17

SYSTEM IDENTIFICATION IN PRACTICE

In this book we have dealt with the theory of system identification. Performing the identification task in practice enhances the "art side" of the topic. Experience, intuition, and insights will then play important roles. In this final chapter we shall discuss the system identification techniques as a toolbox for investigating, understanding, and mastering real-life systems. We shall first, in Section 17.1, describe the system identification tool in the hand of the user: interactive computing. Section 17.2 discusses the practical side of identification; how to approach the task. Then, in Section 17.3 we discuss a few applications to real data sets. Finally, in Section 17.4 we try to answer the ultimate question: What does system identification have to offer for real problems in engineering and applied science?

17.1 THE TOOL: INTERACTIVE SOFTWARE

The work to produce a model by identification is characterized by the following sequence:

1. Specify a model structure.
2. The computer delivers the best model in this structure.
3. Evaluate the properties of this model.
4. Test a new structure, go to step 1.

See Figure 17.1, which is a more elaborate version of Figure 1.10. 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:

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

B *Nonparametric identification methods*
Estimation of covariances, Fourier transforms, correlation and spectral analysis and so on.

1

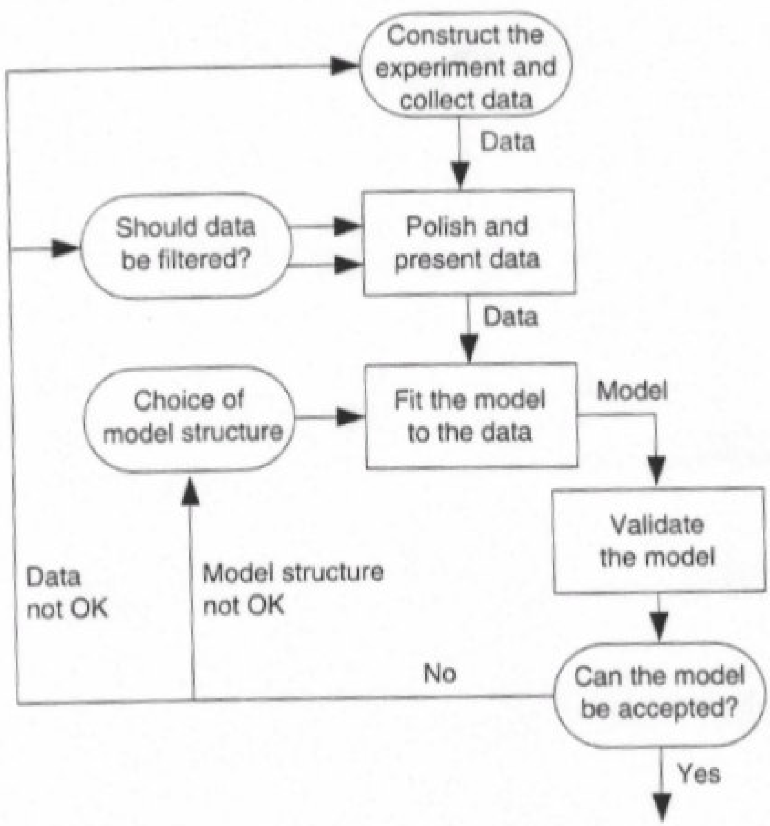

Figure 17.1 Identification cycle. Rectangles: the computer's main responsibility. Ovals: the user's main responsibility.

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), Ljung (1995), 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 described in Section 4.2 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 (4.62), (4.84), and (4.91). A Graphical User Interface helps the user both to keep track of identified models and to guide him or her to available techniques.

Other packages of general type include PIM, Landau (1990), the Identification module in Matrix-X, Van Overschee et.al. (1994), Adapt$_X$, Larimore (1997), and the Frequency-Domain Identification Toolbox in MATLAB, Kollàr (1994). A tool for specifically dealing with gray box identification, IDKIT, is described in Graebe (1990).

17.2 THE PRACTICAL SIDE OF SYSTEM 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 the techniques of Chapter 16, and then validate this model. Typically this has to be repeated with quite a few different structures before a satisfactory model can be found.

The difficulties of this process should not be underestimated, and it will require substantial experience to master it. Here follows a procedure that could prove useful to try out. This is adapted from Ljung (1995).

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. See (16.20).
- **Feedback in Data:** If there is feedback from the output to the input, due to some regulator, then the spectral and correlation analysis estimates, as well as the state-space model, are not reliable. Discrepancies between these estimates and the ARX model 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 again 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 nonlinear 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. See Sections 5.4–5.6.
- **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.** For a good model, the cross correlation function between residuals and input does not go significantly outside the confidence region. See Section 16.6. A clear peak at lag k shows that the effect from input $u(t-k)$ on $y(t)$ is not properly described. A rule of thumb is that a slowly varying cross correlation function outside the confidence region is an indication of too few poles, while sharper peaks indicate too few zeros or wrong delays.

  For models that are to be used for control design, it is quite valuable to display the result of residual analysis in the frequency domain as in Example 16.3.
- **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 ARMAX, OE, or BJ model structures with an A- or F-polynomial of an order equal to that of the number of non-cancelled 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.*

17.3 SOME APPLICATIONS

The Hairdryer; A Laboratory Scale Application

Consider as a real, but laboratory scale process, Feedback's Process Trainer PT326, depicted in Figure 17.2. Its function is like a hairdryer: air is fanned through a tube and heated at the inlet. The input u is the power of the heating device, which is just a mesh of resistor wires. The output is the outlet air temperature. It should be said that the process is well behaved: it has reasonably simple dynamics with quite small disturbances. It also allows measurements with good signal-to-noise ratio.

Transient Response. The step response of the process is given in Figure 17.3. It reveals that the dynamics is simple, with no oscillatory poles, the dominating time constant is around 0.4 seconds, and there is a pure time delay of about 0.14 seconds.

Experiment Design. To collect data for further analysis, a few decisions have to be taken. Following the discussion of Section 13.7, we select a sampling interval of 0.08 s, since Figure 17.3 clearly shows that the dominating time constant is not much less than 0.4 s. A shorter sampling interval would also mean several delays between the (sampled) input and output sequences. The input was chosen to be a binary random signal shifting between 35 and 65 W. The probability of shifting the input at each sample was set to 0.2. A record of 1000 samples was collected, and the data set is shown in Figure 17.4. As a first step, the sample means of the input and

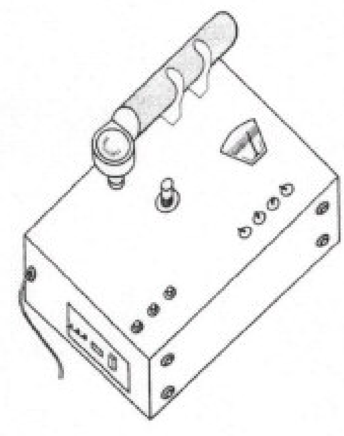

Figure 17.2 The hairdryer process.

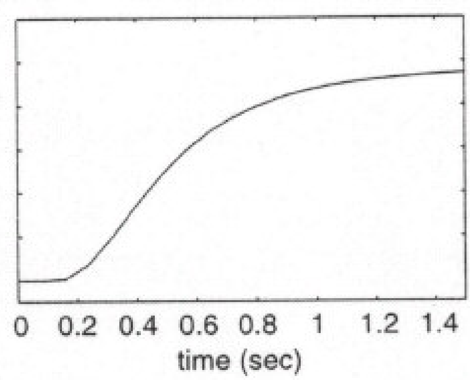

Figure 17.3 The step response from the process.

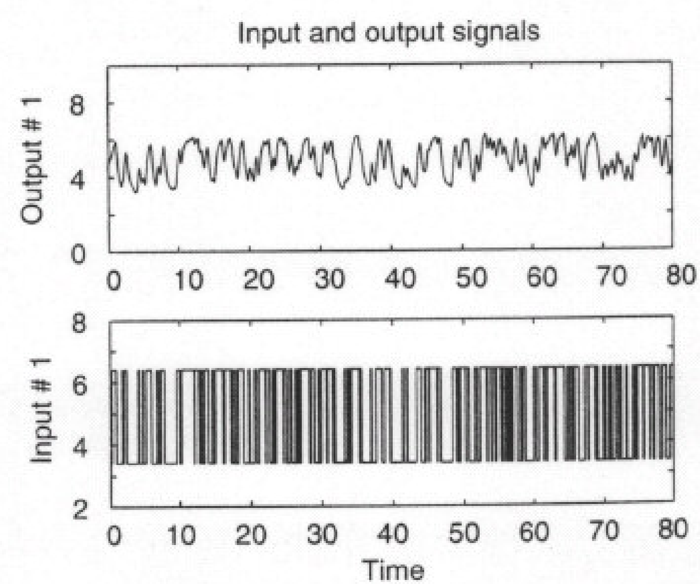

Figure 17.4 The data set from the process trainer. This is the same set as **dryer2** supplied with the System Identification Toolbox.

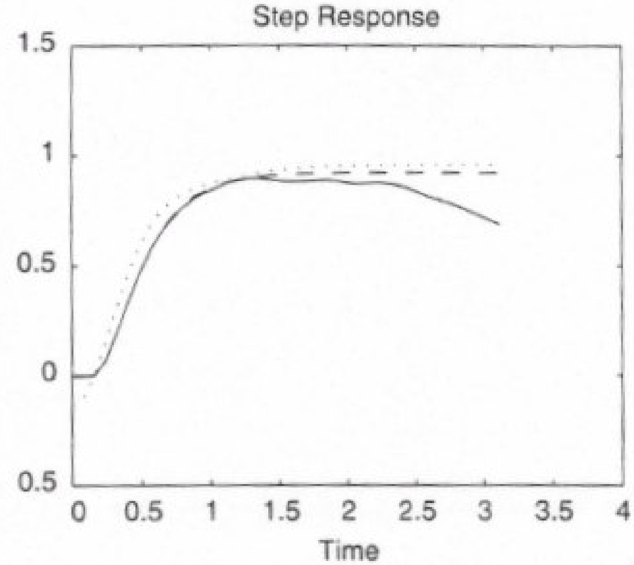

Figure 17.5 The step responses according to the correlation analysis estimate (solid), the fourth order ARX model (dashed), and the 3rd order state-space model (dotted).

output sequences were removed (see the discussion of Section 14.1). Then the data set was split into two halves, the first to be used for estimation, and the second one for validation.

Preliminary Models. Following Step 2 in Section 17.2 we estimate the step/impulse response by correlation analysis, as described in Section 6.1, compute the spectral analysis estimate of the frequency function, as well as a fourth order ARX model and a state-space model (using a subspace method, according to (7.66)). The order of this model is selected automatically, and turned out to be 3 in this case. The results of these calculations are shown in Figures 17.5, 17.6, and 17.7. These

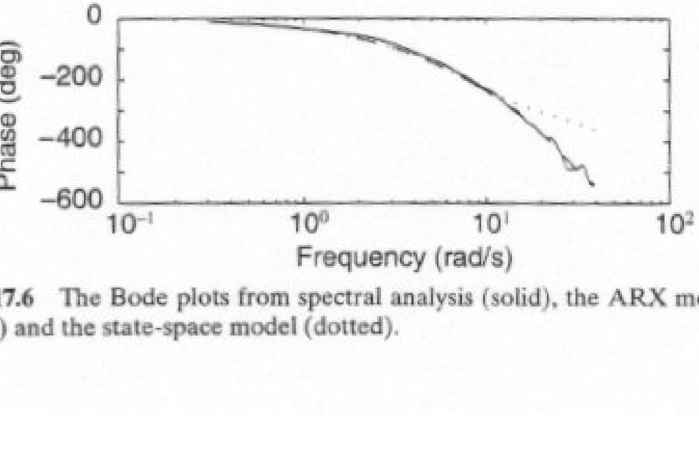

Figure 17.6 The Bode plots from spectral analysis (solid), the ARX model (dashed) and the state-space model (dotted).

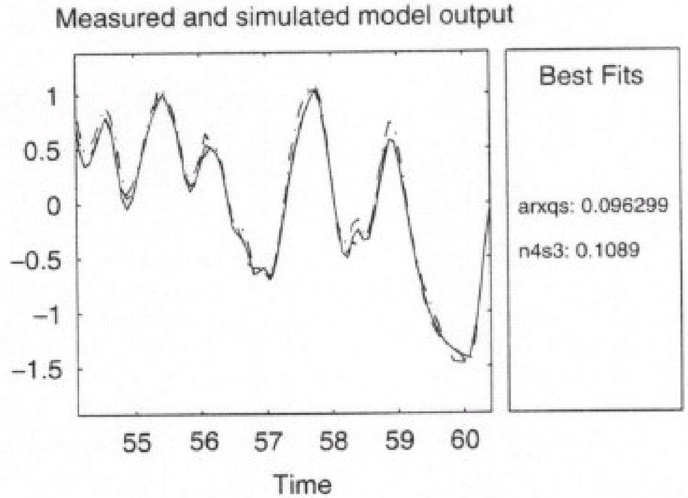

Figure 17.7 The measured output (dash-dotted) for the validation data set, together with the output from the ARX model (arxqs, solid), and the state-space model (n4s3, dashed) when simulated with the input sequence from the validation data set. The figures shown are the RMS-values of the difference between measured and simulated output.

plots show that we have good agreement between the models computed in different ways. This is a clear indication that the models have picked up essential features of the true process. Moreover, the comparisons in Figure 17.7 show that even these "immediate" models are able to reproduce the input-output behavior quite well. All this indicates, according to Step 4 of Section 17.2, that a linear model will do fine, and some further work to fine-tune orders and delays is all that remains.

Further Models. To look into suitable orders and delays, we compute, simultaneously, 1000 ARX-models of the type

$$y(t) + a_1 y(t-1) + \ldots + a_{n_a} y(t-n_a) = b_1 u(t-n_k) + \ldots + b_{n_b} u(t-n_k-n_b+1)$$

consisting of all combinations n_a, n_b, and n_k in the range 1 to 10. The different ARX models will be referred to as ARX(n_a, n_b, n_k). The prediction errors of each model are then computed for the validation data, and their sum of squares is computed. The result is shown in Figure 17.8, where the fit for the models is depicted as a function of the number of parameters used. Only the fit for best model with a given number of parameters is shown. The overall best fit is obtained for a model with 15 parameters, which turns out to be $n_a = 6$, $n_b = 9$ and $n_k = 2$. The figure also shows that almost as good a fit is obtained also for models with much less parameters; like 4. In this case the best orders turn out to be $n_a = 2, n_b = 2$, and $n_k = 3$. These models are added to the model output comparisons in Figure 17.9. We see that the higher order ARX model is able to reproduce the validation data best, but that the differences between the models really are minor. We compute also a state-space model of order 6 as well as an ARMAX-model with $n_a = 3, n_b = 3, n_c = 2$, and

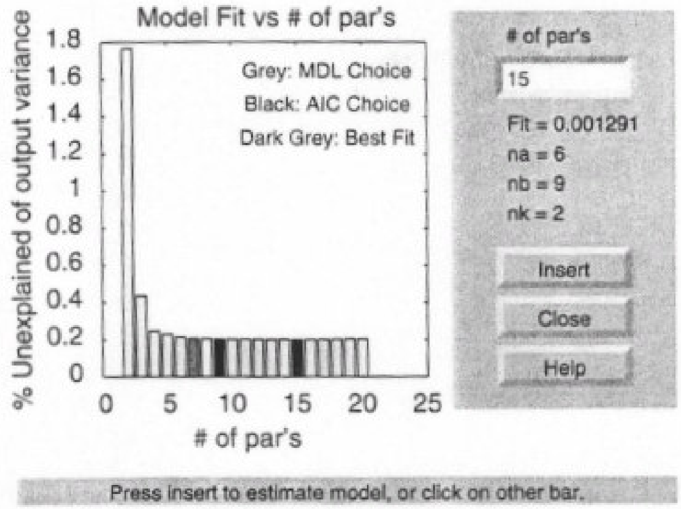

Figure 17.8 The best fit to validation data for ARX-models as a function of the number of used parameters.

$n_k = 2$. These models are also shown in the same plot. The residuals of the models (computed from the validation data) are analyzed in Figure 17.10. It shows that the ARMAX(3,3,2,2) model and the ARX(9,6,2) model both give residuals that pass whiteness and independence tests, while the model ARX(2,2,3) shows statistically significant correlation between past inputs and the residuals.

Final Choice of Model. Based on this analysis we conclude that there are many linear models that give a good fit to the system. The ARX(9,6,2) model shows the best fit to validation data, but is at the same time only marginally better than the simpler, third order ARMAX(3,3,2,2) model. Both also pass the residual analysis

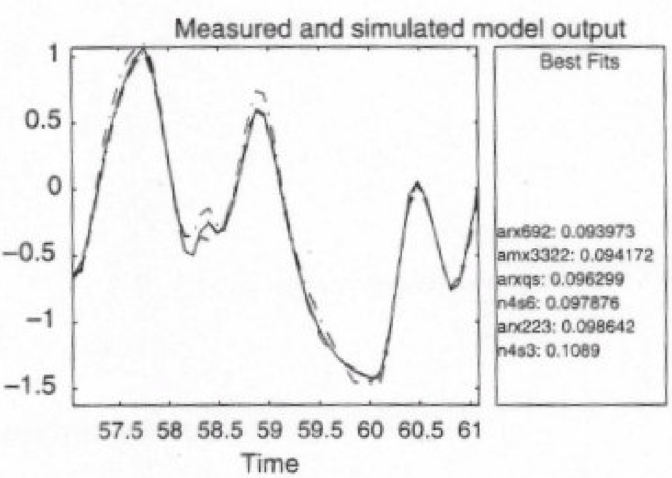

Figure 17.9 Comparisons between several different models, based on the fit between measured and simulated output.

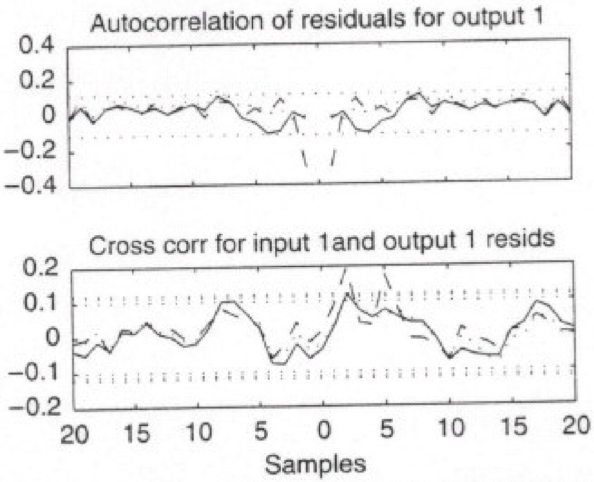

Figure 17.10 Results from the residual analysis of three different models: solid line ARX(9,6,2), dashed line ARX(2,2,3). Dotted lines ARMAX(3,3,2,2). The horizontal lines mark the confidence regions.

tests. It seems reasonable to pick this simple model as the final choice. The numerical value is

$$y(t) - 1.4898y(t-1) + 0.7025y(t-2) - 0.1123y(t-3) = 0.0039u(t-2)$$
$$+ 0.0621u(t-3) + 0.0284u(t-4) + e(t) - 0.5474e(t-1)$$
$$+ 0.2236e(t-2)$$

The estimated standard deviations of the 8 parameters are

$$[0.0574 \quad 0.0849 \quad 0.0333 \quad 0.0015 \quad 0.0023 \quad 0.0055 \quad 0.0710 \quad 0.0523]$$

We see that the coefficient for $u(t-2)$ is on the borderline from being significantly different from zero. This is the reason why models with delay n_k=3 also work well. However, the small effect from the term $u(t-2)$ does give an improved fit.

The estimated standard deviation of the noise source $e(t)$ is 0.0388.

A Fighter Aircraft

Consider the aircraft Example 1.2 with data shown in Figure 1.6. Note that these data were collected under closed loop operation.

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 different models. In these

531

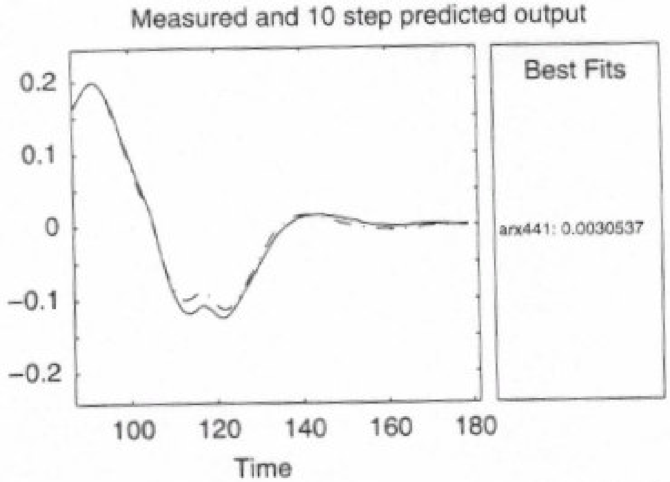

Figure 17.11 Measured output (dash-dotted line) and 10-step ahead predicted output (solid line) for aircraft validation data, using an ARX model with $n_a = 4, n_b^k = 4, n_k^k = 1, k = 1, 2, 3$.

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 17.11. 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 were kept the same.) The best 1-step ahead prediction fit to the validation data turned out to be for a model

$$\begin{aligned} y(t) &+ a_1 y(t-1) + \ldots + a_{n_a} y(t - n_a) \\ &= b_1^{(1)} u_1(t-1) + b_1^{(2)} u_2(t-1) + b_1^{(3)} u_3(t-1) + e(t) \qquad (17.1) \end{aligned}$$

with $n_a = 8$. See Figure 17.14. Note in particular that models that use many parameters are considerably much worse for the validation data. Models of the kind (17.1) 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 17.12. 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 17.13. The result of residual analysis for this model on validation data is shown in Figure 17.15. We see that this simple model with 7 parameters is capable of reproducing new measurements quite well, at the same time it is not falsified by residual analysis.

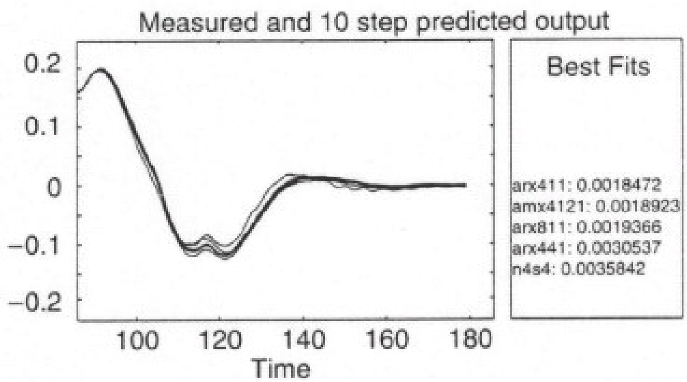

Figure 17.12 As Figure 17.11 but for several different models.

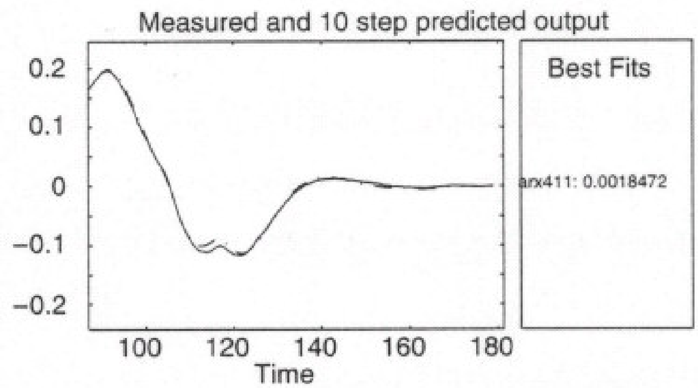

Figure 17.13 As Figure 17.11 but for the best ARX model.

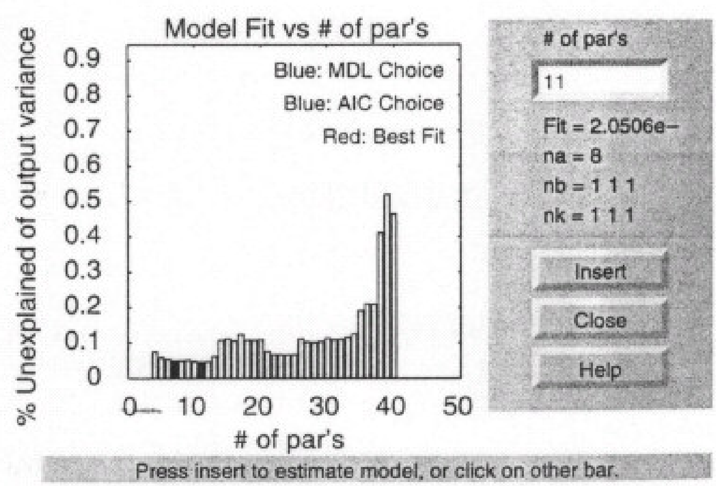

Figure 17.14 Comparisons of the 1-step ahead prediction error for 1000 ARX-models for the aircraft data.

533

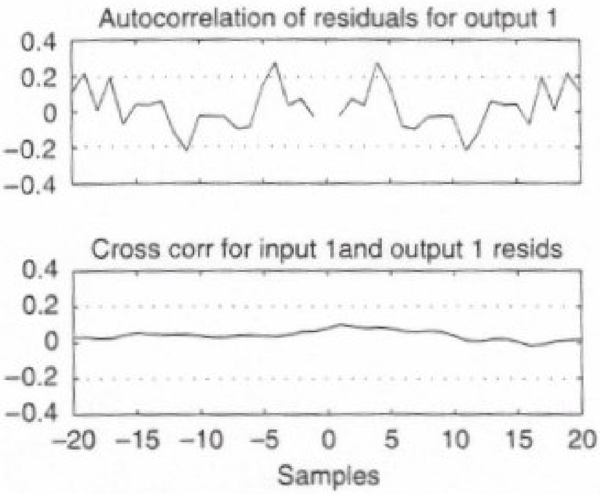

Figure 17.15 Residual analysis the best ARX model for the validation aircraft data.

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

$y(t)$: The κ-number of the pulp flowing out

$u(t)$: The κ-number of the pulp flowing in

$f(t)$: The output flow

$h(t)$: 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

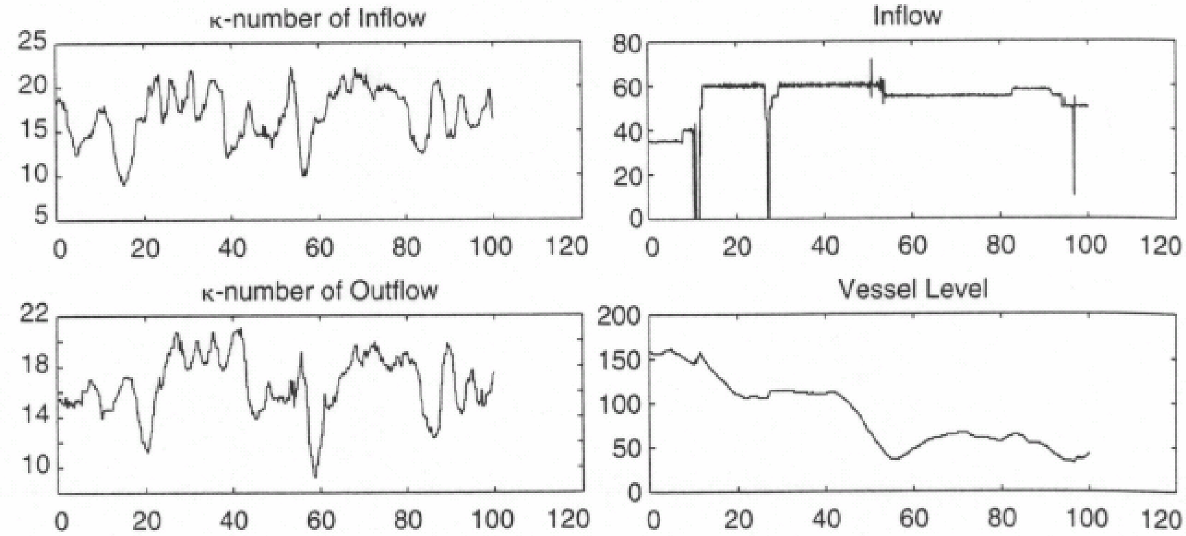

Figure 17.16 From the pulp factory at Skutskär, Sweden. The plots show the κ-number of the pulp flowing into a buffer vessel. The κ-number of the pulp coming out from the buffer vessel. Flow out from the buffer vessel. Level in the buffer vessel. The sampling interval is 4 minutes, and the time scale shown in hours.

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 17.2 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 17.17. This does not look good.

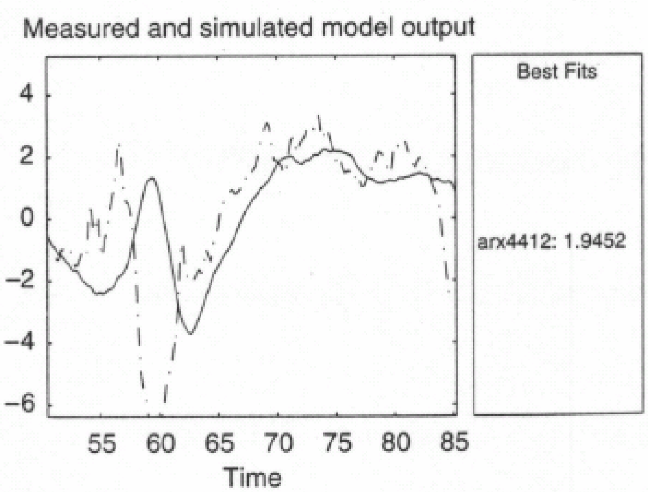

Figure 17.17 The measured validation output y (dash-dotted line) together with the best linear simulated model output for the system from u, f, h to y.

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;
t =1:length(z)
newt = interp1(cumsum(pf+0.00001),t,[pf(1):sum(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 17.18. We now apply the same procedure to the resampled data u_1 and y_1. The best ARX model fit was obtained for

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

Slightly better fit was obtained for an output-error model (4.25) with the same orders ($n_b = 1, n_f = 4, n_k = 9$). The comparison is shown in Figure 17.19. This "looks good."

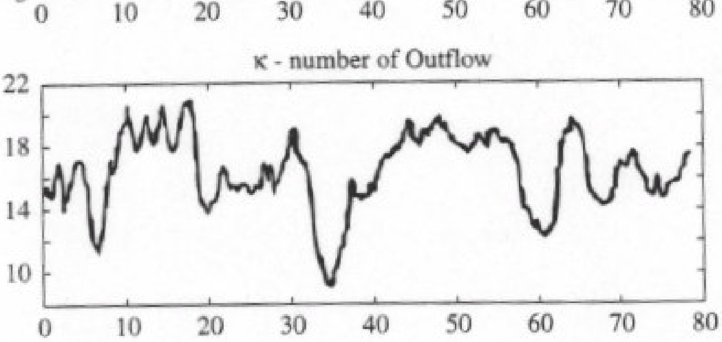

Figure 17.18 The input and output κ-numbers resampled according to the text.

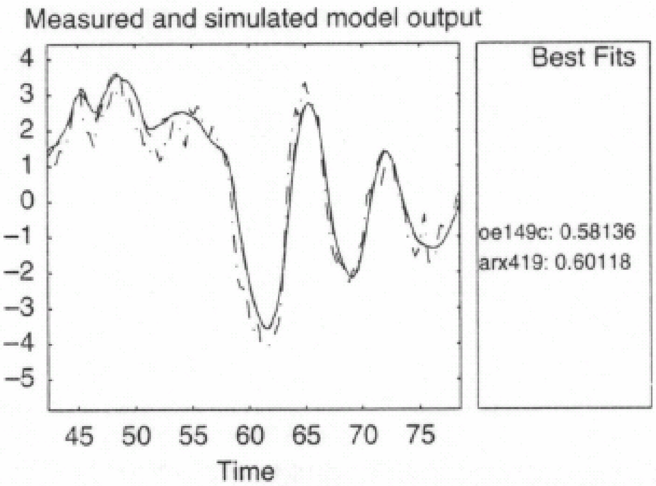

Figure 17.19 The measured validation output y_1 (dash-dotted line) together with simulated model outputs from resampled u_1. An ARX(419) model is shown as well as an OE model of the same orders.

The impulse responses of these models are shown in Figure 17.20. We see a delay of about 1.75 hours and then a time constant of about 2 hours. 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. See Andersson and Pucar (1995)for a more comprehensive treatment of the data in this example.

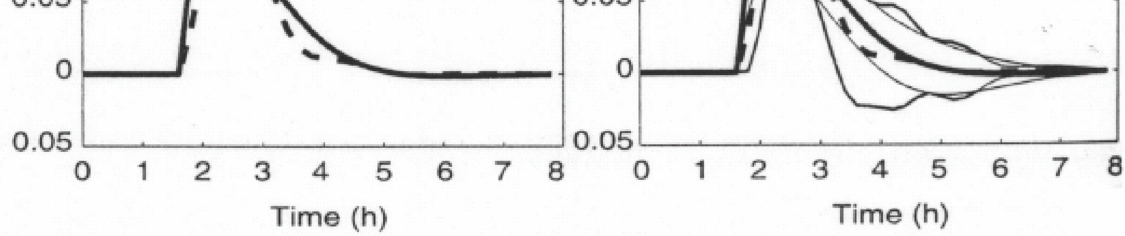

Figure 17.20 The impulse response of the ARX (solid) and OE (dashed) models. The right figure shows also the corresponding estimated 99% confidence intervals.

17.4 WHAT DOES SYSTEM IDENTIFICATION HAVE TO OFFER?

System identification techniques form a versatile tool for many problems in science and engineering. The techniques are, as such, application independent. The value of the tool has been evidenced by numerous applications in diverse fields. For example, the proceedings from the IFAC symposium series in System Identification contain thousands of successful applications from a wide selection of areas. Still, there are some limitations associated with the techniques, and we shall in this final section give some comments on this.

Adaptive and Robust Designs: Have They Made Modeling Obsolete?

As we discussed in Section 1.2, models of dynamical systems are instrumental for many purposes: prediction, control, simulation, filter design, reconstruction of measurements, and so on. It is sometimes claimed that the need for a model can be circumvented by more elaborate solutions: adaptive mechanisms where the decision parameters are directly adjusted or robust designs that are insensitive to the correctness of the underlying model. One should note, though, that adaptive schemes typically can be interpreted as the recursive identification algorithms described in Chapter 11 applied to a specific model structure (e.g., the model parameterized in terms of the corresponding optimal regulator); see Chapter 7 in Ljung and Söderström (1983). The model-building feature is thus very much present also in adaptive mechanisms.

Robust design is based on a nominal model and is determined so that good operation is secured even if the actual system deviates from the nominal model. Usually, a neighborhood around the nominal model can be specified within which performance degradation is acceptable. It is then a very useful fact that models obtained by system identification can be delivered with a quality tag: estimated deviations form a true description in the parameter domain or in the frequency domain. Such models are thus suited for robust design.

Limitations: Data Quality

It is obvious that the limitation of the use of system identification techniques is linked to the availability of good data and good model structures. Without a reasonable data record not much can be done, and there are several reasons why such a record cannot be obtained in certain applications. A first and quite obvious reason is that the time scale of the process is so slow that any informative data records by necessity will be short. Ecological and economical systems may clearly suffer from this problem. Another reason is that the input may not be open to manipulations, either by its nature or due to safety and production requirements. The signal-to-noise ratio could then be bad, and identifiability (informative data sets) perhaps cannot be guaranteed. Bad signal-to-noise ratios can, in theory, be compensated for by longer data records. Even if the plant, as such, admits long experimentation time, it may not always be a feasible way out, due to time variations in the process, drift, slow disturbances, and so on.

Finally, even when we are allowed to manipulate the inputs, can measure for long periods, and have good signal-to-noise ratios, it may still be difficult to obtain a good data record. The prime reason for this is the presence of unmeasurable disturbances that do not fit well into the standard picture of "stationary stochastic processes". We have discussed how to cope with such slow disturbances in Section

14.1 and how to handle occasional "bursts" by robust norms (Section 15.2), and such measures may often be successful. The fact remains though: data quality must be a prime concern in system identification applications. This also determines the cost of the exercise.

Limitations: Model Structures

It is trivial that a bad model structure cannot offer a good model, regardless of the amount and quality of the available data. For example, the ARX model structure in Figure 17.17 can never provide a good description of the buffer vessel dynamics even if fitted to data collected over several years. The crucial nonlinear mechanisms must be built in, and this requires physical insight.

The first problem thus is whether the process (around its operation point of interest) admits a standard, linear, ready-made ("black-box") model description, or whether a tailor-made model set must be constructed. In the first case, our chances of success are good; in the second, we have to resort to some physical insight before a model can be estimated or hoping that the nonlinear dynamics can be picked up by a nonlinear black-box structure. This problem clearly is application dependent and therefore not so much discussed in the identification literature. It cannot, however, be sufficiently stressed that the key to success lies here: Thinking, intuition, and insights cannot be made obsolete by automated model construction.