Fig. 1. This figure shows the solution x of (4) as a function of $\log_2(\alpha)$ and $\log_2(\beta)$, applied to example (5). The higher α, the less "misfit" we allow; the higher β, the less latency we allow.

and $\beta = 1$ in (4), we obtain (2). In Fig. 1, we show the solution x of problem (4) as a function of the weights α and β for the following example:

$$A = [3\ 4\ 4.3\ 3.4\ 2\ 1.09\ \ -3\ \ -2.54\ \ -2.09\ 0.89]^{\mathrm{T}} \quad (5)$$

and

$$b = [3.57\ 4.80\ 4.33\ 3.93\ 2.50\ 2.05\ \ -2.25\ \ -1.99 -\ 1.20\ 1.52]^{\mathrm{T}}.$$

It is clear that by changing the values of the weights, we can go in a continuous way from the LS solution ($x_{LS} = 1.04909$, $\alpha = \infty$, $\beta = 1$) to the TLS solution ($x_{TLS} = 1.0757$, $\alpha = 1$, $\beta = \infty$).

Finally, from a statistical point of view, it is obvious that these three approaches correspond to the maximum likelihood (ML) solution under different noise assumptions, namely independently identically distributed (i.i.d.) Gaussian noise on b (case 1), i.i.d. Gaussian noise on $[A \quad b]$ (case 2), i.i.d. Gaussian noise with standard deviation σ_1 on $[A \quad b]$ and i.i.d. Gaussian latency contribution e with standard deviation σ_2, where $\alpha/\beta = (\sigma_2/\sigma_1)^2$.

3. Dynamic case

In a way similar to the static case, we can create a new framework for dynamic LTI systems by combining the dynamic "latency" approach (i.e. the prediction error approach containing models such as AR, ARX, ARMA, etc. in which "white noise residuals" are introduced to explain the observed input/output data) and the dynamic "misfit" approach (i.e. the behavioral approach containing models such as dynamic TLS or EIV, etc.). The

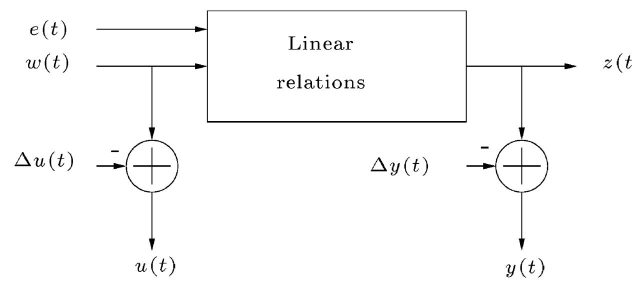

Fig. 2. This figure illustrates the new model set of LTI dynamical systems. The observed data are $u(t)$ and $y(t)$. They are assumed to be corrupted versions of exact data, such that $w(t) = u(t) + \Delta u(t)$ and $z(t) = y(t) + \Delta y(t)$. There are also "unobserved" inputs $e(t)$, that "help explain" the data $z(t)$.

motivation for this framework is shown in Fig. 2. There is no reason why the presence of a latency input $e(t)$ would exclude a noise misfit of the measurements at the input ($\Delta u(t)$) and output ($\Delta y(t)$) of the system or vice versa. Since the statistical properties of latency inputs and noise misfit can be very different, it makes sense to make a distinction in the problem formulation and the consecutive model selection procedure (where using the weights α, β and γ in (7), we can vary the importance of the latency terms versus the misfit terms). Models in this framework are of the form

$$A(q)z(t) = B(q)w(t) + C(q)e(t), \quad (6)$$

with $z(t) = y(t) + \Delta y(t)$ and $w(t) = u(t) + \Delta u(t)$, where $\Delta y(t)$ and $\Delta u(t)$ represent the so-called misfit of the observed output $y(t)$ and the observed input $u(t)$. Further, $e(t)$ is the so-called latent variable, which can be interpreted as an "unobserved" input. In the remainder of the paper, it is assumed that $\Delta u(t)$, $\Delta y(t)$ and $e(t)$ are independent for all time instants t. Furthermore it is assumed that $\Delta u(t)$, $t = 1, \ldots, N$, are zero mean independently and identically distributed random variables with variance $\sigma^2_{\Delta u}$ (where N stands for the number of data points). A similar assumption is made for $\Delta y(t)$ and $e(t)$ but with variances of respectively $\sigma^2_{\Delta y}$ and σ^2_e. $A(q)$, $B(q)$ and $C(q)$ are polynomials of appropriate degree in the delay operator q^{-1}, with $A(q) = a_0 + a_1 q^{-1} + \cdots + a_{n_a} q^{-n_a}$, $B(q) = b_0 + b_1 q^{-1} + b_2 q^{-2} + \cdots + b_{n_b} q^{-n_b}$ and $C(q) = c_0 + c_1 q^{-1} + \cdots + c_{n_c} q^{-n_c}$, where we assume that n_a, n_b and n_c are known. As model selection procedure, i.e. to select the best model from this new model set (6), we propose to minimize the following cost function:

$$J = \alpha J_{\text{output}} + \beta J_{\text{input}} + \gamma J_{\text{latency}} \quad (7)$$

with $J_{\text{output}} = \sum_{t=t_1}^{N} (\Delta y(t))^2$, $J_{\text{input}} = \sum_{t=t_2}^{N} (\Delta u(t))^2$, $J_{\text{latency}} = \sum_{t=t_3}^{N} e(t)^2$, N being the number of data points, and α, β, γ real positive scalars, specified by the user; let $M = \max(n_a, n_b, n_c) + 1$ and $m = N - M + 1$, then $t_1 = M - n_a$, $t_2 = M - n_b$ and $t_3 = M - n_c$.

Table 1
This table shows the different LTI dynamic systems of our general framework. Depending on the triplet (α, β, γ) we obtain 1 of the 27 model sets (note that when e.g. $\beta = *$, this means that there is no input term present in (7), nor in (6)). Some of them correspond to existing model sets, others are new. The last column indicates whether we have a misfit-like model (M), a latency-like model (L) or a model that applies to exact data (E) only

No	(α, β, γ)	Case	Data misfit	Model	M/L
1	$(*,*,*)$	No meaning			
2	$(*,*,1)$	No meaning			
3	$(*,*,\infty)$	See Case 1			
4	$(*,1,*)$	Noisy input realization	$U \to W$	$Wb = 0$	M
5	$(*,1,1)$	ARMA with noisy inputs	$U \to W$	$Wb + Ec = 0$	M + L
6	$(*,1,\infty)$	See Case 4			
7	$(*,\infty,*)$	Input = impulse response		$Ub = 0$	E
8	$(*,\infty,1)$	ARMA for inputs		$Ub + Ec = 0$	L
9	$(*,\infty,\infty)$	See Case 7			
10	$(1,*,*)$	Noisy output realization	$Y \to Z$	$Za = 0$	M
11	$(1,*,1)$	ARMA with noisy outputs	$Y \to Z$	$Za + Ec = 0$	M + L
12	$(1,*,\infty)$	See Case 10			
13	$(1,1,*)$	Dynamic errors-in-variables	$Y \to Z$ $U \to W$	$Za + Wb = 0$	M
14	$(1,1,1)$	ARMAX with noisy in/outputs	$Y \to Z$ $U \to W$	$Za + Wb + Ec = 0$	M + L
15	$(1,1,\infty)$	See Case 13			
16	$(1,\infty,*)$	Output error	$Y \to Z$	$Za + Ub = 0$	M
17	$(1,\infty,1)$	ARMAX with noisy output	$Y \to Z$	$Za + Ub + Ec = 0$	M + L
18	$(1,\infty,\infty)$	See Case 16			
19	$(\infty,*,*)$	Output = impulse response		$Ya = 0$	E
20	$(\infty,*,1)$	ARMA		$Ya + Ec = 0$	L
21	$(\infty,*,\infty)$	See Case 19			
22	$(\infty,1,*)$	Input error	$U \to W$	$Ya + Wb = 0$	M
23	$(\infty,1,1)$	ARMAX with noisy input	$U \to W$	$Ya + Wb + Ec = 0$	M + L
24	$(\infty,1,\infty)$	See Case 22			
25	$(\infty,\infty,*)$	Linear System		$Ya + Ub = 0$	E
26	$(\infty,\infty,1)$	ARMAX		$Ya + Ub + Ec = 0$	L
27	(∞,∞,∞)	See Case 25			

It is clear from the e.g., that a classical ARMAX model fits into this framework by selecting $\alpha \to \infty$, $\beta \to \infty$ and $\gamma = 1$. There are 27 different cases that can be obtained by the particular choices of α, β and γ (they can be finite, infinite, or their corresponding term can be omitted in model (6)). Table 1 shows the different possibilities. We see that the framework covers many existing linear dynamic modeling techniques, but in addition it proposes several new models (e.g. case 23 in Table 1). We now discuss two cases, in order to clarify the table:

- *Case* 17 ($\alpha = 1$, $\beta = \infty$ and $\gamma = 1$): to interpret the values of the weightings α, β and γ, we have to look at the model equations (6) and the cost function of the model selection procedure (7). Setting $\alpha = 1$ we see from (7) that we allow a misfit term $\Delta y(t)$ on the output. The fact that $\beta = \infty$ implies that J_{input} in (7) has to be zero and thus there can be no misfit on the input. Finally, $\gamma = 1$ means that the cost function (7) contains latency variables. As a result the model equation (6) becomes

  $$A(q)z(t) = B(q)u(t) + C(q)e(t),$$

  since there is misfit on the output (this explains why $z(t)$ is used instead of the measured $y(t)$, see Fig. 2), there is no misfit on the input (in that case $u(t) = w(t)$, so we use the measured quantity $u(t)$) and a latency variable is included.
- *Case* 11 ($\alpha = 1$, $\beta = *$ and $\gamma = 1$): again using the cost function (7), we see that there is misfit on the output ($\alpha = 1$), there is no input ($\beta = *$; this "$*$" indicates that the input term in (7) and (6) is simply omitted) and a latency term is included. Obviously this leads to the following model equation: $A(q)z(t) = C(q)e(t)$. Now it should also be clear why this model is called ARMA with noisy output: the only difference with the classical ARMA model $A(q)y(t) = C(q)e(t)$ is the fact that we take a misfit on the output ($y(t) \to z(t)$) into account.

We can write Eq. (6) for $t = M, \ldots, N$ in the following matrix format: $Za - Wb - Ec = 0$, with $a = [a_{n_a}, \ldots, a_0]^{\mathrm{T}}$, $b = [b_{n_b}, \ldots, b_0]^{\mathrm{T}}$, $c = [c_{n_c}, \ldots, c_0]^{\mathrm{T}}$, $Z \in \mathbb{R}^{m \times (n_a + 1)}$ a Hankel matrix constructed from $\tilde{z} \equiv z(t_1 : N)$ (Y is defined in a similar way w.r.t. $\tilde{y} \equiv y(t_1 : N)$), $W \in \mathbb{R}^{m \times (n_b + 1)}$ a Hankel matrix constructed from $\tilde{w} \equiv w(t_2 : N)$ (U is defined in a similar way w.r.t.

$\tilde{u} \equiv u(t_2:N)$) and $E \in \mathbb{R}^{m \times (n_c+1)}$ a Hankel matrix constructed from $\tilde{e} \equiv e(t_3:N)$ (all these Hankel matrices are constructed by putting the corresponding vector in its first column and last row, starting in the upper left corner). The model selection procedure can thus be rephrased as

$$\min_{\Delta y, \Delta u, e, a, b, c} \alpha J_{\text{output}} + \beta J_{\text{input}} + \gamma J_{\text{latency}},$$

$$\text{such that } Za - Wb - Ec = 0,$$

$$\text{such that } a_0 = 1 \text{ and } c_0 = 1. \tag{8}$$

Note the introduction of the constraints $a_0 = 1$ and $c_0 = 1$, which are not mentioned before. These extra constraints are imposed to avoid the trivial solution $a = 0$, $b = 0$ and $c = 0$. We propose a set of linear constraints, which lead—under the statistical assumptions stated at the beginning of this section—to ML solutions for the special cases like AR, ARMA, etc. At the end of this section we will revisit the choice of the constraints. By applying the technique of Lagrange multipliers to the optimization problem in (8) we can derive a set of equations that characterizes the stationary points of problem (8) (for details see Appendix A). Using the latter equations, it is possible to transform (8) into the following optimization problem:

$$\min_{a,b,c} [a^{\mathrm{T}}\ b^{\mathrm{T}}]\Xi^{\mathrm{T}}\left(\frac{\mathbf{B}_1(a)}{\alpha} + \frac{\mathbf{B}_2(b)}{\beta} + \frac{\mathbf{B}_3(c)}{\gamma}\right)^{-1}\Xi\begin{bmatrix} a \\ b \end{bmatrix},$$

$$\text{such that } a_0 = 1,\ c_0 = 1, \tag{9}$$

where

$\Xi = [Y \ \ -U]$, $\mathbf{B}_1(l) \equiv \mathbf{P}_1(l)\mathbf{P}_1(l)^{\mathrm{T}}$,

$\mathbf{B}_2(l) \equiv \mathbf{P}_2(l)\mathbf{P}_2(l)^{\mathrm{T}}$,

$\mathbf{B}_3(l) \equiv \mathbf{P}_3(l)\mathbf{P}_3(l)^{\mathrm{T}}$

and

$\mathbf{P}_1 : \mathbb{R}^{(n_a+1)\times 1} \to \mathbb{R}^{m \times (N-M+n_a+1)} : a \to \mathbf{P}_1(a)$,
defined by $\mathbf{P}_1(a)\tilde{z} = Za$.

$\mathbf{P}_2 : \mathbb{R}^{(n_b+1)\times 1} \to \mathbb{R}^{m \times (N-M+n_b+1)} : b \to \mathbf{P}_2(b)$,
defined by $\mathbf{P}_2(b)\tilde{w} = Wb$.

$\mathbf{P}_3 : \mathbb{R}^{(n_c+1)\times 1} \to \mathbb{R}^{m \times (N-M+n_c+1)} : c \to \mathbf{P}_3(c)$,
defined by $\mathbf{P}_3(c)\tilde{e} = Ec$.

Note that the objective function in (9) only depends on the model parameters a, b and c. The latter objective function is scaling invariant in the parameter vector $[a^{\mathrm{T}}\ b^{\mathrm{T}}\ c^{\mathrm{T}}]^{\mathrm{T}}$ (this is easily verified by noting the quadratic dependence of $\mathbf{B}_1(a)$ on a, $\mathbf{B}_2(b)$ on b and $\mathbf{B}_3(c)$ on c), implying that the constraints in (9) are necessary to single out a solution.

By choosing the constraints $a_0 = 1$, $c_0 = 1$, the solution obtained by solving (9) in the special case of PEM problems, corresponds to the solution obtained with classical methods for solving the PEM approach. From formulation (9) it is also clear that a change of the constraints into $a^{\mathrm{T}}a = 1$, $c^{\mathrm{T}}c = 1$ is bound to yield a different solution.

In this paragraph it is shown that the choice of the constraints—that have to be imposed to the model vectors a and c of problem (8) in order to avoid the trivial solution—is not an obvious one. To this end, we consider a simple AR model, since this is a special case of the framework described in (6). The model selection procedure[1] formulated in (8) then becomes

$$\min_{a,\tilde{e}} \tilde{e}^{\mathrm{T}}\tilde{e} \quad \text{such that } Ya = \tilde{e}. \tag{10}$$

Clearly, if we do not want to end up with the trivial solution $a = 0$, $\tilde{e} = 0$ we have to impose an extra constraint on a. The classical PEM approach adopts the constraint $a_0 = 1$. We now investigate what happens when the latter constraint is replaced by $a^{\mathrm{T}}a = 1$. Problem (10) thus becomes

$$\min_{a} \frac{a^{\mathrm{T}}Y^{\mathrm{T}}Ya}{m} \quad \text{such that } a^{\mathrm{T}}a = 1. \tag{11}$$

We inserted m (the number of rows of Y) in the cost function of (11), since this allows us to let $m \to \infty$ while the cost function stays bounded. Suppose that the latent variables in $\tilde{e}$ are i.i.d. zero-mean Gaussian noise. For stable AR systems it is well-known (Ljung, 1987) that the output will then be stationary too. As a consequence, the matrix $Y^{\mathrm{T}}Y/m$ becomes Toeplitz in the limit for $m \to \infty$. Since it is already symmetric by construction, it belongs to the class of centrosymmetric[2] matrices. It should be clear that the a that results from (11) is the eigenvector corresponding to the smallest eigenvalue of $Y^{\mathrm{T}}Y/m$. Applying Theorem 3.2.6 from Lemmerling (1999) to problem (11) we can conclude that the model vector a is centrosymmetric or skew centrosymmetric[3] and thus we find that the poles of the AR system, i.e. the roots of the following equation in ζ:

$$a_0\zeta^{n_a} + a_1\zeta^{n_a-1} + \cdots + a_{n_a-1}\zeta + a_{n_a} = 0,$$

come in pairs $(\zeta, 1/\zeta)$ (i.e. symmetric w.r.t. the unit circle). Hence by using a quadratic constraint we identify a model that has unstable poles, even if the original system that generated the data was stable. Said in other words, the pole estimates are biased. One can show that

[1] Note that we have left out the constraints on a and c since this will be discussed further on.

[2] A square matrix $A \in \mathbb{R}^{q \times q}$ is called centrosymmetric if $A = E_q A E_q$, with $E_q \in \mathbb{R}^{q \times q}$ the anti-identity matrix.

[3] A vector $v \in \mathbb{R}^{q \times 1}$ is called (skew) centrosymmetric if $v = E_q v$ ($v = -E_q v$) with $E_q \in \mathbb{R}^{q \times q}$ the anti-identity matrix.

with the linear constraint $a_0 = 1$, the pole estimates are unbiased.

4. Properties and interpretations

In this section we discuss some properties of the misfit versus latency framework. In the first subsection some orthogonality properties between the misfit on the input, the misfit on the output and the latent variables are discussed. Such orthogonality properties should come as no surprise. They express orthogonality in a similar way as in a (static) LS approach where the residuals are orthogonal to the data or as in a (dynamic) PEM approach where the residuals satisfy an orthogonality condition w.r.t. the data. The second subsection shows that the misfit on the input, the misfit on the output and the latent variables satisfy themselves linear systems. Proofs are presented in appendix B.

4.1. Orthogonality

The first orthogonality property we prove concerns the "corrected" input $\tilde{w}$ and the input misfit $\tilde{w} - \tilde{u}$.

Property 4.1. The "corrected" input $\tilde{w}$ and the input misfit $\tilde{w} - \tilde{u}$ obtained as the result of the model selection procedure (8) are orthogonal: $\tilde{w}^{\mathrm{T}}(\tilde{w} - \tilde{u}) = 0$.

The next orthogonality property is on the output misfit and the latent variables.

Property 4.2. Let

$$K = \begin{bmatrix} \alpha I_{N-M+n_a+1} & 0 \\ 0 & \gamma I_{N-M+n_c+1} \end{bmatrix}.$$

The vector that contains the "corrected" output and the latent variables is K-orthogonal to the vector containing the output misfit and the latent variables: $[(\tilde{z} - \tilde{y})^{\mathrm{T}}\ \tilde{e}^{\mathrm{T}}]K[\tilde{z}^{\mathrm{T}}\ \tilde{e}^{\mathrm{T}}]^{\mathrm{T}} = 0$.

4.2. LTI systems for misfit and latency

Remember the Rank 1 property of the misfit $[\Delta A\ \Delta b]$ in the static case. Similarly, the latent variables, the input misfit and the output misfit are structured too in the sense that under certain conditions on the model orders, they satisfy themselves model equations of LTI dynamic systems (whose parameter vectors can easily be derived from the parameter vectors obtained in (8)).

Property 4.3. If $n_a = n_b$, the input misfit and the output misfit satisfy the following LTI dynamic system equations:

$$-\beta(W - U)a(n_a + 1 : -1 : 1) = \alpha(Z - Y)b(n_b + 1 : -1 : 1). \quad (12)$$

If $n_a = n_c$, the output misfit and the latent variables satisfy the following LTI dynamic system equations:

$$-\alpha(Z - Y)c(n_c + 1 : -1 : 1) = \gamma E a(n_a + 1 : -1 : 1). \quad (13)$$

If $n_b = n_c$, the input misfit and the latent variables satisfy the following LTI dynamic system equations:

$$\beta(W - U)c(n_c + 1 : -1 : 1) = \gamma E b(n_b + 1 : -1 : 1). \quad (14)$$

If in the previous property, the conditions on the orders are not satisfied, it is still possible to prove similar properties. If e.g. $n_b > n_a$, a formula similar to (12) can be proven by defining a new $\check{\mathbf{P}}_1(a) \in \mathbb{R}^{m \times (N-M+n_b+1)}$, which is simply $\mathbf{P}_1(a)$ padded with $n_b - n_a$ zero columns.

5. Experiments

In this section, we describe two experiments to give an idea of the potential of the misfit versus latency framework. In the first experiment we simulate an ARMAX model. The experiment itself consists in determining models of the new framework using the selection procedure (9). Models are determined for several combinations of the triplet (α,β,γ) in such a way that we obtain a "root locus" of the zeros of the a polynomial (i.e. the poles of the model) with on the one end the PEM approach and on the other end the behavioral modeling approach.

The second experiment shows how the knowledge of the statistical properties of the misfit and latency variables can be exploited in the new framework, thereby leading to an improved statistical accuracy of the estimated parameter vectors. The results are compared to those of the classical PEM approach.

5.1. Experiment 1

For this experiment we start from the following exact model parameters:

$a = [0.81\ \ -1.2541\ \ 1]^{\mathrm{T}}$, $b = [1\ 2.3\ 0.3]^{\mathrm{T}}$, $c = 1$. Using an i.i.d. Gaussian noise sequence $e(t)$ with standard deviation 1, a deterministic input sequence $w(t)$ and initial conditions $z(1) = z(2) = 0$, an output $z(t)$ is generated using the following ARMAX model equation:

$A(q)z(t) = B(q)w(t) + C(q)e(t)$, $t = 3, \ldots, N$. Setting $N = 50$ we obtain Fig. 3, which shows the above mentioned sequences. Since we consider the case that there is no misfit on the input nor on the output, we have that $z(t) = y(t)$ and $w(t) = u(t)$ (see Fig. 2). Using $\tilde{y} = y(1:50)$, $\tilde{u} = u(1:50)$, $\tilde{e} = e(3:50)$, we can construct the corresponding matrices Y, U and E and apply the model selection procedure (9). The optimization is performed using the function leastsq in Matlab. The procedure is repeated for the following triplets (α,β,γ): Case 1: $(\alpha,\beta,\gamma) = (10^5,10^5,1)$; Case 2: $(\alpha,\beta,\gamma) = (10^3,10^3,10^2)$;

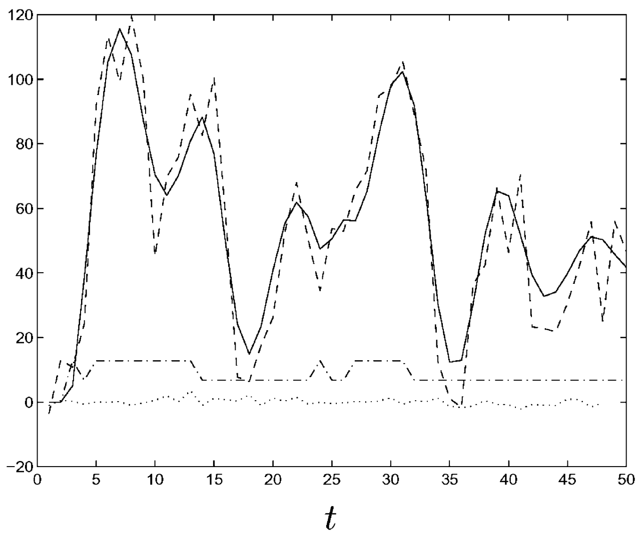

Fig. 3. This figure shows the deterministic input $w(t)$ (dash-dotted line), the i.i.d. Gaussian noise input $e(t)$ (dotted line) of experiment 1. The full line is the sequence $z(t)$, obtained by using the ARMAX model and the initial conditions described in the first experiment. The dashed line is the sequence $y(t)$ as used in the second experiment. It is obtained by adding a noise "misfit" (in this particular case i.i.d. Gaussian noise with standard deviation 10) to the sequence $z(t)$.

Case 3: $(\alpha,\beta,\gamma) = (1,1,10^5)$. Case 1 clearly corresponds to the ARMAX case (See Case 26 of Table 1). Case 3 clearly corresponds to the dynamic TLS model (See Case 15 of Table 1). Case 2 is a model that lies in between an ARMAX model and a dynamic TLS model. To give an idea of the effect of the different choices of the triplet (α,β,γ), we plot in Fig. 4 the poles (zeroes of the polynomial based on the estimated a): poles of Case 1 are indicated by a "x", those of Case 2 by a "o" and for Case 3 we used a " + ". The left hand side shows the location of the poles w.r.t. the unit circle. The right hand side zooms in on the region of interest. We clearly see the "root locus" for the poles of the different models, i.e. the continuous evolution of the poles when going from a PEM approach ("x") towards a behavioral modeling approach (" + ").

5.2. Experiment 2

In this experiment we use the same ARMAX model as in the first experiment. The output z is thus generated in the same way as in the first experiment, but in this case we add noise to z such that $z \neq y$ as opposed to experiment 1. The noise added to z is i.i.d. Gaussian noise with standard deviation 10. The sequence y that is obtained in this way is the dashed line in Fig. 3. We now perform 200 Monte-Carlo simulations in which each time a different realization of the latent variable e and the noise misfit on the output $(y - z)$ is generated. In each run we determine 2 models (using the known model orders $[n_a\ n_b\ n_c] = [2\ 2\ 0]$) based on the available measurements y and u:

(1) a model determined using (9) with $\alpha = 1$, $\beta = 10^5$ and $\gamma = 100$.
(2) a classical ARMAX model (i.e. the PEM approach) using the identification toolbox in Matlab.

The choice of α, β and γ is obviously based on the knowledge of the statistical properties of the latent

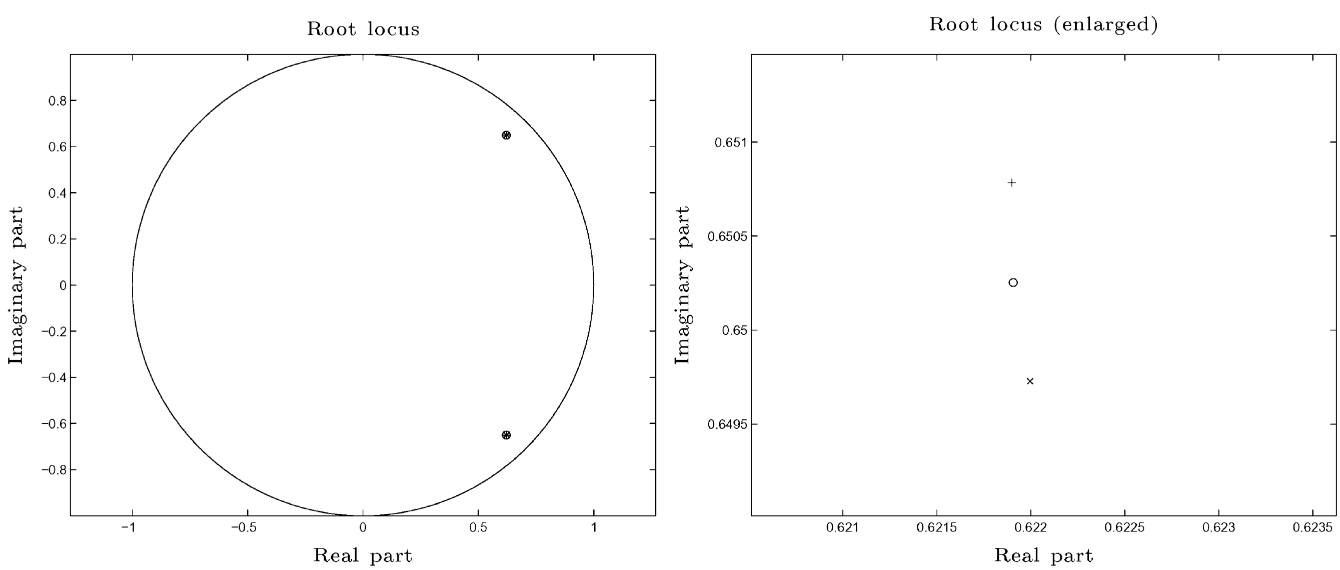

Fig. 4. These figures show the root locus obtained by varying the parameters, α, β and γ in the model selection procedure (9) applied to the example described in experiment 1.

variable and the output misfit. First of all, $\gamma/\alpha = 100$ since the variance on the output misfit is 100 times larger than the variance on the latent variable. Furthermore, $\beta = 10^5$, since no misfit is present on the input (note that the 10^5 is thus used as an approximation of ∞).

The goal of the experiment is in fact to see whether we can get better statistical results with our new framework, if the statistical properties of noise misfit and latent variables are properly taken into account. To get an idea of the statistical accuracy of our new framework and the associated selection procedure (9), we calculate the following relative errors for the "misfit versus latency" model obtained with (9) and the classical PEM ARMAX result

$$err_{\mathrm{mvl}} = \sum_{i=1}^{200} \frac{\|a_{\mathrm{mvl}}^{(i)} - a\|_2}{200\|a\|_2}, \quad err_{\mathrm{PEM}} = \sum_{i=1}^{200} \frac{\|a_{\mathrm{PEM}}^{(i)} - a\|_2}{200\|a\|_2},$$

where a is the exact AR parameter vector, $a_{\mathrm{mvl}}^{(i)}$ is the AR parameter vector obtained using (9) in the ith Monte Carlo run and $a_{\mathrm{PEM}}^{(i)}$ is the AR parameter vector obtained using the classical PEM approach in the ith Monte Carlo run. The relative accuracies obtained in this way are: $err_{\mathrm{mvl}} = 1.84\%$ and $err_{\mathrm{PEM}} = 35.75\%$. This clearly shows the potential of the new framework.

It should be noted that better results can be obtained with the PEM approach when a higher order noise model c is used. The best results were obtained with $[n_a\ n_b\ n_c] = [2\ 2\ 2]: err_{\mathrm{PEM}} = 4.23\%$. It can easily be explained why this model order leads to the best PEM model. The exact model set equations are

$$\sum_{i=0}^{2} a_i(y(t-i) + \Delta y(t-i)) = \sum_{i=0}^{2} b_i u(t-i) + e(t),$$

where $\sigma_{\Delta y} = 10$ and $\sigma_e = 1$. Since $\sigma_e \ll \sigma_{\Delta y}$ a higher order noise model will model in the PEM approach the misfit contribution Δy, which can be represented by the following equation:

$$\sum_{i=0}^{2} a_i y(t-i) \approx \sum_{i=0}^{2} b_i u(t-i) + \sum_{i=0}^{2} c_i \Delta y(t-i),$$

where $\approx$ indicates that the latency contribution $e(t)$ has been neglected. Nevertheless, even the best PEM model yields a more than 100% higher relative error (as defined above) than the obtained misfit versus latency model.

6. Conclusion

In this paper we have presented the misfit versus latency framework that was introduced in De Moor and Lemmerling (1998). The latter framework combines some ideas of the PEM approach and the behavioral modeling approach. We have shown that by using the technique of Lagrange multipliers the proposed model selection procedure can be rephrased as an optimization problem involving only the model parameters as variables. The importance of the constraints imposed on the model parameters has been demonstrated. Some orthogonality properties and interpretations are proven.

Using this new optimization approach we have been able to do some experiments in the new framework. First of all it has been demonstrated—by means of a root locus plot—how the specific values of the weights α, β and γ influence the location of the poles of the selected model. Secondly, it has been shown that knowledge of the statistical properties of the misfit and latency variables can greatly enhance the statistical accuracy of the obtained model parameter vectors, compared to the classical PEM approach.

Acknowledgements

Philippe Lemmerling is supported by a post doctoral KUL scholarship. Bart De Moor is a full professor at the Katholieke Universiteit Leuven. This work is supported by several institutions: the Flemish Government: Research Council K.U. Leuven (Concerted Research Action GOA-MIPS and Mefisto-666); the FWO projects G.0240.99, G.0256.97, and Research Communities: ICCoS and ANMMM; IWT projects: EUREKA 1562-SINOPSYS, EUREKA 2063-IMPACT, STWW the Belgian State, Prime Minister's Office — OSTC —: IUAP P4-02 (1997–2001) and IUAP P4-24 (1997–2001), Sustainable Mobility Programme — Project MD/01/24 (1997–2000); the European Commission: TMR Networks: ALAPEDES and System Identification, Brite/Euram Thematic Network (NICONET); Industrial Contract Research: ISMC, Electrabel, Laborelec, Verhaert, Europay.

Appendix A

A.1. Derivation of (9)

By applying the technique of Lagrange multipliers to the optimization problem in (8) we will now derive a set of equations that characterizes the stationary points of problem (8). The Lagrangian of the latter problem is

$$L(a, b, c, \tilde{w}, \tilde{z}, \tilde{e}, l, \kappa, \lambda) = \alpha \sum_{t=t_1}^{N} (\Delta y(t))^2 + \beta \sum_{t=t_2}^{N} (\Delta u(t))^2 + \gamma \sum_{t=t_3}^{N} (e(t))^2 + l^{\mathrm{T}}(Za - Wb - Ec) + \kappa(a_0 - 1) + \lambda(c_0 - 1).$$

Taking the derivatives of L with respect to a, b, c, $\tilde{w}$, $\tilde{z}$, $\tilde{e}$, l, κ, λ and setting these equal to 0, we obtain the following equations:

$$Z^{\mathrm{T}} l + \kappa[0 \ \ldots \ 0 \ 1]^{\mathrm{T}} = 0, \quad (A.1)$$

$$-W^{\mathrm{T}} l = 0, \quad (A.2)$$

$$-E^{\mathrm{T}} l + \lambda[0 \ \ldots \ 0 \ 1]^{\mathrm{T}} = 0, \quad (A.3)$$

$$\beta(\tilde{w} - \tilde{u}) - \mathbf{P}_2(b)^{\mathrm{T}} l = 0, \quad (A.4)$$

$$\alpha(\tilde{z} - \tilde{y}) + \mathbf{P}_1(a)^{\mathrm{T}} l = 0, \quad (A.5)$$

$$\gamma\tilde{e} - \mathbf{P}_3(c)^{\mathrm{T}} l = 0, \quad (A.6)$$

$$Za - Wb - Ec = 0, \quad (A.7)$$

$$a_0 = 1, \quad (A.8)$$

$$c_0 = 1, \quad (A.9)$$

where $\mathbf{P}_1$, $\mathbf{P}_2$ and $\mathbf{P}_3$ are matrix functions defined as follows:

$\mathbf{P}_1 : \mathbb{R}^{(n_a+1)\times 1} \to \mathbb{R}^{m\times(N-M+n_a+1)} : a \to \mathbf{P}_1(a)$,
defined by $\mathbf{P}_1(a)\tilde{z} = Za$.
$\mathbf{P}_2 : \mathbb{R}^{(n_b+1)\times 1} \to \mathbb{R}^{m\times(N-M+n_b+1)} : b \to \mathbf{P}_2(b)$,
defined by $\mathbf{P}_2(b)\tilde{w} = Wb$.
$\mathbf{P}_3 : \mathbb{R}^{(n_c+1)\times 1} \to \mathbb{R}^{m\times(N-M+n_c+1)} : c \to \mathbf{P}_3(c)$,
defined by $\mathbf{P}_3(c)\tilde{e} = Ec$.
$\mathbf{P}_4 : \mathbb{R}^{m\times 1} \to \mathbb{R}^{(n_a+1)\times(N-M+n_a+1)} : l \to \mathbf{P}_4(l)$,
defined by $\mathbf{P}_4(l)\tilde{z} = Z^{\mathrm{T}} l$.
$\mathbf{P}_5 : \mathbb{R}^{m\times 1} \to \mathbb{R}^{(n_b+1)\times(N-M+n_b+1)} : l \to \mathbf{P}_5(l)$,
defined by $\mathbf{P}_5(l)\tilde{w} = W^{\mathrm{T}} l$.

Furthermore we define $\mathbf{B}_1(l) \equiv \mathbf{P}_1(l)\mathbf{P}_1(l)^{\mathrm{T}}$, $\mathbf{B}_2(l) \equiv \mathbf{P}_2(l)\mathbf{P}_2(l)^{\mathrm{T}}$, $\mathbf{B}_3(l) \equiv \mathbf{P}_3(l)\mathbf{P}_3(l)^{\mathrm{T}}$, $\mathbf{B}_4(l) \equiv \mathbf{P}_4(l)\mathbf{P}_4(l)^{\mathrm{T}}$, $\mathbf{B}_5(l) \equiv \mathbf{P}_5(l)\mathbf{P}_5(l)^{\mathrm{T}}$ and $\mathbf{B}_6(l) \equiv \mathbf{P}_6(l)\mathbf{P}_6(l)^{\mathrm{T}}$. Using (A.4) and the definition of $\mathbf{B}_2$, we find

$$(\tilde{w} - \tilde{u})^{\mathrm{T}}(\tilde{w} - \tilde{u}) = l^{\mathrm{T}}\mathbf{B}_2(b)l/\beta^2. \quad (A.10)$$

Similarly, we find by combining (A.5) with the definition of $\mathbf{B}_1$ that

$$(\tilde{z} - \tilde{y})^{\mathrm{T}}(\tilde{z} - \tilde{y}) = l^{\mathrm{T}}\mathbf{B}_1(a)l/\alpha^2, \quad (A.11)$$

and finally, proper use of (A.6) and the definition of $\mathbf{B}_3$ reveals that

$$\tilde{e}^{\mathrm{T}}\tilde{e} = l^{\mathrm{T}}\mathbf{B}_3(c)l/\gamma^2. \quad (A.12)$$

From (A.7), the definition of $\mathbf{P}_1$, $\mathbf{P}_2$ and $\mathbf{P}_3$, we know that

$$\mathbf{P}_1(a)\tilde{z} - \mathbf{P}_2(b)\tilde{w} - \mathbf{P}_3(c)\tilde{e} = 0.$$

Using (A.4)–(A.6), the definitions of $\mathbf{P}_1$, $\mathbf{P}_2$, $\mathbf{B}_1$, $\mathbf{B}_2$ and $\mathbf{B}_3$, the last equation becomes

$$Ya - Ub = \left(\frac{\mathbf{B}_1(a)}{\alpha} + \frac{\mathbf{B}_2(b)}{\beta} + \frac{\mathbf{B}_3(c)}{\gamma}\right) l$$

or

$$l = \left(\frac{\mathbf{B}_1(a)}{\alpha} + \frac{\mathbf{B}_2(b)}{\beta} + \frac{\mathbf{B}_3(c)}{\gamma}\right)^{-1} (Ya - Ub). \quad (A.13)$$

Combining (A.10)–(A.13), the cost function in the model selection procedure can be rewritten as follows:

$$[a^{\mathrm{T}} \ b^{\mathrm{T}}]\begin{bmatrix} Y^{\mathrm{T}} \\ -U^{\mathrm{T}} \end{bmatrix}\left(\frac{\mathbf{B}_1}{\alpha} + \frac{\mathbf{B}_2}{\beta} + \frac{\mathbf{B}_3}{\gamma}\right)^{-1} [Y \ -U]\begin{bmatrix} a \\ b \end{bmatrix}. \quad (A.14)$$

Note that (A.14) only depends on the model parameters a, b and c. Careful inspection of the Lagrange multiplier equations (A.1)–(A.9) that have been used to arrive at (A.14) reveals that only constraint independent multiplier equations were used. The latter means that if the constraints on a and c (see Eqs. (A.8) and (A.9)) were changed, we would still find the same cost function (A.14). The cost function (A.14) is scaling invariant in the parameter vector $[a^{\mathrm{T}} \ b^{\mathrm{T}} \ c^{\mathrm{T}}]^{\mathrm{T}}$ (this is easily verified by noting the quadratic dependence of $\mathbf{B}_1(a)$ on a, $\mathbf{B}_2(b)$ on b and $\mathbf{B}_3(c)$ on c), implying that we have to impose extra constraints on the parameter vectors in order to single out a solution. Note that this scaling invariance property only holds if none of the misfit terms is missing. If we write down (A.14) e.g. for an ARMAX model (i.e. a special case of our new framework), it becomes obvious that (A.14) no longer is scaling invariant, because the terms containing $\mathbf{B}_1(a)$ and $\mathbf{B}_2(b)$ have disappeared. In those cases, the cost function is no longer scaling invariant and the constraints that we impose do not just single out one of the solutions along a line of solutions, but they can lead to solutions lying in different directions. To obtain the same results as in our misfit versus latency framework, we also impose the constraints $a_0 = 1$ and $c_0 = 1$. Summarizing, we have derived an optimization formulation that yields the same solution as the model selection procedure (8):

$$\min_{a,b,c} \ [a^{\mathrm{T}} \ b^{\mathrm{T}}]\Xi^{\mathrm{T}}\left(\frac{\mathbf{B}_1(a)}{\alpha} + \frac{\mathbf{B}_2(b)}{\beta} + \frac{\mathbf{B}_3(c)}{\gamma}\right)^{-1} \Xi\begin{bmatrix} a \\ b \end{bmatrix},$$

such that $a_0 = 1$, $c_0 = 1$,
where $\Xi = [Y \quad -U]$.

Appendix B

Proof of property 4.1. Using (A.4), the definition of $\mathbf{P}_2$ and (A.2), we find

$$\tilde{w}^{\mathrm{T}}(\tilde{w} - \tilde{u}) = \tilde{w}^{\mathrm{T}}\mathbf{P}_2(b)^{\mathrm{T}} l/\beta = b^{\mathrm{T}}W^{\mathrm{T}} l/\beta = 0,$$

thereby proving the first orthogonality property.

Proof of property 4.2. Consecutive application of (A.5), the definition of $\mathbf{P}_1$, (A.1) and (A.8) yields:

$$\begin{aligned} \tilde{z}^{\mathrm{T}}(\alpha I_{N-M+n_a+1})(\tilde{z} - \tilde{y}) &= -\tilde{z}^{\mathrm{T}}\mathbf{P}_1(a)^{\mathrm{T}} l \\ &= -a^{\mathrm{T}}Z^{\mathrm{T}} l = \kappa. \end{aligned} \quad (B.1)$$

On the other hand, using (A.6), the definition of $\mathbf{P}_3$, (A.3) and (A.9), we obtain:

$$\tilde{e}^{\mathrm{T}}(\gamma I_{N-M+n_c+1})\tilde{e} = \tilde{e}^{\mathrm{T}}\mathbf{P}_3(c)^{\mathrm{T}} l = c^{\mathrm{T}} E^{\mathrm{T}} l = \lambda. \tag{B.2}$$

Multiplying (A.1) with a^{T} to the left, (A.2) with b^{T} to the left and (A.3) with c^{T} to the left and adding the results, we find by using (A.7) that $-\kappa a_0 = \lambda c_0$ or by taking (A.8) and (A.9) into account: $\lambda = -\kappa$. Since $\kappa = -\lambda$ the property follows by adding (B.1) and (B.2).

Proof of property 4.3. First note that the following properties are readily checked if $n_a = n_b$, simply by using the definitions of $\mathbf{P}_1$ and $\mathbf{P}_2$:

$$\mathbf{P}_1 = \mathbf{P}_2, \tag{B.3}$$

$$\begin{aligned}&\mathbf{P}_1(b(n_b+1:-1:1))\mathbf{P}_1(a)^{\mathrm{T}}\\ &\quad = \mathbf{P}_1(a(n_a+1:-1:1))\mathbf{P}_1(b)^{\mathrm{T}}.\end{aligned} \tag{B.4}$$

Using (A.4), (A.5), (B.3) we find

$$\begin{aligned}&\beta\mathbf{P}_1(a(n_a+1:-1:1))(\tilde{w}-\tilde{u})\\ &\quad = \mathbf{P}_1(a(n_a+1:-1:1))\mathbf{P}_1(b)^{\mathrm{T}} l\end{aligned}$$

and

$$\begin{aligned}&\alpha\mathbf{P}_1(b(n_b+1:-1:1))(\tilde{z}-\tilde{y})\\ &\quad = -\mathbf{P}_1(b(n_b+1:-1:1))\mathbf{P}_1(a)^{\mathrm{T}} l.\end{aligned}$$

From the latter 2 equations, the definition of $\mathbf{P}_1$ and (B.4) we obtain:

$$\begin{aligned}&-\beta(W-U)a(n_a+1:-1:1)\\ &\quad = \alpha(Z-Y)b(n_b+1:-1:1).\end{aligned}$$

Similarly, if $n_a = n_c$ the following is easily verified by using the definitions of $\mathbf{P}_1$ and $\mathbf{P}_3$:

$$\mathbf{P}_1 = \mathbf{P}_3, \tag{B.5}$$

$$\begin{aligned}&\mathbf{P}_1(c(n_c+1:-1:1))\mathbf{P}_1(a)^{\mathrm{T}}\\ &\quad = \mathbf{P}_1(a(n_a+1:-1:1))\mathbf{P}_1(c)^{\mathrm{T}}.\end{aligned} \tag{B.6}$$

Combining (A.5), (A.6), (B.5) it is clear that

$$\begin{aligned}&-\alpha\mathbf{P}_1(c(n_c+1:-1:1))(\tilde{z}-\tilde{y})\\ &\quad = \mathbf{P}_1(c(n_c+1:-1:1))\mathbf{P}_1(a)^{\mathrm{T}} l\end{aligned}$$

and

$$\begin{aligned}&\gamma\mathbf{P}_1(a(n_a+1:-1:1))\tilde{e}\\ &\quad = \mathbf{P}_1(a(n_a+1:-1:1))\mathbf{P}_1(c)^{\mathrm{T}} l.\end{aligned}$$

Using the latter 2 equations, the definition of $\mathbf{P}_1$ and (B.6) we find:

$$-\alpha(Z-Y)c(n_c+1:-1:1) = \gamma E a(n_a+1:-1:1).$$

To obtain the third LTI dynamic system of this property, note that if $n_b = n_c$ straightforward application of the definitions of $\mathbf{P}_2$ and $\mathbf{P}_3$ gives:

$$\mathbf{P}_2 = \mathbf{P}_3, \tag{B.7}$$

$$\begin{aligned}&\mathbf{P}_2(c(n_c+1:-1:1))\mathbf{P}_2(b)^{\mathrm{T}}\\ &\quad = \mathbf{P}_2(b(n_b+1:-1:1))\mathbf{P}_2(c)^{\mathrm{T}}.\end{aligned} \tag{B.8}$$

Using (A.4), (A.6), (B.7) yields:

$$\begin{aligned}&\beta\mathbf{P}_2(c(n_c+1:-1:1))(\tilde{w}-\tilde{u})\\ &\quad = \mathbf{P}_2(c(n_c+1:-1:1))\mathbf{P}_2(b)^{\mathrm{T}} l,\end{aligned}$$

$$\begin{aligned}&\gamma\mathbf{P}_2(b(n_b+1:-1:1))\tilde{e}\\ &\quad = \mathbf{P}_2(b(n_b+1:-1:1))\mathbf{P}_2(c)^{\mathrm{T}} l.\end{aligned}$$

Combining the latter 2 equations, the definition of $\mathbf{P}_2$ and (B.8), it follows that:

$$\beta(W-U)c(n_c+1:-1:1) = \gamma E b(n_b+1:-1:1).$$

References

De Moor, B. (1993). Structured total least squares and l_2 approximation problems. In Van Dooren, Ammar, Nichols, & Mehrmann (Eds.), *Linear algebra and its applications*, *special issue on numerical linear algebra methods in control*, *signals and systems*. (pp. 188–189, 163–207).

De Moor, B. (1994). Total least squares for affinely structured matrices and the noisy realization problem. *IEEE Transactions on Signal Processing*, *42*, 3004–3113.

De Moor, B. (1997). Linear system identification, structured total least squares and the riemannian svd. In S. Van Huffel, (Ed.), *Recent advances in total least squares techniques and errors-in-variables modeling*. Philadelphia: SIAM. (pp. 225–239).

De Moor, B., & Lemmerling, P. (1998). A new trade-off in linear system identification: misfit versus latency. In *Proceedings of the MTNS98*, Padova, Italy, June.

De Moor, B., & Roorda, B. (1994). l_2-optimal linear system identification: Dynamic total least squares for siso systems. In *Proceedings of the 33rd IEEE CDC*. (pp. 2874–2879), Florida, December.

Golub, G. H., & Van Loan, C. E. (1996). *Matrix computations* (3rd ed.). Baltimore: Johns Hopkins University Press.

Lemmerling, P. (1999). *Structured total least squares*: *analysis*, *algorithms and applications*. Ph.D. thesis, Department Electrical Engineering, ESAT-SISTA/COSIC, Katholieke Universiteit Leuven, May.

Ljung, L. (1987). *System identification: theory for the user*. Englewood Cliffs, NJ: Prentice-Hall.

Roorda, B. (1995). Algorithms for global total least squares modelling of finite multi-variable time series. *Automatica*, *31*(3), 391–404.

Roorda, B., & Heij, C. (1995). Global total least squares modelling of multivariable time series. *IEEE Transactions on Automatic Control*, *40*(1), 50–63.

Söderström, T., & Stoica, P. (1989). *System identification*. Englewood Cliffs, NJ: Prentice-Hall.

Van Huffel, S., & Vandewalle, J. (1991). *The total least squares problem*: *computational aspects and analysis*, Vol. 9. Philadelphia: SIAM.

Willems, J. C. (1986). From time series to linear system, Part i. Finite dimensional linear time invariant systems, Part ii. Exact modelling, Part iii. Approximate modelling. *Automatica*, *22/23*, 561–580, 675–694, 87–115.

Philippe Lemmerling received the M.Sc. and Ph.D. degree in Electrical Engineering from the Katholieke Universiteit Leuven, in 1994 and 1999, respectively. Currently he is a postdoctoral researcher at the Katholieke Universiteit Leuven. His research interests are in the areas of linear algebra, optimization and their application in biomedical engineering (www.esat.kuleuven.ac.be/ ~ biomed).

Bart De Moor (www.esat.kuleuven.ac.be/ ~ demoor) is a full professor at the Department of Electrical Engineering of the Katholieke Universiteit Leuven in Belgium, in the research group SISTA. He was born on Tuesday July 12, 1960 and received his doctoral degree in Applied Sciences in 1988 at the Katholieke Universiteit Leuven, Belgium. He was a visiting research associate (1988–1989) at the Departments of Computer Science (Gene Golub) and Electrical Engineering (Thomas Kailath) of Stanford University, California, USA. His research interests include numerical linear algebra (generalized svd, complementarity problems, structured total least squares, multilinear algebra), system identification (subspace methods), datamining (least squares support vector machines) and bio-informatics. He has more than 300 (co-)authored papers in international journals and conference proceedings and received several awards for his work (Leybold-Heraeus Prize (1986), the Leslie Fox Prize (1989), the Guillemin-Cauer Best Paper Award (1990) of the IEEE Transaction on Circuits and Systems, the bi-annual Siemens prize (1994), the tri-annual best paper award of Automatica (1996), the IEEE Signal Processing Society Best Paper Award (1999) and he was a Laureate of the Belgian Royal Academy of Sciences (1992)). He is serving on several boards of (inter)national scientific, cultural and commercial organisations, including the Flemish Interuniversity Institute for Biotechnology (www.vib.be), the Belgian Study Center for Nuclear Energy, the Flemish Center for Postharvest Technology. He is also one of the founders of ISMC NV (Intelligent Systems, Modelling and Control, www.ismc.be), a spin-off company specialized in modelling and control of multivariable industrial processes (chemical and power plants, oil exploration) and Data4S NV (www.data4s.com), a spin-off company active in datamining products and services in telecommunications and customer followup. In 1991–1992 he was the chief of staff of the Belgian federal minister of science Ms. Wivina Demeester-DeMeyer and later on of the Belgian prime minister Wilfried Martens. From 1994–1999 he was the main Advisor on Science and Technology policy of the Flemish minister-president Luc Van den Brande.