1994-08-18

# FROM DATA TO MODEL: A GUIDED TOUR OF SYSTEM IDENTIFICATION

Lennart Ljung

REPORT
LiTH-ISY-R-1652


Department of Electrical Engineering

Linköpings tekniska högskola

# From Data to Model: A Guided Tour of System Identification

Lennart Ljung
Department of Electrical Engineering
Linköping University
S-581 83 Linköping, Sweden
E-mail: ljung@isy.liu.se

June 16, 1994

**Abstract** In this contribution we give an overview and discussion of the basic steps of System Identification. The four main ingredients of the process that takes us from observed data to a validated model are: (1) The data itself, (2) The set of candidate models, (3) The criterion of fit and (4) The validation procedure. We discuss how these ingredients can be blended to a useful mix for model-building in practice.

## 1 Introduction

Here is an archetypical problem in science and human learning:"We are shown a collection of vector pairs $\{[y(t); x(t)]; t = 1, \dots N\}$. Call this "the training set". We are then shown a new value $x(N+1)$ and are asked to name a corresponding value $y(N+1)$." The variable $t$ could be thought of as time, but could be anything. The vectors $y(t)$ and $x(t)$ may take values in any sets (finite sets or subsets of $\Re^n$ or anything else) and the dimension of $x(t)$ could very well depend on $t$ (and could be unbounded). The formulation covers most kinds of classification and model building problems.

How to solve this problem? The mathematical modelling approach is to construct a function $\hat{g}_N(t, x(t))$ based on the "training" set, and to use this function for pairing $y(t)$ to new $x(t)$:

$$\hat{y}(t) = \hat{g}_N(t, x(t)) \tag{1}$$

Where do we get the function $g$ from? Essentially we have to search for it in a family of functions that is described (parametrized) in terms of a finite number of parameters. These parameters will be denoted by $\theta$. The family of candidate model functions will be called a *model structure*, and we write the function as

$$g(t, \theta, x(t)) \tag{2}$$

The value $y(t)$ is thus matched against the "candidate" $g(t, \theta, x(t))$ :

$$y(t) \sim g(t, \theta, x(t)) \tag{3}$$

We shall also use the notation

$$\hat{y}(t|\theta) = g(t, \theta, x(t)) \tag{4}$$

to stress that $g$ is a "predicted" or "guessed" $y$-value. The search for a good model function is then carried out in terms of the parameters $\theta$, and the chosen value $\hat{\theta}_N$ gives us

$$\hat{g}_N(t, x(t)) = g(t, \hat{\theta}_N, x(t)) \tag{5}$$

The process of going from observed data to a mathematical model is thus fundamental in science and engineering. In the control area this process has been termed "System Identification" and the objective is then to find dynamical models (difference or differential equations) from observed input and output signals. Its basic features are however common with general model building processes in statistics and other sciences.

System Identification has been an active research area for more than thirty years. It has matured and many of the techniques have become standard tools in control and signal processing engineering. The "mainstream approach" is described e.g. in [14] and [27]. Over the past few years there has been a significantly renewed interest in the area with topics like "unknown-but-bounded" disturbances, [24] [17], set membership techniques [5], [21] subspace techniques [22], $H_\infty$-identification [23], [10], worst case analysis [8], [16], as well as how to deal with unmodeled dynamics [20].

The procedure is characterized by four basic ingredients:

1. The observed data
2. A set of candidate models
3. A criterion of fit
4. Validation

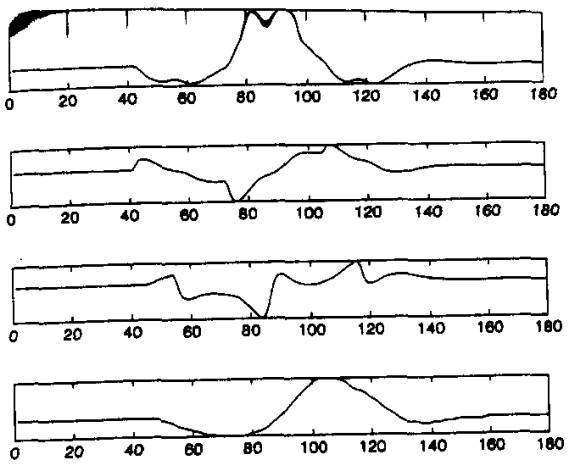

Figure 1: Results form test flights of the new Swedish aircraft JAS-Gripen, developed by SAAB Military Aircraft AB, Sweden. From above) Pitch rate. b) Elevator angle. c) Canard angle. d) Leading edge flap.

We shall in the sequel discuss these items more closely.

## 2 The Data

The area of system identification begins and ends with real data. Data required to build models and to validate models. The result of the modelling process can be no better than what corresponds to the information contents in the data.

Let us take a look at two data sets:

**Example 1 An unstable aircraft.** *Figure 1 shows some results from test flights of the new Swedish aircraft JAS-Gripen, developed by SAAB Military Aircraft AB, Sweden. The problem is to use the information in these data to determine the dynamical properties of the aircraft for fine-tuning regulators, for simulations, and so on. Of particular interest are the aerodynamical derivatives.*

**Example 2 Vessel dynamics.** *See Figure 2. The problem is to determine the residence time in the buffer vessel. The pulp spends about 48 hours total in the process, and knowing the residence time in the different vessels is important in order to associate various portions of the pulp with the different chemical actions that have taken place in the vessel at different times. (The $\kappa$-number is a quality property that in this context can be seen as a marker allowing us to trace the pulp.)*

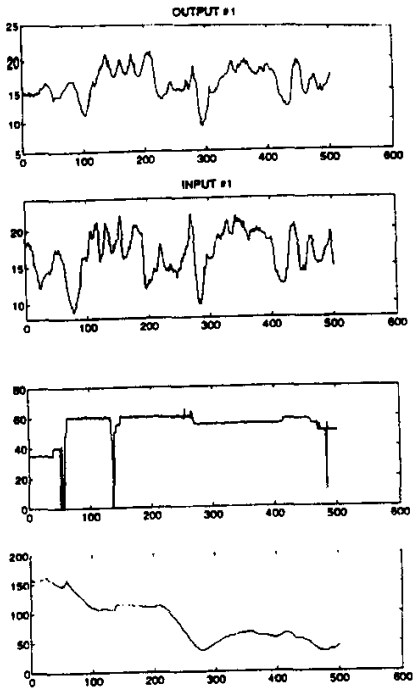


Figure 2: From the pulp factory at Skutskär, Sweden. The pulp flows continuously through the plant via several buffer tanks. From above: a) The $\kappa$-number of the pulp flowing into a buffer vessel. b) The $\kappa$-number of the pulp coming out from the buffer vessel. c) Flow out from the buffer vessel. d) Level in the buffer vessel.

## 3 The Set of Models: Model Structures

The single most important step in the identification process is to decide upon a model structure, i.e., a set of candidate models. In practice typically a whole lot of them are tried out and the process of identification really becomes the process of evaluating and choosing between the resulting models in these different structures.

It is natural to distinguish between three types of model structures:

1. Black-box structures
2. Structures from physical modelling
3. Structures from semi-physical modelling

## 3.1 Black-box structures

A black-box structure is one where the parameterization in terms of a parameter vector $\theta$ is chosen so that the family of candidate models covers as "many common and interesting" ones as possible. No particular attention to the actual application is then paid. For a linear system (a linear mapping from past data to future ones) we could for example think of choosing the parameters as the impulse response coefficients, of a finite impulse response model

$$\hat{y}(t|\theta) = \sum_{k=1}^{M} \theta_k u(t-k) \tag{6}$$

Here $u(t)$ is the input to the process and $\hat{y}(t|\theta)$ is the model's predicted or "guessed" output at time $t$. $y(t)$ will be the actual output. More common in control applications is the ARX black box structure for linear systems:

$$\hat{y}(t|\theta) = -a_1 y(t-1) - a_2 y(t-2) - \ldots$$

$$- a_n y(t-n) + b_1 u(t-1) + \ldots + b_m u(t-m) \tag{7}$$

"the mother of all dynamical model structures".

In general we can write a black box structure conceptually as

$$\hat{y}(t|\theta) = \sum_{k=1}^{M} \theta_k h_k(z^{t-1}) \tag{8}$$

i.e. as some kind of function expansion. In the general case the basis functions $\{h_k\}$ may also depend on $\theta$. In most cases the $h_k$ are also constrained to be functions of a fixed dimensional vector $\varphi(z^t)$ (like

$$h_k(z^{t-1}) = h_k(\varphi(z^{t-1})),$$

$$\varphi(z^t) = (y(t-1), \ldots, y(t-u)u(t-1), \ldots, u(t-u))$$

It is instructive to distinguish between two principally different basis functions:

- Global: Each of the $h_k$ have support in the whole $\varphi$-space
- Local: Each of the $h_k$ has support only in a small local box in the $\varphi$-space.That is, $h_k(\varphi)$ is zero unless $\varphi$ belongs to a certain neighborhood (that depends on $k$).

Among black-box structures that use global basis functions are all the usual linear black box models, Volterra series expansions and so on.

The local basic functions models can be visualized as a multidimensional table: The $\varphi$-space has been split up into a number of boxes. A new observation $\varphi(t)$ then falls into one of these boxes, the one corresponding to say $h_k$, and the predicted output is then taken as $\theta_k$ (or possibly interpolated, taking into account few neighboring boxes). The sizes and locations of the boxes can be determined with the aid of estimation data. The extreme case is when the boxes are determined so that exactly one data point $\varphi(t)\, t = 1 \ldots, N$ has fallen in each box: this is the so called nearest neighbor approach [30]. All this is well established in the statistical literature under names of "non-parametric regression" and "density estimation" [28], [3].

Neural network model structures, e.g. [19] represent a spectacular revival of these techniques. So called radial basis networks correspond to localized bases (where the "boxes" overlap like Gaussian distribution functions), while the feed-forward sigmoid network formally would use global basis functions (although the "dynamic effects" really are localized). Fuzzy modelling [13] is again an example of localized basis functions with typically polynomial interpolation rules, which are inherited by the "membership function".

It is worth stressing that these new techniques of neural net modelling and fuzzy identification represent useful realization of non-linear black box modelling with some new particular structures, but at the same time they definitely fall into a very old and classical framework of estimation techniques (See, e.g. [15], [2].)

## 3.2 Structures from physical modelling

In case we have physical insight into the properties of the system to be identified, it is natural to exploit this: *"Don't estimate what you already know!"* Basically we then write down those physical laws and relationships that describe the system. Most often they are then summarized in a state space form where the parameters are unknown physical constants in the description. The identification process is then to estimate these constants.

## 3.3 Semi-physical model structures

The logical route to utilize available physical knowledge may be quite laborious. It is then tempting to instead try some simple black-box structures, such as the ARX model (7) *("Try Simple Things First")*. This is quite OK, but it should in any case be combined with physical insight. Here is a toy example to illustrate the point:

*"Suppose we want to build a model for how the voltage applied to an electric heater affects the temperature of the room. Physical modelling entails writing down all equations relating to the power of the heater, heat transfer, heat convection and so on. This involves several equations, expressions and unknown heat transfer coefficients and so on. A simple black-box approach would instead be to use, say the ARX-model (7) with u as the applied voltage and y the room temperature. But that's too simple! A moment's reflection reveals that it's the heater power rather than the voltage that gives the temperature change. Thus use (7) with u= squared voltage and y= room temperature"*

I would like to coin the term *semi-physical modelling* for introducing non-linear transformation of the raw measurement, based on high-school physics and common sense. The transformed measurements are then used in black-box structures such as the ARX structure.

Clearly semi-physical modelling is in frequent use. It is however also true that many failures of identification are indeed to be blamed on not applying this principle.

### 3.4 Hybrid structures

Of particular current interest is to conceive model structures that are capable of dealing both with dynamic effects, described by differential/difference equations and with logical constraints, "the ifs and the buts" of the system. Not so many concrete results have yet been obtained in this area, but quite intense work is going on now. We may point to some work on using three models and pattern recognition for these hybrid model structures: [29], [26].

## 4 The Criterion of Fit

The system identification problem really is a variant of the following archetypical problem in science and human learning: "We are shown a collection of vector pairs

$$z^N\{[y(t);\varphi(t)], \quad t=1,2\ldots N\}$$

Call this "the training set". We are then shown a new value $\varphi(N+1)$ and are asked to name a corresponding value $y(N+1)$"

The variable $t$ could be thought of as time, but could be anything. The vectors $y(t)$ and $\varphi(t)$ may take values in any sets (finite sets or subsets of $\Re^n$ or anything else) and the dimension of $\varphi(t)$ could very well depend on $t$ (and could be unbounded). The formulation covers most kinds of classification and model building problems.

How to solve this problem? The mathematical modelling approach is to construct a function $\hat{g}_N(t,\varphi(t))$ based on the "training" set, and to use this function for pairing $y(t)$ to new $\varphi(t)$:

$$\hat{y}(t)=\hat{g}_N(t,\varphi(t)) \tag{9}$$

Where do we get the function $g$ from?

Basically, we have to follow the process described in the previous section, and carry out the search for $g$ in a family of functions that is parameterized in terms of a finite number of parameters, i.e., a model structure. We thus match the measured value $y(t)$ against the candidate $g(t,\theta,\varphi(t))=\hat{y}(t|\theta)$

$$y(t)\sim\hat{y}(t|\theta)$$

How shall we proceed to match these two? There are essentially two approaches. We can assume that the variables are related by

$$y(t)=\hat{y}(t|\theta)+v(t) \tag{10}$$

where $v(t)$ is the effect of *unmeasured inputs* that influence the system

The unmeasured input $v$ is usually thought of as "disturbances and noise". Clearly we need some sort of assumptions about the character of $v$, in order to proceed to find a good value of $\theta$, based on the information in $z^t$. There are two basic approaches to such assumptions.

- *Non-probabilistic*: Constrain the set of possible signals $\{v(t)\}$ in some way, like

  $$|v(t)|\le C \quad \forall t \tag{11}$$

  In general we may write for the "allowed" disturbaces:

  $$v\in V(\theta) \tag{12}$$

- *Probabilistic:* Assign probabilities to the different possible $\{v(t)\}$ sequences. That is, describe $\{v(t)\}$ as a random process with known or parameterized probability distribution:

  $$v \text{ has pdf } p_v(\cdot,\theta) \tag{13}$$

### The non-probabilistic approach

Given a model description (10) and some constraint on possible $\{v(t)\}$-sequences (such as (11)) the identification problem is conceptually very simple:

Find all those values of $\theta$, such that (10) holds with a $v$ subject to (12). This gives

$$\theta \in \Omega_\theta \tag{14}$$

Calculating $\Omega_\theta$ could of course be an overwhelming task. There is a rather extensive literature on various ways to do this. Most often one has to be content with an over bound $\bar{\Omega}_\theta$

$$\Omega_\theta \subset \bar{\Omega}_\theta.$$

with a more simple version of (10), viz

$$y(t) = \theta^T \varphi(t) + v(t) \tag{15}$$

$$\varphi(t) = [-y(t-1), \ldots, -y(t-n), u(t-n) \ldots, u(t-m)]^T \tag{16}$$

and with the simple constraint (11). The approach has been called "unknown-but-bounded" noise, "set membership identification" and "optimal algorithms". See among many references [25] [18], [12], [11].

### The probabilistic approach

Given a model description like (10) and a probabilistic measure (13) for $v$, we have of course indirectly specified a probabilistic measure. Indeed, the measure can be explicitly given as

$$\log p_{Z^t}(z^t, \theta) = \sum_{k=1}^{t} \log f_v(\epsilon(k, \theta), \theta) \tag{17}$$

where

$$\epsilon(t, \theta) = y(t) - \hat{y}(t|\theta) \tag{18}$$

which gives the logarithm of the likelihood criterion.

### A pragmatic approach

A more pragmatic approach to estimating the dynamics of a system is simply to postulate a *predictor model structure*, i.e. look for a description of the observed data within a family of models

$$\hat{y}(t|\theta) = g_t(z^t, \theta) \tag{19}$$

where the prediction of $y(t)$ is denoted by $\hat{y}(t|\theta)$. The function $g$ is based on observations available at time $t-1$,

$$z^t = [y(t-1), u(t-1), \ldots, y(0), u(0)] \tag{20}$$

and is an arbitrary (differentiable) function of these data and of the parameter vector $\theta$. The actual output will then differ from the prediction by an error $e(t)$

$$y(t) = \hat{y}(t|\theta) + e(t) \tag{21}$$

We then seek that value of $\theta$ that has the best track record in achieving good predictions

$$\hat{\theta}_N = \arg\min_\theta \frac{1}{N} \sum_{t=1}^{N} \ell(t, \theta, \epsilon(t, \theta)) \tag{22}$$

$$\epsilon(t, \theta) = y(t) - \hat{y}(t|\theta) \tag{23}$$

It is clear that by invoking a probabilistic framework, i.e. by assigning a pdf to $\{e(t)\}$ in (21) the pragmatic estimate (22) can be seen as an ML estimate (17) – (18).

It is also clear that by choosing

$$\ell(t, \theta, x) = \begin{cases} 0 & |x| \leq C \\ \infty & |x| > C \end{cases} \tag{24}$$

the method (22) will pick out those $\theta$ which are consistent with the assumption $|e(t)| \leq C$ in (20) for all $0 \leq 1 \leq N$. Thus the non-probabilistic approach also fits into (22).

Most "traditional" control oriented descriptions of System Identification follow this mixture of pragmatic and probabilistic approaches. See e.g. [14] and [27].

## 5 Model Validation

It is not enough to come up with a nominal model $\hat{\theta}_N$ from (22) – we must also have a measure of its reliability. *Model validation* is the process of examining the model, assessing its quality and possibly rejecting its use for the purpose in question. In a sense this could be viewed as the *essential process of identification* – the estimation phase is really just a means to provide candidate models that might pass the needle's eye of validation.

Model validation has at least these different objectives:

1. To decide if the model is "good enough" for the intended application
2. To decide how "far from the true system description" the model might be
3. To decide whether the model and the data indeed are consistent with assumptions about the model structure.

These objectives partly overlap, but it is still possible to single out basic techniques:

1. The most obvious and pragmatic way to decide if a model is good enough is to test how well it is

able to reproduce validation data (data that were not used to estimate the model) in simulation or prediction. The user can then by eye inspection decide if the fit is "good enough". In my mind this is the prime validation tool.

2. To determine *error bounds* – how far is the true system from the model – is a fundamentally difficult question. If we adopt a probabilistic setting and assume that the true system is to be found within the chosen structure it becomes a matter to see how much the stochastic disturbances might have affected the model. The covariance matrix of the asymptotic distribution is classically used for the error bounds in this case, This covariance matrix is generally given by

$$cov\{\hat{\theta}_N\} \sim \frac{1}{N} E e^2(t) [cov\{\frac{d}{d\theta}\hat{y}(t|\theta)\}]^{-1} \quad (25)$$

for the structure (21). If we (according to 3) below) cannot disprove that the true system can be represented in the chosen structure it is still reasonable to use the measure (25).

3. To test if the data and the model are consistent with the model structure assumptions, is again a more straightforward task. Basically we compute the residuals $y(t) - \hat{y}(t|\hat{\theta}_N) = \epsilon(t)$ from the model and a (validation) data set and check if

   (a) $|\epsilon(t)| < C$ in the deterministic setting (11)

   (b) $\epsilon(t)$ and $u(t-\tau)$ are independent random variables, in the probabilistic setting ($u$ is the input to the system).

   The latter test is one of many *residual* analysis tests that can be performed, and this is standard statistical practice, see e.g. [4].

A quite important aspect of Model Validation that has raised considerable current interest is that of how to deal with *unmodeled dynamics*. See, e.g. [6] and [7]. A very insightful recent discussion is given in [20].

A model structure is always too simple to describe a true system. Thus we always have to deal with model errors. These consist of two parts

1. Errors arising from the "unmeasured input". We call these *the random error* or *the variance error*

2. Errors arising from too simple a model structure. We call these errors *bias error* (or unmodeled dynamics).

The total error of the model is from a statistical point of view the sum of (the squared) bias error and the variance error:

$$f_e(d) = f_b(d) + f_v(d) \quad (26)$$

Here $d$ is a variable that reflects the model complexity. (such that a larger value of $d$ means a more flexible model structure)

The following chain of reasoning is taken from [9]: It is natural to assume that $f_b$ decreases and $f_v$ increases with $d$, and that the product $f_b f_v$ decreases with $d$ (the bias error decreases more rapidly than the variance error increases). It is also natural to seek a model complexity that minimizes the mean square error fit. However, under the given assumptions

$$d^* = \arg\min f_e(d^*) \Rightarrow f_b(d^*) \leq f_v(d^*) \quad (27)$$

that is, the variance error dominates the bias error for such a choice of model structure.

In [9] it is also proven that a model that passes a typical residual analysis test is also such that the bias error is dominated by the variance error.

The message of this analysis is consequently that *for a validated model the bias error is no larger than the variance error*. Moreover, the variance error can be estimated from data. It is then another matter that the user may like a simpler model for design/simulation purposes. But the total error for such a simplified model can then be assessed via its distance to a validated model.

# 6 Back to Data: The Practical Side of Identification

It follows from our discussion that the most essential element in the process of identification – once the data have been recorded – is to try out various model structures, compute the best model in the structures; using (22), and then validate this model. Typically this has to be repeated with quite a few different structures before a satisfactory model can be found.

While one should not underestimate the difficulties of this process, I suggest the following simple procedure to get started and gain insight into the models.

1. Find out a good value for the delay between input and output, e.g. by using correlation analysis.

2. Estimate a fourth order linear model with this delay using part of the data, and simulate this model with the input and compare the model's simulated output with the measured output over the whole data record. In MATLAB language this is simple,

```
z = [y u];
compare(z,arx(z(1:200,:),[4 4 1]));
```

If the model/system is unstable or has integrators, use prediction over a reasonable large time horizon instead of simulation.

Now, either of two things happen:

- *The comparison "looks good".* Then we can be confident that with some extra work – trying out different orders, and various noise models – we can fine tune the model and have an acceptable model quite soon. Let me add here that I am amazed by the large amount of applications that fall into this category.
- *The comparison "does not look good".* Then we must do further work. There are three basic reasons for the failure.
  1. *A good description needs higher order linear dynamics.* This is actually in practice the least likely reason, except for systems with mechanical resonances. One then obviously has to try higher order models or focus on certain frequency bands by band pass filtering.
  2. *There are more signals that significantly affect the output.* We must then look for what these signals might be, check if they can be measured and if so include them among the inputs. Signal sources that cannot be traced or measured are called "disturbances" and we simply have to live with the fact that they will have an adverse effect on the comparisons.
  3. *Some important non-linearities have been overlooked.* We must then resort to semi-physical modelling to find out if some of the measured signals should be subjected to non-linear transformations. If no such transformations suggest themselves, one might have to try some non-linear black-box model, like a neural network.

Clearly, this advice does not cover all the art of identification, but it is the best half page summary of the practical process of identification that I can offer.

**Example 3** *Aircraft dynamics*

*Let us try the recipe on the aircraft data in figure 1! Picking the canard angle only as the input, estimating a fourth order model based on the data points 90 to 180, gives figure 3. (We use 10-step ahead prediction in this example since the models are unstable – as they should be, JAS has unstable dynamics in this flight case). It does not "look good". Let us try alternative 2: More inputs. We repeat the procedure*

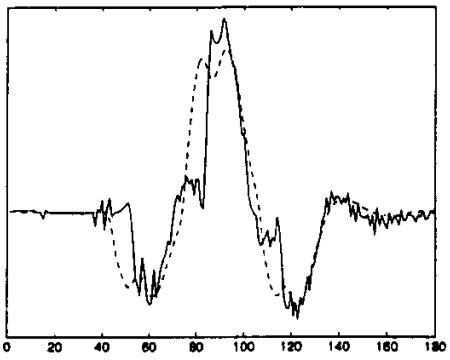


Figure 3: Dashed line: Actual Pitch rate. Solid line: 10 step ahead predicted pitch rate, based on the fourth order model from canard angle only.

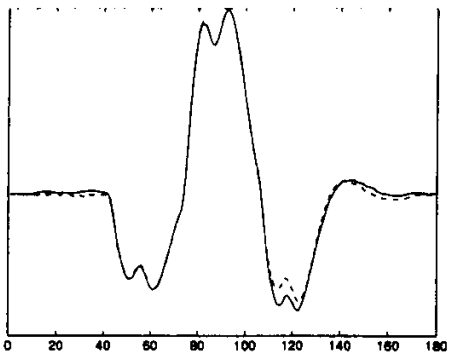


Figure 4: As figure 3 but using all three inputs.

*using all three inputs in figure 1. That is, the model is computed as*

```
arx([y u1 u2 u3], [4 4 4 4 1 1 1])
```

*on the same data set. The comparison is shown in figure 4. It "looks good". By further fine-tuning, as well as using model structures from physical modeling, only slight improvements are obtained.*

**Example 4** *Buffer vessel dynamics*

*Let us now consider the pulp process of figure 2. We use the $\kappa$-number before the vessel as input and the $\kappa$-number after the vessel as output. The delay is preliminarily estimated to 12 samples. Our recipe, where a fourth order linear model is estimated using the first 200 samples and then simulated over the whole record gives figure 5. It does not look good.*

*Some reflection shows that this process indeed must be non-linear (or time-varying): the flow and the vessel level definitely affect the dynamics. For example, if*

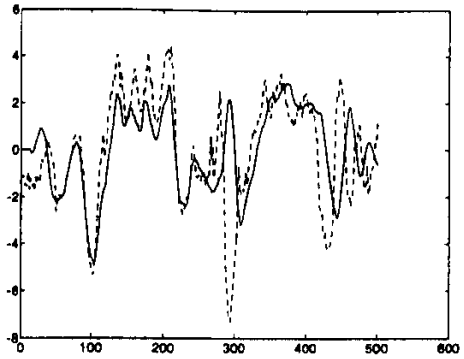


Figure 5: Dashed line: $\kappa$-number after the vessel, actual measurements. Solid line: Simulated $\kappa$-number using the input only and a fourth order linear model with delay 12, estimated using the first 200 data points.

*the flow was a plug flow (no mixing) the vessel would have a dynamics of a pure delay equal to vessel volume divided by flow.*

*Let us thus resample the date accordingly, i.e. so that a new sample is taken (by interpolation from the original measurement) equidistantly in terms of integrated flows divided by volume. In MATLAB terms this will be*

```
z = [y,u]; pf = flow./level;
t =1:length(z)
newt =
table1([cumsum(pf),t],[pf(1)sum:(pf)]' );
newz = table1([t,z], newt);
```

*We now apply the same procedure to the resampled data. This gives figure 6. This "looks good". Somewhat better numbers can then be obtained by fine-tuning the orders.*

# 7 Conclusions

The area of process identification is one where real practical application and rather advanced mathematical tools and perspectives meet. The meeting place is really the software into which many years' research has been packaged. There are now many successful such packages commercially available. They have become standard tools in many industrial applications. This again stresses that it is the engineer's perspective that is the ultimate one in this area.

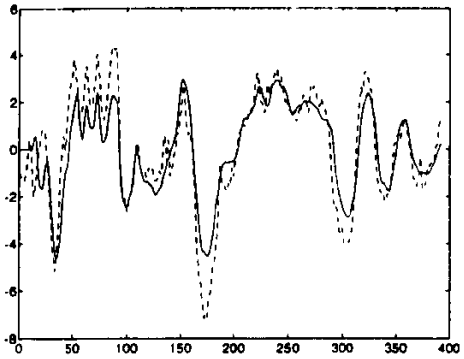


Figure 6: Same as figure 5 but applied to resampled data

# 8 Acknowledgements

I am happy to acknowledge many years' financial support from the National Swedish Board for Technical Development, and the Swedish Research Council for Engineering Sciences.

I am also grateful to SAAB Military Aircraft AB for allowing me to show the test flight data from JAS-Gripen and to Skutskär Paper and Pulp Mill for the process data. The analysis of the latter data has been carried out by Torbjörn Andersson and Predrag Pucar [1].

# References

[1] T. Andersson and P. Pucar. Estimation of residence time in continuous flow systems with varying flow and volume. Technical Report LiTH-ISY-I-1460, Department of Electrical Engineering, Linköping University, Sweden, 1993.

[2] A.R. Barron. Statistical properteis of artificial neural networks. In *Proceedings of the 28th IEEE Conference on Decision and Control*, pages 280–285, 1989.

[3] L. Devroye and L. Györfi. *Non-parametric density estimation.* Wiley, New York, 1985.

[4] N.R. Draper and H. Smith. *Applied Regression Analysis, 2nd ed.* Wiley, New York, 1981.

[5] E. Fogel. System identification via membership set constraints with energy constrained noise. *IEEE Trans on Automatic Control*, AC-24:615–622, 1979.

[6] G.C. Goodwin, M. Gevers, and D.Q. Mayne. Bias and variance distribution in transfer function estimates. In *Proc. 9th IFAC Symp. Identif. Syst. Parameter Estimation*, Budapest, Hungary, 1991.

[7] G.C. Goodwin, M. Gevers, and B. Ninness. Quantifying the error in estimated transfer functions with application to model order selection. *IEEE Trans. Automatic Control*, 37(7):913–929, 1992.

[8] L. Guo and P. Khargonekar. A class of algorithms for identification in $h_\infty$. *Automatica*, 1992.

[9] L. Guo and L. Ljung. Exponential stability of general tracking algorithms. *IEEE Trans. Autom. Control*, 1994. Submitted.

[10] A.J. Helmicki. Control oriented system identification: A worst case/deterministic approach in $h_\infty$. *IEEE Transactions on Automatic Control*, 36(October):1163–1176, 1991.

[11] H. Hjalmarsson and L. Ljung. Estimating models variance in the case of undermodelling. *IEEE Trans. Automoatic Control*, 37(7), 1992.

[12] R.L. Kosut, M.K. Lau, and S.P. Boyd. Set-membership identification of systems with parametric and nonparametric uncertainty. *IEEE Trans. Autom. Control*, 37(7):929–942, 1992.

[13] C.W. Ku and Y.Z. Lu. Fuzzy model identification and self-learning. *IEEE Trans. on SMC*, 17, 1987.

[14] L. Ljung. *System Identification - Theory for the User*. Prentice-Hall, Englewood Cliffs, N.J., 1987.

[15] L. Ljung and T. Söderström. *Theory and Practice of Recursive Identification*. MIT press, Cambridge, Mass., 1983.

[16] P. Mäkilä and J. Partington. Robust approximation and identification in $h_\infty$. pages 70–76. Proc. American Control Conference, 1991.

[17] M. Milanese and G. Belforte. Estimations theory and uncertainty intervals evaluation in the presence of unkown but bounded errors: Linear families of models and estimators. *IEEE Trans. on Automatic Control*, AC-27:408–414, 1982.

[18] M. Milanese and R. Tempo. Optimal algorithms for robust estimation and prediction. *IEEE Trans. Automatic Control*, AC-30:730–738, 1985.

[19] K.S. Narendra and K. Parathasarathy. Identification and control of dynamical systems using neural networks. *IEEE Trans. Neural Networks*, 1:4–27, 1990.

[20] B. M. Ninness. *Stochastic and Deterministic Modeling*. PhD thesis, Dept. of Electrical Engineering, Univerisyt of Newcastle, NSW, Australia, August, 1993.

[21] J. Norton. Identification and application of bounded parameter modles. *Automatica*. 23:497–507, 1987.

[22] P. Van Overschee and B. De Moor. N4sid: Subspace algorithms for the identification of combined deterministic-stochastic systems . *Automatica, (Special Issue)*, 29(5):to appear, 1993.

[23] P. Parker and R. Bitmead. Adaptive frequency response estimation. In *Proc. Conference on Decision and Control*, pages 348–353, 1987.

[24] F.C. Schweppe. Recursive state estimation - unknown but bounded errors and system inputs. *IEEE Trans. on Automatic Control*, (37):22 28. 1968.

[25] F.C. Schweppe. *Uncertain Dynamical Systems*. Prentice-Hall, Englewood Cliffs, 1973.

[26] A. Skeppstedt, L. Ljung, and M. Millnert. Construction of composite models from observed data. *Int. j. Control*, 55(1):141-152, 1992.

[27] T. Söderström and P. Stoica. *System Identification*. Prentice-Hall Int., London, 1989.

[28] C.J. Stone. Consistent non-parametric regression (with discussion). *Ann. Statist.*, 5:595–645, 1977.

[29] J.E. Strömberg, F. Gustafsson, and L. Ljung. Trees as black-box model structures for dynamical systems. In *Proc. 1st European Control conference (ECC'91)*, pages 1175–1180, Grenoble, France, 1991.

[30] T.M.Cover and P.E. Hart. Nearest neigbor pattern classification. *Trans. IEEE Info. Theory*, IT-13:21-27, 1967.

/home/rt/ingegerd/report/aium94.tex