Practical Issues of System Identification

Lennart Ljung
Division of Automatic Control
E-mail: ljung@isy.liu.se
29th June 2007

Report no.: LiTH-ISY-R-2810
Accepted for publication in UNESCO encyclopedia EOLSS: Encyclopedia of LifeSupport Systems

Address:
Department of Electrical Engineering
Linköpings universitet
SE-581 83 Linköping, Sweden

WWW: http://www.control.isy.liu.se
Abstract
This is a survey of practical issues in System Identification.

Keywords: identification
PRACTICAL ISSUES OF SYSTEM IDENTIFICATION

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

Keywords: Parameter estimation, model structure, input design, drift, trends, identification applications, PRBS, chirp signal, ARX models, state-space models, output-error models.

Contents

1. The Framework
   1.1. Starting Point
   1.2. Some Typical Model Structures
   1.3. Estimating the Parameters
2. The User and the System Identification Problem
   2.1. The Tool: Interactive Software
3. Choice of Input Signals
4. Preprocessing Data
   4.1. Drifts and Detrending
   4.2. Prefiltering
5. Selecting Model Structures
6. Some Applications

Bibliography

Glossary

System Identification: The methodology to build mathematical models of dynamical systems based on observed input–output signals.

Prediction Error Identification: Building mathematical models of dynamical systems by choosing that model that shows the best predictions of output signals, when applied to measured data.

Model Structure Selection: The process of selecting a suitable model structure to represent a given system. This process is governed both by prior physical knowledge and information in the data.

Input Design: The process of selecting a suitable input signal for identification. This amounts to determine a signal that excites the appropriate dynamics of the system, subject to given constraints.

PRBS: A particular periodic input signal, Pseudo Random Binary Signal, that is generated as a deterministic signal, but has several features of white noise.

Prefiltering, Data Preprocessing: The process of conditioning measured input and output signals before using them to estimate models.

Summary

System Identification concerns the problem of building mathematical models of dynamical systems. This involves a fair amount of theory and algorithms. Equally important, though, is
the practical side of the methodology. This contribution deals with the issues that are essential to construct a good model in practice. Such issues include the problem of input and experiment design. Typical essential features of the input are discussed as well as examples of commonly used inputs. Next comes the question of how to condition the measured signals. This involves issues of removing trends and disturbances outside of the frequency ranges of interest for the model. The most demanding task is to find a suitable model structure, guided by information in the observed data. A first cut methodology for this is described. For a successful application, it is in the end a matter of combining intuition, and information from various data test. Two applications, a fighter aircraft and a buffer vessel in process industry illustrate the process.

1. The Framework

System Identification is both a science and an art. In several articles in this encyclopedia, the science of System Identification has been developed. This involves techniques for parameter estimation, the statistical framework, choice of model structures, techniques for non-parametric techniques etc. The art of System Identification concerns how all these techniques are applied in practice: What problems have to be solved, what choices have to be made when the user faces a real-life plant and need a mathematical model of how it works. Some issues that are critical for this situation will be reviewed in this article.

1.1. Starting Point

We consider the following set-up for system identification. The set of models that can be used consists of linear time invariant (LTI) descriptions of dynamical systems. They will generally be described as

\[ y(t) = G(q, \theta)u(t) + H(q, \theta)e(t) \]  

(1)

Here \( y(t) \) is the output at time \( t \), \( u(t) \) is the input signal at time \( t \), and \( e(t) \) is a disturbance source, typically described as a sequence of independent random variables. \( q \) is the shift operator, and \( G \) and \( H \) are rational transfer functions in \( q \). In other words, (1) describes a set of linear difference equations relating the input, output, and disturbances to each other.

The signals \( y \) and \( u \) are observed, while \( e \) is not measurable. The measurements are made in discrete time, and they will generally just be enumerated as \( Z^N = \{ y(1), u(1), y(2), u(2), \ldots, y(N), u(N) \} \). The model could very well be multivariable, i.e. \( y(t) \) and \( u(t) \) could be vectors containing several input and output variables.

The system identification problem is to

- generate a suitable input signal \( u \)
- measure the corresponding \( y \)
- find an appropriate model parameterization (structure) (1)
- determine the “best values” of the corresponding parameters \( \theta \)
- determine if the resulting model is adequate for its intended purpose
1.2. Some Typical Model Structures

The general model (1) contains all possible LTI models. It is just a question of how to parameterize the transfer functions. See other entries in this encyclopedia for more details on how this can be done. Some common special cases are

- **ARX models**: \( A(q)y(t) = B(q) + e(t) \). This corresponds to the parameterization of \( G(q, \theta) \) as a rational function

\[
G(q, \theta) = \frac{B(q)}{A(q)}
\]

with the parameters \( \theta \) being the coefficients of the numerator and denominator polynomials. This structure has a noise model

\[
H(q, \theta) = \frac{1}{A(q)}
\]

that does not have any extra degree of freedom. The most important reason for the use of such ARX models is that \( \theta \) can be estimated by a simple linear least squares method.

- **Output Error (OE) models**: These use a fixed noise model \( H(q, \theta) = 1 \) and thus assume all measurement error be white noise at the output of the system.

- **State-space models**: These correspond to the parameterization of \( G \) and \( H \) in terms of matrices \( A, B, C, D, K \) as in

\[
G(q, \theta) = C(\theta)(qI - A(\theta))^{-1}B(\theta) + D(\theta)
\]

\[
H(q, \theta) = C(\theta)(qI - A(\theta))^{-1}K(\theta) + I
\]

This means that the input-output relationship can be written as a state-space model (in innovations form):

\[
x(t + 1) = A(\theta)x(t) + B(\theta) + K(\theta)e(t)
\]

\[
y(t) = C(\theta)x(t) + D(\theta) + e(t)
\]

(2)

(3)

The parameterization of the state space matrices can be done in a arbitrary way. They could for example be constructed from an underlying continuous-time model with parameter entries corresponding to unknown constants of physical significance.

1.3. Estimating the Parameters

There are several possibilities for estimating the parameters in (1). A generic method is the prediction error approach: Form the error between the model output \( G(q, \theta)u(t) \) and the measured output \( y(t) \) and filter it with the inverse noise model:

\[
\varepsilon(t, \theta) = H^{-1}(q, \theta)(y(t) - G(q, \theta)u(t))
\]

(4)

Note that \( \varepsilon \) can be seen as an estimate of \( e \) in (1). Then select the parameter estimate \( \hat{\theta}_N \) so that the size of these weighted errors becomes as small as possible:

\[
\hat{\theta}_N = \arg \min_{\theta} \frac{1}{N} \sum_{t=1}^{N} |\varepsilon(t, \theta)|^2
\]
The minimization of this criterion, and hence the computation of the estimate must normally be made by iterative numerical search. It is essentially only the ARX model that allows a closed form expression for $\hat{\theta}_N$.

For the state-space model (2) there is another possibility to estimate the system matrices, known as subspace methods. In short, this method first estimates the states $x$ from ARX-like expressions, and then treats (2) as a linear regression with $x$ (pretended as) known. The advantage is a numerically efficient, non-iterative algorithm.

2. The User and the System Identification Problem

To turn to the art of System Identification, we can pose as users faced with a physical process to experiment with, with the aim of constructing a reliable mathematical model. This task involves several subproblems:

1. Select an input signal to apply to the process
2. Collect the corresponding output data
3. Scrutinize the obtained data to find out if some preprocessing will be necessary
4. Specify a model structure.
5. Let the computer deliver the best model in this structure, when applied to the collected data.
6. Evaluate the properties of this model.
7. Test a new structure, go to step 4.
8. If the models obtained in this way are not adequate, go back to step 3 to try some other data preprocessing, or to step 1 to carry out a new experiment with another and more "revealing" input.

See Figure 1 which illustrates this process.

2.1. The Tool: Interactive Software

To have any chance of success it will be necessary to have good computer support. Figure 1 shows – in rectangles – what support the computer can supply. The techniques and algorithms behind these rectangles belong to the “science” of System Identification. The remaining decisions, that have to be taken by the user, – marked in ovals in the figure –is what this article will focus on.

The first thing that requires help is to compute the model and to evaluate its properties. There are now many commercially available program packages for identification that supply such help. They typically contain the following routines:
Figure 1: Identification cycle. Rectangles: the computer’s main responsibility. Ovals: the user’s main responsibility.

A Handling of data, plotting, and the like
Filtering of data, removal of drift, choice of data segments, and so on.

B Non-parametric identification methods
Estimation of covariances, Fourier transforms, correlation and spectral analysis, and so on.

C Parametric estimation methods
Calculation of parametric estimates in different model structures.

D Presentation of models
Simulation of models, estimation and plotting of poles and zeros, computation of frequency functions and plotting in Bode diagrams, and so on.

E Model validation
Computation and analysis of residuals \( \varepsilon(t, \hat{\theta}_N) \); comparison between different models’ properties, and the like.

The existing program packages differ mainly by various user interfaces and by different options regarding the choice of model structure according to item C.

One of the most used packages is MathWork’s System Identification Toolbox (SITB), which is used together with MATLAB. The command structure is given by MATLAB’s programming environment with the work-space concept and MACRO possibilities in the form of m-files. SITB gives the possibility to use all model structures of the black-box type (1) with an arbitrary number of inputs. ARX-models and state-space models with an arbitrary number of inputs and outputs are also covered. Moreover, the user can define arbitrary tailor-made linear
state-space models in discrete and continuous time as in (2). A Graphical User Interface helps the user both to keep track of identified models and to guide him or her to available techniques.

The remainder of this article will deal with the issues in the ovals of Figure 1: In Section 3 the choice of input will be discussed, while Section 4 deals with data preprocessing. The difficult task to find a good model structure is then discussed in Section 5. Finally two applications will be reviewed in Section 6.

3. Choice of Input Signals

The requirement from the previous section that the data should be informative means for open loop operation that the input should be persistently exciting (p.e.) of a certain order; i.e. that it contains sufficiently many distinct frequencies. This leaves a substantial amount of freedom for the actual choice, and we shall in this section discuss good and typical choices of input signals.

For the identification of linear systems, there are three basic facts that govern the choices:

1. The asymptotic properties of the estimate (bias and variance) depend only on the input spectrum – not the actual waveform of the input.

2. The input must have limited amplitude: \( u \leq u(t) \leq \pi \). The crest factor measures how well a given signal utilizes such a given amplitude span: It is essentially defined as the maximum amplitude divided by the standard deviation of the signal.

3. Periodic inputs may have certain advantages.

Common Input Signals

For linear system identification it is desirable to achieve a desired input spectrum for a signal with as small crest factor as possible. Unfortunately these properties are somewhat in conflict: If it is easy to manipulate a signal’s spectrum, it tends to have a high crest factor and vice versa. We shall now describe typical choices of waveforms, and how to achieve desired spectra.

A general comment is that it is always advisable to generate the signal and study its properties, off-line, before using it as an input in an identification experiment.

Filtered Gaussian White Noise. A simple choice is to let the signal be generated as white Gaussian noise, filtered through a linear filter. With this we can achieve virtually any signal spectrum (that does not have too narrow pass bands) by proper choice of filters. Since the signal is generated off-line, non-causal filters can be applied and transient effects can be eliminated, which gives even better spectral behavior. The Gaussian signal is theoretically unbounded, so it has to be saturated ("clipped") at a certain amplitude. Picking that, e.g., to be at 3 standard deviations gives a crest factor of 3, and at the same time, only an average of 1 \% of the time points are affected. This should lead to quite minor distortions of the spectrum.
Random Binary Noise. A random binary signal is a random process which assumes only two values. It can be generated in a number of different ways. The telegraph signal is generated as a random process which at any given sample has a certain probability to change from the current level to the other one. Apparently, the most common way is to simply generate white, zero mean Gaussian noise, filter it by an appropriately chosen linear filter, and then just take the sign of the filtered signal. It can then be adjusted to any desired binary levels. The crest factor is thus the ideal 1. The problem is that taking the sign of the filtered Gaussian signal will change its spectrum. We therefore do not have full control of shaping the spectrum. In the off-line situation we can however always check the spectrum of the signal before using it as input to the process to see if it is acceptable.

Pseudo-Random Binary Noise, PRBS. A Pseudo-Random Binary Signal is a periodic, deterministic signal with white noise like properties. It is generated by the difference equation

\[ u(t) = \text{rem}(A(q)u(t), 2) = \text{rem}(a_1u(t - 1) + \ldots + a_nu(t - n), 2) \]  \hspace{1cm} (5)

Here \( \text{rem}(x, 2) \) is the remainder as \( x \) is divided by 2, i.e., the calculations in (5) should be carried out modulo 2. \( u(t) \) thus only assumes the values 0 and 1. After \( u \) is generated, we can of course change that to any two levels. The vector of past inputs \( [u(t - 1) \ldots u(t - n)] \) can only assume \( 2^n \) different values. The sequence \( u \) must thus be periodic with a period of at most \( 2^n \). In fact, since \( n \) consecutive zeros would make further \( u \)s identically zero, we can eliminate that state, and the maximum period length is \( M = 2^n - 1 \). Now the actual period of the signal will depend on the choice of \( A(q) \), but it can be shown that for each \( n \) there exists choices of \( A(q) \) that give this maximum length, and the corresponding inputs are called Maximum length PRBS. The interest in maximum length PRBS follows from the following property:

Any maximum length PRBS shifting between \( \pm \pi \) has the spectrum

\[ \Phi_u(\omega) = \frac{2\pi^2}{M} \sum_{k=1}^{M-1} \delta(\omega - 2\pi k/M), \quad 0 \leq \omega < 2\pi \]  \hspace{1cm} (6)

where we ignored terms proportional to \( 1/M^2 \). Here \( \delta \) is the Dirac delta function. In the region \(-\pi \leq \omega < \pi \) there will be \( M - 1 \) frequency peaks (\( \omega = 0 \) excluded). This shows that maximum length PRBS behaves like “periodic white noise,” and is persistently exciting of order \( M - 1 \). Figure 2 shows one period of a PRBS and its spectrum.

Notice that it is essential to perform these calculations over whole periods. Generating just a part of a period of a PRBS will not give a signal with properties (6).

Like white random binary noise, PRBS has an optimal crest factor. The advantages and disadvantages of PRBS compared to binary random noise can be summarized as follows:

- If the PRBS contains whole periods, its covariance matrix will have a very special pattern. It can be analytically inverted, which will facilitate certain computations.

- As a deterministic signal, PRBS has its second order properties secured when evaluated over whole periods. For random signals, one must rely upon the law of large numbers to have good second order properties for finite samples.
For PRBS one should work with an integer number of periods to enjoy its good properties, which limits the choice of experiment length.

Multi-Sines. A natural choice of input is to form it as a sum of sinusoids:

$$u(t) = \sum_{k=1}^{d} a_k \cos(\omega_k t + \phi_k)$$  

Apart from transient effects this gives a spectrum

$$\Phi_u(\omega) = \sum_{k=1}^{d} \frac{a_k^2}{4} [\delta(\omega - \omega_k) + \delta(\omega + \omega_k)]$$

With $d, a_k$ and $\omega_k$ we can thus place the signal power very precisely to desired frequencies. The only problem with this input is the crest factor. The power of the signal is $\sum a_k^2/2$. If all sinusoids are in phase, the squared amplitude will be $(\sum a_k)^2$. The crest factor can thus be up to $\sqrt{2d}$ (if all $a_k$ are equal). The way to control the crest factor is to choose the phases $\phi_k$ so that the cosines are “as much out of phase” as possible. A simple solution is the so-called Schroeder phase choice, which means that the phases are spread as follows when the amplitudes $a_k$ are equal:

$$\phi_1 \text{ arbitrary}$$
$$\phi_k = \phi_1 - \frac{k(k-1)}{d} \pi; \quad 2 \leq k \leq d.$$  

Chirp Signals or Swept Sinusoids. A chirp signal is a sinusoid with a frequency that changes continuously over a certain band $\Omega : \omega_1 \leq \omega \leq \omega_2$ over a certain time period $0 \leq t \leq N$:

$$u(t) = A \cos(\omega_1 t + (\omega_2 - \omega_1) t^2/(2N))$$
The “instantaneous frequency” \( \omega_i \) in this signal is obtained by differentiating the argument w.r.t. time \( t \):

\[
\omega_i = \omega_1 + \frac{t}{N}(\omega_2 - \omega_1)
\]

and we see that it increases from \( \omega_1 \) to \( \omega_2 \). This signal has the same crest factor as a pure sinusoid, i.e., \( \sqrt{2} \), and it gives good control over the excited frequency band. Due to the sliding frequency, there will however also be power contributions outside the band \( \Omega \).

**Periodic Inputs**

Some of the signals above are inherently periodic, like the PRBS, or the sum of sinusoids. All of them can in any case be made periodic by simple repetition. To retain the nice frequency properties they have been designed for, the following facts must be taken into account when creating periodic signals:

- The PRBS signal must be generated over one full period, \( M = 2^n - 1 \), and then be repeated. This follows from the discussion of its second order properties.
- To create a multi-tone of period \( M \), the frequencies \( \omega_k \) in (7) must be chosen from the DFT-grid \( \omega_k = 2\pi \ell / M, \quad \ell = 0, 1, \ldots, M - 1 \).
- To make the chirp signal (10) nicely periodic with period \( N = M \), \( \omega_1 \) and \( \omega_2 \) must be chosen as \( 2\pi k / M \) for some integer \( k \). The signal generated by (10) can then be repeated an arbitrary number of times.
- To display the spectrum of a periodic signal of length \( N \) with an even number of periods without any leakage, it should be computed for the (DFT) frequencies \( \omega_k = 2\pi k / N \). (“Leakage” means that the Fourier transform is distorted by boundary effects.)

What are the advantages and disadvantages with periodic inputs?

- A signal with period \( M \) can have at most \( M \) distinct frequencies in its spectrum. It is thus persistently exciting of, at most, order \( M \). In this sense, non-periodic inputs inject more excitation into the system over a given time span.
- When a periodic input has been applied, say \( K \) periods each of length \( M \) \((N = KM)\), it is usually advisable to average the output over the periods, and work only with one period of input-output data in the model building session. This gives less data to handle. The signal to noise ratio is improved by a factor of \( K \) by this operation, at the same time as the data record is reduced by the same factor. No difference in asymptotic properties should thus result from this (unless the noise model and the dynamics model share parameters). However, several methods have a performance threshold for finite samples and poor signal-to-noise ratios, so in practice there might also be an accuracy benefit from averaging the measurement over the periods.
- A periodic input allows both formal and informal estimates of the noise level in the system. After transient effects have disappeared, the differences in the output response over the different periods must be attributed to the noise sources. This could be quite helpful in the model validation process for the important distinction between model errors and noise.
4. Preprocessing Data

When the data have been collected from the identification experiment, they are not likely to be in shape for immediate use in identification algorithms. There are several possible deficiencies in the data that should be attended to:

1. High-frequency disturbances in the data record, above the frequencies of interest to the system dynamics
2. Occasional bursts and outliers, missing data, non-continuous data records
3. Drift and offset, low-frequency disturbances, possibly of periodic character

It must be stressed that in off-line applications, one should always first plot the data in order to inspect them for these deficiencies. In this section we shall discuss how to preprocess the data so as to avoid problems in the identification procedures later.

4.1. Drifts and Detrending

Low-frequency disturbances, offsets, trends, drift, and periodic (seasonal) variations are not uncommon in data. They typically stem from external sources that we may or may not prefer to include in the modeling. There are basically two different approaches to dealing with such problems:

1. Removing the disturbances by explicit pretreatment of the data.
2. Letting the noise model take care of the disturbances.

The first approach involves removing trends and offsets by direct subtraction, while the second relies on noise models with poles on or close to the unit circle, like the ARIMA models (I for integration) much used in the so called Box-Jenkins approach.

Drifts and trends can be seen as time-varying equilibria. Straight lines or curve segments can be fitted to the data, and deviations from these time-varying means are considered. For seasonal variations, several techniques of this character have been developed for economic time series. Periodic signals are adjusted to data, and then subtracted.

Another approach would be to difference the data or, equivalently, to use ARIMA model structures, which include an integrator in the noise model. Alternatively, the noise model could be given extra flexibility to find the integrator or a complex pair of poles on the unit circle to account for periodic variations. With some knowledge of the frequencies of these slow variations, a better alternative may be to high-pass filter the data. This has the same effect of removing offsets and slow drifts, but does not push the model fit into the high frequency range as differencing does. See Section 4.2 for further comments on this.
4.2. Prefiltering

Prefiltering the input and the output data through the same filter will not change the input-output relation for a linear system:

\[ y(t) = G_o(q)u(t) + H_o \varepsilon(t) \Rightarrow L(q)y(t) = G_o(q)L(q)u(t) + L(q)H_o(q)\varepsilon(t) \]

(In the multivariable case all signals must be subjected to the same filter, so that \( L(q) \) is a multiple of the identity matrix.) The filtering however changes the noise characteristics, so the estimated model will still be affected by the prefiltering. In this section we shall discuss the role and use of this feature.

From an estimation point of view, filtering the prediction errors before making the fit, as in \( (4) \), is an important option:

\[ \varepsilon_F(t, \theta) = L(q)\varepsilon(t, \theta) = \frac{L(q)}{H(q, \theta)} (y(t) - G(q, \theta)u(t)) \]

\[ = \frac{1}{H(q, \theta)} (L(q)y(t) - G(q, \theta)L(q)u(t)) \quad (11) \]

From these expressions we see a few things:

- Filtering prediction errors is the same as filtering the observed input-output data. In the multivariable case the same filter must then be applied to all signals.

- A prefilter \( L(q) \) is equivalent to a noise model \( H(q) = 1/L(q) \). We can thus interchangeably talk about prefilters and noise models.

The basic purpose of prefiltering is to remove disturbances in the data that we do not want to include in the modeling. This actually goes hand in hand with the noise modeling aspect of prefiltering: Removing, say, a seasonal variation of a certain frequency by a band-stop filter, can also be interpreted as fixing a noise model with very high gain in this frequency band, which is a way of expressing the presence of the seasonal variation.

High-Frequency Disturbances. High frequency disturbances in the data, above the frequencies of interest for the system dynamics, indicate that the choices of sampling interval and presampling filters were not thoughtful enough. This can however be remedied by low pass filtering of the data. Also if it turns out that the sampling interval was unnecessarily short, one may always resample the data by picking every \( s \)th sample from the original record. Then, however, a digital antialiasing filter must be applied before the resampling.

Low-Frequency Disturbances. Low frequency disturbances in terms of offset, drift, and slow seasonal variations were discussed in Section 4.1. A very suitable method to deal with such problems is to apply high pass filtering. This must be considered as a clearly better alternative to data differencing.
5. Selecting Model Structures

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, and then validate this model. Typically this has to be repeated with quite a few different structures before a satisfactory model can be found.

An important feature is to take into account the intended use of the model. There is really no such thing as "a best model" in general terms. Rather, we should seek a model that is appropriate for its purpose. For example, a model that is going to be used for control design could be much simpler than one that is intended for accurate simulation studies.

The difficulties of the model structure process should not be underestimated, and it will require substantial experience to master it. Here follows a simple first cut procedure that could prove useful to try out.

Step 1: Looking at the Data
Plot the data. Look at them carefully. Try to see the dynamics with your own eyes. Can you see the effects in the outputs of the changes in the input? Can nonlinear effects be seen, like different responses at different levels, or different responses to a step up and a step down? Are there portions of the data that appear to be "messy" or carry no information? Use this insight to select portions of the data for estimation and validation purposes.

Do physical levels play a role in the model? If not, detrend the data by removing their mean values. The models will then describe how changes in the input give changes in output, but not explain the actual levels of the signals. This is the normal situation. The default situation, with good data, is to detrend by removing means, and then select the first two thirds or so of the data record for estimation purposes, and use the remaining data for validation. (All of this corresponds to the "Data Quickstart" in the MATLAB Identification Toolbox.)

Step 2: Getting a Feel for the Difficulties
Compute and display the spectral analysis frequency response estimate, the correlation analysis impulse response estimate, as well as a fourth order ARX model with a delay estimated from the correlation analysis, and a default order state-space model computed by a subspace method. (All of this corresponds to the "Estimate Quickstart" in the MATLAB Identification Toolbox.) Look at the agreement between the

- Spectral Analysis estimate and the ARX and state-space models' frequency functions.
- Correlation Analysis estimate and the ARX and state-space models' transient responses.
- Measured Validation Data output and the ARX and state-space models' simulated outputs. We call this the Model Output Plot.

If these agreements are reasonable, the problem is not so difficult, and a relatively simple linear model will do a good job. Some fine tuning of model orders and noise models may have to be made, and we can proceed to Step 4. Otherwise go to Step 3.
Step 3: Examining the Difficulties

There may be several reasons why the comparisons in Step 2 did not look good. This step discusses the most common ones, and how they can be handled:

- **Model Unstable**: The ARX or state-space model may turn out to be unstable, but could still be useful for control purposes. Then change to a 5- or 10-step ahead prediction instead of simulation when the agreement between measured and model outputs is considered.

- **Feedback in Data**: If there is feedback from the output to the input, due to some regulator, then the spectral and correlations analysis estimates are not reliable. Discrepancies between these estimates and the ARX and state-space models can therefore be disregarded in this case. In residual analysis of the parametric models, feedback in data can also be visible as correlation between residuals and input for negative lags.

- **Noise Model**: If the state-space model is clearly better than the ARX model at reproducing the measured output this is an indication that the disturbances have a substantial influence, and it will be necessary to carefully model them.

- **Model Order**: If a fourth order model does not give a good Model Output plot, try eighth order. If the fit clearly improves, it follows that higher order models will be required, but that linear models could be sufficient.

- **Additional Inputs**: If the Model Output fit has not significantly improved by the tests so far, think over the physics of the application. Are there more signals that have been, or could be, measured that might influence the output? If so, include these among the inputs and try again a fourth order ARX model from all the inputs. (Note that the inputs need not at all be control signals; anything measurable, including disturbances, should be treated as inputs).

- **Nonlinear Effects**: If the fit between measured and model output is still bad, consider the physics of the application. Are there nonlinear effects in the system? In that case, form the nonlinearities from the measured data. This could be as simple as forming the product of voltage and current measurements, if it is the electrical power that is the driving stimulus in, say, a heating process, and temperature is the output. This is of course application dependent. It does not cost very much work, however, to form a number of additional inputs by reasonable non-linear transformations of the measured signals, and just test whether inclusion of them improves the fit.

- **General Nonlinear Mappings**: In some applications physical insight may be lacking, so it is difficult to come up with structured non-linearities on physical grounds. In such cases, nonlinear, black box models could be a solution, such as Artificial Neural Networks, etc.

- **Still Problems?** If none of these tests leads to a model that is able to reproduce the validation data reasonably well, the conclusion might be that a sufficiently good model cannot be produced from the data. There may be many reasons for this. The most important one is that the data simply do not contain sufficient information, e.g., due to bad signal to noise ratios, large and non-stationary disturbances, varying system properties, etc.

Otherwise, use the insights on which inputs to use and which model orders to expect and proceed to Step 4.

Step 4: Fine Tuning Orders and Noise Structures

For real data there is no such thing as a "correct model structure." However, different
structures can give quite different model quality. The only way to find this out is to try out a number of different structures and compare the properties of the obtained models. There are a few things to look for in these comparisons:

- **Fit Between Simulated and Measured Output.** Look at the fit between the model’s simulated output and the measured one for the validation data. Formally, pick that model, for which this number is the lowest. In practice, it is better to be more pragmatic, and also take into account the model complexity, and whether the important features of the output response are captured.

- **Residual Analysis Test.** Form the model residuals, i.e. the prediction errors as in (4) for the resulting model. For a good model the cross correlation function between residuals and input does not go significantly outside the confidence region.

- **Pole Zero Cancellations.** If the pole-zero plot (including confidence intervals) indicates pole-zero cancellations in the dynamics, this suggests that lower order models can be used. In particular, if it turns out that the order of ARX models has to be increased to get a good fit, but that pole-zero cancellations are indicated, then the extra poles are just introduced to describe the noise. Then try other model structures $G(q, \theta)$ with this number of poles.

What Model Structures Should be Tested?
Well, any amount of time can be spent on checking out a very large number of structures. It often takes just a few seconds to compute and evaluate a model in a certain structure, so one should have a generous attitude to the testing. However, experience shows that when the basic properties of the system’s behavior have been picked up, it is not much use to fine tune orders in absurdum just to improve the fit by fractions of percents. For ARX models and state-space models estimated by subspace methods there are also efficient algorithms for handling many model structures in parallel.

**Multivariable Systems**
Multivariable systems are often more challenging to model. In particular, systems with several outputs could be difficult. A basic reason for the difficulties is that the couplings between several inputs and outputs leads to more complex models: The structures involved are richer and more parameters will be required to obtain a good fit.

Generally speaking, it is preferable to work with state-space models in the multivariable case, since the model structure complexity is easier to deal with. It is essentially just a matter of choosing the model order.

**Working with Subsets of the Input-Output Channels:**
In the process of identifying good models of a system it is often useful to select subsets of the input and output channels. Partial models of the system’s behavior will then be constructed. It might not, for example, be clear if all measured inputs have a significant influence on the outputs. That is most easily tested by removing an input channel from the data, building a model for how the output(s) depend on the remaining input channels, and checking if there is a significant deterioration in the model output’s fit to the measured one. See also the discussion under Step 3 above. Generally speaking, the fit gets better when more inputs are included and worse when more outputs are included. To understand the latter fact, it should be realized that a model that has to explain the behavior of several outputs has a tougher job than one that simply must account for a single output. If there are difficulties to obtain good models for a multi-output system, it might thus be wise to model one output at a time, to find out which are the difficult ones to handle. Models that just are to be used for simulations could very well be built up from single-output
models, for one output at a time. However, models for prediction and control will be able to produce better results if constructed for all outputs simultaneously. This follows from the fact that knowing the set of all previous output channels gives a better basis for prediction than just knowing the past outputs in one channel.

**Step 5: Accepting the model**

The final step is to accept, at least for the time being, the model to be used for its intended application. Note the following, though: *No matter how good an estimated model looks on the computer screen, it has only picked up a simple reflection of reality. Surprisingly often, however, this is sufficient for rational decision making.*

6. Some Applications

**A Fighter Aircraft**

For the development of aircraft, a substantial amount of work is allocated to construct a mathematical model of its dynamic behavior. This is required both for the simulators, for the synthesis of autopilots and for the analysis of its properties. Substantial physical insight is utilized, as well as wind tunnel experiments is used in the course of this work, and a most important source of information comes from the test flights. Figure 6 shows some results from test flights of the Swedish aircraft Gripen, developed by SAAB Military Aircraft AB, Sweden. We will use these data to build a model of the pitch channel, i.e. how the pitch rate is affected by the three control signals, elevator, canard, and leading edge flap. The elevator in this case corresponds to aileron combinations at the back of the wings, while separate action is achieved from the ailerons at the leading edge (the front of the wings). The canards are a separate set of rudders at the front of the wings. The aircraft is unstable in the pitch channel at this flight condition, so clearly the experiment was carried out under closed loop control.

To develop models of the aircraft’s pitch channel from these data, we proceed as follows. The data set is first detrended, so that the means of each signal is removed. Then the data is split into one set consisting of the first 90 samples, to be used for estimation, and a validation data set consisting of the remaining 90 samples. As a main tool to screen models we computed the RMS fit between the measured output and the 10-step ahead predicted output according to the

![Pitch rate, Leading Edge Flap, Elevator, Canard](image-url)
different models. In these calculation the whole data set was used – in order to let transients die out – but the fit was computed only for the validation part of the data. The reason for using 10 step ahead predictions rather than simulations is that the pitch cannel of the aircraft is unstable, and so will most of the estimated models also be. A simulation comparison may therefore be misleading.

A typical starting ARX model, using 4 past outputs and 4 past values of each of the 3 inputs, gave a fit according to Figure 4. We see that we get a good fit, so it seems reasonable that we can do a good modeling job with fairly simple models. As a next step we calculate 1000 ARX models corresponding to orders in inputs and outputs and delays ranging between 1 and 10. (In this case all 3 input orders we kept the same.) The best 1-step ahead prediction fit to the validation data turned out to be for a model

\[ y(t) + a_1 y(t - 1) + \ldots + a_{n_a} y(t - n_a) = b_1^{(1)} u_1(t - 1) + b_2^{(2)} u_2(t - 1) + b_3^{(3)} u_3(t - 1) + e(t) \]  

with \( n_a = 8 \). See Figure 7. Note in particular that models that use many parameters are considerably much worse on the validation data. Models of the kind (12) with other values of \( n_a \) were also estimated, as well as ARMAX models and state-space models using the N4SID method. A comparison plot based for several such models is shown in Figure 5. The best 10-step ahead prediction fit is obtained for the ARX model with \( n_a = 4 \). (Note, though, that the best 1-step ahead prediction is obtained for \( n_a = 8 \), as was said above.) The comparison for that model is shown in Figure 6. The result of residual analysis for this model on validation data is shown in Figure 8. We see that this simple model with 7 parameters is capable

![Figure 4: Measured output (dash-dotted line) and 10-step ahead predicted output (solid line) for aircraft validation data, using an ARX model with \( n_a = 4, n_k^1 = 4, n_k^2 = 1, k = 1, 2, 3 \).](image)

of reproducing new measurements quite well, at the same time it is not falsified by residual analysis.
Figure 5: As figure 4 but for several different models.

Figure 6: As figure 4 but for the best ARX-model.
Figure 7: Comparisons of the 1-step ahead prediction error for 1000 ARX-models for the aircraft data.

Figure 8: Residual analysis the best ARX-model for the validation aircraft data.
Figure 9: From the pulp factory at Skutskär, Sweden. The plots show The κ-number of the pulp flowing into a buffer vessel. The κ-number of the pulp coming out from the buffer vessel. Flow out from the buffer vessel. Level in the buffer vessel. The sampling interval is 4 minutes, and the time scale shown in hours.

Buffer Vessel Dynamics

This example concerns a typical problem in process industry. It is taken from the pulp factory in Skutskär, Sweden. Wood chips are cooked in the digester and the resulting pulp travels through several vessels where it is washed, bleached etc.

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. Figure 9 shows data from one buffer vessel. We denote the measurements as follows:

\[ y(t) : \text{The } \kappa\text{-number of the pulp flowing out} \]
\[ u(t) : \text{The } \kappa\text{-number of the pulp flowing in} \]
\[ f(t) : \text{The output flow} \]
\[ h(t) : \text{The level of the vessel} \]

The problem is to determine the residence time in the buffer vessel. (The κ-number is a quality property that in this context can be seen as a marker allowing us to trace the pulp.)

To estimate the residence time of the vessel it is natural to estimate the dynamics from \( u \) to \( y \). That should show how long time it takes for a change in the input to have an effect on the output.

We can visually inspect the input-output data and see that the delay seems to be at least an hour or two. The sampling rate may therefore be too fast and we resample the data (decimate it) by a factor of 3, thus giving a sampling interval of 12 minutes. We proceed as before, remove the means from the κ-number signals, split into estimation and validation data and estimate simple ARX-models. This turns out to give quite bad results.

According to the recipe of Section 5 we should then contemplate if there are more input signals that may affect the process. Yes, clearly the flow and level of the vessel should have something to do with the dynamics, so we include these two inputs. The best model output comparison was achieved for an ARX model with 4 parameters associated with the output and each of the
inputs, a delay of 12 from \( u \) and a delay of 1 from \( f \) and \( h \). This comparison is shown in Figure 10. This does not look good.

![Figure 10: Measured and simulated model output](image)

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 the flow was a plug flow (no mixing in the vessel) the vessel would have a dynamics of a pure delay equal to vessel volume divided by flow. This ratio, which has dimension time, is really the natural time scale of the process, in the sense that the delay would be constant in this time scale for a plug flow, even if vessel flow and level vary.

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 flow divided by volume. In MATLAB terms this will be \( z = [y,u] \); \( pf = f/h \);

```matlab
    t = 1:length(z)
    newt = interp1(cumsum(pf+0.00001),t,[pf(1);cumsum(pf)]');
    newz = interp1(t,z,newt);
    y1=newz(:,1); u1=newz(:,2)
```

(The small added number to \( pf \) is in order to overcome those time points where the flow is zero.) The resampled data are shown in Figure 11. We now apply the same procedure to the resampled data \( u_1 \) to \( y_1 \). The best ARX model fit was obtained for

\[
y_1(t) + a_1 y(t - 1) + \ldots + a_4 y(t - 1) = b_1 u_1(t - 9) + e(t)
\]

The comparison is shown in Figure 12. This “looks good”.

The impulse responses of these models are shown in Figure 13. We see a delay of about 2 hours and then a time constant of about 1 hour. The vessel thus gives a pure delay as well as some mixing of the contents. The two impulse responses are in good agreement, if we take into account their uncertainties.
Figure 11: The input and output $\kappa$-numbers resampled according to the text.

Figure 12: The measured validation output $y_1$ (dash-dotted line) together with simulated model outputs from resampled $u_1$. An ARX(419) model is shown as well as an OE model of the same orders.

Figure 13: The impulse response of the ARX (solid) and OE (dashed) models. The right figure shows also the corresponding estimated 99% confidence intervals.
Bibliography


Practical Issues of System Identification

This is a survey of practical issues in System Identification.

Keywords: identification