# Misfit versus latency

Philippe Lemmerling and Bart De Moor

*Abstract*— **In this paper we present a framework that combines the 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 modeling, behavioral modeling, dynamic linear time-invariant systems**

## I. Introduction

Mathematical models are the heart of many applications in areas such as control system design, prediction, simulation, fault detection,... In this paper a new framework [7] for modeling dynamic linear time-invariant (LTI) systems is developed. It combines the ideas of the prediction error modeling (PEM) approach [3], [11] and the behavioral modeling approach [13], [9], [10], [8], [4], [5], [6]. Both approaches –the PEM approach and the behavioral modeling approach– aim to fit a model of the proposed model set to the 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 labelled 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 thanks 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 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 is 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 [7].
The paper is structured as follows. The next section introduces the idea of misfit versus latency for static LTI models. Section III extends this 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 the technique of 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 V), 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 IV derives some properties and interpretations of the new framework. The potential of the misfit versus latency framework is illustrated with some simulation examples in section V.

## II. Static case

In this section we address static LTI models of the form:

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

with $A \in \mathbb{R}^{m\times n}, m \geq n, b \in \mathbb{R}^{m\times 1}$ the data and $x \in \mathbb{R}^{n\times 1}$ the parameter vector. In general, equation (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.
1. The first method consists of introducing a so called "latent variable" $e \in \mathbb{R}^{m\times 1}$, such that $Ax = b + e$, and the appropriate model is selected by the following optimization problem:

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

where $\|.\|_2$ indicates the vector 2−norm. The latter approach is called the "latency" approach and corresponds to the well-known LS solution of the overdetermined system of equations $Ax \approx b$. As can be seen from (2), the inconsistency of the equations is explained by the introduction of a new variable $e$. The minimization of $\|e\|_2$ can heuristically

The authors are with the Department of Electrical Engineering, ESAT-SISTA, Katholieke Universiteit Leuven, Kardinaal Mercierlaan 94, 3001 Heverlee, Belgium. Phone: +32-(0)16-321709. Fax: +32-(0)16-321970. Philippe Lemmerling is a Research Assistant with the K.U.Leuven. Email: philippe.lemmerling@esat.kuleuven.ac.be.

Bart De Moor is a Research Associate at the F.W.O. (Fund for Scientific Research-Flanders). Email: bart.demoor@esat.kuleuven.ac.be. 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

be explained by the desire to find the "minimum energy" $e$ so that $Ax = b + e$.

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

$$\begin{aligned} &\min_{\Delta A, \Delta b, x} \|[\Delta A \;\; \Delta b]\|_F^2 \\ &\text{such that } \; (A + \Delta A)x = b + \Delta b, \end{aligned} \tag{3}$$

where $\|.\|_F$ 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 [1], [12] that the solution to (3) follows from the SVD of $[A \;\; b]$. The correction $[\Delta A \; \Delta b]$ is a rank one matrix obtained from the SVD of $[A \; b]$. Further on, this type of approach will be referred to as the misfit approach.

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:

$$\begin{aligned} &\min_{\Delta A, \Delta b, e, x} \alpha \|[\Delta A \;\; \Delta b]\|_F^2 + \beta \|e\|_2^2 \\ &\text{such that } \; (A + \Delta A)x = b + \Delta b + e, \end{aligned} \tag{4}$$

with $\alpha$ and $\beta$ user-specified positive scalar weights that are inversely proportional to the importance assigned to the misfit respectively the latency contributions. 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 $[A \;\; b]W^{-1/2} = U\Sigma Z^T$, the SVD of $[A \;\; b]W^{-1/2}$, $V = W^{-1/2}Z$, $W = \begin{bmatrix} \frac{1}{\alpha} I_q & 0 \\ 0 & \frac{1}{\alpha} + \frac{1}{\beta} \end{bmatrix}$ under the assumption that $V(n+1, n+1) \neq 0$. We see that the approach (4)

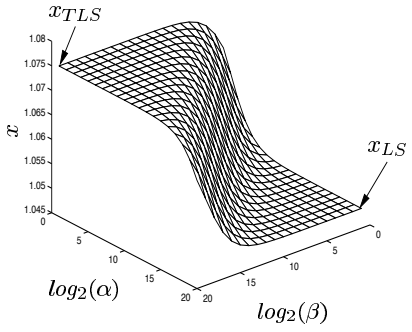



Fig. 1. *This figure shows the solution $x$ of (4) as a function of $log_2(\alpha)$ and $log_2(\beta)$, applied to example 5. The higher $\alpha$, the less "misfit" we allow; the higher $\beta$, the less latency we allow.*

is a generalization of the approaches described in (2) and (3). By taking e.g. $\alpha = \infty$ and $\beta = 1$ in (4), we obtain (2). In Figure 1, we show the solution $x$ of problem (4) as a function of the weights $\alpha$ and $\beta$ for the following example.

$$\begin{aligned} A &= [3 \;\; 4 \;\; 4.3 \;\; 3.4 \;\; 2 \;\; 1.09 \;\; -3 \;\; -2.54 \;\; -2.09 \;\; 0.89]^T \text{ and} \\ b &= [3.57 \;\; 4.80 \;\; 4.33 \;\; 3.93 \;\; 2.50 \;\; 2.05 \;\; -2.25 \;\; -1.99 \;\; -1.20 \;\; 1.52]^T. \end{aligned} \tag{5}$$

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.

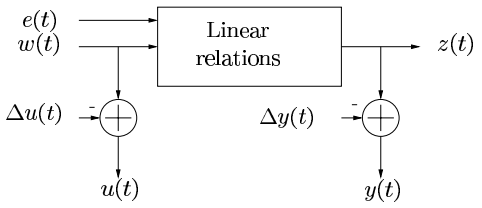



Fig. 2. *This figure illustrates the new model set of LTI dynamical systems. The observed data are $u(t)$ and $y(t)$. They are assumed to be corrupted versions of exact data, such that $w(t) = u(t) + \Delta u(t)$ and $z(t) = y(t) + \Delta y(t)$. There are also "unobserved" inputs $e(t)$, that "help explain"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

## III. 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, ... 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,...). The motivation for this framework is shown in Figure 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 (using the weights $\alpha$, $\beta$ and $\gamma$ in (7) we can vary the importance of the latency terms vs. the misfit terms). Models in this framework are of the form

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

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. $A(q), B(q)$ and $C(q)$ are polynomials of appropriate degree in the delay operator $q^{-1}$, with $A(q) = a_0 + a_1 q^{-1} + \ldots + a_{n_a} q^{-n_a}$, $B(q) = b_1 + b_2 q^{-1} + \ldots + b_{n_b} q^{-n_b+1}$ and $C(q) = c_0 + c_1 q^{-1} + \ldots + c_{n_c} q^{-n_c}$,

where we assume that $n_a$, $n_b$ and $n_c$ are user-defined. To select the best model from this new model set (6), we propose to minimize the following cost function:

$$J = \alpha J_{\text{output}} + \beta J_{\text{input}} + \gamma J_{\text{latency}} , \tag{7}$$

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 $\alpha, \beta, \gamma$ real positive scalars, specified by the user; let $M = max(n_a, n_b - 1, n_c) + 1$ and $m = N - M + 1$, then $t_1 = M - n_a$, $t_2 = M - n_b + 1$ and $t_3 = M - n_c$.
It is clear that, e.g., 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 $\alpha, \beta$ and $\gamma$ (they can be finite, infinite, or their corresponding term can be omitted in the model (6)). Table I 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 I).

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 $\alpha$, $\beta$ and $\gamma$, 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 a latency variable. 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 figure 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 equations (6) for $t = M, \ldots, N$ in the following matrix format: $Za - Wb - Ec = 0$, with $a = [a_{n_a}, \ldots, a_0]^T, b = [b_{n_b}, \ldots, b_1]^T, c = [c_{n_c}, \ldots, c_0]^T$, $Z \in \mathbb{R}^{m \times (n_a+1)}$ a Hankel matrix constructed from $\tilde{z} \equiv z(t_1 : N)$ ($Y$ is defined in a similar way w.r.t. $\tilde{y} \equiv y(t_1 : N)$), $W \in \mathbb{R}^{m \times n_b}$ a Hankel matrix constructed from $\tilde{w} \equiv w(t_2 : N-1)$ ($U$ is defined in a similar way w.r.t. $\tilde{u} \equiv u(t_2 : N-1)$) 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:

$$\begin{aligned} &\min_{\Delta y, \Delta u, e, a, b, c} \alpha J_{output} + \beta J_{input} + \gamma J_{latency} \\ &\text{such that } \; Za - Wb - Ec = 0. \end{aligned} \tag{8}$$

To avoid the trivial solution $a = 0$, $b = 0$ and $c = 0$, we need to impose extra constraints. We propose a set of linear constraints, which lead to ML solutions for the special cases like AR, ARMA, ... The constraints imposed are: $a_0 = 1$ and $c_0 = 1$. 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 chapter 7 of [2]). Using the latter equations, it is possible to transform (8) into the following optimization problem:

$$\begin{aligned} &\min_{a,b,c} [a^T \; b^T] \begin{bmatrix} Y^T \\ -U^T \end{bmatrix} \left( \frac{\mathbf{B_1}(a)}{\alpha} + \frac{\mathbf{B_2}(b)}{\beta} + \frac{\mathbf{B_3}(c)}{\gamma} \right)^{-1} [Y \; -U] \begin{bmatrix} a \\ b \end{bmatrix} \\ &\text{such that } a_0 = 1, c_0 = 1, \end{aligned} \tag{9}$$

where $\mathbf{B_1}(l) \equiv \mathbf{P_1}(l)\mathbf{P_1}(l)^T$, $\mathbf{B_2}(l) \equiv \mathbf{P_2}(l)\mathbf{P_2}(l)^T$, $\mathbf{B_3}(l) \equiv \mathbf{P_3}(l)\mathbf{P_3}(l)^T$ and

$$\begin{aligned} &\mathbf{P_1} : \mathbb{R}^{(n_a+1)\times 1} \to \mathbb{R}^{m \times (N-M+n_a+1)} : a \to \mathbf{P_1}(a), \\ &\text{defined by } \mathbf{P_1}(a)\tilde{z} = Za \\ &\mathbf{P_2} : \mathbb{R}^{n_b \times 1} \to \mathbb{R}^{m \times (N-M+n_b)} : b \to \mathbf{P_2}(b), \\ &\text{defined by } \mathbf{P_2}(b)\tilde{w} = Wb \\ &\mathbf{P_3} : \mathbb{R}^{(n_c+1)\times 1} \to \mathbb{R}^{m \times (N-M+n_c+1)} : c \to \mathbf{P_3}(c), \\ &\text{defined by } \mathbf{P_3}(c)\tilde{e} = Ec. \end{aligned}$$

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 $[a^T \; b^T \; c^T]^T$ (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 $a^T a = 1$, $c^T c = 1$ 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 formulated in (8) then becomes:

$$\begin{aligned} &\min_{a, \tilde{e}} \tilde{e}^T \tilde{e} \\ &\text{such that } Ya = \tilde{e}. \end{aligned} \tag{10}$$

| No | $(\alpha, \beta, \gamma)$ | 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 | $Y \to Z$<br>$U \to W$ | $Za + Wb = 0$ | M |
| 14 | (1,1,1) | ARMAX with noisy in/outputs | $Y \to Z$<br>$U \to W$ | $Za + Wb + Ec = 0$ | 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 | ($\infty$,$\infty$,$\infty$) | See Case 25 | | | |

TABLE I

*This table shows the different LTI dynamic systems of our general framework. Depending on the triplet $(\alpha, \beta, \gamma)$ 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.*

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 $a^T a = 1$. Problem (10) thus becomes:

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

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 indepently identically distributed (i.i.d.) zero-mean white Gaussian noise. For stable AR systems it is well-known [3] that the output will then be stationary too. As a consequence, the matrix $\frac{Y^T Y}{m}$ becomes Toeplitz in the limit for $m \to \infty$. Since it is already symmetric by construction, it belongs to the class of centrosymmetric[1] matrices. It should be clear that the $a$ that results from (11) is the eigenvector corresponding to the smallest eigenvalue of $\frac{Y^T Y}{m}$. Applying theorem 3.2.6 in [2] to problem (11) we can conclude that the model vector $a$ is centrosymmetric or skew centrosymmetric[2] and thus we find that the poles of the AR system, i.e. the roots of the following equation in $\zeta$

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

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

[1] 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.

[2] 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.

pole estimates are biased. One can show that with the lineair constraint $a_0 = 1$, the pole estimates are unbiased.

## IV. Properties and interpretations

In this section we discuss some properties and new interpretations 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. The second subsection shows that the misfit on the input, the misfit on the output and the latent variables satisfy themselves linear systems. Finally subsection IV-C indicates that it is possible to interprete the misfit versus latency framework discussed in this paper, as a pure latency framework applied to corrected input and output data matrices $U$ and $Y$. Proofs are omitted but can be found in [2].

### A. Orthogonality

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

*Property IV.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:

$$\tilde{w}^T(\tilde{w} - \tilde{u}) = 0.$$

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

*Property IV.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:

$$[(\tilde{z} - \tilde{y})^T \ \tilde{e}^T] K \begin{bmatrix} \tilde{z} \\ \tilde{e} \end{bmatrix} = 0.$$

### B. 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 IV.3:* If $n_a + 1 = n_b$, the input misfit and the output misfit satisfy the following LTI dynamic system equations:

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

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

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

If $n_b = n_c + 1$, the input misfit and the latent variables satisfy the following LTI dynamic system equations:

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

If in the previous property, the conditions on the orders are not satisfied, it is still possible to prove similar properties. If e.g. $n_b > n_a + 1$, a formula similar to (12) can be proven by defining a new $\check{\mathbf{P}}_1(a) \in \mathbb{R}^{m \times (N-M+n_b)}$, which is simply $\mathbf{P_1}(a)$ padded with $n_b - n_a - 1$ zero columns.

### C. Pure latency interpretation

In this subsection we demonstrate how the misfit versus latency framework can be interpreted as a pure latency framework.

*Property IV.4:* The misfit versus latency framework can be interpreted as a pure latency framework involving corrected output and input data matrices $Y_{corr}$ and $U_{corr}$ (for details see chapter 7 of [2]).

## V. Experiments

In this section we describe 2 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 $(\alpha, \beta, \gamma)$ 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 one end the PEM approach and on the other end the behavioral modeling approach.

The second experiment shows how 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 that of the classical PEM approach.

### A. Experiment 1

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

$$\begin{aligned} a &= [0.81 \ -1.2541 \ 1]^T \\ b &= [1 \ 2.3 \ 0.3]^T \\ c &= 1. \end{aligned}$$

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 to 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 figure 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 figure 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 $(\alpha, \beta, \gamma)$:

- case 1: $(\alpha, \beta, \gamma) = (10^5, 10^5, 1)$

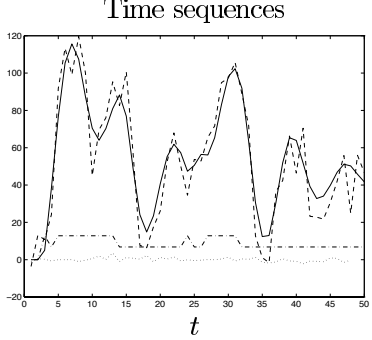



Fig. 3. *This figure shows the deterministic input $w(t)$ (dash-dotted line), the i.i.d. Gaussian noise input $e(t)$ (dotted line) of experiment* 1. *The full line is the sequence $z(t)$, obtained by using the ARMAX model and the initial conditions described in the first experiment. The dashed line is 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 2: $(\alpha, \beta, \gamma) = (10^3, 10^3, 10^2)$
- case 3: $(\alpha, \beta, \gamma) = (1, 1, 10^5)$.

Case 1 clearly corresponds to the ARMAX case (see case 26 of table I). Case 3 clearly corresponds to the dynamic TLS model (see case 15 of table I). 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 $(\alpha, \beta, \gamma)$, we plot in figure 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 ("+").

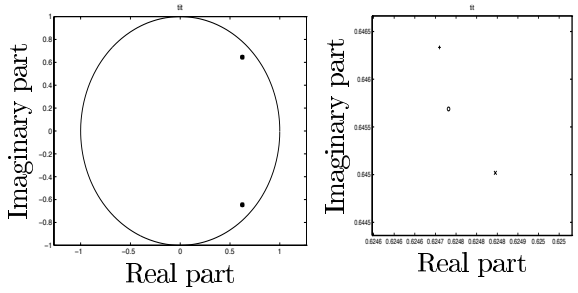



Fig. 4. *These figures show the root locus obtained by varying the parameters $\alpha$, $\beta$ and $\gamma$ in the model selection procedure (9) applied to the example described in experiment* 1.

### B. 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 figure 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 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 $\alpha$, $\beta$ and $\gamma$ is obviously based on the knowledge of the statistical properties of the latent 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 $\infty$).
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:

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

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

where $a$ is the exact AR parameter vector, $a_{mvl}^{(i)}$ is the AR parameter vector obtained using (9) in the $i$th Monte Carlo run and $a_{PEM}^{(i)}$ is the AR parameter vector obtained using the classical PEM approach in the $i$th Monte Carlo run. The relative accuracies obtained in this way are:

$$err_{mvl} = 1.84\% \text{ and } err_{PEM} = 35.75\%.$$

This clearly shows the potential of the new framework.

## VI. Conclusion

In this paper we have presented the misfit versus latency framework that was introduced in [7]. The latter framework combines the 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 $\alpha$, $\beta$ and $\gamma$ 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.

## References


[1] G.H. Golub and C.F. Van Loan. *Matrix Computations*. Johns Hopkins University Press, Baltimore, 1989.

[2] P. Lemmerling. *Structured total least squares: analysis,algorithms and applications*. PhD thesis, Dept. of Elect. Eng., ESAT-SISTA, Katholieke Universiteit Leuven, May 1999.

[3] L. Ljung. *System identification: theory for the user*. Prentice-Hall, Inc., 1987.

[4] B. De Moor. Structured total least squares and $l_2$ approximation problems. *Linear Algebra and its Applications, Special Issue on Numerical Linear Algebra Methods in Control, Signals and Systems, (Van Dooren, Ammar, Nichols, and Mehrmann, eds.)*, 188-189:163–207, July 1993.

[5] B. De Moor. Total least squares for affinely structured matrices and the noisy realization problem. *IEEE Transactions on Signal Processing*, 42:3104–3114, November 1994.

[6] B. De Moor. Linear system identification, structured total least squares and the riemannian svd. In S. Van Huffel, editor, *Recent Advances in Total Least Squares Techniques and Errors-in-Variables modeling*, pages 225–239. SIAM, 1997.

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

[8] B. De Moor and B. Roorda. $l_2$-optimal linear system identification: Dynamic total least squares for siso systems. In *Proc. of 33rd IEEE CDC*, pages 2874–2879, Florida, December 1994.

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

[10] B. Roorda and C. Heij. Global total least squares modelling of multivariable time series. *IEEE Transcations on Automatic Control*, 40(1):50–63, 1995.

[11] T. Söderström and P. Stoica. *System Identification*. Prentice Hall, 1989.

[12] S. Van Huffel and J. Vandewalle. *The Total Least Squares Problem: computational aspects and analysis*, volume 9. SIAM, Philadelphia, 1991.

[13] J. C. Willems. 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, 1986,1987.