89-39I

An Introduction to System Identification

Marc Moonen*, Bart De Moor
ESAT Katholieke Universiteit Leuven
K.Mercierlaan 94, 3030 Heverlee, Belgium
tel 016/22 09 31 telex 25941 elekul fax 32/16/221855

May 1989

1 Introduction

Inferring models from observations and measurements, and studying their properties, is one of the central issues in all scientific disciplines. **System identification** deals with the problem of building mathematical models of dynamical systems based on observed data from these systems.

A **system** is an object in which variables of different kinds interact. These variables can be divided in inputs, outputs, disturbances and states. Typically, *inputs* are those variables that can be manipulated and affect the system as external stimuli. *Outputs* are the direct observations. The *disturbances* can be divided into those that are directly measurable and those that are only observed through their influence on the output. Disturbances include measurement noise, uncontrollable inputs, etc. The *state* of a system is the minimal information that is needed to determine the output, once the inputs and disturbances are known.

Mathematical models are derived and applied for several reasons:

Simulation: Using mathematical models, one can analyse the behavior of a system via simulations, when experiments on the *real* system are too dangerous (nuclear power plants), too expensive (loss of production), too time consuming, too complicated or simply impossible (ecological systems).

Prediction: In some situations, one is interested in predicting the future behavior of a system, possibly under several different scenarios on the inputs and disturbance variables.

Optimal Filtering: A mathematical model can be used for obtaining information concerning variables that are not directly accessible or observable. This includes estimation of state variables via Luenberger or Kalman filtering.

Control Applications: Once a mathematical model of a system is available, one can develop controllers that achieve prespecified control tasks. A simple approach to the design of an automatic control system consists in combining a certain system identification scheme with any control law (Figure 1). The principle of using the estimated model as if it were the true system for the purpose of design, is called the *certainty equivalence principle.*

*supported by the Belgian National Fund for Scientific Research (N.F.W.O.)

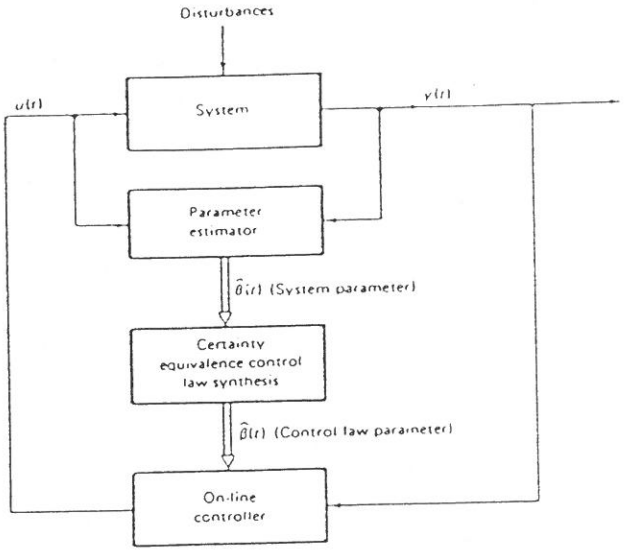

Figure 1: Certainty Equivalence Control Law

In this paper, a brief overview is given of conventional "black box identification" techniques. In **section 2**, we discuss mathematical models in general terms and present models that are suited for system identification. In **section 3**, the system identification experiment set up is discussed. *Input-output models* (difference equations) are mostly used for simple single input—single output systems. **Section 4** deals with identifying such models. These techniques equally well apply to more complex multivariable systems, but the required parametrizations become hardly elegant and lead to numerically ill conditioned computations. Therefore, one should then preferably make use of *state space models* instead of I/O-models. In **Section 5** the identification of state space models is surveyed, while finally in **section 6** a few applications of these schemes on industrial plants are presented.

2 Mathematical Models.

Mathematical models may be phrased with varying degrees of mathematical formalism. A rough classification can be obtained from the following list of qualifications:

lumped/distributed

discrete/continuous time

linear/non-linear

time-invariant / time-varying

In this survey, we shall employ only

lumped, discrete time, linear, time-invariant models.

The preference for this model class is dictated by several reasons:

The **lumpedness** arises from the fact that in most cases, the *sensors* only collect *local* measurements that "sample" the system only in the immediate neighbourhood of the

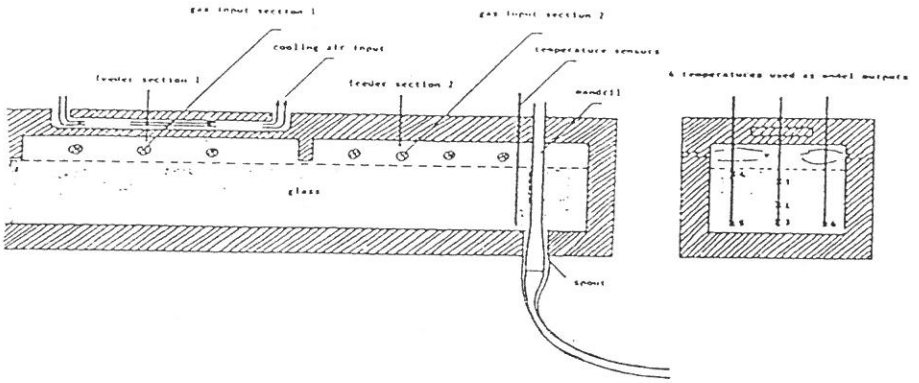

Figure 2: Cross section of a feeder

sensor. As an example, consider the temperature measurement in a glass feeder by using several sensors (Figure 2). The inputs correspond to heating and ventilation, while the outputs are the temperatures at the very locations of the sensors. This system is certainly a *distributed parameter system*, the mathematical model of which is in terms of partial differential equations, when derived from physical laws. Yet, the several sensors represent a spatial *discretization* of the system by a 3-dimensional grid of temperature measurements. Hence the corresponding mathematical model becomes lumped.

While for most physical systems it is most natural to work with a continuous time representation (e.g. differential equations), the increasing use of *digital computers*, enforces the use of **discrete time models**. Mathematically, it is possible to convert any continuous time behavior into discrete time under fairly general conditions (while the reverse is not necessarily true).

For *non-linear* systems, we shall exploit the idea of **local linearization in an operating point**. The behavior is considered to originate in a linear system, within an observation window of finite length. Obviously, the quality and reliability of the derived model will depend upon the relation between the length of the finite window, the number of observations (the sampling rate) and the time constants that characterize the time-variance within this finite window. The restriction to linearity is a self-imposed limitation to the kind of mathematical operations and devices that will be used. Not only is the theory of linear models well developed, the algebraic and numerical tools that are needed are abundantly avalaible (and frequently reduce to the solution of a set of (overdetermined) linear equations, or to (generalized) eigenvalue problems).

Since in most practical situations, the observation-window is of finite length and the number of observations is finite as well, the behavior of the system can be approximated sufficiently well within this finite window by a **time invariant system**. Also, one can develop *adaptive strategies*, that update the model from one time window to another with only few additional computations.

3 System identification experiment set up

The construction of a model from data involves three basic steps:

1. collecting useful data,
2. chosing a convenient model set,
3. computing the (best) model within the model set, possibly following a certain identification criterion.

The Data

The acquisition of 'good' data, is not at all a trivial task. The following issues should be kept in mind:

Determination of inputs and outputs: The appropriate choice of variables that will be measured, may be determined by the ultimate goal of the model.

Choice of the input signals: In some cases, experiments on the real plant are impossible and data should be obtained from *normal operating records*. In other cases, one can freely chose the input sequences. In any case, for a reliable identification, the inputs should satisfy necessary conditions. A formal description of this is in terms of the concept of *persistancy of excitation*. The inputs should be *sufficiently rich* such that all modes of the system are excited and observable in the output sequence. As a rule of thumb, an input signal should at least contain n different sinusoids, in order to identify an n-th order system.

Data sampling rate: Data sampling is inherent in computer based data acquisition systems. It is unavoidable that sampling as such leads to information losses and it is important to select the sampling instances such that these losses are insignificant. Typically (and most effectively), sampling is carried out at equidistant sampling instants. In principle, if sampling is performed at a sampling frequency f_s, no information is 'lost' as far as frequency components are concerned below the so-called Nyquist frequency $f_s/2$. Hence, in order to avoid distortion (*aliasing* or *frequency folding*), one should apply analog *anti-aliasing (pre-sampling) filtering*, in order to eliminate all high-frequency components above the Nyquist frequency. Reversely, the sampling rate should be chosen in principle twice the highest frequency of interest. However, in practical cases, one often uses sampling rates that are 4 to 10 times higher than the minimal frequency of interest. The redundancy that is introduced in this way, can be used for further digital data preprocessing (see below). A detailed analysis and practical guidelines for appropriate determination of the sampling rate, may be found in [1, p.29, p.71] [10, p.385-386].

Data preprocessing: In a lot of applications, it is necessary to 'clean' the measurement data before any identification algorithm can be applied. Preprocessing includes the elimination of occasional bursts and outliers, 'peak shaving', trend removal, estimation of drift and offset, periodical interference, the analysis of disturbances, such as day-night phenomena etc... Useful guidelines and algorithms can be found in [2].

Estimation of time delays: Time delays are common industrial processes. As an example, consider the measurement of the tube wall thickness of a glass tube after shaping. This can only be measured with suffucient accuracy if the tube itself is sufficiently cooled. This introduces a considerable time delay. A delay of one sampling period highers the system order with one, introducing an additional pole equal to zero. Therefore, delays should be avoided and compensated as much as possible, by shifting the output data accordingly. Delays can be estimated via a physical investigation of the origin of the delays, via cross-correlation techniques or from inspection of the impulse responses. Details can be found in [1, p.42].

The Model Set

A set of candidate models is obtained by specifying within which collection of models we are going to look for a suitable one. This is no doubt the most important and, at the same time, often the most difficult choice of any identification procedure. As much as possible, any *a priori available information* on the system, should be reflected in the choice of a certain model. If for instance, certain *physical* laws are known to hold true for the system, one could impose a certain equation structure and identify the unknown physical parameters. In other cases, standard linear models may be employed, without reference to the physical background (*black box* approach). In this paper, we shall briefly review two possible black box models, viz. **input-output models (section 4)** and **state space models (section 5)**, and show how such models can be identified from input-output data.

The Identification Step

Having determined the set of models, one should determine within this set, the model that is the 'best' approximation or provides the 'best' explanation of the observed data. The assessment of model quality is typically based upon how the models perform when they attempt to reproduce the measured data. Typically, it is desirable to have models that are as *simple* as possible, yet that at the same time *explain* as much as possible of the observed data, i.e. that minimize the *misfit*. These two requirements are in a certain sense conflicting: Intuitively, it is obvious that a simple model will not be able to explain or simulate complex behaviors while a complex model will explain a lot but will be difficult to identify or to use appropriately.

While the system identification procedure has the logical flow just described (collect data, fix a model set, pick out the best model), it is possible that the model does not pass the validation test, so that several steps in the identification procedure have to be revised. The resulting *system identification loop* is depicted in Figure 3.

For this reason, good interactive software is an important tool for handling the interactive character of the problem. The qualification 'good' implies the availibility of graphic possibilities, the guarantee of numerical reliablitity and acceptable levels of computational speed and memory requirements. As an example of such software packages, let us mention *MATLAB* [11] and *SIMNON* [1], *Matrix-X*, *Control C*, among others, besides the more classical software libraries like NAG, LINPACK and EISPACK.

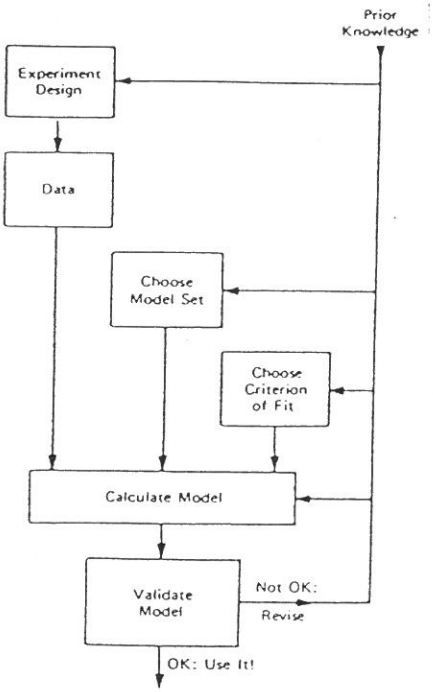

Figure 3: The system identification loop.

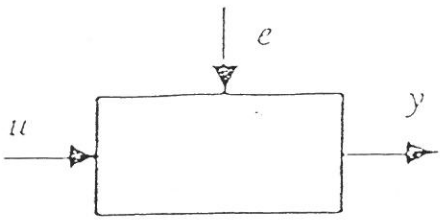

Figure 4: single input–single output black box.

4 Input-output Models

Preliminaries

The basic input-output description of a single input–single output linear system with an additive disturbance (Figure 4) is:

$$y_k = G(q)u_k + H(q)e_k$$

where the following notations are applied:

y_k is the output at time k, u_k is the input, e_k is a sequence of independent random variables (white noise) with zero mean and (mostly Gaussian) probability density function $f_e(.)$.

q is the *forward shift operator*

$$\begin{aligned} qu_k &= u_{k+1} \\ q^{-1}u_k &= u_{k-1} \end{aligned}$$

$G(q)$ and $H(q)$ represent the dynamic relation between respectively input and output, and disturbance and output.

$$\begin{aligned} G(q) &= \sum_{k=1}^{\infty} g(k)q^{-k} \\ H(q) &= 1 + \sum_{k=1}^{\infty} h(k)q^{-k} \end{aligned}$$

The rationale behind this model is the following: The output sequence y_k is being thought of as to originate from the parallel connection of two systems: The first system models the causal, dynamic dependence of the outputs on the observed or predetermined inputs. Everything that can not be explained by this first system, is then contributed to the second system (disturbance). It turns out that the disturbance model $H(q)e_k$, where $H(q)$ is a linear filter, and e_k is a white noise sequence, provides enough flexibility for most practical applications.

In the above description, a few parameters might be unknown and thus subject to identification. If we summarize these into an unknown parameter vector θ, the system description reads

$$y_k = G(q,\theta).u_k + H(q,\theta).e_k$$

The most obvious parametrization is in terms of rational functions

$$\begin{aligned} G(q) &= \frac{B(q)}{A(q)} \\ &= \frac{b_1 q^{-1} + b_2 q^{-2} + \ldots + b_{n_b} q^{-n_b}}{1 + a_1 q^{-1} + a_2 q^{-2} + \ldots + a_{n_a} q^{-n_a}} \\ H(q) &= \frac{D(q)}{C(q)} \\ &= \frac{1 + d_1 q^{-1} + d_2 q^{-2} + \ldots + d_{n_d} q^{-n_d}}{1 + c_1 q^{-1} + c_2 q^{-2} + \ldots + c_{n_c} q^{-n_c}} \end{aligned}$$

where the parameter vector then equals

$$\theta = \left[\begin{array}{cccccccccccc} a_1 & \ldots & a_{n_a} & b_1 & \ldots & b_{n_b} & c_1 & \ldots & c_{n_c} & d_1 & \ldots & d_{n_d} \end{array} \right]^t$$

Several special cases may arise:

linear difference equation:

$$A(q)y_k = B(q)u_k$$

ARX (equation error) model structure:

$$A(q)y_k = B(q)u_k + e_k$$

The additive white noise term acts as a direct error in the linear difference equation. These models are called ARX, where AR refers to the *autoregressive part* $A(q)y_k$, and X refers to the *eXogeneous* input $B(q)u_k$. Alternatively, the ARX description can be written as

$$y_k = \frac{B(q)}{A(q)}u_k + \frac{1}{A(q)}e_k$$

from which it follows that the white noise term is filtered through the the denominator $A(q)$. Although this might rarely correspond to reality, ARX models are frequently used, as together with a quadratic identification criterion (see below), they give rise to a set of *linear equations* that can readily be solved, whereas in the general case, more complex numerical optimisation techniques are necessary.

ARMAX model structure: In order to allow for more flexibility in the description of the disturbance term, the ARX model can be extended to an ARMAX model, where MA refers to the *moving average* term $C(q)e_k$

$$A(q)y_k = B(q)u_k + C(q)e_k$$

output error model structure:

$$y_k = \frac{B(q)}{A(q)}u_k + C(q)e_k$$

Box Jenkins model structure: This is the most general case,

$$y_k = \frac{B(q)}{A(q)}u_k + \frac{C(q)}{D(q)}e_k$$

For most practical applications however, this structure is much too complex, and requires too many computations when being identified.

The presented models can be generalized towards **multivariable systems** (systems with several inputs and outputs). Instead of using rational functions, one should then employ rational matrices, i.e. matrices of which the elements are rational functions of the shift operator q. While these models can be embedded in a generalizing approach [7] [10], they frequently lead to numerically unstable computations because of the ill conditioning of certain canonical parametrizations. When dealing with multivariable systems (as well as high-order monovariable systems), one should preferably make use of **state space models** (to be presented in section 5).

Finally, let us mention that in general, data-aided model structure selection is a largely underdeveloped research field for input-output models. The choice of an appropriate model structure, as well as the determination of optimal polynomial degrees (n_a, n_b, etc.), although crucial for a successful identification, is mostly a matter of trial and error (see [10] for details).

Identification of input-output models.

From the general description, one can formally compute the **prediction error** e_k

$$e_k = H^{-1}(q,\theta).[y_k - G(q,\theta)u_k]$$

For a given I/O data set $u_1, u_2, \ldots, u_N$ and $y_1, y_2, \ldots, y_N$, and for each specific choice for the parameter vector θ, one can compute a series of prediction errors $e_1(\theta), e_2(\theta), \ldots, e_N(\theta)$. The **optimal choice** for θ in a way minimizes these prediction errors. In the sequel, we will

only discuss *prediction error methods.* Related *correlation methods* (like e.g. instrumental variables methods) will not be discussed for the sake of brevity.

A mathematical description of the **prediction error methods** is a follows. First of all, one can apply a linear filter $L(q)$ to the previously computed prediction errors

$$e_k^f(\theta) = L(q)e_k(\theta)$$

As will be demonstrated, this imposes a frequency weighting on the misfit.
Making use of a predifined norm

$$V(\theta) = \frac{1}{N}\sum_{k=1}^{N} l(e_k^f(\theta))$$

where $l(.)$ is a scalar (positive) function, the optimal θ is chosen to be the one that minimizes this norm

$$\theta_{opt} = \arg_\theta \min V(\theta)$$

The most obvious choice for $l(.)$ is a quadratic function

$$l(e) = \frac{1}{2}e^2$$

Prediction error methods can then elegantly be interpreted in the frequency domain. It turns out that for this case the specified norm approximately equals

$$V(\theta) \cong \frac{1}{2\pi}\int_{-\pi}^{\pi} \frac{1}{2}|G_0(e^{j\omega}) - G(e^{j\omega},\theta)|^2 . \frac{|U(\omega)|^2}{|H(e^{j\omega},\theta)|^2}|L(e^{j\omega})|^2 d\omega$$

where

$G_0(q)$ is the (unknown) exact transfer from u_k to y_k

$G(q,\theta)$ and $H(q,\theta)$ are the modelled transfers

$U(\omega)$ is the spectrum of the input signal u_k

As $H(e^{j\omega},\theta)$ corresponds to the modelled noise spectrum, a prediction error method can thus be interpreted as an optimal "fitting" in the frequency domain of the modelled transfer $G(q,\theta)$ onto the exact transfer $G_0(q)$, with a frequency weighting that for each frequency equals the signal–to–noise ratio for that frequency, modified by the prefilter $L(q)$. Prediction error methods are therefore closely related to spectrum analysis methods.

Many different prediction error methods can be devised, combining different choices for the model set, the filter $L(q)$, the criterion function $l(.)$, and sometimes also the numerical technique, used to solve the optimization problem. In the sequel, the most common techniques, viz. the *Least Squares Method* and the *Maximum Likelihood Method*, will be discussed.

Least Squares method for ARX models :

A quadratic optimisation criterion $l(e) = \frac{1}{2}e^2$ applied to an ARX model set, gives rise to a simple (overdetermined) set of linear equations, that can readily be solved (hence the popularity of ARX models). ARX models can be written as

$$y_k = -a_1 y_{k-1} - \ldots - a_{n_a} y_{k-n_a} + b_1 u_{k-1} + \ldots + b_{n_b} u_{k-n_b} + e_k$$

The *parameter vector* θ was defined as

$$\theta = [a_1\ a_2 \ldots a_{n_a}\ b_1\ b_2 \ldots b_{n_b}]^t$$

If we also define the *regression vector* φ_k as follows

$$\varphi_k = [-y_{k-1} \ldots - y_{k-n_a}\ u_{k-1} \ldots u_{k-n_b}]^t,$$

one can easily prove that the identification problem reduces to solving (in a least squares sense) an overdetermined set of linear equations

$$\begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_N \end{bmatrix} = \begin{bmatrix} \varphi_1^T \\ \varphi_2^T \\ \vdots \\ \varphi_N^T \end{bmatrix} .\theta$$

with an optimal solution

$$\theta_{opt} = [\frac{1}{N}\sum_{k=1}^{N} \varphi_k . \varphi_k^T]^{-1} \frac{1}{N}\sum_{k=1}^{N} \varphi_k . y_k$$

The Least Squares method is extremely simple in a sense that the function that needs to be optimized, has only one minimum, that can readily be computed, without making use of numerical optimization techniques. A major drawback of course is its limited applicability (only ARX model). Correlation methods (like e.g. instrumental variable methods) in a way extend this approach to ARMAX models (see [10] for details)

Maximum Likelihood Methode :

So far, we did not make use of the probability density function $f_e(.)$. Alternatively, the optimal choice for θ can be defined as the one for which the available data u_k and y_k correspond to a "highest probability" for the sequence of prediction errors $e_1(\theta), e_2(\theta), \ldots, e_N(\theta)$.

A mathematical description is as follows. Again, from the general I/O-description, one can formally compute the sequence of prediction errors, for each specific choice for θ

$$e_k = H^{-1}(q,\theta).[y_k - G(q,\theta)u_k]$$

The combined probability for this sequence can be computed from the probability distribution function $f_e(.)$

$$f(e_1(\theta), e_2(\theta), \ldots, e_N(\theta)) = \prod_{k=1}^{N} f_e(e_k(\theta))$$

The optimal choice for θ then equals

$$\theta_{opt} = \arg_\theta \max \prod_{k=1}^{N} f_e(e_k(\theta))$$

The maximum of this function also maximizes its logarithm

$$\begin{aligned}\theta_{opt} &= \arg_\theta \max \frac{1}{N} \log(\prod_{k=1}^{N} f_e(e_k(\theta)) \\ &= \arg_\theta \max \frac{1}{N} \sum_{k=1}^{N} (\log f_e(e_k(\theta)) \\ &= \arg_\theta \min \frac{1}{N} \sum_{k=1}^{N} l(e_k(\theta)\end{aligned}$$

which thus corresponds to a prediction error criterion, where

$$l(e_k(\theta)) = -\log f_e(e_k(\theta))$$

The Maximum Likelihood Method thus indeed belongs to the family of prediction error methods. Furthermore, if the distribution $f_e(.)$ is Gaussian, with zero mean and covariance λ, one can prove that

$$l(e_k, \theta) = \log f_e(e_k(\theta)) = const + \frac{1}{2} \log \lambda + \frac{1}{2} \frac{\epsilon^2}{\lambda}$$

If λ is a known constant, this reduces to a quadratic criterion. For ARX models for instance, one will then again compute the Least Squares Solution.

Finally, let us mention that the Maximum Likelihood Method is mostly used because of its optimal asymptotic properties ($N \to \infty$, where N is the number of observations). A drawback stems form the use of $f_e(.)$ which is often unknown, whereas small changes in $f_e(.)$ can introduce major changes in the identification results.

Recursive Identification.

In many cases, it is necessary or useful to have a model of the system available while the system is in operation, typically when the system to be modelled is non-linear or time-varying. In these applications, one will perform an on-line adjustment of the model each time a new measurement becomes available. Such identification techniques are called *recursive*. They exploit in the model adjustment, as much as possible the already obtained model and update the new one by a minimal modification. As an example, a Recursive Least Squares Procedure can be described as follows (see [10] [1] [7] for details).

$$\hat{\theta}_k = \hat{\theta}_{k-1} + L_k(y_k - \varphi_k \hat{\theta}_{k-1})$$

with

$$\begin{aligned}L_k &= \frac{P_{k-1}\varphi_k}{\lambda + \varphi_k^t P_{k-1} \varphi_k} \\ P_k &= \frac{1}{\lambda}(P_{k-1} - \frac{P_{k-1}\varphi_k\varphi_k^t P_{k-1}}{\lambda + \varphi_k^t P_{k-1}\varphi_k})\end{aligned}$$

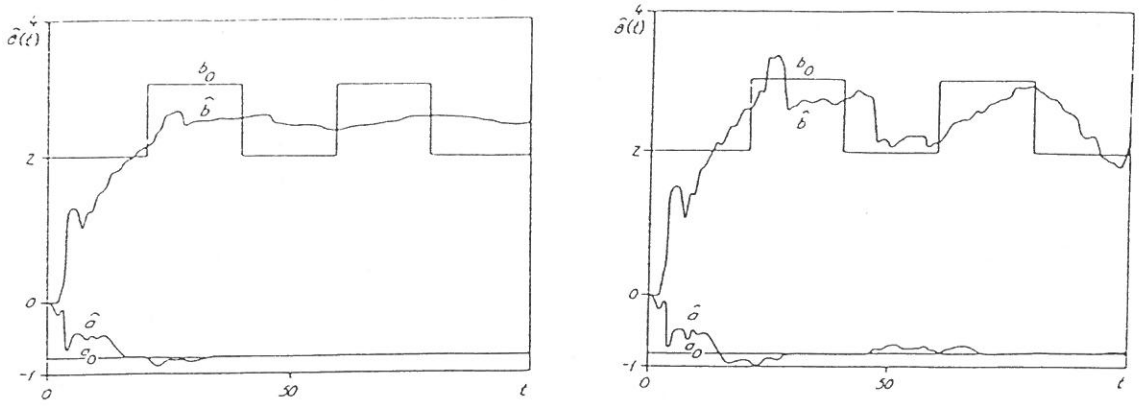

Figure 5: Adaptivity and forgetting factor

At each time instance k, the parameter vector θ is updated by modifying the previous value. The modification is determined by the *gain* L_k, which weights the amount by which the prediction error $y_k - \varphi_k \hat{\theta}_{k-1}$ will affect the update. P_k is the covariance matrix of the parameter vector, that reflects the confidence in the parameter estimation, and can be determined recursively from the previous estimation of the covariance matrix, taking into account a 'new' regression vector φ_k, as given by the above formula. Finally, λ is a *forgetting factor*, which is chosen as $0 < \lambda < 1$. It represents an *exponential weighting*, by which older measurements are discounted. If λ is close to 1, the estimation will be *noise insensitive*, while if it is close to 0, the estimation of parameters will adopt itself much faster to modifications of the system. Hence, an appropriate choice of λ will always be a matter of *compromise* between *fast adaptivity* and *noise insensitivity*.
An example is given in Figure 5. The first figure represents the identification of two parameters (one constant and one time varying) with no forgetting. For the second figure, the forgetting factor is $\lambda = 0.9$.

5 State Space models

As already mentioned, I/O-models give rise to ill conditioned mathematical problems, when applied to complex systems (multivariable or high order monovariable). In these cases, one should preferably use **state space models**. In this section, both direct and indirect state space identification methods are dealt with.

Preliminaries

A lumped linear time-invariant discrete time system can be represented by a state space model of the form:

$$\begin{aligned} x_{k+1} &= Ax_k + Bu_k + w_k \\ y_k &= Cx_k + Du_k + v_k \end{aligned}$$

The first equation is referred to as the state equation while the second one is called the output equation. u_k is a vector with m components, that contains the m input observations at time k. The vector y_k contains the l outputs. The vector x_k contains the n states at time k. The vectors v_k and w_k both represent disturbances. v_k is referred to as *measurement noise* while w_k is called the *process noise*. However, if one wants to take into account these noise contributions, one pretty soon runs into unsolved mathematical problems Therefore, the noise contributions are mostly left out at the outset, where it is assumed that their influence is negligible. The state space identification schemes to be presented then in the first place apply to the simplified (noise-free) model

$$\begin{aligned} x_{k+1} &= Ax_k + Bu_k \\ y_k &= Cx_k + Du_k \end{aligned}$$

One can then prove that the identification schemes deliver *consistent* results under certain assumptions regarding the noise contributions, so that again the *best model* is obtained (although in a slightly different sense then before).

A state space model is not unique with respect to an observed input-output behavior. Indeed, it is easily verified that an 'equivalent' state space model is obtained by inserting a non-singular matrix T as follows:

$$\begin{aligned} Tx_{k+1} &= (TAT^{-1})(Tx_k) + (TB)u_k \\ y_k &= (CT^{-1})(Tx_k) + Du_k \end{aligned}$$

Alternatively (with $z_k = Tx_k$)

$$\begin{aligned} z_{k+1} &= A^\star z_k + B^\star u_k \\ y_k &= C^\star z_k + D^\star u_k \end{aligned}$$

The matrices have been changed but the input-output pairs were not affected. This implies that only for very specific choices of coordinate systems in the vector space of states, the components of the state vectors have physical meaning.

The identification techniques to be described essentially exploit three basic techniques from numerical linear algebra:

1. The solution of a set of linear equations
2. The eigenvalue decomposition
3. The singular value decomposition (SVD)

While the first two of them may be qualified as classical, the third one was introduced only recently as a practical numerical tool by the development of an efficient algorithm ([6] and the references therein). This important decomposition is introduced, and briefly analysed in an appendix.

Indirect identification schemes differ from direct schemes in a sense that first an impulse response (Markov parameters) is to be computed, whereafter the system matrices are computed from the SVD of a block Hankel matrix constructed with these Markov parameters. With **direct methods** on the other hand, the system matrices are computed directly from the SVD of a block Hankel matrix constructed with input-output data.

Indirect State Space Identification techniques

An indirect identification scheme consists in two steps. First, an approximate impulse response is computed. Next, the system matrices are identified making use of the computed Markov parameters.

First step : Deconvolution .

The output of a linear system can be computed as a convolution of the input, with the impulse response $H_0, H_1, H_2, \ldots$ ($H_i = CA^{i-1}B$, $i > 0$ is an $l \times m$ matrix, where m is the number of inputs, l is the number of outputs, whereas $H_0 = D$.)

$$y_k = \sum_{i=0}^{\infty} H_i . u_{k-i}$$

If we assume that $H_i < \epsilon$, $i > K$, where ϵ is a small number (for a *stable* system), the convolution sum approximately equals

$$y_k \simeq \sum_{i=0}^{K} H_i . u_{k-i}$$

Therefore, if a set of I/O-data is available, the sequence of Markov parameters can approximately be computed from an (overdetermined) set of linear equations (*deconvolution*)

$$\begin{bmatrix} u_K & u_{K-1} & \cdots & \cdots & \cdots & u_0 \\ u_{K+1} & u_K & \cdots & \cdots & \cdots & u_1 \\ u_{K+2} & u_{K+1} & \cdots & \cdots & \cdots & u_2 \\ \cdots & \cdots & \cdots & \cdots & \cdots & \cdots \\ u_{K+i-1} & u_{K+i-2} & \cdots & \cdots & \cdots & u_{i-1} \end{bmatrix} . \begin{bmatrix} H_0 \\ H_1 \\ H_2 \\ \vdots \\ H_K \end{bmatrix} \simeq \begin{bmatrix} y_K \\ y_{K+1} \\ y_{K+2} \\ \vdots \\ y_{K+i-1} \end{bmatrix}$$

(Note that the general deconvolution problem for unstable systems remained unsolved so far.)

Second step : Realization .

Once a sequence of Markov parameters is known, the system matrices can be computed as follows. Matrix D is known to be the first Markov parameter

$$D = H_0$$

In order to compute A, B and C, a block Hankel matrix $H_{p,q}$ is constructed (where the block dimensions p and q are chosen to be larger than the systeem order n).

$$H_{p,q} = \begin{bmatrix} H_1 & H_2 & H_3 & \cdots & H_q \\ H_2 & H_3 & H_4 & \cdots & H_{q+1} \\ H_3 & H_4 & H_5 & \cdots & H_{q+2} \\ H_4 & H_5 & H_6 & \cdots & H_{q+3} \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ H_p & H_{p+1} & H_{p+2} & \cdots & H_{q+p-1} \end{bmatrix}$$

One can then prove that **the rank of $H_{p,q}$, equals the system order n.** If the Markov parameters are inaccurate, due to noise on the data, this matrix will have full rank. The system order can then be estimated from the singular value spectrum, that reveals kind of an effective matrix rank (see appendix).

From $H_i = CA^{i-1}B$ it then follows that

$$\begin{aligned} H_{p,q} &= \begin{bmatrix} C \\ CA \\ CA^2 \\ \dots \\ CA^{p-1} \end{bmatrix} . \begin{bmatrix} B & AB & A^2B & \dots & A^{q-1}B \end{bmatrix} \\ &= \Gamma_p . \Delta_q \end{aligned}$$

where Γ_p and Δ_q are the observability and controlability matrices. Similarly, the *shifted* matrix $H^*_{p,q}$

$$H^*_{p,q} = \begin{bmatrix} H_2 & H_3 & H_4 & \dots & H_{q+1} \\ H_3 & H_4 & H_5 & \dots & H_{q+2} \\ H_4 & H_5 & H_6 & \dots & H_{q+3} \\ H_5 & H_6 & H_7 & \dots & H_{q+4} \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ H_{p+1} & H_{p+2} & H_{p+3} & \dots & H_{q+p} \end{bmatrix}$$

satisfies

$$H^*_{p,q} = \Gamma_p . A . \Delta_q$$

so that the system matrix A in principle can be computed from

$$A = \Gamma_p^+ . H^*_{p,q} . \Delta_q^+ \qquad (*)$$

(where M^+ is the pseudo-inverse of M.)
One can now prove that Γ_p and Δ_q can be computed from the SVD $H_{p,q} = U.\Sigma.V^T$

$$\begin{aligned} \Gamma_p &= U.\Sigma^{1/2} \\ \Delta_q &= \Sigma^{1/2}.V^t \end{aligned}$$

Finally, B and C equal the first block column and the first block row in Δ_q and Γ_p respectively, whereas the system matrix A equals

$$A = \Sigma^{-1/2} . U^t . H^*_{p,q} . V^T . \Sigma^{1/2}$$

This algorithm is due to **Zeiger and McEwen.** A major drawback is the computational complexity, that for instance seems to rule out an elegant adaptive implementation. Furthermore, if the available I/O-data are noisy, it is not clear in which sense this algorithm delivers the "best" model (if it does!). Direct methods seem to offer a reasonable alternative in this respect.

Direct State Space Identification techniques

The direct identification approach for state space models will be based upon two basic properties. The first property allows for an estimation of the system order, while the second one shows how to obtain a state vector sequence. Once a state vector sequence is known, the system matrices (and the disturbance vectors w_k and v_k) follow from the solution of a set of linear equations.

Let us consider the measurement vectors m_k that are constructed from input-output pairs by simple concatenation:

$$m_k = \begin{pmatrix} u_k \\ y_k \end{pmatrix}$$

and construct two block Hankel matrices with these input-output pairs:

$$H_1 = \begin{pmatrix} m_k & m_{k+1} & m_{k+2} & m_{k+3} & \cdots & \cdots & m_{k+j-1} \\ m_{k+1} & m_{k+2} & m_{k+3} & m_{k+4} & \cdots & \cdots & m_{k+j} \\ m_{k+2} & m_{k+3} & m_{k+4} & m_{k+5} & \cdots & \cdots & m_{k+j+1} \\ m_{k+3} & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots \\ \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots \\ m_{k+i-1} & m_{k+i} & \cdots & \cdots & \cdots & \cdots & m_{k+i+j-2} \end{pmatrix}$$

$$H_2 = \begin{pmatrix} m_{k+i} & m_{k+i+1} & m_{k+i+2} & m_{k+i+3} & \cdots & \cdots & m_{k+i+j-1} \\ m_{k+i+1} & m_{k+i+2} & m_{k+i+3} & m_{k+i+4} & \cdots & \cdots & m_{k+i+j} \\ m_{k+i+2} & m_{k+i+3} & m_{k+i+4} & m_{k+i+5} & \cdots & \cdots & m_{k+i+j+1} \\ m_{k+i+3} & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots \\ \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots \\ m_{k+2i-1} & m_{k+2i} & \cdots & \cdots & \cdots & \cdots & m_{k+2i+j-2} \end{pmatrix}$$

(The *block dimensions i and j* are user determined and should be chosen 'sufficiently large', see [3] for more detail.)

H_1 is called the *past input-output block Hankel matrix* while H_2 is the *future input-output block Hankel matrix.* They are both submatrices of the 'large' *input-output block Hankel matrix* H:

$$H = \begin{pmatrix} H_1 \\ H_2 \end{pmatrix}$$

The following theorem allows to estimate the order of the system from the singular values of the input-output block Hankel matrices.

Theorem 1 Rank property
Under fairly general conclusions, it holds that:

$$rank(H_1) = rank(H_2) = mi + n$$

and

$$rank(H) = 2mi + n$$

where n is the system order, m is the number of inputs, and i is the number of block rows in H_1 and H_2

For a proof and a detailed analysis of the necessary conditions, the reader is referred to [3].

The next theorem demonstrates how an appropriate state vector sequence can be computed from the past and future input-output block Hankel matrices.

Theorem 2 The state as the intersection of past and future.
Let the state vector sequence $\mathcal{X}$ be defined as

$$\mathcal{X} = [x_{k+i}\ x_{k+i+1}\ \cdots\ x_{k+i+j-1}]$$

then, under certain conditions

$$\text{span}_{\text{row}}(\mathcal{X}) = \text{span}_{\text{row}}(H_1) \cap \text{span}_{\text{row}}(H_2)$$

so that any basis for this intersection constitutes a valid state vector sequence $\mathcal{X}$ with the basis vectors as the consecutive row vectors.

Again the SVD of H allows for an elegant computation of the required intersection

$$\begin{aligned} H &= \begin{bmatrix} H_1 \\ H_2 \end{bmatrix} \\ &= U_H.S_H.V_H^t \\ &= \begin{bmatrix} U_{11} & U_{12} \\ U_{21} & U_{22} \end{bmatrix} \cdot \begin{bmatrix} S_{11} & 0 \\ 0 & 0 \end{bmatrix} .V_H^t \end{aligned}$$

$$\begin{aligned} \dim(U_{11}) &= (mi + li) \times (2mi + n) \\ \dim(U_{12}) &= (mi + li) \times (2li - n) \\ \dim(U_{21}) &= (mi + li) \times (2mi + n) \\ \dim(U_{22}) &= (mi + li) \times (2li - n) \\ \dim(S_{11}) &= (2mi + n) \times (2mi + n) \end{aligned}$$

From

$$U_{12}^t.H_1 = -U_{22}^t.H_2$$

it follows that the row space of $U_{12}^t.H_1$ equals the required intersection.

Once $\mathcal{X} = [x_{k+i}\ x_{k+i+1}\ \cdots\ x_{k+i+j-1}]$ is known, the system matrices can be identified by solving an (overdetermined) set of linear equations:

$$\begin{bmatrix} x_{k+i+1} & \cdots & x_{k+i+j-1} \\ y_{k+i} & \cdots & y_{k+i+j-2} \end{bmatrix} = \begin{bmatrix} A & B \\ C & D \end{bmatrix} \cdot \begin{bmatrix} x_{k+i} & \cdots & x_{k+i+j-2} \\ u_{k+i} & \cdots & u_{k+i+j-2} \end{bmatrix}$$

The above results constitute the heart of a two-step identification scheme. First a state vector sequence is realized as the intersection of the row spaces of two block Hankel matrices, constructed with I/O-data. Then the system matrices are obtained at once from the least squares solution of a set of linear equations.

This identification procedure is proven to be **consistent** if the number of columns in H tends to infinity and if the input-output measurements are corrupted with additive white measurement noise, or in other words, if the columns in H are subject to independently and identically distributed errors with zero mean and common error covariance matrix equal to the identity matrix, up to a factor of proportionality. Furthermore, elegant adaptive implementations, based on SVD-updating, are conceivable (see [4] for details).

6 Model Validation and Examples

Model validation is concerned with the question whether the model is 'good enough' for its purpose. For instance, if the analysed system is known to be time-invariant and approximatively linear, then one expects at least that the model will sufficiently well *predict* the behavior when inputs are applied. Hence a validation criterion could be the difference between the observed and the simulated output. Other measures could be formulated in terms of frequency behavior (amplitude and the phase spectrum), etc. See [10] [1] [7]) for details. In this section, we present two **examples** of identifications of industrial plants, together with the corresponding validations.

A power plant .

Input-output measurements on a 120 MW power plant (Pont-Sur-Sambre, France) were obtained under *normal operating conditions* (five inputs, three outputs). Inputs and outputs are depicted in Figure 6. Inputs include gas flow, turbine valves openings, super heater spray flow, gas dampers and air flow. The outputs are the steam pressure, the main steam temperature and the reheat steam temperature. Using direct state space identification techniques, 4 different models were derived with an increasing complexity. It can be concluded from Figure 7 that the quality of the model gets better as the complexity increases. The block dimensions used were $i = 5, j = 90$.

A glass production installation .

A feeder is the final part of a process installation that is used for melting glass. Its main task is to realize a homogeneous temperature distribution. See [2, p.193] for an in depth discussion of the data acquisition and preprocessing procedures. Input 1 is the gas input of the first feeder, input 2 is a cooling air input while input 3 is the gas input of the second feeder. Pseudo-random binary sequences were applied as inputs. The first 300 samples of these inputs are depicted in Figure 8. The outputs of the process are the glass temperature at 6 different locations in a cross section of the feeder. The block dimensions of the input-output block Hankel matrix used, were $i = 10, j = 300$. The singular spectrum of the $2(m+l)i \times j$ ($=180 \times 300$) input output block Hankel matrix is depicted in Figure 8. The singular values from the $(2mi + 1) = 61$-th on are depicted in the right hand side of Figure 8. They allow to determine an approximate order, which for this example was chosen to be 4. Results of simulations based on the 300 used input-output vectors can be found in Figure 9. In Figure 9, one also finds a prediction of the first and the 4-th output from time step 800 to 1000. The prediction of output 1 was the worst, while that

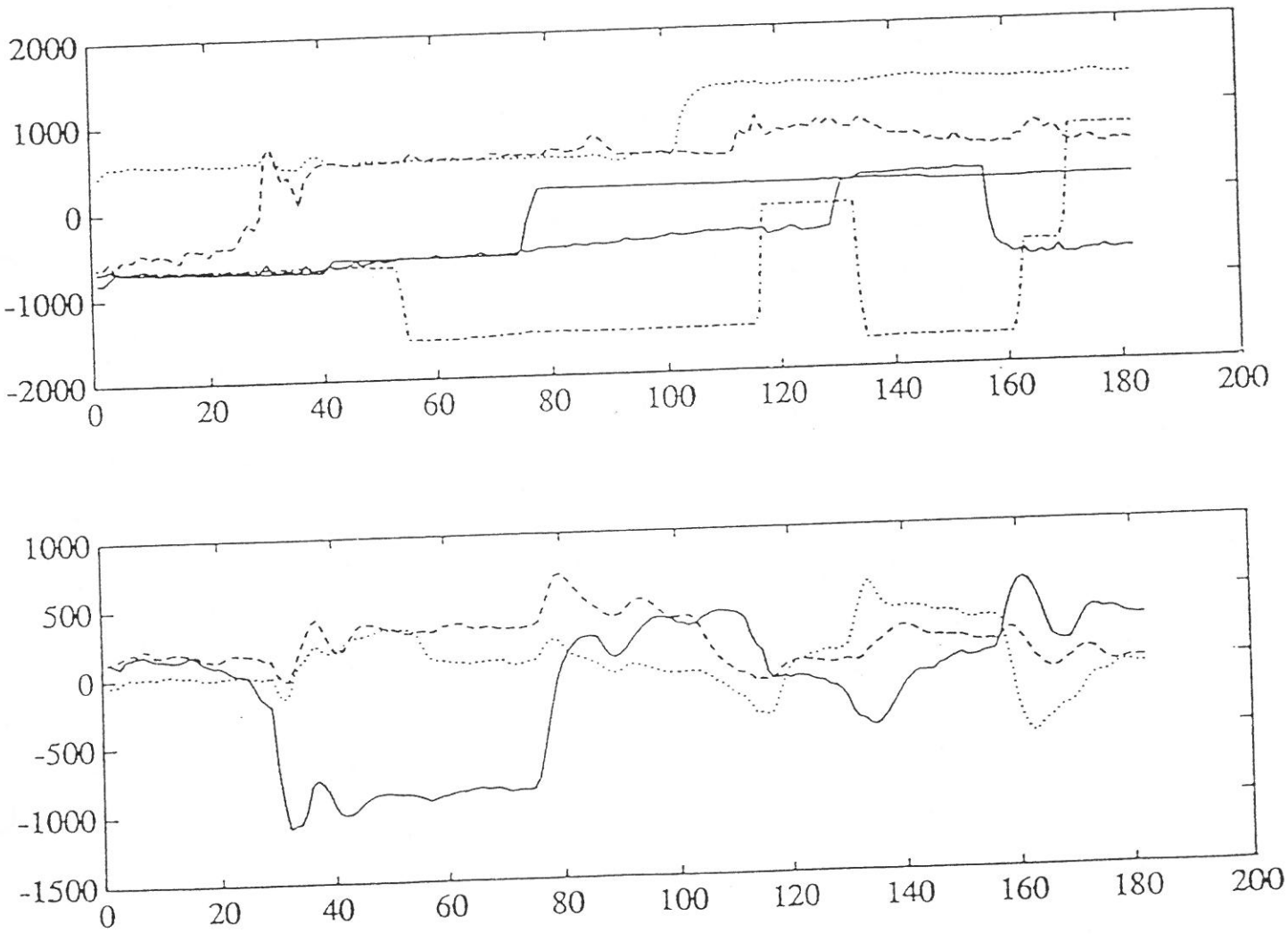

Figure 6: Inputs and outputs of a power plant

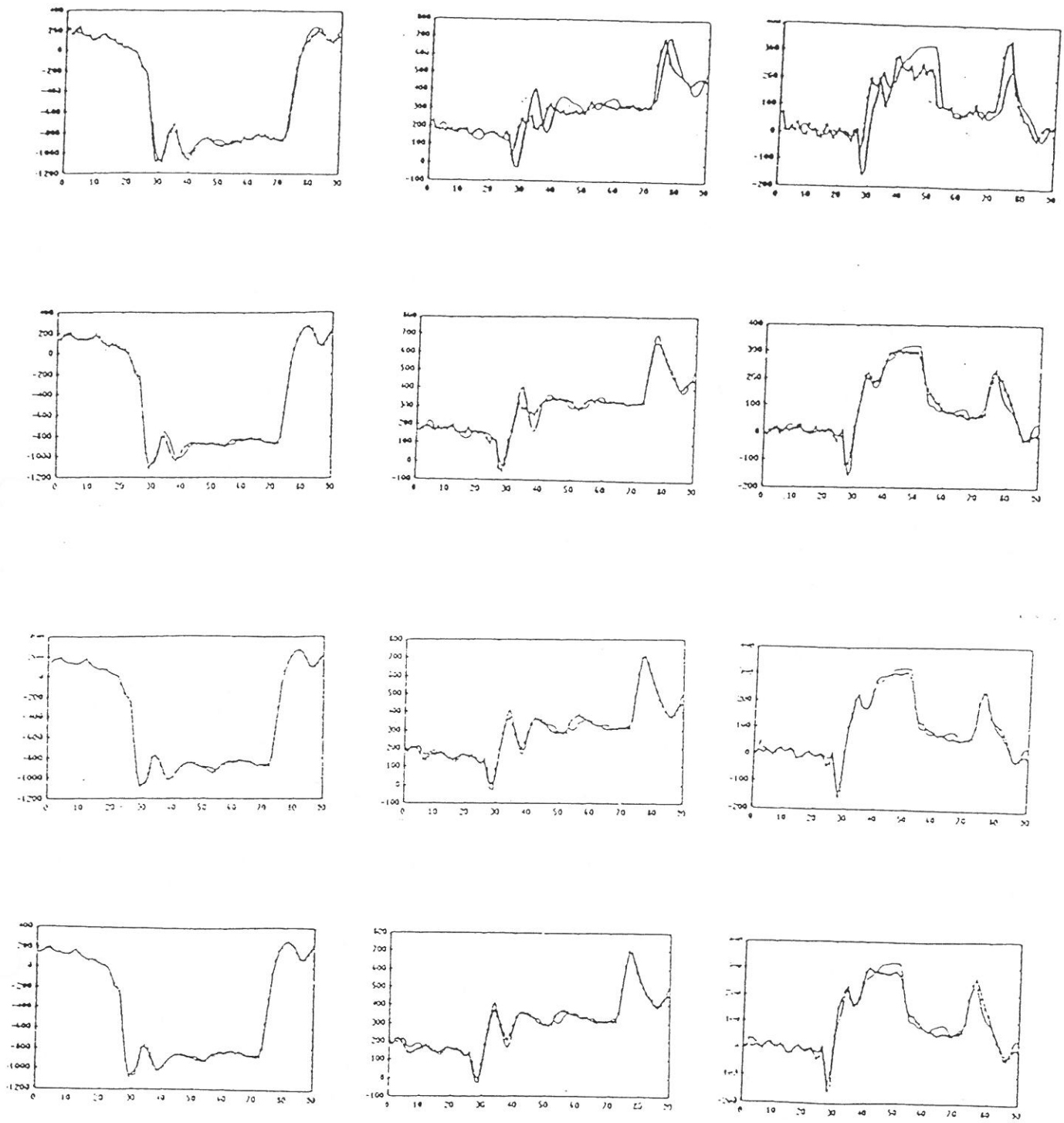

Figure 7: Measured outputs (full line) and simulations (stars) for a (a) first order, (b) 4-th order, (c) 7-th order, (d) 9-th order approximation

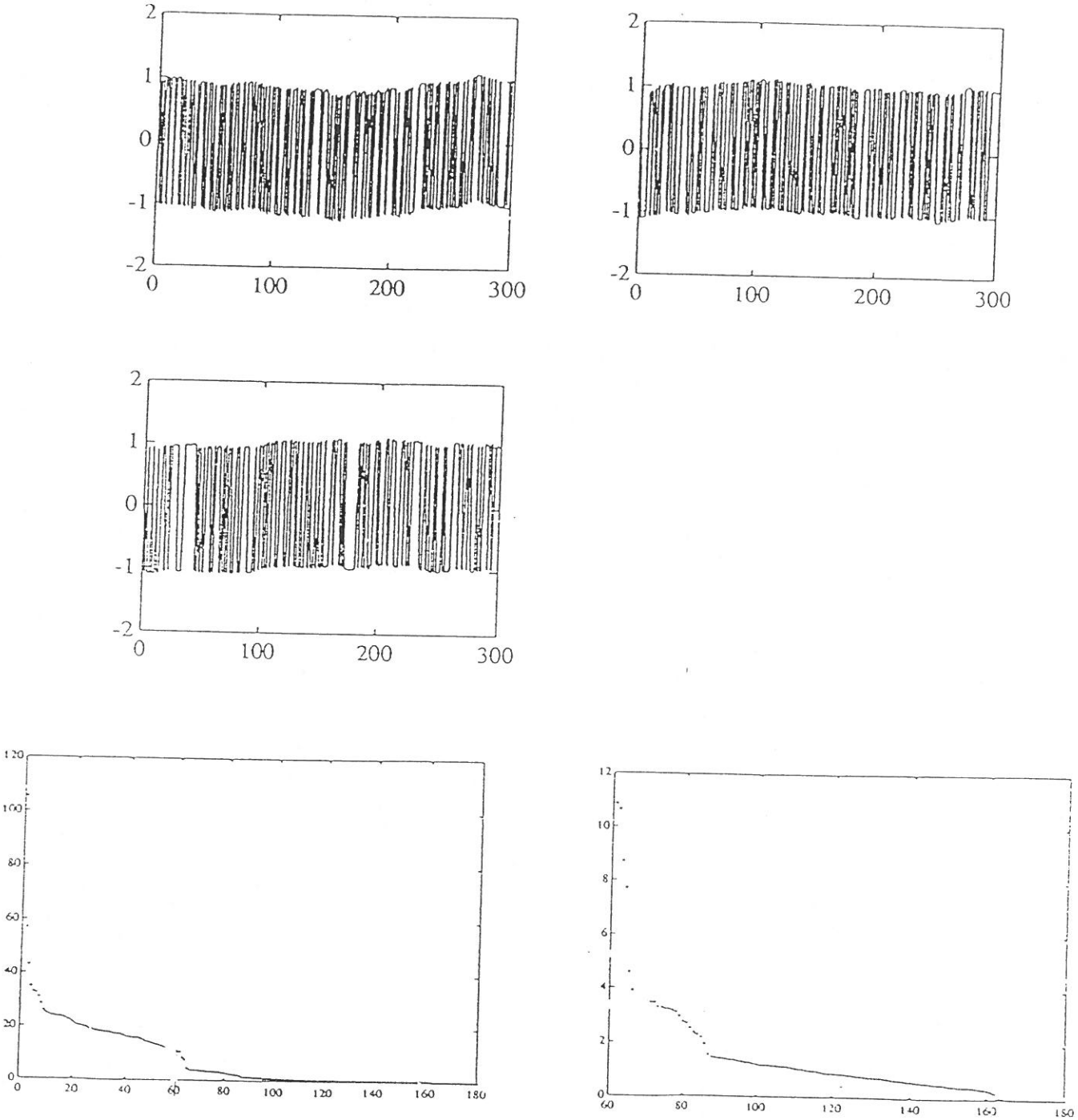

Figure 8: Three inputs of the feeder, singular spectra of the input-output block Hankel matrix.

of output 4 was the best of all predictions. Obviosuly, the model is rather good despite an offset in simulation of the first output.

7 Conclusions

In this brief introduction to system identification, we have surveyed common black box identification techniques. Different kinds of models were reviewed (input-output models, state space models), and corresponding identification schemes were presented. Finally, we have provided a few examples of identification applications in an industrial environment.

The authors are indebted to *Jan Swevers, Lieven Vandenberghe and Joos Vandewalle* for many interesting discussions and stimulating suggestions, and for the preparation of some of the figures.

References

[1] Astrom K., Wittenmark B. *Computer Controlled Systems.* Prentice-Hall, Englewood Cliffs, N.J., 1984.

[2] Backx T. *Identification of an industrial process: A Markov Parameter Approach.* Doctoral Thesis, Technische Universiteit Eindhoven, November 1987.

[3] De Moor B. *Mathematical Concepts and Techniques for Modelling of Static and Dynamic Systems.* Doctoral Dissertation, Katholieke Universiteit Leuven, Electrical Department, June 1988.

[4] Moonen M., De Moor B., Vandenberghe L., Vandewalle J. *On- and off-line identification of linear state space models.* International Journal of Control, January 1989.

[5] Eykhoff P. *System Identification.* Wiley, New York, 1974.

[6] Golub G., Van Loan C. *Matrix Computations.* North Oxford Academic Publ. Co., John Hopkins University Press, 1983.

[7] Goodwin G.C., Sin K.S. *Adaptive filtering, prediction and control.* Prentice-Hall Information and System Sciences Series, (Ed.T. Kailath), Prentice Hall, Inc., Englewood Cliffs, New Jersey 07632, 1984.

[8] Kailath T. *Linear Systems.* Prentice-Hall, Englewood Cliffs, NJ, 1980.

[9] Kailath T. *A view on three decades of linear filtering theory.* IEEE Trans. on Information Theory, vol.IT-20, pp.146-181, 1974.

[10] Ljung L. *System Identification: Theory for the User.* Prentice Hall Information and System Sciences Series (Ed. T. Kailath), Prentice-Hall, Inc., Englewood Cliffs, New Jersey 07632, 1987.

[11] Pro-Matlab, VAX/VMS version 2.1., VMS. December 1986, Mathworks Inc. MA, USA.

[12] Norton J.P. *An introduction to identification.* Academic Press, Harcourt Brace Jovanovich Publishers, London, 1986.

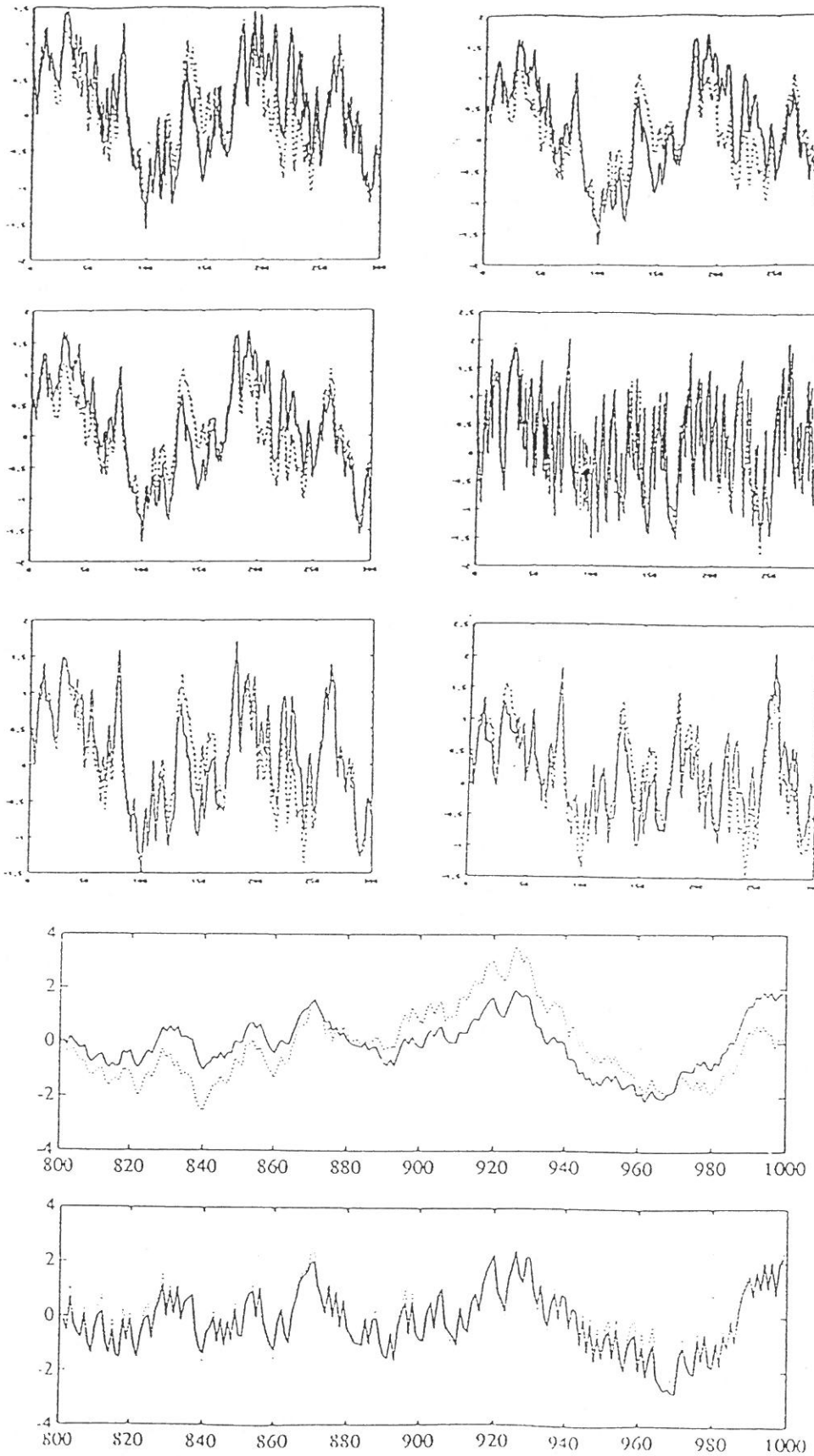

Figure 9: Simulation of the 6 outputs (measured=full line) and predicition of output 1 and 4 (measured=full line)

Appendix
The Singular Value Decomposition

the singular value decomposition (SVD) is a matrix factorization, where a $p \times q$ matrix X is factorized as the product of three matrices U, S, V, each of which has some interesting features:

$$X = USV^t$$

In this factorization:

S is a $p \times q$ diagonal matrix, with the singular values on its main diagonal:

$$S = \begin{pmatrix} s_1 & 0 & \dots & \dots & 0 \\ 0 & s_2 & 0 & \dots & \dots \\ \dots & \dots & \dots & \dots & \dots \\ 0 & \dots & \dots & s_r & \dots \\ 0 & \dots & \dots & \dots & 0 \\ 0 & \dots & \dots & \dots & 0 \end{pmatrix}$$

The singular values are ordered in non-decreasing order:

$$s_1 \geq s_2 \geq \ldots \geq s_r > 0$$

The smallest non-zero singular value s_r reveals the *algebraic rank* r of the matrix X, equal to the number of linearly independent rows (columns) in the matrix X. (Recall the a row (column) is linearly independent of all other rows (columns) if it cannot be written as a linear combination of these other rows (columns)).

The matrix U is a $p \times p$ orthonormal matrix:

$$U^tU = I_p = UU^t$$

Its columns $u^i, i = 1, \ldots, p$ are called the left singular vectors.

The matrix V is a $q \times q$ orthonormal matrix:

$$V^tV = I_q = VV^t$$

Its columns $v^i, i = 1, \ldots, q$ are called the right singular vectors.

The so-called *dyadic decomposition* is merely an alternative formulation of this:

$$X = u^1 s_1 (v^1)^t + \ldots + u^r s_r (v^r)^t$$

In this way, the matrix X of rank r is decomposed into r rank one matrices of decreasing importance (as $s_i \geq s_{i+1}$). An important optimality property (with respect to *data reduction applications*) is the following. Denote by $\|X\|_F^2$ the (squared) *Frobenius norm* of a matrix, which is the sum of its elements squared, then the SVD provides the solution to the following optimization problem:

$$\min_{\hat{X}} \|X - \hat{X}\|_F^2$$

subject to the constraint that

$$rank(\hat{X}) = k$$

If the dyadic decomposition of the matrix X is given as above, the solution is simply:

$$\hat{X} = \sum_{i=1}^{k} u^i s_i (v^i)^t$$

One of the most interesting properties of the singular values is their extreme insensitivity to additive perturbations. This implies that, when the data in the matrix are noisy, the singular values will still reveal the rank of the unperturbed matrix if of course the signal-to-noise ratio is not too small. As an example, consider the following 50×5 matrix:

$$X = \begin{pmatrix} 1 & 50 & 51 & -102 & 52 \\ 2 & 49 & 51 & -102 & 53 \\ 3 & 48 & 51 & -102 & 54 \\ \dots & \dots & \dots & \dots & \dots \\ 50 & 1 & 51 & -102 & 101 \end{pmatrix}$$

It is easy to verify that the third column is the sum of the first two columns, the 4-th one is 2 times the 3-th one, the last one is the sum of the first and the third column. Hence the algebraic rank of the matrix X is 2. Now random noise is added to this matrix. Each element is corrupted by normally distributed zero mean random noise with variance 1. The singular values of the exact and the perturbed matrix are:

	exact	noisy
1	1005.5	1004.7
2	167.7	167.7
3	0	6.57
4	0	6.23
5	0	5.51

Hence, from the 'gap' in the singular spectrum, one may still conclude that the perturbed noisy version is close to a matrix of rank 2. Typical is the so-called noise threshold: The smallest singular values correspond to the noise and are all of the same order of magnitude.
More properties and algorithms can be found in [6].