2 A Journey into Identification

An overview of identification is provided by means of examples and a simulation case study. The objective is to take the reader through an illustrative tour of identification while giving glimpses of the technical aspects. The main topics of interest are identifiability, quality of parameter estimates and model development.

In Chapter 1, we learned that the measured response usually consists of both the effects of the input as well as that of the unmeasured disturbances and/or measurement errors, known as the *deterministic* and *stochastic* effects, respectively (recall Figure 1.6). Further, from §1.3.1, recall that the ability to identify a model (of a specified structure) uniquely and precisely depends on two factors: (i) the input characteristics and (ii) the relative proportions of the deterministic and stochastic effects in the measurement, quantified by the *signal-to-noise ratio* (SNR). In a broader sense, the uniqueness of identified models and is not only connected to the input design but also with the estimation algorithm and the model structure itself. Collectively, these factors are formally analyzed using the concept of *identifiability*.

In the following section, we offer brief glimpses of the aforementioned technical concepts, especially the roles of model and input in identification, primarily through illustrations on simple examples[1]. In addition, the phenomenon of **overfitting**, a technical term for "excessive" mathematical modeling, is also elucidated through the example of a polynomial fit. The centerpiece of this chapter is a complete case study on the identification of a liquid level system using the systematic procedure presented in §1.4.

2.1 IDENTIFIABILITY

As remarked earlier, the ability to identify a unique model for a given system depends on three critical aspects:

1. **Model:** This is solely a property of the model and is concerned with *whether there exists a unique (one-to-one) mapping between the model and the parameters being estimated*[2].
2. **Experimental conditions:** Specifically we refer to the type of input and sampling rate, among which the first factor is the most influential and is a significant design factor. The question is then *whether the input has generated the requisite information required to distinguish between two candidate models*.
3. **Estimation method:** A number of methods are available for parameter estimation as discussed in §1.4.4. The question is *whether the estimation method is capable of estimating the "true" parameters if infinite samples are available*, which is termed as an *asymptotic property* of the estimator. The technical term is *consistency*.

Furthermore, it is necessary to know whether the estimated model, in principle, converges to the "true" system. This aspect depends on the true system description itself. The three factors above

[1]Formal discussions on identifiability appear in §18.6 and §22.3.

[2]An additional concern that remains is if the model is complex enough to explain the underlying system, but we shall defer a discussion on this issue until Chapter 21.

and the system's characteristics together determine the ability to discover the true system from observations, which is formally termed as *system identifiability*.

Model identifiability (or sometimes *parameter identifiability*) is the term that is used to formally describe the ability to obtain a unique model. In analyzing model identifiability, the analysis is carried out at a theoretical level, where the characteristics of the input and estimation algorithms are assumed to be ideal. Thus, we are primarily concerned with the model itself.

The role of input in system identifiability is studied through the concept of *informative experiments*. Example 2.2 below is aimed at illustrating the underlying ideas. The third property, which is solely known as *consistency*, will be taken up for illustration at a later stage, in Chapters 12 and 13. System identifiability is formally studied

For simplicity we shall consider only deterministic systems. The first example below illustrates the notion of model identifiability, specifically for so-called *parametrized* models.

Example 2.1: Model or Parameter Identifiability

Consider fitting the model $y[k,\boldsymbol{\theta}] = \theta_1\theta_2 u[k]$ to a given data, where $u[k]$ and $y[k]$ are the input and output of a system, while $\boldsymbol{\theta} = \begin{bmatrix}\theta_1 & \theta_2\end{bmatrix}^T$ is the parameter vector to be identified.

The prediction (denoted by a hat) of this model to a given input is

$$\hat{y}[k,\boldsymbol{\theta}] = \theta_1\theta_2 u[k] \tag{2.1}$$

Regardless of the input, two different parameter values $\boldsymbol{\theta}_1$ and $\boldsymbol{\theta}_2$ produce identical predictions. Stated mathematically,

$$\hat{y}[k,\boldsymbol{\theta}_1] = \hat{y}[k,\boldsymbol{\theta}_2] \implies \boldsymbol{\theta}_1 = \boldsymbol{\theta}_2 \tag{2.2}$$

Therefore, the mapping from the parameter space to the model (predictor) space is not one-to-one. Formally, the model is said to be not (globally) identifiable. Consequently, it is not possible to arrive at a unique estimate of $\boldsymbol{\theta}$.

On the other hand, if the model is re-parametrized in terms of a single parameter $\beta = \theta_1\theta_2$, then the model is identifiable at all points in the β space.

The above example suggests that *re-parametrization of a model*, in this case from a higher-dimensional to a lower-dimensional parameter space, *can improve identifiability for that model*. Further, a model can be rendered locally identifiable by adding suitable constraints on the parameters (see Exercise **E2.2.**). A formal treatment of these concepts, particularly for linear dynamical systems, is presented in §18.6. It must be also mentioned that, on several occasions, models may not be in the parametrized forms, known as *non-parametric models*; the concept of identifiability even applies to such models.

We now turn our attention to the second aspect of identifiability, which is concerned with recovery of the true system from the data and is largely governed by how the experiment has been performed.

The basic principle in identification is that *good models can only be obtained from "good" data*. If the data is generated by a poor excitation, then there is very little scope for building a good model simply because the data lacks *information* (recall the analogy of an interview process).

The model realized by the user heavily depends on the experimental conditions that prevailed at the data acquisition stage. The process may not have been excited over the entire input-output range. As a result, the model does not "see" all aspects of the relationship. In the case of high-order linear systems, the input may have only excited the lower-order dynamics or for non-linear systems it is that the input has excited only a subset of non-linearities.

The following example illustrates how an improper choice of input can lead to loss of identifiability. Once again, the illustration is in a deterministic setting.

Example 2.2: Role of Input in Identifiability

Consider a linear time-invariant (LTI) system governed by the following input-output relationship (3^{rd}-order finite impulse response system):

$$y[k] = b_1 u[k-1] + b_2 u[k-2] + b_3 u[k-3] \tag{2.3}$$

with $b_1 = 1$, $b_2 = 0.6$ and $b_3 = 0.3$. Suppose a sinusoidal input of the form $u[k] = \sin(2\pi(0.1)k) = \sin(\omega_0 k)$ is applied to the system.

Under the input, the output in (2.3) is

$$y[k] = b_1 \sin(\omega_0 k - \phi) + b_2 \sin(\omega_0 k - 2\phi) + b_3 \sin(\omega_0 - 3\phi)$$

which can be re-written as

$$\begin{aligned} y[k] &= \left(b_1 + \frac{b_2}{2\cos\phi}\right)\sin(\omega_0 k - \phi) + \left(b_3 + \frac{b_2}{2\cos\phi}\right)\sin(\omega_0 k - 3\phi) \\ &= b_1' \sin(\omega_0 k - \phi) + b_3' \sin(\omega_0 k - 3\phi) \end{aligned} \tag{2.4}$$

Thus, a 3-parameter model manifests as a 2-parameter model when viewed through the lens of a mono-frequency input. Unfortunately, it is not possible to uniquely recover b_1, b_2 and b_3 from b_1' and b_3'.

Observe that one can re-write (2.4) in terms of the pair $(\sin(\omega_0 k - \phi), \sin(\omega_0 k - 2\phi))$ or $(\sin(\omega_0 k - 2\phi), \sin(\omega_0 k - 3\phi))$[3].

Regardless of the way (2.4) is written, with a sinusoid of single frequency, only two of the three explanatory variables $u[k-1]$, $u[k-2]$ and $u[k-3]$ are unique. Consequently, only two of the three parameters b_1, b_2 and b_3, can be uniquely estimated.

The situation encountered in the above example is formally termed as loss of *identifiability* due to insufficient information, which here is due to lack of sufficient input excitation. From a statistical viewpoint, the data *does not have sufficient information* or *evidence* to discriminate between a second-order and higher-order finite impulse response models.

The observations above offer a preview of an important fact in identification that we shall learn in §22.3 - *identifiability is guaranteed (at least for linear time-invariant systems) when the input is persistently exciting*, which roughly translates to an input containing sinusoids of almost all frequencies. These concepts are foundational to input design (see §22.3).

Observe that in Example 2.2 the input generates enough information in the data to estimate a two-parameter model. Further, it may be verified that if the input contains additional frequency components, for e.g., $u[k] = \sin(\omega_0 k) + \sin(\omega_1 k)$, then it is possible to estimate all three parameters uniquely (see Exercise **E2.5.**).

Identifiability can also be at loss due to improper choice of sampling rates. At the scale of observation, only those time constants can be identified that are commensurate with the sampling interval T_s. For instance, it is not possible to detect modes of the process (or even delays) that have settling times less than T_s. Collectively, poor input design and sampling rate lead to *lack of sufficient information*.

The following section addresses the role of SNR in the precision of model estimates, i.e., it is not sufficient to have good excitation in the input but it is also important to have a significant input strength relative to noise for obtaining reliable models.

---

[3]Equation (2.4) is due to the trigonometric relationship $\sin(\omega_0 k - 2\phi) = \frac{1}{2\cos\omega_0}(\sin(\omega_0 k - \phi) + \sin(\omega_0 k - 3\phi))$.

2.2 SIGNAL-TO-NOISE RATIO

Even with good excitation, the stochastic effects in the measurements can be high enough to be detrimental to model quality. For example, choosing sampling rates much faster than the pace at which outputs change can bring in more noise than actual process variation.

The relative contributions of deterministic excitation and random variations are quantified by a measure known as the *signal-to-noise ratio* (SNR) .

$$\mathsf{SNR} = \frac{\text{Variance of signal}}{\text{Variance of noise}} \tag{2.5}$$

The term *signal* here refers to the *true response* of the system. Having a good SNR is critical to obtaining reliable parameter estimates, regardless of the estimation method. This is qualitatively understood as follows. *Fitting a model intuitively amounts to explaining variations in the output*. If a significant portion of the variations in the measurement is due to noise, then the contribution of the input weakens and hence the ability to precisely estimate the model as well. An alternative viewpoint is that the SNR represents the ratio of effects due to known variable versus the uncertainties. Thus, the lower the SNR, the more ambiguous is the estimate of the input-output model.

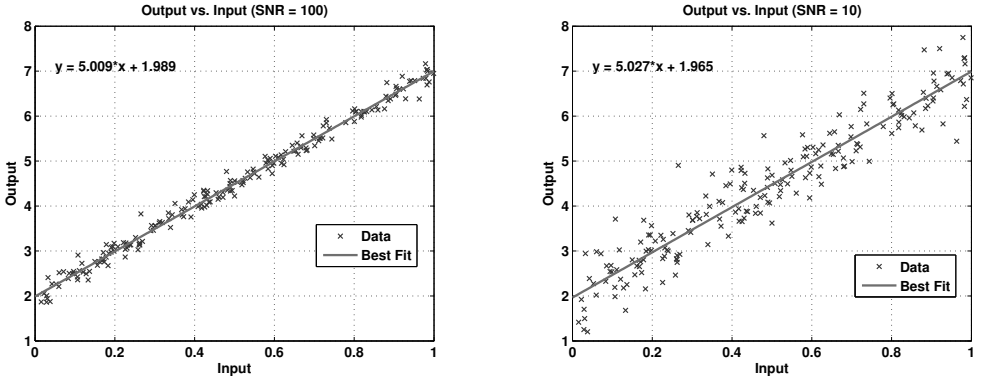
The following example aids in understanding the above concepts.

Example 2.3: Effect of SNR on Parameter Estimation

The relationship between the output $y[k]$ and the input $u[k]$ of a system is known to be

$$y[k] = b_1 u[k] + b_0$$

with $b_1 = 5$ and $b_0 = 2$. Relationships such as the one above are common in linear calibration sensors, proportional controllers and so on. Assume that the input and the measured output $y_m[k] = y[k] + v[k]$ is available, where the measurement error $v[k]$ is assumed to be random.



(a) $\sigma_{\hat{b}_1} = 0.036, \sigma_{\hat{b}_0} = 0.02$ (b) $\sigma_{\hat{b}_1} = 0.114, \sigma_{\hat{b}_0} = 0.064$

FIGURE 2.1 The best fit and the error in the parameter estimates depend on the SNR.

The best linear fit and the parameter estimates for two different settings of $\mathsf{SNR} = \frac{\sigma_y^2}{\sigma_v^2}$, namely (i) $\mathsf{SNR} = 100$ and (ii) $\mathsf{SNR} = 10$ obtained from $N = 200$ samples of $(y_m[k], u[k])$ data are shown in Figures 2.1(a) and 2.1(b), respectively. The quantities $\hat{b}_i$ and $\sigma_{\hat{b}_i}$ are the estimates and standard errors in the estimates of b_i, $i = 0, 1$, respectively. While the estimates do not vary significantly with the change in SNR, the errors in $\hat{b}_i$, i.e., $\sigma_{\hat{b}_i}$ in the estimates increase roughly by a factor of three. The increase in fact, is theoretically given by $\sqrt{\frac{100}{10}} = 3.162$. The lower the SNR, the lower the reliability (confidence) of the resulting parameter estimate.

□

Listing 2.1 MATLAB code for Example 2.3

```
% Create input-output data
uk = rand(200,1); yk = 5*uk + 2;

% Add noise to get the measurement
ek = randn(200,1);
ykm1 = yk + ek*sqrt(var(yk)/100);       % SNR 100
ykm2 = yk + ek*sqrt(var(yk)/10);        % SNR 10

% Plot data and use curve fitting to obtain the best fits
figure; plot(uk,ykm1,'x','linewidth',1);
figure; plot(uk,ykm2,'x','linewidth',1);

% Compute LS estimates and their standard errors
Phi = [uk ones(200,1)]
thetah1 = Phi \ ykm1;
err1 = ykm1 - Phi*thetah1;
errth1 = sqrt(diag(inv(Phi'*Phi)*sum(err1.^2)/198));

thetah2 = Phi \ ykm2;
err2 = ykm2 - Phi*thetah2;
errth2 = sqrt(diag(inv(Phi'*Phi)*sum(err2.^2)/198));
```

In practice, the SNR is not known a priori. However, in several applications, the noise variance can be estimated from steady-state data and the input amplitude can be adjusted to achieve a desired SNR in an identification experiment.

The signal-to-noise ratio has a significant role to play in all estimation exercises. For instance, in signal estimation, it quantifies the separability of the signal and noise components of a measurement.

2.3 OVERFITTING

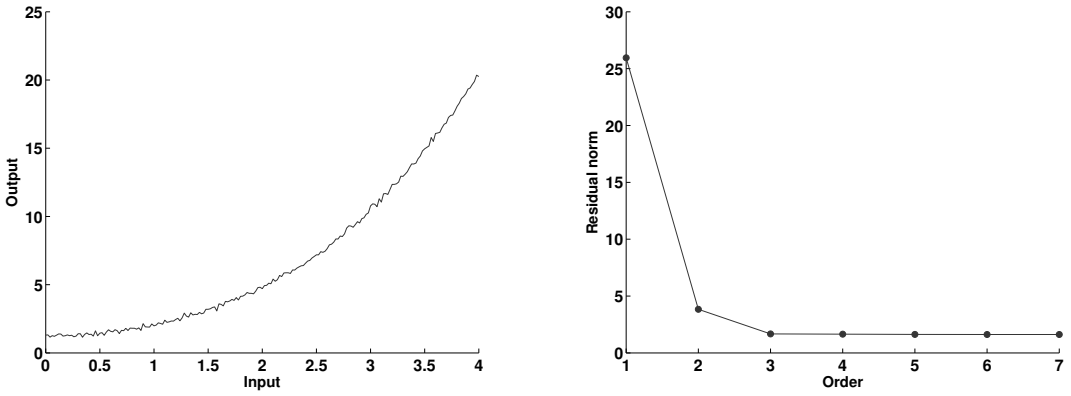
Overfitting occurs when the model is trained to capture the "local" features of the data rather than the "global" characteristics. In identification, this situation arises when one misconstrues the stochastic effects in the data as a part of the deterministic (input-output) effects, i.e., when the chance variations in the response are attributed to the changes in the input variables. This situation occurs when the user "over-specifies" the complexity of the deterministic portion in a bid to explain the output as accurately as possible. The extreme case is when the entire variability in the output is attributed solely to the variations in the input. An obvious benefit from increasing the complexity of the deterministic model is the improved fit on the training data, i.e., lower prediction error. However, the reduction in the bias (of the prediction) comes with the risk of high standard errors (variance) in the model (parameter) estimates. Further, with such models we are led to poor predictions on a fresh data set, on some occasions even unstable (unbounded) ones.

The following example illustrates the risks associated with overfitting on a simple *static system*. Section 2.4 demonstrates this phenomenon on a dynamic system.

Example 2.4: Overfitting

The input-output data of a system is shown in Figure 2.2(a). A visual inspection reveals that a polynomial fit can capture the relationship reasonably well. In order to choose the appropriate order of the polynomial, the sum squared error of residual is drafted against the order of the polynomial, shown in Figure 2.2(b). A good choice of the order is three, since the improvement in the residual norm at higher orders is minuscule.

It would be useful to know if the third order is indeed the optimal choice. This cannot



(a) Input-output data of Example 2.4 (b) Norm of residuals vs. order of polynomial fit

FIGURE 2.2 Training data and order determination in Example 2.4.

be determined using the training data alone; a test data set is also required to assess its predictive abilities (the procedure of cross-validation). Polynomials of order three to five are fit and tested on a fresh data set. Figure 2.3 shows the predictions of the respective models. It is clear that the third-order polynomial model offers the best compromise between the predictions on the training and test data. The fitted polynomial (predictor) is

$$\hat{y}[k] = \underset{(\pm 0.03)}{1.183} + \underset{(\pm 0.07)}{0.384}\, u[k] + \underset{(\pm 0.04)}{0.314}\, u^2[k] + \underset{(\pm 0.007)}{0.198}\, u^3[k]$$

where the values in brackets are the standard errors in the coefficient estimates (§14.3.3 derives the necessary expressions).

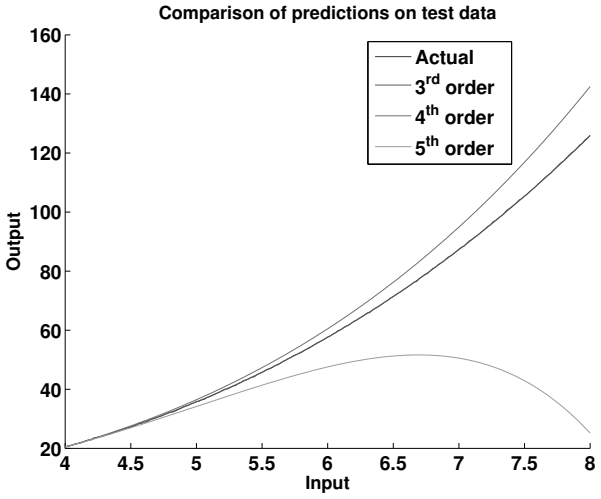


FIGURE 2.3 Cross-validation of polynomial models in Example 2.4.

It is useful to compare the estimated model with the true model used for data generation

$$y[k] = 1.2 + 0.4u[k] + 0.3u^2[k] + 0.2u^3[k] + v[k]$$

where $v[k]$ is an ideal random noise (unpredictable stochastic signal) such that the SNR is set to 10. The chosen order and the coefficient estimates agree very well with the true ones. Note that we have discovered the true model without any prior knowledge.

The fourth and fifth order models *negligibly* lower residual norms on the training data and therefore fail miserably on the test data. In fact the fifth-order model produces unstable

predictions[4]. The reason for the poor performance of the higher-order models is that they have attempted to explain the chance variations using the input. All empirical modeling exercises potentially run into this risk, but as the example shows, it can be avoided by a careful cross-validation.

The instability of predictions is one of the perils in overfitting, which can be avoided by examining the errors in parameter estimates of these models in conjunction with plot in Figure 2.2(b); the errors in parameters of 4^{th} order fit are quite high as shown below,

$$\hat{y}[k] = \underset{(\pm 0.04)}{1.23} + \underset{(\pm 0.13)}{0.14}\, u[k] + \underset{(\pm 0.6)}{0.14}\, u^2[k] + \underset{(\pm 0.054)}{0.085}\, u^3[k] + \underset{(\pm 0.007)}{0.015}\, u^4[k]$$

re-affirming that this model is not suited to the given data.

It must be noted that the parameter estimates and the error values change with each run of the data (due to the use of a different noise realization) - but the message remains the same.

An important remark is in order here. *Over-parametrization has to be viewed not with reference to the "true" model, but with respect to the model that can be identifiable*. There are two reasons for this. Firstly, the process is usually much more complicated than the model that is being fit. Therefore a comparison of the model with the process has little meaning. Secondly, as we have seen in Section 2.1, the number of parameters and the order of the model that can be estimated is largely governed by the input excitation.

Listing 2.2 MATLAB code for Example 2.4

```
% Generate training data
xvec = (0:0.02:4)';
yvec = 1.2 + 0.4*xvec + 0.3*xvec.^2 + 0.2*xvec.^3;

% Add measurement noise
ymeas = yvec + 0.4*randn(length(yvec),1);

for i = 1:7,
eval(['[pvechat' num2str(i) ',S' num2str(i) ']=␣polyfit(xvec,ymeas,1);']);
end

% Calculation of standard error in estimates
Rinv3 = inv(S3.R);
covmat3 = (Rinv3*Rinv3')*S3.normr/S3.df;
errvec3 = sqrt(diag(covmat3));

Rinv4 = inv(S4.R);
covmat4 = (Rinv4*Rinv4')*S4.normr/S4.df;
errvec4 = sqrt(diag(covmat4))';

% Test data for model validation
xtest = (4:0.02:8)';
ytest = 1.2 + 0.4*xtest + 0.3*xtest.^2 + 0.2*xtest.^3;
ytmeas = ytest + 0.4*randn(length(ytest),1);

% Predictions of the model on test data
[yhat3,delta3] = polyval(pvechat3,xtest,S3);
[yhat4,delta4] = polyval(pvechat4,xtest,S4);
[yhat5,delta5] = polyval(pvechat5,xtest,S5);

% Plot training data
figure; plot(xvec,ymeas)
```

[4]Not all higher-order models necessarily produce unstable predictions, but this example points to one such possibility.

```
% Plot the residual norm vs. order
errnorm = [S1.normr S2.normr S3.normr S4.normr S5.normr S6.normr S7.normr]';
figure; plot((1:7),errnorm,'-o','Markerface','blue')

% Plot test results
figure; plot(xtest,[ytmeas yhat3 yhat4 yhat5])
legend({'Actual' ; '3^{rd}␣order' ; '4^{th}␣order' ; '5^{th}␣order'})
```

To summarize, overfitting arises when the proposed model contains parameters in excess of what are *identifiable*. The appropriate level of model complexity is determined by ensuring an acceptable trade-off between predictions on the training and test data.

A systematic model estimation exercise can greatly aid in preventing the model from running into the risks of overfitting. While there exist no strict rules that can completely prevent this phenomenon, identification theory offers useful guidelines based on an approximate analysis. The method adopted for determining the optimal order in the previous example, although heuristic, is representative of the rigorous information-theoretic approaches that are widely used for order selection (Section 22.6.3). The polynomial fits and the errors in estimates have been computed using the LS method and the associated expressions provided in §14.3.

Next, we turn to the focal example of this chapter. An empirical model of a simple dynamic first-order buffer process is developed. It is shown how one can systematically build input-output and state-space models from data with as little knowledge of the process as possible. The main objective is to provide the precursory experience of the options and challenges in a typical identification problem.

2.4 A MODELING EXAMPLE: LIQUID LEVEL SYSTEM

2.4.1 THE PHYSICAL PROCESS

The process of interest is a buffer system as shown in Figure 2.4 consisting of a cylindrical tank, with inlet and outlet flow rates of F_i m^3/hr and F_o m^3/hr, respectively.

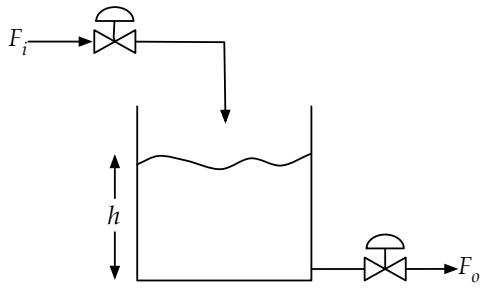


FIGURE 2.4 Schematic of the liquid level system discussed in Section 2.4.

The objective is to build a model that explains changes in level $h(t)$ with respect to changes in inlet flow rate $F_i(t)$. The end-use of this model could be in simulation, control and/or monitoring of the liquid level system.

2.4.2 DATA GENERATION

In a realistic situation, one would perform real-time experiments on the liquid level process with a custom-designed input. The input design would be usually preceded by a preliminary experiment to obtain essential process characteristics for determining the appropriate sampling rate and the frequency content of the input. Keeping in view the main objective of the example, the experiment is

replaced by the *simulation of a first-principles model* with an appropriate input design and sampling interval in place.

Model for simulation

The *deterministic* first-principles model of the liquid level system (based on conservation of mass) is

$$\frac{dh(t)}{dt} + \frac{1}{A_c} C_v \sqrt{h(t)} \quad = \quad \frac{1}{A_c} F_i(t) \tag{2.6}$$

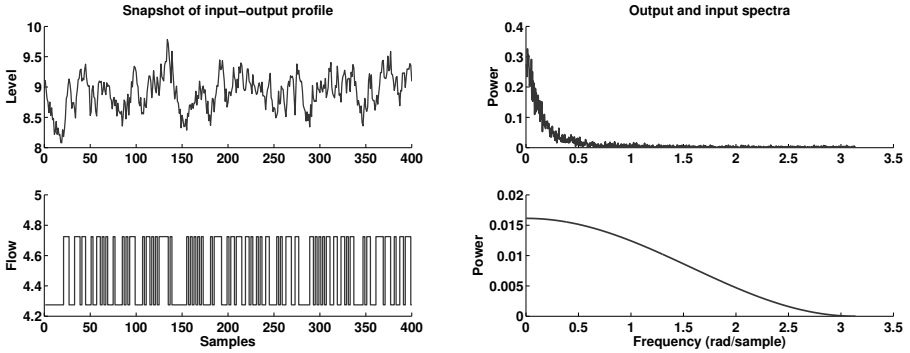
with the usual assumptions of (i) incompressible fluid and (ii) the outlet flow rate being proportional to the square root of the pressure head (valve equation)

$$F_o(t) = C_v \sqrt{h(t)} \tag{2.7}$$

where C_v is the valve coefficient at the outlet. The quantity A_c in (2.6) is the cross-sectional area of the cylindrical tank. The system is brought to a steady state before exciting it with the designed input. With the operating conditions set to $F_i(t) = 4.5$ cu. ft. / min., $C_v = 1.5$ and $A_c = 0.5$ ft^2, the nominal level is 9 ft.

Input used for data generation

The input used for simulation is a discrete-time *pseudo-random binary signal* (PRBS), which consists of short- and long-duration pulses switching between two levels around the nominal operating value. The PRBS input has certain advantages over other inputs as we shall learn later (in §22.3). Figure 2.5(a) (bottom panel) shows the input profile. The long duration pulses (low frequency) secure the gain information while the not-so-long ones are useful in capturing the dynamics of the process.



(a) Snapshot of the flow rate and level data for the liquid level system example

(b) Power spectra of the input and output signals

FIGURE 2.5 Time-trends and spectra of flow and level measurements in the identification of the liquid level system.

The response of the sampled-data (ZOH-process-sampler) system is obtained by numerically integrating the non-linear ODE in (2.6) from time $t = 0$ to $t = 2045$ and observing at $T_s = 1$ min. sampling intervals (in the MATLAB / SIMULINK environment). To render the simulation realistic, a measurement error in the form of an *ideal random* (unpredictable) noise sequence is added to the true response such that the output SNR is set to a value of 10. A snapshot of the liquid level response is shown in the top panel of Figure 2.5(a).

```
% Perform experiment, collect input-output data
sim('liqlevel_exch01.mdl');
dataset = iddata(yk,uk,1);
dataset.OutputName = 'Level'; dataset.InputName = 'Flow';
dataset.TimeUnit = 'minutes';
```

The rest of the section exemplifies the development of a *linear, time-invariant discrete-time numerical* model that can explain the dynamics of the sampled output (level *measurement*) to changes in discrete-time input (flow) following the steps outlined in Section 1.4 and Figure 1.7.

2.4.3 DATA VISUALIZATION AND PRELIMINARY ANALYSIS

As a first step, the input-output data is visually examined for the presence of any drifts, outliers, etc. Figure 2.5(a) shows the absence of any polynomial trends and other anomalies of concern (such as outliers, missing data). Some minimal pre-processing is still necessary, however, to fulfill the assumption of linearity, which holds only for deviations of variables around a nominal operating point (typically steady-state) and not absolute values themselves (essentially zero intercept models).

Steady-state can be determined experimentally before introducing changes to the input. When such experiments have not been performed (we shall assume this to be the case here), an alternative is to use the average of the readings as a nominal operating point. Consequently, a simple mean-centering operation of the input-output data is used to generate the required deviation variables. Therefore, for the rest of the analysis we shall work with

$$y[k] = \tilde{y}[k] - \bar{y}; \quad u[k] = \tilde{u}[k] - \bar{u}$$

where the variables with $\tilde{\cdot}$ are absolute-valued and the quantities under the bar are simple averages of the respective variables.

```
figure
subplot(211)
plot(dataset.y(1:400),'linewidth',1.5);
subplot(212)
stairs(dataset.u(1:400),'linewidth',1.5);

figure
subplot(211)
[Pyy,wvec] = periodogram(yk - mean(yk),[],length(yk),'twosided');
plot(wvec(2:end/2),Pyy(2:end/2),'linewidth',2);
subplot(212)
[Puu,wvec] = periodogram(uk - mean(uk),[],length(uk),'twosided');
plot(wvec(2:end/2),Puu(2:end/2),'linewidth',2);
```

Spectral analysis

It is useful to examine the frequency content of the output signal so as to obtain insights into the *filtering* nature of the process. A *spectral plot* consisting of a plot of power vs. frequency (on the x-axis) is widely used for this purpose. High power in a frequency band implies the strong presence of those frequency components. Figure 2.5(b) shows the spectra of the input and output signals where the power on the vertical axis is a measure of its strength. Most of the input power has been packed in the low- to mid-frequency band. This is in fact a part of the input design strategy because liquid level systems are *low-pass filters*, meaning that any high frequency fluctuations in the input will yield poor response. The output spectrum shown in the top panel of Figure 2.5(b) bears testimony to this fact. All input components with frequencies greater than approximately 0.7 rad/sample have been significantly attenuated by the system.

Partitioning the data

For modeling purposes, the data is *partitioned* into two sets: one partition consisting of the first $N = 1500$ samples for training the model, while the second set consisting of the remainder of the data, used in cross-validation of the model. Both data sets are expressed in terms of deviation variables with the nominal operating point determined from the training data.

```
% Partition data into training and test data
% Use means of training data as the reference point

datatrain = dataset(1:1500); datatest = dataset(1501:end);

[Ztrain,Tr] = detrend(datatrain,0);
Ztest = detrend(datatest,Tr);
```

Following the systematic procedure outlined in Figure 1.7, we begin the model development with the estimation of elementary response or *non-parametric* models (e.g., impulse and step responses).

2.4.4 BUILDING NON-PARAMETRIC MODELS

Non-parametric models provide insights into several important (deterministic) process characteristics with *minimal* assumptions:

i. *Time-delay*: The impulse response (cross-correlation) method is the classical approach for delay estimation and we shall follow it here. At a later stage in the text, (in §22.5) we shall learn the use of frequency-domain methods to obtain efficient estimates of time-delay.
ii. *Gain*: It is the final change in the output to a unit change in the input. Step response coefficients are ideally suited for estimating gain.
iii. *Time-constant*: When the process is approximated to have first-order dynamics, the time constant is roughly the time taken for the response to reach 63% of the final value to a step change in the input. Once again the step response is naturally suited for estimating this parameter.

The step response coefficients can also be used to detect what is known as an *integrating effect*, a term used to characterize unusually very long settling times (very large time constants). This situation does not apply to the present case study. One can also infer the disturbance characteristics using non-parametric methods, but that is reserved for a later discussion.

Impulse Response estimates

The IR estimates are obtained by fitting a finite-length impulse response (FIR) model:

$$y[k] \approx \sum_{l=0}^{M} g[l]u[k-l] \tag{2.8}$$

to the data using the *least squares* method. The coefficients $\{g[.]\}$ form the $(M+1)$ long impulse response sequence that is of interest. When the system has a delay of D samples, then the first D IR coefficients are identically zero. The corresponding estimated coefficients will be however "small" non-zero values. In practice, a test of significance is performed wherein the estimated coefficients lower than a statistically determined threshold are termed as insignificant.

```
% Estimate impulse response and plot it
irestoptions = impulseestOptions;
irestoptions.Advanced.AROrder = 0;

firmod_liql = impulseest(Ztrain,irestoptions);
```

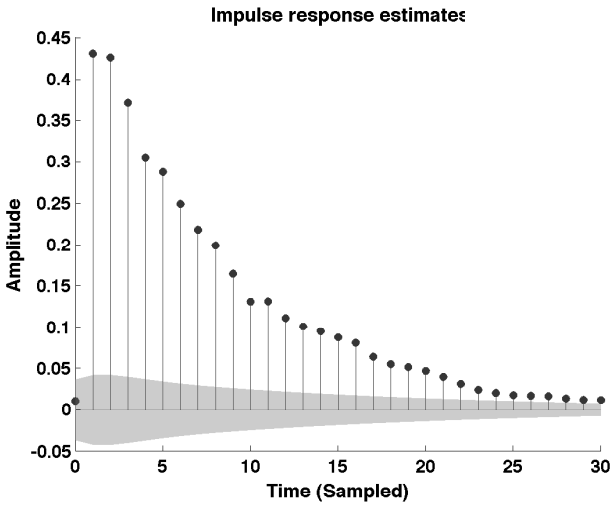


FIGURE 2.6 Impulse response estimates of the liquid level system.

```
[ircoeff,kvec,~,ir_sd] = impulse(firmod_liql,30);
stem(kvec,ircoeff,'Markerfacecolor','blue');
ir_3sd = 2.58*ir_sd;
fillxvec = [kvec(1) kvec(1) kvec(2:end)' kvec(end:-1:2)'];
fillyvec = [-ir_3sd(1) ir_3sd(1) ir_3sd(2:end)' -ir_3sd(end:-1:2)'];
hold
fill(fillxvec,fillyvec,[0.5 0.58 1],'FaceAlpha',0.4,'Linestyle','none');
```

Impulse response estimates obtained from the training data are shown in Figure 2.6. Any estimate falling within the shaded region is treated as statistically insignificant. The discrete-time liquid level system thus has a unit sample delay (delays in discrete-time systems are expressed in terms of samples). Physically there is no delay between flow and level, i.e., the continuous-time system has zero delay. However, the unit delay in the sampled-data system is due to the presence of zero-order hold. This is explained later in Chapter 6.

The decaying nature of the IR estimates is a clear indication that the sampled-data system is *stable*, which agrees very well with our physical knowledge of the process.

Step response model

Estimates of *unit* step response coefficients are obtained from the IR coefficients using a simple relationship

$$y_s[k] = \sum_{n=0}^{k} g[n] \tag{2.9}$$

The resulting estimates are shown in Figure 2.7, which is strongly suggestive of a first-order (or an overdamped higher-order) dynamics with a gain of approximately 3.7 units. If the system is approximated as a first-order process, the time-constant is about 7 samples (minutes). The fact that the step response steadies out is an indicator of the absence of any instabilities and non-stationarities in the process.

```
% Estimate step response and plot it
[srcoeff,kvec,~,sr_sd] = step(firmod_liql,40);

figure; stairs(kvec,srcoeff);
```

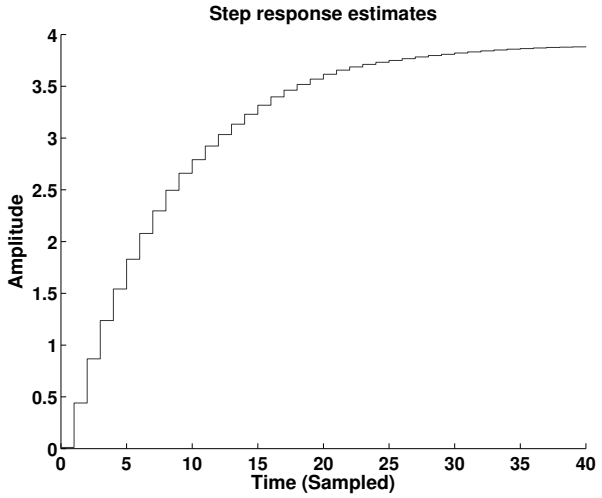


FIGURE 2.7 Step response estimates of the liquid level system.

A third elementary response that is also of interest is the frequency response function, which shall be taken up for estimation at a later point in the text (in Chapter 20).

Identification of response models is afflicted by the estimation of a large number of unknowns, which is not preferred in any estimation exercise. *Parametrizing*, these responses, meaning fitting mathematical expressions to the curves, aids in considerably reducing the size of the unknowns because one estimates the "parameters" instead of the response coefficients. An important point is that *parametrization of response models leads to difference equation form of descriptions*. The following section demonstrates the development of a parametric model for the liquid level system.

2.4.5 BUILDING PARAMETRIC MODELS

The objective is to identify a difference equation (DE) model for the *deterministic* process from the given data. The key information that is to be provided by the user is a suitable "guess" of the delay and order of the DE form.

From the previous section, it is meaningful to premise a first-order difference equation with a delay of 1 unit sample for the *deterministic* process.

$$x[k] + a_1 x[k-1] = b_1 u[k-1] \tag{2.10}$$

where $x[k]$ denotes the **unobserved** *true discrete-time* (deterministic) response of the process.

The parameters $\boldsymbol{\theta} = \begin{bmatrix} a_1 & b_1 \end{bmatrix}^T$ are estimated such that the sum squared one-step ahead prediction errors is minimized,

$$\min_{\boldsymbol{\theta}} \sum_{k=0}^{N-1} (x[k] - \hat{x}[k|k-1])^2 \tag{2.11}$$

where

$$\hat{x}[k|k-1] = -a_1 x[k-1] + b_1 u[k-1] \tag{2.12}$$

is the "prediction of $x[k]$ given the knowledge of $x[.]$ and $u[.]$ *until* the $(k-1)^{\text{th}}$ instant."

However, this is only possible when the true response is known. In reality, only a measurement of $x[k]$ is available. Therefore, it is natural re-write the minimization in (2.11) in terms of the

measurement prediction error:

$$\min_{\boldsymbol{\theta}} \sum_{k=0}^{N-1} (y[k] - \hat{y}[k|k-1])^2 \tag{2.13}$$

where now $\hat{y}[k|k-1]$ is the "prediction of the measurement $y[k]$ given the knowledge of measurements and inputs *until* the $(k-1)^{\text{th}}$ instant. For the rest of the presentation, $\hat{y}[k]$ is used to denote $\hat{y}[k|k-1]$.

Clearly, an expression for the predictor $\hat{y}[k]$ is required. Since we are dealing with measurements here, *it is intuitive that models for both deterministic and stochastic effects are required*.

To construct the overall model for $y[k]$, we first assume that stochastic effects, collectively denoted by $v[k]$, are *additive*

$$y[k] = x[k] + v[k] \tag{2.14}$$

Naturally, the (one-step ahead) prediction of the measurement is

$$\hat{y}[k] = \hat{x}[k] + \hat{v}[k] \tag{2.15}$$

where $\hat{v}[k]$ is the (one-step ahead) prediction of the disturbances and noise.

The deterministic signal $x[k]$ is modeled by the difference equation (2.10). A model for $v[k]$ is required to complete the picture. Different models exist depending on the assumptions made on the predictability of $v[k]$, leading to different descriptions for $y[k]$. Two common models, namely, the **output-error** and the **equation-error** models are explored in this case study. We shall see that these methods make somewhat contrasting assumptions about $v[k]$ and that they can significantly influence the choice of the final input-output model.

Output-error model

A simple assumption is that **the error in the measurement is absolutely unpredictable** given any amount of past. For zero-mean noise this means,

$$\hat{v}[k] = 0 \tag{2.16}$$

Such a stochastic signal is termed as the *white-noise* signal, denoted by $e[k]$ hereafter. Plugging (2.16) into (2.15), we have

$$\hat{y}[k] = \hat{x}[k] = -a_1 x[k-1] + b_1 u[k-1] \tag{2.17}$$

The unknowns on the RHS of the equation are a_1, b_1 and $x[k-1]$. It is clear that the predictor in (2.17) is *non-linear in unknowns*. Since the white-noise error *directly* enters the output, the model producing the predictor above is termed as the *output-error* model.

Applying the non-linear least squares method to solve (2.13) using the predictor in (2.17) on the training data produces,

$$\hat{a}_1 = -0.8826(\pm 0.002),\ \hat{b}_1 = 0.4621(\pm 0.006) \quad \text{(Output-Error Model)} \tag{2.18}$$

where the hat on the parameters $\hat{[.]}$ denotes the respective estimates and the values in the bracket are the standard errors in those estimates.

```
% Estimate output-error model of specified order and delay
mod_oe = oe(Ztrain,[1 1 1]);
present(mod_oe)
```

The goodness of this model will be assessed shortly. Now, we explore an alternative model structure that arises out of relaxing the unpredictable assumption on the stochastic component $v[k]$.

Equation-error model

For these class of models, the assumption is that $v[k]$ is **predictable**, *but* with the additional requirement of a *linear predictor* for the measurement, unlike the one in (2.17).

To understand the basic idea, first rewrite the DE for the true response $x[k]$ (2.12) in terms of the *measurement* using (2.15) to obtain

$$y[k] = -a_1 y[k-1] + b_1 u[k-1] + \overbrace{(v[k] + \underbrace{a_1 v[k-1]}_{w[k]})}$$
$$\implies \hat{y}[k|k-1] = -a_1 y[k-1] + b_1 u[k-1] + \hat{w}[k|k-1]$$

In order to have a **linear predictor** for the measurement $y[k-1]$, we require

$$\hat{w}[k|k-1] = 0 \tag{2.19}$$

meaning $w[k]$ is the white-noise signal $e[k]$. It follows that

$$\hat{y}[k] = -a_1 y[k-1] + b_1 u[k-1] \tag{2.20}$$
$$\hat{v}[k] = -a_1 v[k-1] \tag{2.21}$$

leading to a first-order *auto-regressive* predictor for $v[k]$. Since the error in the *difference equation for the measurement* is white (in contrast to the OE case), the description in (2.21) is known as the *equation-error* model, specifically the auto-regressive eXogenous (ARX) model.

Notice that the model for the measurement error in (2.21) has a *coefficient identical to the one for the deterministic response* in (2.12). A major implication of (2.21) is that *the deterministic and noise process share the same dynamics*.

From an estimation viewpoint, the ARX model offers a significant benefit over the OE model counterpart in that unique estimates of the ARX model parameters can be obtained. However, this luxury comes at a sacrifice that may be fairly serious as we shall shortly learn.

The *linear* least-squares estimates of the equation-error model using the training data are obtained as

$$\hat{a}_1 = -0.7788(\pm 0.012),\ \hat{b}_1 = 0.472(\pm 0.016) \quad \text{(Equation-Error Model)} \tag{2.22}$$

```
% Estimate equation-error model of specified order and delay
mod_arx = arx(datatrain,[1 1 1]);
present(mod_arx)
% Verify using the LS method described in Chapters 14 and 22
```

Both the estimated models must now be tested for their "goodness" to make improvements (if necessary) and to eventually select the "better" model.

2.4.6 GOODNESS OF THE MODEL

The *prediction error*, technically termed as *residual*, serves as the key quantity of interest in assessing the goodness of the model. It is formally defined as

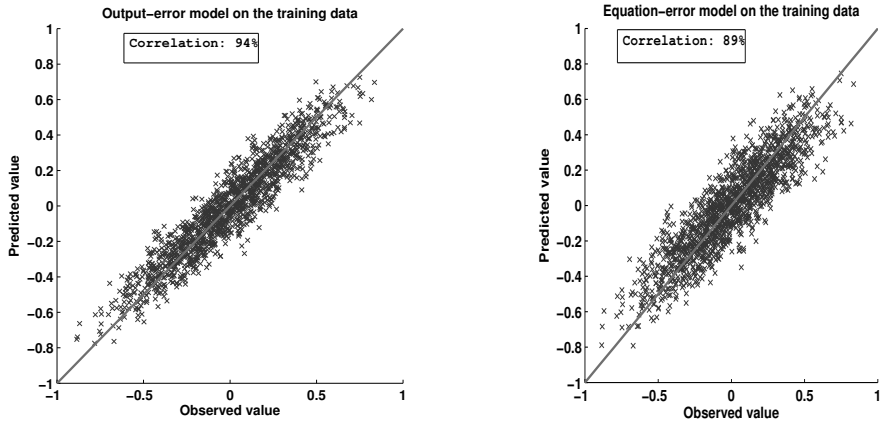
$$\varepsilon[k] = y[k] - \hat{y}[k] \tag{2.23}$$

A "good" model should not leave behind residuals (from training) that offer further scope for predictions, while avoiding overfitting (recall Example 2.4). Consequently, the following have to be fulfilled:

i. the residuals cannot be explained (predicted) by the input (test for the deterministic model),
ii. the residuals cannot be predicted using its own past, i.e., it is is truly unpredictable (test for the stochastic model), and
iii. the errors in parameter estimates are small or negligible relative to the estimates themselves (test for over-parametrization).

Predictions

The *one-step ahead* predictions from the estimated models are computed on the training data. The scatter plots comparing predictions vs. observed values are shown in Figure 2.8.



(a) Output-error model (b) Equation-error model

FIGURE 2.8 Comparing one-step ahead predictions (deviations from steady-state) of the identified models on the training data.

The output-error model fares better than its equation-error counterpart in this respect as evident from the correlation coefficient between the predictions $\hat{y}[k]$, and the observed response $y[k]$. A visual examination of Figure 2.8 also confirms this fact because the predictions of the output-error model have a relatively lower scatter around the unity line[5].

```
% Predictions on training data
yhat_oe = predict(mod_oe,datatrain);
yhat_arx = predict(mod_arx,datatrain);

% Predictions vs. observations
figure;
plot(yhat_oe,datatrain.y,'x',datatrain.y,datatrain.y,'r-')
figure;
plot(yhat_arx,datatrain.y,'x',datatrain.y,datatrain.y,'r-')
```

Test for deterministic model: Correlating residuals with inputs

The foregoing prediction analysis puts the models in close contest. However, to test whether these models leave behind any unexplained input effects, the correlation between the residuals and the *lagged* (time-shifted) inputs are computed for each of these models. This is also known as the *cross-correlation function*. Figure 2.9 shows the cross-correlation between the residuals and lagged inputs. A significant correlation between $\varepsilon[k]$ and input $u[k]$ at positive lags directly implies that the effects of input on the process response have not been completely explained. From Figure 2.9, it is clear that there exists no significant correlation between the residuals of the OE model and the inputs at any lag. Thus, the estimated OE model in (2.18) has satisfactorily captured the dynamics of the deterministic process.

On the other hand, the residuals from the ARX model are significantly correlated with inputs implying that it has *not managed to adequately capture the deterministic effects*. Therefore, it becomes

[5]An alternative plot superposing the predicted and observed outputs can also be used.

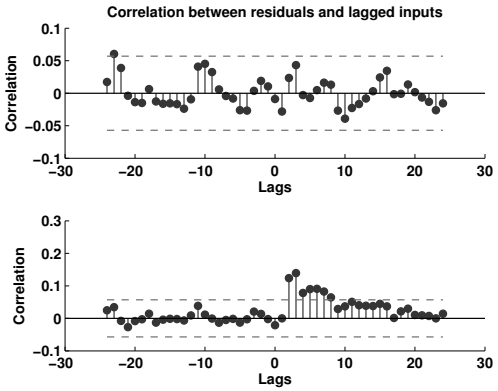


FIGURE 2.9 Correlation between residuals and lagged inputs for the output-error (bottom) and equation-error (top) models.

necessary to consider higher-order ARX models as possible candidates. A few iterations of model fitting followed by correlation analysis of the residuals result in a fifth-order ARX model as being satisfactory. Figure 2.10 displays the correlation between the residuals and inputs for this model. The prediction errors do not contain any significant effects of the inputs. The parameter estimates for this model are reported in Table 2.1. Observe that one of the corresponding parameter estimates ($\hat{a}_1$) is marked by a significant error, thereby reducing the reliability of the model. This should be expected since we have *overparametrized* the model (keep in mind that the deterministic process is a first-order).

The OE model is clearly the winner here in two respects, namely, *parameter estimation error* and *parsimony* (parameter dimensionality).

```
% Compute one-step ahead prediction errors
err_oe = pe(mod_oe,datatrain); % Returned as an iddata object
err_arx = pe(mod_arx,datatrain);

% Correlation between residuals and inputs
corrye_oe = xcov(err_oe.y,datatrain.u,24,'coeff');
corrye_arx = xcov(err_arx.y,datatrain.u,24,'coeff');

% 99 % significance levels
clim = 2.58/sqrt(length(datatrain.y));
% Plot
figure
subplot(211); stem(-24:24,corrye_oe); hold on
plot([-24 24],[1 1]*clim,'r--',[-24 24],[-1 -1]*clim,'r--')
subplot(212); stem(-24:24,corrye_arx); hold on
plot([-24 24],[1 1]*clim,'r--',[-24 24],[-1 -1]*clim,'r--')
```

```
% Fit the best equation-error model
mod_arxb = arx(datatrain,[5 5 1]);
present(mod_arxb)

% Compute residuals
err_arxb = pe(mod_arxb,datatrain); % Compute prediction errors
% ACF of residuals and CCF with inputs
[acf_errb,lags] = xcov(err_arxb.y,20,'coeff');
ccf_errbu = xcov(err_arxb.y,datatrain.u,20,'coeff');

figure
subplot(211); stem(lags(21:end),acf_errb(21:end))
subplot(212); stem(lags,ccf_errbu)
```

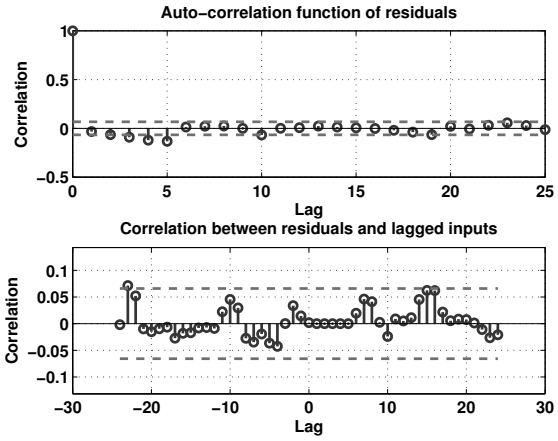


FIGURE 2.10 Correlation analysis of residuals obtained from the best equation-error model.

TABLE 2.1

Parameter estimates (errors) of the best equation-error model for the liquid-level system

Parameter	a_1	a_2	a_3	a_4	a_5
Estimate	0.067 (±0.03)	0.112 (±0.03)	0.102 (±0.03)	0.14 (±0.02)	0.19 (±0.02)
Parameter	b_1	b_2	b_3	b_4	b_5
Estimate	0.451 (±0.02)	0.377 (±0.02)	0.311 (±0.03)	0.17 (±0.02)	0.136 (±0.02)

Test for noise model: Predictability in residuals

Given that the output-error model has captured the deterministic effects adequately well, a natural step is to assess if the stochastic part of the model has satisfactorily explained the random effects. A plot of the *auto-correlation function* of residuals is used for this purpose. It is essentially the correlation between any two samples separated in time by a lag l. Any predictability in the sequence manifests as non-zero correlation at a non-zero lag l. The ACF of the residuals from the OE model is shown in Figure 2.11. The absence of any statistically significant correlation at any non-zero lag indicates no scope for predictability within the residuals. Note that by definition, the ACF is unity at lag zero, i.e., any sample is best correlated with itself.

The stochastic model assumed in (2.16), thus sufficiently explains the noise characteristics of the measured output.

For completeness of analysis, the auto-correlation function of the residuals from the fifth-order

```
% Plot the ACF of residuals
acf_err = xcov(err_oe.y,20,'coeff');
figure; stem((0:20),acf_err(21:end)); hold on

% 99 % significance levels
clim = 2.58/sqrt(length(err_oe.y));
plot([0 20],[1 1]*clim,'r--',[0 20],[-1 -1]*clim,'g--')
```

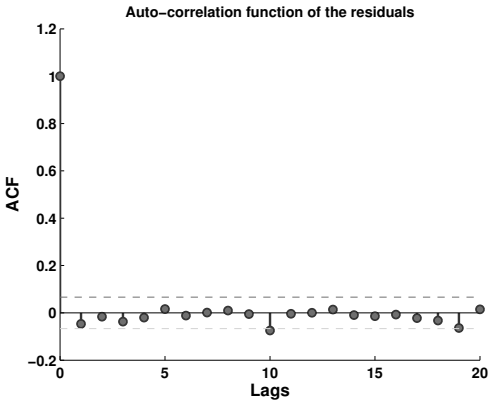


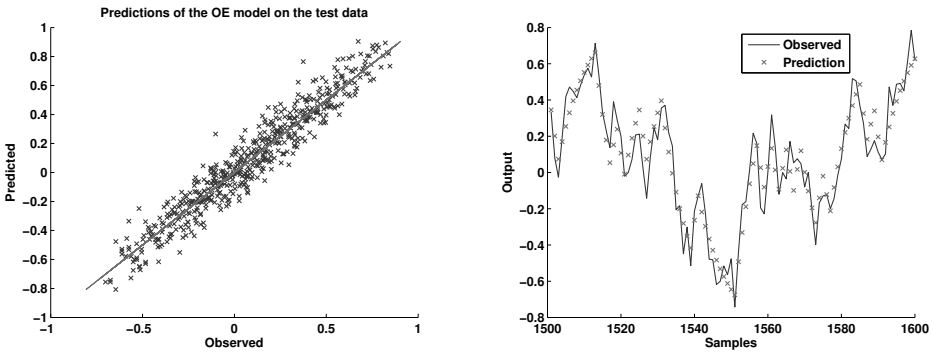
FIGURE 2.11 Auto-correlation function of the residuals from the output-error model.

ARX model is also evaluated. The top panel of Figure 2.10 display the ACF. Some significant correlation exists at non-zero lags implying that the stochastic terms are not adequately modeled.

The results of the assessment tests are convincingly in favor of the OE model in all respects. Thus, solving a non-linear optimization problem was worth the effort.

Cross-validation

A good model is one which yields good predictions on a fresh data set. The predictions of the output error model on a fresh data are shown in Figure 2.12. The acid test for a model is its ability to make good long-range predictions. The predictions (deviations from steady-state) shown in Figure 2.12 are the infinite-step ahead predictions. The one-step ahead predictions of the output at k^{th} instant are computed with the knowledge of the observations until the previous instant, i.e., the $(k-1)^{th}$ instant whereas the infinite-step ahead predictions are calculated based on the observations provided at infinite time in the past, i.e., practically without any information about the observations, but only based on the input profile.



(a) Scatter plot (b) Snapshot of predictions

FIGURE 2.12 Infinite-step ahead predictions from the output-error model on the test data set.

As mentioned earlier, the test data set constitutes samples $N = 1501$ onwards. Figure 2.12(a) shows that the output-error model is able to predict very well on a fresh data set. A common measure of goodness-of-predictions is the normalized root mean square (NRMS) measure of fit (see also §14.3.2):

$$R_f = 1 - \frac{||\mathbf{y} - \hat{\mathbf{y}}||_2}{||\mathbf{y} - \bar{\mathbf{y}}||_2}$$

where $\mathbf{y}$ and $\hat{\mathbf{y}}$ are vectors of measurements and predictions, respectively, and $\bar{\mathbf{y}}$ is the mean value of $\mathbf{y}$. The notation $||.||_2$ stands for the 2-norm of the vector. The NRMS value for predictions calculates to about 72%. A snapshot of the predictions superimposed on the observed values is shown in Figure 2.12(b). The graph visually confirms the goodness of predictions.

```
% Infinite-step ahead predictions
yhat_inf = predict(mod_oe,datatest,Inf);
yhat = yhat_inf.y;

% Compare against measurements
figure
plot(yhat,datatest.y,'x',datatest.y,datatest.y,'r-')
figure
plot((1501:1600),datatest.y(1:100),(1501:1600),yhat(1:100),'gx')
```

Final model

Based on the results of the model assessment tests, namely, the residual analysis, analysis of errors in estimates and cross-validation, the first-order OE model with a delay of unit sample is deemed as the most appropriate:

$$\boxed{\begin{aligned} y[k] &= x[k] + e[k] && \text{(2.24a)}\\ x[k] &= -\hat{a}_1 x[k-1] + \hat{b}_1 u[k-1] && \text{(2.24b)}\\ \hat{a}_1 &= -0.8826(\pm 0.0019), \quad \hat{b}_1 = 0.4621(\pm 0.0052) && \text{(2.24c)} \end{aligned}}$$

It is also a common practice to express the difference equation in what is known as a *transfer function (TF) operator* form:

$$y[k] = G(q^{-1})u[k] + H(q^{-1})e[k] \tag{2.25}$$

where $G(q^{-1})$ and $H(q^{-1})$ are known as *plant* and *noise* models, respectively. The quantity q^{-1} is known as the *backward shift operator*, meaning $q^{-1}x[k] = x[k-1]$.

The OE model in (2.24) in the TF form is written as:

$$\boxed{y[k] = \frac{\overset{(\pm 0.005)}{0.4621q^{-1}}}{1 - \underset{(\pm 0.002)}{0.8826q^{-1}}}u[k] + e[k]} \tag{2.26}$$

The transfer function representation has certain nice features, which are discussed in Chapter 4.

The presentation until this point illustrated the procedure to develop an empirical input-output model from data. Now we turn to show how a state-space description can be identified. Identification of state-space models, as remarked in §1.2, has certain advantages over the input-output forms, specifically in terms of numerical efficiency and order determination.

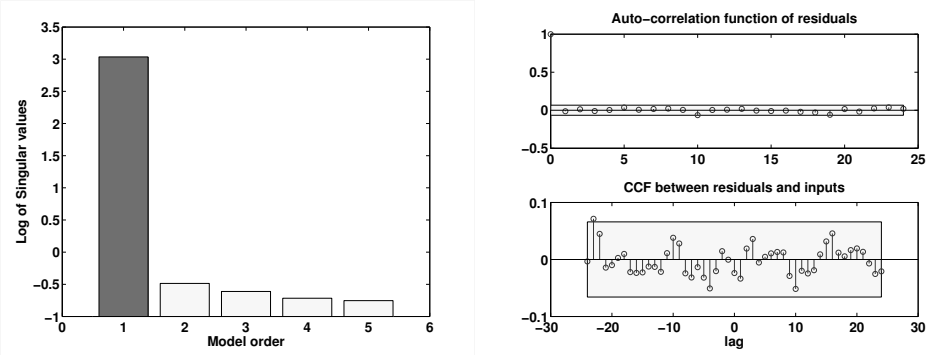
2.4.7 DEVELOPING A STATE-SPACE MODEL

A generic state-space model that is typically used in identification has the form

$$\mathbf{x}[k+1] = \mathbf{A}\mathbf{x}[k] + \mathbf{B}\mathbf{u}[k] + \mathbf{K}\mathbf{e}[k] \qquad \text{(State equation)} \tag{2.27a}$$

$$\mathbf{y}[k] = \mathbf{C}\mathbf{x}[k] + \mathbf{D}\mathbf{u}[k] + \mathbf{e}[k] \qquad \text{(Output equation)} \tag{2.27b}$$

where $\mathbf{x}[k]$ is the vector of *states*, or hidden or unobserved variables. The variable $\mathbf{e}[k]$ is the vector of *unpredictable* random errors (as in the output-error model). The state-space form in (2.27) is



(a) log Hankel singular values (b) Residual analysis of the first-order model

FIGURE 2.13 Plot of (log) Hankel singular values and correlation functions from the state-space model.

also known as the *innovations form*. When the structures of state-space matrices in (2.27) are *not* constrained in any manner, we have a *non-parametrized* state-space form.

The state-space and input-output descriptions are interconvertible. With a suitable *parametrization* (constraints on state-space matrices) of the innovations form followed by conversion into input-output form, one can arrive at the OE and ARX models in (2.24) and (2.20), respectively. For instance, with a single-state and setting $K = 0$ in i.e., by only letting the error enter the output only, leads to the OE model form in (2.24). It is also possible to do the reverse, i.e., write a state-space model from the identified input-output OE model. Such state-space models are then known as *parametrized* state-space models. However, we shall reserve that discussion for Chapter 23. Below we demonstrate a widely used approach for state-space identification which directly estimates a state-space model from data.

In order to directly estimate the state-space model of the form (2.27), we use a popular subspace method known as the N4SID algorithm, where the user is not required to key in any guess of the order. As mentioned earlier, the algorithm automatically determines this parameter using what are known as *Hankel singular values*. In practice, the user can restrict the order to a range, say from first to fifth order. The subspace algorithm is supplied with the delay information that we estimated earlier.

Setting the range of orders as one to five and the delay to unity, the algorithm estimates a *first-order discrete-time* state-space model.

$$\hat{A} = 0.881(\pm 0.002); \qquad \hat{B} = 0.043(\pm 5 \times 10^{-4}) \tag{2.28a}$$

$$\hat{K} = -8.0552 \times 10^{-4}(\pm - 1.2 \times 10^{-3}) \tag{2.28b}$$

$$\hat{C} = 10.71(\pm 4.4 \times 10^{-6}); \qquad D = 0 \tag{2.28c}$$

where the values in brackets are (numerically computed) errors in the respective estimates. The first-order choice is arrived at by examining the (logarithm of) Hankel singular values for the specified range of orders, as shown in Figure 2.13(a). States that have significant Hankel singular values are retained in the model.

Residual analysis plots shown in Figure 2.13(b) shows that the first-order model has satisfactorily explained the input-output relationship as well as the stochastic terms.

Identifying a structured state-space model

Barring $\hat{K}$, all the estimates in (2.28) appear to be significant[6]. The high error in $\hat{K}$ relative to its estimate indicates that the true K is most likely to be zero, i.e., the disturbance term in the *state*

[6]The matrix D represents the instantaneous effect of input on output. A non-zero delay automatically implies that $D = 0$.

Listing 2.3 MATLAB code for estimating the state-space models.

```
% Load the input-output data
load liqleveldata

% Create IDDATA object and training / test objects
Z = iddata(yk,uk,1);
[datatrain,Tr] = detrend(Z(1:1500),'constant');
datatest = detrend(Z(1501:end),Tr);

% Fit a state-space model and analyse the residuals
mod_ss = ssest(datatrain,1:5,'Ts',1);
resid(datatrain,mod_ss)
present(mod_ss)

% Re-estimate the SS model w/o the state disturbance
mod_ss2 = ssest(datatrain,1,'Ts',1,'disturbancemodel','none');
present(mod_ss2)

% Estimate a structured SS model corresponding to OE model
mod_ss3 = ssest(datatrain,1,'Ts',1,'disturbancemodel','none','form','canonical...
    ');
present(mod_ss3)
```

is absent[7]. Thus, we could omit that term in the state equation and fit the model once again. The resulting estimate is:

$$x[k+1] = 0.8826(\pm 0.001375)x[k] + 0.04152(\pm 2.4 \times 10^{-4})u[k] \tag{2.29a}$$

$$y[k] = 11.13(\pm 9.126 \times 10^{-7})x[k] + e[k] \tag{2.29b}$$

The estimates have clearly improved, as evident from the lowering of errors relative to the model in (2.28).

One could further force the state-space model to have a structure similar to (2.24) so that it is possible to compare the estimates. Forcing $C = 1$ (in addition to $K = 0$) produces the estimate

$$x[k+1] = 0.8826(\pm 0.00188)x[k] + 0.04621(\pm 0.0052)u[k] \tag{2.30a}$$

$$y[k] = x[k] + e[k] \tag{2.30b}$$

resulting in almost congruent models because state-space models with structural constraints (a.k.a. structured state-space models) are identified using the same method (prediction-error minimization) as that for input-output models. The only difference is in the initial guess of these estimates.

Identified model vs. true model

In closing, it is instructive to compare the identified models in (2.18) and (2.30) with the "true" *linearized* discrete-time model. *This is only for the purpose of illustration. In a practical situation it is obvious that such a privilege does not exist.*

The *linearized*, **discretized** model of the non-linear continuous-time liquid-level system is:

$$x[k] = -0.8825x[k-1] + 0.47u[k-1] \tag{2.31}$$

[7]The errors reported for the state-space model are computed using a numerical sensitivity analysis, unlike for the OE and ARX models. Therefore, in general, a great deal of caution is required in using these error estimates.

at a sampling interval of $T_s = 1$ min. The estimated output-error models in (2.18) and (2.30) are a very close match to the discretized model of the process in (2.31) in structure and parameters, reproduced below for convenience,

$$\hat{a}_1 = -0.8826(\pm 0.002),\ \hat{b}_1 = 0.4621(\pm 0.006) \qquad \text{(2.18 revisited)}$$

It is interesting to compare (2.31) with the first-order ARX model of (2.22). The input coefficient is estimated with reasonable accuracy but the regression parameter estimate is significantly different from the true value. In other words, a *systematic error* (bias) is incurred in the estimation of a_1. This is true for all ARX model estimates whenever a mismatch exists between the noise model of ARX structure and that of the true process (a demonstration of this fact is given in Chapter 21).

In the simulation of the system, a *stationary unpredictable* noise (*white-noise*) was added to the response with the variance adjusted to obtain a SNR of 10. In retrospect, the deterministic parts of the first-order OE and ARX model matched with the process (in structure), while only the stochastic part of the OE model agreed with that of the data generating process. The latter factor is the reason why a first-order ARX model failed to deliver. A higher-order model was satisfactory in some respects, but then the deterministic part was over-modeled. The message to carry forth is that *the characterization of the stochastic terms is important in rightly modeling the deterministic sub-systems of the process*.

2.5 REFLECTIONS AND SUMMARY

It is appropriate to conclude this chapter with an introspective summary of the important lessons learned herein. In these reflections, we take the opportunity to raise generic questions that are encountered in identification and provide answers as pointers to the appropriate portions in this text.

1. In Example 2.1, a model, which was linear in input, but non-linear in parameters, was found to be non-identifiable. Is this true of all models (strictly speaking, predictors) that are non-linear in parameters? The answer is not necessarily yes. For linear dynamical systems, we shall study in §18.6 identifiability of different model structures.
2. Example 2.2 of a FIR system showed the role of input design in *identifiability*, which is the ability to identify a model uniquely. *In general also, does this hold good*? The answer is, *yes*. Section 22.3 discusses these concepts in the context of input design.
3. The relation between variability in parameter estimates and SNR was illustrated in Example 2.3. *What is the exact relation in general*? The fundamental *Cramer-Rao's inequality* provides bounds on lowest achievable error in parameter estimates. It involves the Fisher's information metric. These theoretical concepts are defined and illustrated in Chapter 13. Specific expressions for errors in parameter estimates can be derived under large-sample conditions for certain estimators such as LS and MLE (see §14.3.1 and §15.1.4).
4. In generating the data for the liquid level system, we have used a PRBS input. Naturally, we could have interrogated the system with a different input sequence, e.g., a step or a mixture of sine waves. Once again, the key governing factors are identifiability and SNR. Section 22.3, as mentioned earlier, is devoted to the topic of input design.
5. The sampling interval of 1 min. for the level measurement was based on the general guidelines given in Chapter 6, which discusses fundamental results on *how should one choose the sampling rate*? While the general answer is based on the maximum frequency contained in the signal, in identification and control, the connection is with the pace of the dynamics or the bandwidth (both related to time-constants) of the system.
6. Pre-processing and apportioning of data for training and cross-validation purposes can involve more generic approaches than the ones used in the case study. See Chapter 22 for related discussions.

7. Models for the liquid level system were selected from the LTI family. A non-linear model would have certainly been a better choice given the physics of the process, but would have opened up a number of challenges in the estimation stage. A general recommendation is to begin with a LTI family because they can be estimated relatively easily, unless there are compelling reasons to select otherwise.
8. Is it possible to determine whether an LTI model suits a given process? The answer to this question is provided by *coherence* (see §11.4), which not only examines the suitability of an LTI relationship but also suggests the frequency range over which such a model can be built. The results from this analysis can be used in other stages of identification, such as delay estimation and *pre-filtering* the data.
9. Non-parametric descriptions provide a wealth of information. While impulse and step response models offer good insights, frequency-domain descriptions present a broader and deeper picture of the system dynamics. These aspects are discussed in Chapters 5, 11 and 17. The non-parametric models in §2.4.4 were estimated using the methods described in Chapter 20.
10. We considered two forms of parametric model structures, namely the OE and the EE model structures, since they are the two widely used starting points in identification of parametric models. See §17.5 for a comprehensive description of the gamut of parametric models used in identification. Non-linear least squares and the linear version used for parameter estimation are special cases of the more versatile and powerful *prediction-error minimization methods* discussed in Chapter 21.
11. The primary reason for the poor performance of ARX model is that the parametrization of the noise model in (2.21) is completely tied to that of the deterministic part. This kind of parametrization does not suit the case study, but may suit several other processes. A converse argument can also be made for the OE models. See the ensuing point.
12. The output-error model structure turned out to be an appropriate one for the liquid level system. *In general is this true*? *If the measurement noise possessed some predictability, would the output-error model structure still produce good results*? The answer is *yes*, only under *open-loop conditions* and for the *deterministic* part. See Chapter 21 for a theoretical proof.
13. If the stochastic part of the output has some predictability, auto-correlation analysis of the residuals from the best OE model would clearly indicate those characteristics. In such a case, a time-series model for the residuals is also constructed. Subsequently, the deterministic and time-series models are used as initial guesses for estimating the more general Box-Jenkins model. See Chapter 21 for a case study.
14. When the parametrization is such that it results in a linear predictor (e.g., ARX), there are certain advantages. On the other hand, having independent "tunable" parameters in the noise and deterministic models (as in Box-Jenkins structures) always results in non-linear predictors, but a broader class of processes can be modeled. *Which one is a better approach*? The example shows that mathematical convenience need not always be given the top priority. Notwithstanding this fact, a common practice is to first develop the "best possible" ARX model for the given data and then seek other model structures lest they fail.
15. The residual analysis is the key to model refinement. *What information does the correlation between residuals and inputs contain*? *How can we use this information to improve the model*? These aspects are discussed at length in Chapter 22. The auto-correlation and cross-correlation functions are the main tools, as was evident from the case study. These concepts are theoretically defined in Chapter 8 while estimators of these functions are provided in §16.4.
16. Order determination for input-output models is traditionally done using information-theoretic measures such as AIC, BIC, etc. (see §22.6.3). State-space models offered a clear advantage in that it replaced the lengthy trial-and-error approach to order determination. It also obviates the need for the use of aforementioned measures. *Do these advantages always exist*? In general, state-space models almost always offer valuable preliminary insights into the order and structure

of input-output models. One of the disadvantages, among a few others (e.g., non-uniqueness of representation) of state-space models is that a non-parametric state-space model has more parameters to estimate than an identifiable input-output model of the same order. Thus, do we run into identifiability issues with these models? Chapter 4 presents a vivid comparison of deterministic state-space vs. input-output models. See also Chapter 23.

17. Constraining the state-space matrices structurally reduces the number of unknowns to be estimated, thereby improving identifiability. In the case study, we were able to discover the underlying structure of the process by a systematic analysis of the *freely* parametrized model. (2.28). Generalizing these ideas leads us into the expanses of grey box modeling. See §23.7.2 for expansions and illustrations of these ideas.

For many of the questions above, generally speaking, no definitive answers exist. Only guiding principles for building good quality models can be given. The responsibility rests with the user in making "appropriate" choices using a pragmatic blend of theory and intuition at all stages of identification; from a careful design of experiments to a systematic analysis of data.

EXERCISES

E2.1. Highlight the two fundamental challenges in identification.

E2.2. Give two examples of a model that is not identifiable in the entire parameter space, i.e., is not globally identifiable, but is locally identifiable.

E2.3. Explain the two different model structures, namely, *output-error* model and *equation-error* model structures.

E2.4. Describe the iterative procedure for identification.

E2.5. Show that for Example 2.2, using the input $u[k] = \sin(\omega_0 k) + \sin(\omega_1 k)$ does not result in loss of identifiability.

E2.6. Re-estimate the parameters of the OE model with different sample sizes of training data $N = 100$ and $N = 500$. Compare the errors in the estimated parameters with those reported in (2.24). What do you conclude?

E2.7. Re-work the entire liquid level case study for two different signal-to-noise ratios, 5 and 1 respectively. What do you conclude on the accuracy and precision of the resulting parameter estimates?

E2.8. In the exercise above, does the subspace algorithm still rightly identify the model order when a state-space model is being identified?

E2.9. Generate a fresh flow-level response data by changing the measurement noise from *white* to *filtered* noise by placing a *discrete-time* filter block in the disturbance channel path, in the SIMULINK block diagram. Use the filter coefficients `num = 1`, `den = [1 -0.5]` and set the SNR to 10. For this purpose, switch off the noise, simulate and compute the variance of the true response. Subsequently, turn the white-block noise on and adjust the variance parameter in the block such that the desired SNR is achieved.

E2.10. Using the data from the above simulation, determine the best OE model (remember the only tests are correlation between inputs and residuals, as well as good parameter estimates) still manages to explain the deterministic process correctly.

E2.11. How does the ARX model (of a suitably high-order) fare in the above situation, i.e., data generated with colored noise?

E2.12. Develop a state-space model for the process using the above data. Compare it with the model in (2.29) and comment on the differences, if any.