1

Introduction

In this chapter, we introduce the idea of model-based identification, starting with the basic notions of signal processing and estimation. Once defined, we introduce the concepts of model-based signal processing, that lead to the development and application of subspace identification. Next, we show that the essential ingredient of the model-based processor is the "model" that must be available either through the underlying science (first principles) or through the core of this text – model-based identification.

1.1 Background

The development of processors capable of extracting information from noisy sensor measurement data is essential in a wide variety of applications, whether it be locating a hostile target using radar or sonar systems or locating a tumor in breast tissue or even locating a seismic source in the case of an earthquake. The nondestructive evaluation (NDE) of a wing or hull of a ship provides a challenging medium even in the simplest of arrangements requiring sophisticated processing especially if the medium is heterogeneous. Designing a controller for a smart car or a drone or for that matter a delicate robotic surgical instrument also depends on providing enhanced signals for feedback and error corrections. Robots replacing humans in assembly lines or providing assistance in mundane tasks must sense their surroundings to function in a such a noisy environment. Most "hi-tech" applications require the incorporation of "smart" processors capable of sensing their operational environment, enhancing noisy measurements and extracting critical information in order to perform a preassigned task such as detecting a hostile target and launching a weapon or detecting a tumor and extracting it. In order to design a processor with the required capability, it is necessary to utilize as much available a priori information as possible. The design may incorporate a variety of disciplines to achieve the desired results. For instance, the processor must be able to sense the operational environment, whether it be highly cluttered electromagnetic propagation

Model-Based Processing: An Applied Subspace Identification Approach, First Edition. James V. Candy.

at an airport or a noisy ocean acoustic environment in a busy harbor. Array radiation measurements in the case of an active radar system targeting signals of great interest can detect incoming threats, while passive listening provided by an acoustic array aids in the detection of submarines or similarly tumors in the human body for ultrasonics as well. The ability of the processor to operate effectively in such harsh environments requires more and more sophistication, rather than just simple filtering techniques. It is here that we address not only the need, but also the a priori requirements for a design. For instance, the detection and localization of a quiet diesel submarine cannot be achieved without some representation of the noisy, varying ocean incorporated in the processing scheme. How does such information get embedded? This is the question for not only the signal processor, but also the ocean acoustician and sensor designer to ponder. The solution boils down to the melding of this information enabling the development of a processor capable of performing well. So we see that except in an exceptional case, the knowledge of the underlying phenomenology that governs just how a signal propagates in an uncertain medium or environment coupled with that of how a sensor can make a reasonable measurement to provide the desired information and a processor capable of extracting that information defines a "team" consisting of a phenomenologist, sensor designer and signal processor that can enable a solution to the problem at hand. In this text, we discuss such an approach that incorporates all of these capabilities. We start with the basic processor and then progress to a scheme capable of incorporating the underlying phenomenology, measurement systems, and uncertainties into the processor. In order to do so, we start with defining signal processing and signal estimation, followed by the fundamental model-based signal processor and then approaches to obtain the required model from experimental as well as application data sets.

1.2 Signal Estimation

Signal processing is based on one fundamental concept – extracting critical information from uncertain measurement data [1, 2]. Processing problems can lead to some complex and intricate paradigms to perform this extraction especially from noisy, sometimes inadequate measurements. Whether the data are created using a seismic geophone sensor from a monitoring network or an array of hydrophone transducers located on the hull of an ocean-going vessel, the basic processing problem remains the same – extract the useful information. Techniques in signal processing (e.g. filtering, Fourier transforms, time–frequency and wavelet transforms) are effective; however, as the underlying process generating the measurements becomes more complex, the resulting processor may require more and more information about the process phenomenology to extract the desired information. The challenge is

to formulate a meaningful strategy that is aimed at performing the processing required, even in the face of these high uncertainties. This strategy can be as simple as a transformation of the measured data to another domain for analysis or as complex as embedding a full-scale propagation model into the processor [3]. For example, think of trying to extract a set of resonances (damped sinusoids) from accelerometer time series. It is nearly impossible to calculate zero-crossings from the time series, but it is a simple matter to transform the data to the spectral domain using a Fourier transform and then applying the property that sinusoids are impulse-like in Fourier space facilitating their extraction through peak detection. Finding a sinusoidal source propagating in the ocean is another matter that is quite complex due to the attenuation and dispersion characteristics of this harsh, variable environment. Here, a complex propagation model must be developed and applied to "unravel" the highly distorted data to reveal the source – a simple Fourier transform will no longer work. The aims of both approaches are the same – to extract the desired information and reject the extraneous and, therefore, develop a processing scheme to achieve this goal. The underlying signal processing philosophy is a "bottoms-up" perspective enabling the problem to dictate the solution, rather than vice versa.

More specifically, signal processing forms the basic nucleus of many applications. It is a specialty area that many researchers/practitioners apply in their daily technical regimen with great success such as the simplicity in Fourier analysis of resonance data or in the complexity of analyzing the time–frequency response of dolphin sounds. Applications abound with unique signal processing approaches offering solutions to the underlying problem. For instance, the localization of a target in the hostile underwater ocean acoustic environment not only challenges the phenomenologist but also taxes the core of signal processing basics, thereby requiring that more sophistication and a priori knowledge be incorporated into the processor. This particular application has led to many advances both in underwater signal processing and in the development of a wide variety of so-called model-based or physics-based processors. A prime example of this technology is the advent of the model-based, matched-field processor [3–5] that has led not only to a solution of the target localization problem, but also to many applications in other areas such as nondestructive evaluation and biomedical imaging. Therefore, the conclusion remains the same, signal processing is a necessary ingredient as a working tool that must be mastered by phenomenologists to extract the useful information from uncertain measurements. In fact, we define signal processing as a set of techniques "to extract the desired information and reject the extraneous from uncertain measurement data."

Signal processing relies on any prior knowledge of the phenomenology generating the underlying measurements. Characterizing this phenomenology and propagation physics along with the accompanying measurement

instrumentation and noise are the preliminaries that all phenomenologists must tackle to solve such a processing problem. In many cases, this is much easier said than done. The first step is to determine what the desired information is and typically this is not the task of the signal processor, but that of the phenomenologist performing the study. In our case, we assume that the investigation is to extract information stemming from signals emanating from a source, whether it be an autonomous unmanned vehicle (AUV) on the highway or passively operating in the deep ocean, or a vibrating structure responding to ground motion. Applications can be very complex especially in the case of ultrasound propagating through complex media such as tissue in biomedical applications or through heterogeneous materials of critical parts in nondestructive evaluation (NDE) investigations or photons emanating from a radiation source [6]. In any case, the processing usually involves manipulating the measured data to extract the desired information, such as location and tracking of the AUV, failure detection for the structure, or tumor/flaw detection, and localization in both biomedical and NDE [3].

If a measured signal is free from extraneous variations and is repeatable from measurement to measurement, then it is defined as a *deterministic* signal (Chapter 2). However, if it varies extraneously and is no longer repeatable, then it is defined as *random* signal. This text is concerned with the development of processing techniques to extract pertinent information from random signals utilizing any a priori information available. We call these techniques signal estimation or signal enhancement, and we call a particular algorithm a *signal estimator* or just *estimator*. Symbolically, we use the "caret" ($\hat{}$) notation to annotate an estimate (e.g. $s \rightarrow \hat{s}$). Sometimes, estimators are called filters (e.g. Wiener filter) because they perform the same function as a deterministic (signal) filter except for the fact that the signals are random; that is, they remove unwanted disturbances. Noisy measurements are processed by the estimator to produce "filtered" data. To solidify these concepts, consider the following examples.

Example 1.1 A deterministic signal composed of two sinusoids: the information at 10 Hz and the disturbance at 20 Hz, with its corresponding Fourier spectrum shown in Figure 1.1a. From a priori information, it is known that the desired signal has no frequencies above 15 Hz; however, the raw spectrum reveals the disturbance at 20 Hz. Since the data are deterministic, a low-pass filter with a cutoff frequency of 12.5 Hz is designed to extract the desired information (10 Hz signal) and reject the extraneous (20 Hz disturbance). The filtered data are shown in Figure 1.1b, where we can see the filtered signal and the resulting spectrum. □

Consider the output of the estimation filter designed to eliminate random noise from a transient signal; that is, the estimator is a function of the signal

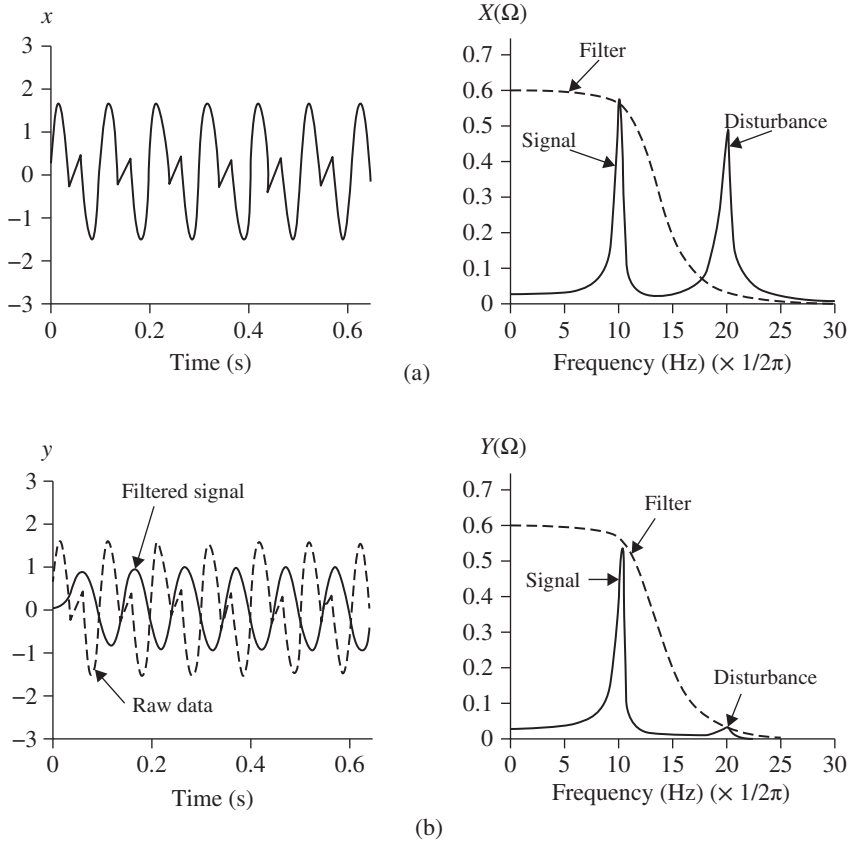

Figure 1.1 Processing of a *deterministic* signal. (a) Raw data and spectrum with signal at 10 Hz and disturbance at 20 Hz. (b) Processed data extracting the 10 Hz signal (desired) and rejecting the extraneous (20 Hz disturbance).

$x(t)$ and noise $n(t)$

$$\hat{y}(t) = a(x(t), n(t)).$$

Consider the following example to illustrate this processor.

Example 1.2 A random pulse-like signal contaminated by noise is shown in Figure 1.2a. Here, we see the measured data along with its Fourier spectrum. We design a signal estimator to extract the desired signal and remove the noise. The processed data are shown in Figure 1.2b, where we observe the results of the estimation *filter* and the corresponding enhanced spectrum. Here, we see how the filtered response has eliminated the random noise. We discuss the concepts of signal estimation using the modern parametric design methods in Chapter 4. □

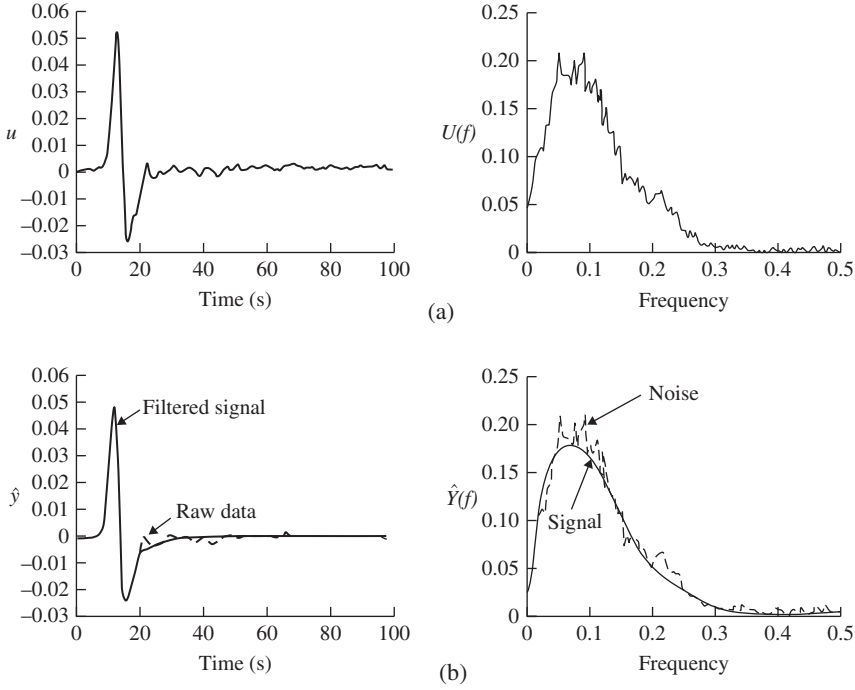

Figure 1.2 Processing of a *random* signal and noise. (a) Raw data and spectrum with noise. (b) Processed data extracting the signal (estimate) and rejecting the extraneous (noise).

If the estimation filter or more commonly the *estimator* employs a model of phenomenology or process under investigation, then it is considered a model-based processor. For example, suppose we use a so-called *Gauss–Markov* model (see Chapter 3) in our estimator design, that is, if we use

$$x(t) = Ax(t-1) + w(t-1)$$
$$y(t) = Cx(t) + v(t)$$

then the resulting signal estimator,

$$\hat{x}(t) = A\hat{x}(t-1) + K(t)e(t)$$

is called a model-based signal processor and in this case a *Kalman filter*. Model-based signal processing is discussed in Chapter 4. We shall see that random signals can be characterized by stochastic processes (Chapter 2), transformed to equivalent deterministic representations (covariance and power spectrum) and processed (model-based processors, Chapters 4 and 5) much the same as a deterministic signal.

Estimation can be thought of as a procedure made up of three primary parts:

- Criterion function
- Models
- Algorithm.

The criterion function can take many forms and can also be classified as deterministic or stochastic. Models represent a broad class of information, formalizing the a priori knowledge about the process generating the signal, measurement instrumentation, noise characterization, and underlying probabilistic structure. Finally, the algorithm or technique chosen to minimize (or maximize) the criterion can take many different forms depending on (i) the models, (ii) the criterion, and (iii) the choice of solution. For example, one may choose to solve the well-known least-squares problem recursively or with a numerical-optimization algorithm. Another important aspect of most estimation algorithms is that they provide a "measure of quality" of the estimator. Usually, what this means is that the estimator also predicts vital statistical information about how well it is performing.

Intuitively, we can think of the estimation procedure as follows:

- The specification of a criterion
- The selection of models from a priori knowledge
- The development and implementation of an algorithm.

Criterion functions are usually selected on the basis of information that is meaningful about the process or the ease with which an estimator can be developed. Criterion functions that are useful in estimation can be classified as deterministic and probabilistic. Some typical functions are as follows:

- *Deterministic:*
  - Squared error
  - Absolute error
  - Integral absolute error
  - Integral squared error
- *Probabilistic:*
  - Maximum likelihood
  - Maximum a posteriori (Bayesian)
  - Maximum entropy
  - Minimum (error) variance.

Models can also be deterministic as well as probabilistic; however, here we prefer to limit their basis to knowledge of the process phenomenology (physics) and the underlying probability density functions as well as the necessary statistics to describe the functions. Phenomenological models fall into the usual

classes defined by the type of underlying mathematical equations and their structure, namely, linear or nonlinear, differential or difference, ordinary or partial, time invariant or varying. Usually, these models evolve to a stochastic model by the inclusion of uncertainty or noise processes.

Finally, the *estimation algorithm* can evolve from various influences. A preconceived notion of the structure of the estimator heavily influences the resulting algorithm. We may choose, based on computational considerations, to calculate an estimate recursively, rather than as a result of a batch process because we require an online, pseudo-real-time estimate. Also each algorithm must provide a measure of *estimation quality*, usually in terms of the expected estimation error. This measure provides a means for comparing estimators. Thus, the estimation procedure is a combination of these three major ingredients: criterion, models, and algorithm.

1.3 Model-Based Processing

Another view of the underlying processing problem is to decompose it into a set of steps that capture the strategic essence of the processing scheme. Inherently, we believe that the more "a priori" knowledge about the measurement and its underlying phenomenology we can incorporate into the processor, the better we can expect the processor to perform – as long as the information that is included is correct! One strategy called the *model-based approach* provides the essence of model-based signal processing (MBSP) [3]. Some believe that all signal processing schemes can be cast into this generic framework. Simply, the model-based approach is "incorporating mathematical models of both physical phenomenology and the measurement process (including noise) into the processor to extract the desired information." This approach provides a mechanism to incorporate knowledge of the underlying physics or dynamics in the form of mathematical process models along with measurement system models and accompanying noise as well as model uncertainties directly into the resulting processor. In this way, the model-based processor (MBP) enables the interpretation of results directly in terms of the problem physics. It is actually a modeler's tool enabling the incorporation of any a priori information about the problem to extract the desired information. This approach of selecting the appropriate model is depicted in the signal processing staircase of Figure 1.3. We note that as we progress up the "modeling" steps to increase the (SNR), the complexity of the model increases to achieve the desired results. This is our roadmap [3]. As depicted, the fidelity of the model incorporated into the processor determines the complexity of the model-based processor with the ultimate goal of increasing the inherent signal-to-noise ratio (SNR). These models can range from simple, implicit, nonphysical representations of the measurement data such as the Fourier or wavelet transforms to

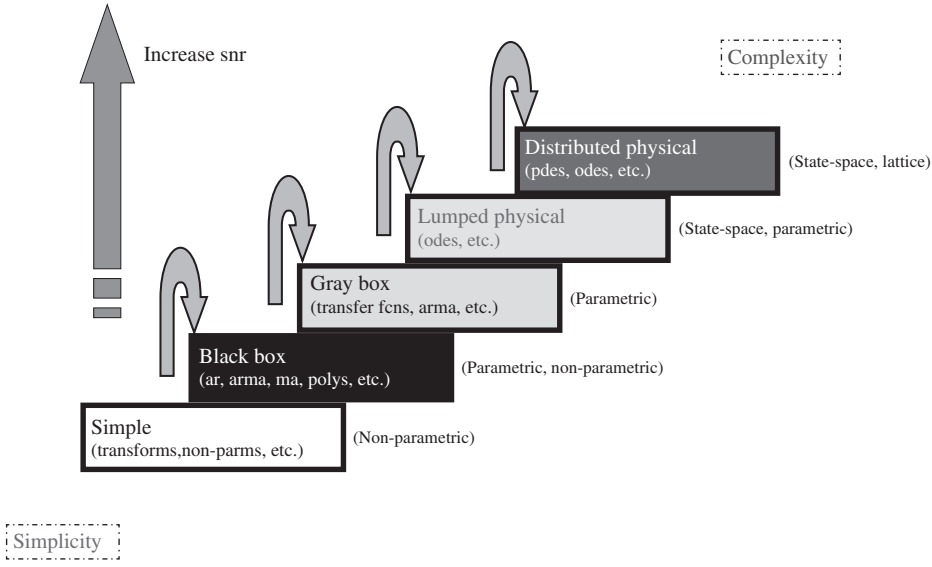

Figure 1.3 Model-based signal processing "model staircase": Step 1 – simple models. Step 2 – black-box models. Step 3 – gray-box models. Step 4 – lumped physical models. Step 5 – distributed physical models.

parametric black-box models used for data prediction, to lumped mathematical representations characterized by ordinary differential equations, to distributed representations characterized by partial differential equation models to capture the underlying physics of the process under investigation. The dominating factor of which model is the most appropriate is usually determined by how severe the measurements are contaminated with noise and the underlying uncertainties. In many cases, if the SNR of the measurements is high, then simple non-physical techniques can be used to extract the desired information; however, for low SNR measurements, more and more of the physics and instrumentation must be incorporated for the extraction. As we progress up the *modeling steps* to increase SNR, the model and algorithm complexity increases proportionally to achieve the desired results. Examining each of the steps individually leads us to realize that the lowest step using no explicit model (*simple*) essentially incorporates little a priori information and is used to analyze the information content (spectrum, time–frequency, etc.) of the raw measurement data to attempt to draw some rough conclusions about the nature of the signals under investigation. Progressing up to the next step, *black-box* models are basically used as data prediction mechanisms. They have a parametric form (e.g. polynomial, transfer function), but again there is little physical information that can be gleaned from their outputs. At the next step, the *gray-box* models evolve that can use the underlying black-box structures; however, now the parameters can be used

extract limited physical information from the data. For instance, a black-box transfer function model fit to the data yields coefficient polynomials, which can be factored to extract resonance frequencies and damping coefficients characterizing the overall system response being measured. Progressing further up the steps, we finally reach the true model-based techniques that explicitly incorporate the process physics using a *lumped physical* model structure usually characterized by ordinary differential or difference equations. The top step leads us to processes that are captured by *distributed physical* model structures in the form of partial differential equations. This level is clearly the most complex, since much computer horsepower is devoted to solving the physical propagation problem. So we see that model-based signal processing offers the ability to operate directly in the physical space of the phenomenologist with the additional convenience of providing a one-to-one correspondence between the underlying phenomenology and the model embedded in the processor.

Various levels of models can be used for processing purposes. It is important to investigate what, if anything, can be gained by filtering the data. The amount of information available in the data is related to the precision (variance) of the particular measurement instrumentation used as well as to any signal-processing devices or algorithms employed to improve the estimates. As we utilize more and more information about the physical principles underlying the given data, we expect to improve our estimates (decrease estimation error) significantly.

A typical measurement Y_{meas} is depicted in Figure 1.4. Accordingly, a reasonable model of this simple measurement instrument might be given by

$$Y_{\text{meas}} = S_{\text{true}} + N_{\text{noise}} \qquad \text{(Measurement)}$$

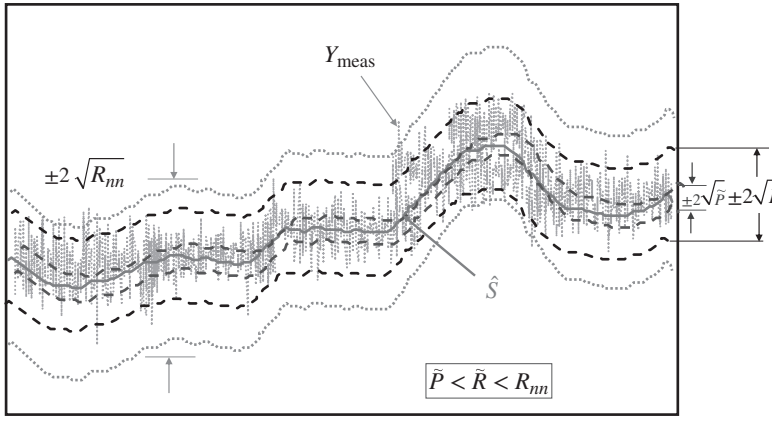

Figure 1.4 Model-based signal processing of a noisy measurement conceptually illustrating that the more "prior" information incorporated into the processor enables a reduction of error variance.

If we were to use Y_{meas} to estimate S_{true} (that is, $\hat{S}$), we have the noise lying within the $\pm 2\sqrt{\tilde{R}}$ confidence limits superimposed on the signal (see Figure 1.4). The best estimate of S_{true} we can hope for is only within the accuracy (bias) and precision (variance) of the instrument. If we include a model of the measurement instrument as well as its associated uncertainties, then we can improve the SNR of the noisy measurements. This technique represents the processing based on our instrument and noise (statistical) models given by

$$Y_{\text{meas}} = c(S_{\text{true}}) + N_{\text{noise}}, \quad N \sim \mathcal{N}(0, R_{nn}) \qquad \text{(Measurement and Noise)}$$

where $c(S_{\text{true}})$ is the measurement system model and $\mathcal{N}(0, R_{nn})$ is the noise statistics captured by a *Gaussian* or *normal distribution* of zero-mean and variance specified by R_{nn}.

We can also specify the estimation error variance $\tilde{R}$ or equivalently the quality of this processor in estimating S. Finally, if we incorporate not only instrumentation knowledge but also knowledge of the physical process, then we expect to do even better; that is, we expect the signal estimation error defined by $(\tilde{S} := S_{\text{true}} - \hat{S})$ variance $\tilde{P}$ to be small (see Figure 1.4 for $\pm 2\sqrt{\tilde{R}}$ confidence limits or bounds) as we incorporate more and more knowledge into the processor, that is

$$\begin{aligned} \hat{S} &= a(S_{\text{true}}) + W_{\text{noise}}, \quad W \sim \mathcal{N}(0, R_{ww}) \\ &\qquad \text{(Process Model and Noise)} \\ Y_{\text{meas}} &= c(S_{\text{true}}) + N_{\text{noise}}, \quad N \sim \mathcal{N}(0, R_{nn}) \\ &\qquad \text{(Measurement Model and Noise)} \end{aligned}$$

where $a(S_{\text{true}})$ is the process system model and $\mathcal{N}(0, R_{ww})$ is the process noise statistics captured by a zero-mean, Gaussian distribution with variance R_{ww}.

In fact, this is the case because it can be shown that

$$\text{Instrument variance} > \tilde{R} > \tilde{P}$$

This is the basic idea in model-based signal processing: "The more a priori information we can incorporate into the algorithm, the smaller the resulting error variance."

Consider the following example to illustrate these ideas.

Example 1.3 The voltage at the output of an RC-circuit is to be measured using a high impedance voltmeter shown in Figure 1.5. The measurement is contaminated with random instrumentation noise, which can be modeled as

$$e_{\text{out}} = K_e e + n$$

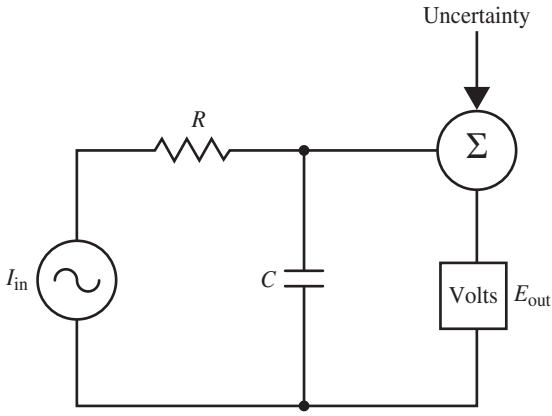

Figure 1.5 Model-based signal processing "model" for an RC-circuit.

where

e_{out} = measured voltage
K_e = instrument amplification factor
e = true voltage
n = zero-mean random noise of variance R_{nn}

This model corresponds to those described previously. A processor is to be designed to improve the precision of the instrument. Then, we have

- Measurement:

$$s \to e$$
$$c(\cdot) \to K_e$$
$$v \to n$$
$$R_{vv} \to R_{nn}$$

- and for the filter,

$$\hat{s} \to \hat{e}$$
$$\tilde{R}_{ss} \to \tilde{R}_{ee}$$

The precision of the instrument can be improved even further by including a model of the process (circuit). Writing the Kirchhoff node equations, we have

$$\dot{e} = \frac{1}{C} I_{in} - \frac{e}{RC} + q$$

where

$$R = \text{resistance}$$
$$C = \text{capacitance}$$
$$I_{\text{in}} = \text{excitation current}$$
$$q = \text{zero-mean random noise of variance } R_{qq}$$

The more sophisticated model-based processor employs both measurement and process models. Thus, we have

- Process:

$$\dot{s} \to \dot{e}$$
$$a(\cdot) \to -\frac{e}{RC} + \frac{1}{C} I_{\text{in}}$$
$$w \to q$$
$$R_{ww} \to R_{qq}$$

- Measurement:

$$s \to e$$
$$c(\cdot) \to K_e$$
$$v \to n$$
$$R_{vv} \to R_{nn}$$

- Therefore, the filter becomes

$$\hat{s} \to \hat{e}$$
$$\tilde{P}_{ss} \to \tilde{P}_{ee}$$

- such that

$$R_{vv} > \tilde{R}_{ee} > \tilde{P}_{ee}$$

This completes the example. □

Next, we illustrate the design of the model-based processor for this circuit.

Example 1.4 Suppose we are asked to design a simple RC-circuit as shown in Figure 1.5. The measurement is contaminated with random measurement noise as well as uncertainties in the values of the resistor and capacitor. Writing the Kirchhoff node equations, we have

$$I_{\text{in}} - \frac{e}{R} - C\dot{e} = 0$$

and the measurement given by the voltmeter is

$$e_{\text{out}} = K_e e.$$

Discretizing this equation using first differences with *sampling interval* ΔT and including white Gaussian noise sources for the parameter and measurement uncertainties, we develop the following discrete Gauss–Markov model:

$$e(t) = \underbrace{\left(1 - \frac{\triangle T}{RC}\right)}_{A} e(t-1) + \underbrace{\frac{\triangle T}{C}}_{B} I_{\text{in}}(t-1) + w(t-1)$$

$$e_{\text{out}}(t) = \underbrace{K_e}_{C} e(t) + v(t)$$

The resulting model-based processor (Kalman filter) is given by

$$\hat{e}(t|t) = \underbrace{\left(1 - \frac{\triangle T}{RC}\right)}_{A} \hat{e}(t-1|t-1) + K(t)e(t)$$

For $R = 3.3$ kΩ and $C = 1000$ μF, $\triangle T = 100$ ms, $K_e = 2.0$, $e_0 = 2.5$ V, $I_{\text{in}} = 100$ μA (negative step) with $w \sim \mathcal{N}(0, 0.0001)$ and $v \sim \mathcal{N}(0, 4.0)$ or

$$e(t) = 0.97e(t-1) + I_{\text{in}}(t-1) + w(t-1)$$
$$e_{\text{out}}(t) = 2e(t) + v(t)$$

The noisy measurement and corresponding model-based random signal estimate is shown in Figure 1.6. Here we see the noisy measurement data, the

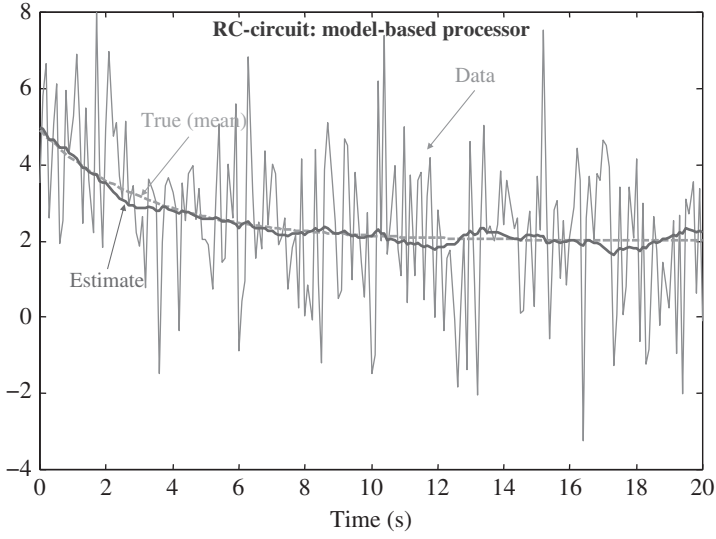

Figure 1.6 Model-based processor for RC-circuit: raw data (solid), true measurement (dashed), and model-based estimate (thick solid).

noise-free (true) measurement (dashed line) along with the corresponding "filtered" or model-based estimate. Note how the estimate *tracks* the true measurement significantly increasing the SNR enhancing the signal. This completes the example. □

There are many different forms of model-based processors depending on the models used and the manner in which the estimates are calculated. For example, there are process model-based processors (Kalman filters [3]), statistical model-based processors (Box–Jenkins filters [7], Bayesian filters [6]), statistical model-based processors (covariance filters [2]), or even optimization-based processors (gradient filters [8]). In any case, many processors can be placed in a recursive form with various subtleties emerging in the calculation of the current estimate ($\hat{S}_{\text{old}}$). The standard technique employed is based on correcting or updating the current estimate as a new piece of measurement data becomes available. The estimates generally take the *recursive form*:

$$\hat{S}_{\text{new}} = \hat{S}_{\text{old}} + KE_{\text{new}}$$

where

$$E_{\text{new}} = Y - \hat{Y}_{\text{old}} = Y - C\hat{S}_{\text{old}}$$

Here we see that the new estimate is obtained by correcting the old estimate by a K-weighted amount. The error term E_{new} is the new information or innovation; that is, it is the difference between the actual measurement and the predicted measurement ($\hat{Y}_{\text{old}}$) based on the old estimate ($\hat{S}_{\text{old}}$). The computation of the weight K depends on the criterion used (e.g. mean-squared error, absolute error).

Consider the following example, which demonstrates how to recursively estimate the sample mean.

Example 1.5 The sample mean estimator can easily be put in the recursive form. The estimator is given by

$$\hat{S}(N) = \frac{1}{N}\sum_{t=1}^{N} y(t)$$

Extracting the Nth term from the sum, we obtain

$$\hat{S}(N) = \frac{1}{N}y(N) + \frac{1}{N}\sum_{t=1}^{N-1} y(t)$$

Identify $\hat{S}(N-1)$ from the last term, that is,

$$\hat{S}(N) = \frac{1}{N}y(N) + \left(\frac{N-1}{N}\right)\hat{S}(N-1)$$

The recursive form is given by

$$\underbrace{\hat{S}(N)}_{S_{\text{new}}} = \underbrace{\hat{S}(N-1)}_{S_{\text{old}}} + \underbrace{\frac{1}{N}}_{K} \underbrace{[y(N) - \hat{S}(N-1)]}_{E_{\text{new}}} \qquad \square$$

This completes the introductory discussion on model-based signal processing from a heuristic point of view. Next we discuss methods of *extracting* a model from measurement data in order to construct an MBP.

1.4 Model-Based Identification

Identification provides a solution to the problem of extracting a model from measured data sequences either time series, frequency data or simply an ordered set of indexed values. As illustrated in Figure 1.7, both inputs (when available) and measured outputs are used to perform the extraction in spite of any disturbances or parametric uncertainties. Since its fundamental evolution from control theory, identification or "system" identification is based on estimating a model of a *dynamic* system [7] from data. Models can be of many varieties ranging from simple polynomials to highly complex constructs evolving from nonlinear distributed systems. The extraction of a model from data is critical for a large number of applications evolving from the detection of submarines in a varying ocean, to tumor localization in breast tissue, to pinpointing the epicenter of a highly destructive earthquake or to simply monitoring the condition of a motor as it drives a critical system component [3].

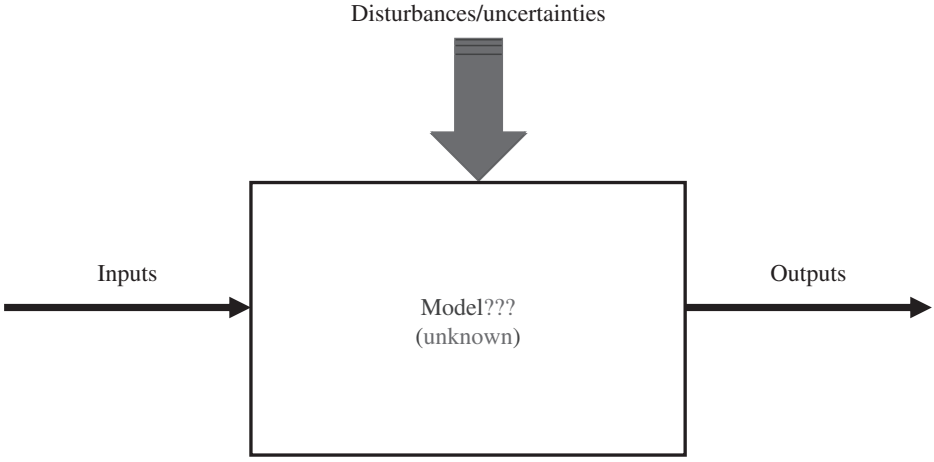

Figure 1.7 System identification problem: input (deterministic/random), disturbances (noise/parametric uncertainty), output (noisy measurement), model (class selected).

Each of these applications requires an aspect of modeling and fundamental understanding (when possible) of the underlying phenomenology governing the process as well as the measurement instrumentation extracting the data along with the accompanying uncertainties. Some of these problems can be solved with simply a "black-box" representation that faithfully reproduces the data in some manner without the need to capture the underlying dynamics (e.g. common check book entries) or to a "gray box" model that has been extracted, but has parameters of great interest (e.g. unknown mass of a toxic material). However, when the true need exists to obtain an accurate representation of the underlying phenomenology like the structural dynamics of an aircraft wing or the untimely vibrations of a turbine in a nuclear power plant, then more sophisticated representations of the system and uncertainties are clearly required. In cases such as these, models that capture the dynamics must be developed and "fit" to the data in order to perform applications such as condition monitoring of the structure or failure detection/prediction of a rotating machine. Here models can evolve from lumped characterizations governed by sets of ordinary differential equations, linear or nonlinear, or to distributed representations evolved from sets of partial differential equations. All of these representations have one thing in common, when the need to perform a critical task is at hand – they are represented by a mathematical model that captures their underlying phenomenology that must somehow be extracted from noisy measurements. This is the fundamental problem that we address in this text, but we must restrict our attention to a more manageable set of representations, since many monographs have addressed problem sets targeting specific applications [9, 10].

In fact, this concept of specialty solutions leads us to the generic *state-space* model (Chapter 3) of systems theory and controls. Here the basic idea is that all of the theoretical properties of a system are characterized by this fundamental model set that enables the theory to be developed and then applied to any system that can be represented in the state-space. Many models naturally evolve in the state-space, since it is essentially the representation of a set of nth order differential equations (ordinary or partial, linear or nonlinear, time (space) invariant or time (space) varying, scalar or multivariable) that are converted into a set of first-order equations, each one of which is a state. For example, a simple mechanical system consisting of a single mass, spring, damper construct is characterized by a set of second-order, linear, time-invariant, differential equations that can simply be represented in state-space form by a set of two first-order equations, each one representing a state: one for displacement and one for velocity [9]. We employ the state-space representation *throughout* this text and provide sufficient background in Chapter 3.

Here we are primarily focused on the development of models for the design and application of model-based signal processors (MBSP) using

estimation/identification techniques to achieve a meaningful design. For model-based identification (MBID), we incorporate a *known structure*, rather than strictly a generic class of model (e.g. state-space, transfer function, polynomial); that is, we have possibly a physics-based structure and would like to "jointly" estimate the unknown or poorly known parameters embedded within that structure. The problem is similar to what the identification community might term a "gray-box" model [7] and processor to estimate parameters. Here we call it a parametrically adaptive processor in which the distinction can be made between three fundamental, but different, problems. The first problem is simply a parameter estimation, in which no particular model is embedded, such as in classic least-squares estimation [3, 11]. The second can be thought of as a simple signal estimation problem such as a model-based processor with known model as in the RC-circuit example. The third is the problem that we are most concerned and defines what we mean by model-based identification compared to subspace identification (discussed in the next section). We have defined this problem and the resulting processor in various contexts and applications as a *parametrically adaptive* processor [3, 6] in which both signal and parameters are estimated simultaneously. Therefore, we define the *model-based identification problem* as the joint estimation of unknown signal/parameters of a known structure from noisy measurement data. In essence, it is the development of a parametrically adaptive processor of known structure and capable of solving the joint signal/parametric estimation problem. This processor is known as a *model-based identifier* and is well known in the literature that was developed for a wealth of various applications [3, 6].

Identification is broad in the sense that it does not limit the problem to various classes of models directly. For instance, for an unknown system, a model set is selected with some perception that it is capable of representing the underlying phenomenology adequately, then this set is identified directly from the data and validated for its accuracy. There is clearly a well-defined procedure that captures this approach to solve the identification problem [7, 10]. In some cases, the class structure of the model may be known a priori, but the order or equivalently the number of independent equations to capture its evolution is not (e.g. number of oceanic modes). Here, techniques to perform order estimation precede the fitting of model parameters first and are followed by the parameter estimation to extract the desired model [12]. In other cases, the order is known from prior information and parameter estimation follows directly (e.g. a designed mechanical structure). In any case, these constraints govern the approach to solving the identification problem and extracting the model for application. Many applications exist, where it is desired to monitor a process and track a variety of parameters as they evolve in time (e.g. radiation detection), but in order to accomplish this on-line, the model-based processor or more appropriately the model-based identifier must update the model

parameters sequentially in order to accomplish its designated task. We develop these processors for both linear and nonlinear models in Chapters 4 and 5.

In Chapter 5, we present a different approach that incorporates the solution into the identification problem as the integral part of the model-based signal processor (Kalman filter) that can be applied to a large number of applications, but with little success unless a reliable model is available or can be adapted to a changing environment [13]. Using various approaches, it is possible to identify the model very rapidly and incorporate it into a variety of processing problems such as state estimation, tracking, detection, classification, controls, and communications to mention a few [14, 15].

As an example, consider the development of a processor to extract the unknown parameters from the RC-circuit.

Example 1.6 Suppose we investigate the RC-circuit and would like to dynamically estimate the circuit parameters as well as its output voltage. The model in state-space form with $x := e$ and $y := e_{\text{out}}$ is given by (see Example 1.4)

$$\begin{aligned} e(t) &= \theta_1 e(t-1) + \theta_2 I_{\text{in}}(t-1) + w(t-1) \\ e_{\text{out}}(t) &= \theta_3 e(t) + v(t) \end{aligned}$$

The processor can be implemented in a number of ways, one of which as a least-squares parameter estimator using a strictly parametric estimator based on

$$\begin{aligned} \theta(t) &= \theta(t-1) + w(t-1) \\ e_{\text{out}}(t) &= \theta(t) + v(t) \end{aligned}$$

or as a MBID-processor (nonlinear) embedding the circuit model and jointly estimating both states and parameters based on the augmented model $x := [e \;\; | \;\; \theta_1 \;\; | \;\; \theta_2 \;\; | \;\; \theta_3]'$

$$\begin{aligned} e(t) &= \theta_1(t-1)e(t-1) + \theta_2(t-1)I_{\text{in}}(t-1) + w_x(t-1) \\ \theta(t) &= \theta(t-1) + w_\theta(t-1) \\ e_{\text{out}}(t) &= \theta_3(t)e(t) + v(t) \end{aligned}$$

The results are shown in Figure 1.8, where we see each of the parameter estimates as well as the filtered measurement. In Figure 1.8a, we see the parameter estimates for both the system ($\hat{\theta}_1 \approx -0.97$) and input ($\hat{\theta}_2 \approx -0.1$) model parameters, while in Figure 1.8b we observe both the output ($\hat{\theta}_3 \approx 2.0$) parameter and enhanced (filtered) output voltage measurement ($\hat{y} \approx \hat{e}_{\text{out}}$). This completes the design of the MBID. □

Next we consider the heart of the text – subspace identification.

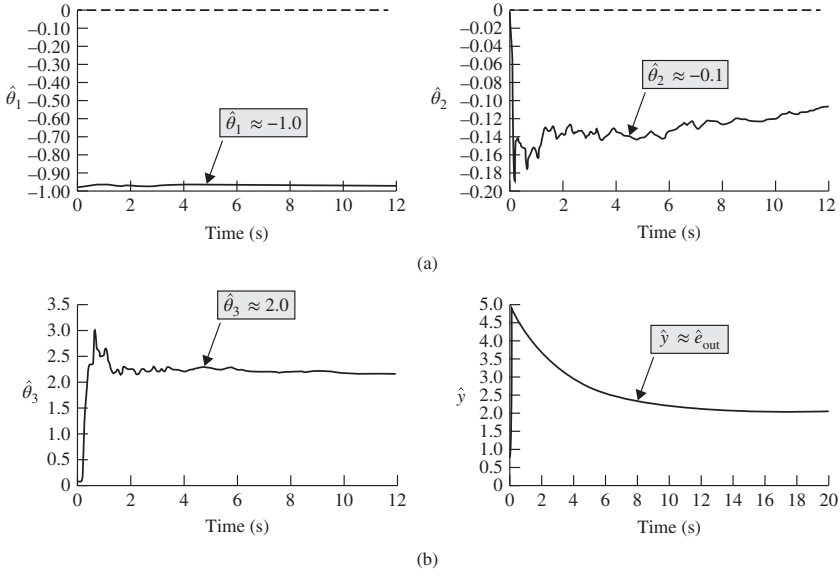

Figure 1.8 Model-based identification of RC-circuit parameters: $\hat{\theta}_1 \approx -0.97$, $\hat{\theta}_2 \approx -0.1$, $\hat{\theta}_3 \approx 2.0$ and $\hat{y} \approx \hat{e}_{out}$.

1.5 Subspace Identification

Models for the processor evolve in a variety of ways, either from first principles accompanied by estimating its inherent uncertain parameters as in parametrically adaptive schemes [3], or by extracting constrained model sets employing direct optimization methodologies [7], or by simply fitting a black-box structure to noisy data [8]. Once the model is extracted from controlled experimental data or data synthesized from a highly complex truth model, the long-term processor can be developed for direct application [6]. Since many real-world applications seek a real-time solution, we concentrate primarily on the development of fast, reliable identification methods that enable such an implementation [16–18]. Model extraction/development must be followed by validation and testing to ensure that the model reliably represents the underlying phenomenology – a bad model can only lead to failure!

The concept of subspace has a well-defined meaning in system identification. Simply stated, "a *subspace* is a space or subset of a parent space that inherits some or all of its characteristics." For instance, we can consider each layer of an onion a subset (subspace) of the onion itself, each inheriting its generic properties. In mathematics, we can consider the decomposition of a topological space into subsets, each inheriting properties of the previous region or subregion creating a hierarchy. For example, a topological space with

all of its properties can be decomposed into subregions (layers): first into a topological-metric-space, next into a topological-normalized-metric-space, and finally into a topological-normalized-metric-inner product-space with each subregion (layer) inheriting some of the properties of its predecessor. In identification, subspace refers primarily to a vector (state) space in which its essential characteristics can be captured by a "smaller" space – the subspace. For example, an Nth order state-space dynamic system can be approximated by an Mth order ($M < N$) system capturing the majority of its properties. This is the concept that we will employ throughout this text.

Subspace identification (SID) is a technique to extract (identify) a black-box model in generic state-space form from uncertain input/output data using robust, numerically stable, linear algebraic methods. SID (simply) performs a black-box identification of a generic state-space model (A, B, C, D) from noisy measurements as illustrated in Figure 1.9. Unlike MBID discussed previously, it does *not* require any specific structure of the state-space model, only its order (number of states) that can be derived directly from the embedded linear algebraic decomposition. SID methods are constrained to *linear* state-space models.

More specifically, compared to the MBID processors that iteratively estimate parameters of the previous section [3, 7], we investigate subspace identification techniques [16–18]. These techniques enable the capability to extract a large number of parameters compared to the prediction error (P-E) methods that are quite time-consuming and not practical for large parametric problems coupled with the need for potential real-time applications. Subspace identification techniques are robust and reliable and capable of real-time application,

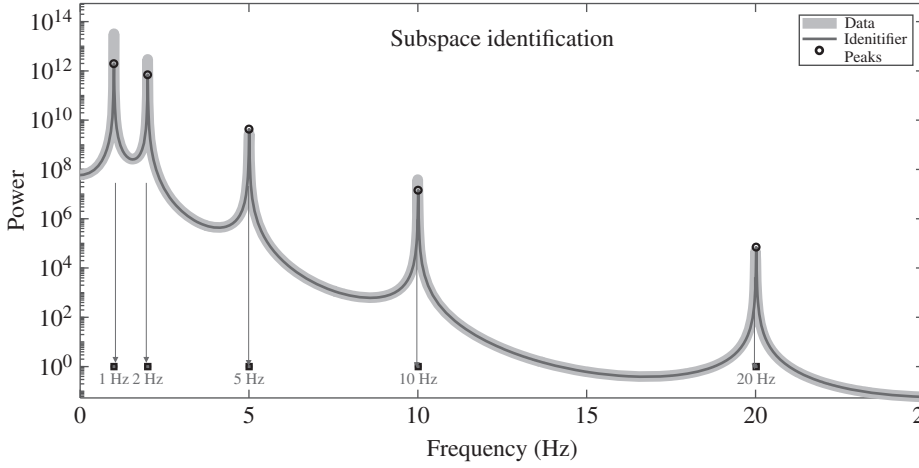

Figure 1.9 Identification of a sinusoidal model from noise-free data: spectral frequencies (identified) at 1 Hz, 2, 5, 10, and 20 Hz.

but they are not as accurate as the optimization-based P-E methods [8]. In fact, the SID methods are typically used as "starters" for P-E processors enabling a reasonable starting point for the numerical optimizers, thereby ensuring faster convergence. In this text, we concentrate on the SID approach, presented in Chapters 6 and 7.

Consider the following example of sinusoids in noise to illustrate this process.

Example 1.7 Suppose we are given an ideal measurement consisting of five sinusoids at frequencies ranging from 1 to 20 Hz and we would like to develop a model of this process, after performing an identification, that is, our underlying model is simply a sum of sinusoids of varying amplitudes such that

$$y(t) = \sum_{i=1}^{5} A_i \sin 2\pi \ f_i t \ \text{ with } f_i = \{1, 2, 5, 10, 20\} \text{ Hz} \tag{1.1}$$

We could simply perform a Fourier transform in this noise-free case and locate the peaks and magnitudes to extract the sinusoids or perform a parametric estimation using a nonlinear (sinusoid) optimization technique yielding a model-based identification. However, another approach is to perform a *black-box* identification choosing a generic model-class (e.g. state-space) that captures these dynamics in the form of a transfer function, estimate the underlying state-space model, and using it, calculate the corresponding impulse response. With this representation, the corresponding frequencies can be extracted from the eigenvalues (poles) of the identified model. The data were processed by an identifier using the proper order (20-states or $2 \times$ No. sinusoids), and the results are shown in Figure 1.9, both in the Fourier (power) spectrum obtained from the simulated noise-free data along with the estimated (identified) frequencies (squares) annotated on the plot. There is perfect agreement. This example illustrates the concept of identification (black-box) and becomes more and more important as the SNR decreases and the spectral peaks are no longer available for detection and location. □

This completes the introduction to subspace identification. Next we discuss then notation that will be employed in this text.

1.6 Notation and Terminology

The notation used throughout this text is standard in the literature. Where necessary, vectors are represented by boldface, lowercase, $\mathbf{x}$, and matrices by boldface, uppercase, $\mathbf{A}$. The *adjoint* of the matrix $\mathbf{A}$ is annotated as adj($\mathbf{A}$) and its corresponding *determinant* by det($\mathbf{A}$). We define the notation $\underline{N}$ to be a shorthand way of writing $1, 2, \ldots, N$. It will be used in matrices, $A(\underline{N})$ to mean

there are N-columns of A. We also use the equivalent MATLAB notation $(1:N)$. We denote the real part of a signal by Re and its imaginary part by Im. As mentioned previously, estimators are annotated by the caret, such as $\hat{x}$. We also define partial derivatives at the component level by $\frac{\partial}{\partial \theta_i}$, the N_θ- gradient vector by ∇_θ and higher order partials by ∇^2_θ.

The most difficult notational problem will be with the "time" indices. Since this text is predominantly discrete-time, we will use the usual time symbol t to mean a discrete-time index, i.e. $t \in \mathcal{I}$ for $\mathcal{I}$ the set of integers. However, and hopefully not too confusing, t will also be used for continuous-time, i.e. $t \in \mathcal{R}$ for $\mathcal{R}$ the set of real numbers denoting the continuum. When used as a continuous-time variable, $t \in \mathcal{R}$, it will be represented as a subscript to distinguish it, i.e. $x_{\mathbf{t}}$. This approach of choosing $t \in \mathcal{I}$ primarily follows the system identification literature and for the ease of recognizing discrete-time variable in transform relations (e.g. discrete Fourier transform). The rule of thumb is therefore to "interpret t as a discrete-time index unless noted by a subscript as continuous in the text." With this in mind, we will define a variety of discrete estimator notations as $\hat{x}(t|t-1)$ to mean the estimate at time (discrete) t based upon all of the previous data up to $t-1$. We will define these symbols prior to their use with the text to assure no misunderstanding of its meaning.

With a slight abuse of notation, we will use the terminology *distribution* of X, $\Pr(X)$ in general, so as not to have to differentiate between density for continuous random variables or processes and mass for discrete variates. It will be obvious from the context which is meant. In some cases, we will be required to make the distinction between cumulative distribution function (CDF) and density (PDF) or mass (PMF) functions. Here we use the uppercase notation $P_X(x)$ for the CDF and lowercase $p_X(x)$ for the PDF or PMF.

Subsequently, we will also need to express a discrete PMF as a continuous PDF using impulse or delta functions as "samplers" much the same as in signal processing when we assume there exists an impulse sampler that leads to the well-known Nyquist sampling theorem [3]. Thus, corresponding to a discrete PMF we can define a continuous PDF through the concept of an *impulse sampler*, that is, given a discrete PMF defined by

$$p_X(x) \approx p(X = x_i) = \sum_i p_i \;\; \delta(x - x_i) \tag{1.2}$$

then we define the *equivalent continuous* PDF as $p_X(x)$. Moments follow from the usual definitions associated with a continuous PDF; for instance, consider the definition of the expectation or mean. Substituting the equivalent PDF and utilizing the sifting property of the impulse function gives

$$E\{x\} = \int_{-\infty}^{\infty} x \;\; p_X(x) \;\; dx = \int_{-\infty}^{\infty} x \;\; \left(\sum_i p_i \delta(x - x_i) \right) dx = \sum_i x_i p_i \tag{1.3}$$

which is precisely the mean of the discrete PMF(see Appendix A for more details).

Also, as mentioned, we will use the symbol $\sim$ to mean "distributed according to" as in $x \sim \mathcal{N}(m, v)$ defining the random variable x as Gaussian distributed with mean m and variance v. We may also use the extended notation: $\mathcal{N}(x : m, v)$ to include the random variable x as well. When *sampling*, we use the *non-conventional* right arrow "action" notation $\rightarrow$ to mean "draw a sample from" a particular distribution such as $x_i \rightarrow \Pr(x)$ – this again will be clear from the context. When *resampling*, that is, replacing samples with new ones, we use the "block" right arrow such as $x_j \leftrightarrow x_i$ meaning new sample x_j replaces current sample x_i. In a discrete (finite) probabilistic representation, we define a purely discrete variate as $x_k(t) := \Pr(x(t) = \mathcal{X}_k)$ meaning that x can only take on values (integers) k from a known set $\mathcal{X} = \{\mathcal{X}_1, \ldots, \mathcal{X}_k, \ldots, \mathcal{X}_N\}$ at time t. We also use the symbol $\square$ to mark the end of an example.

We define two projection operator projection operators: (i) orthogonal and (ii) oblique or parallel. The *orthogonal projection operator* "projects" $\bullet$ onto $\mathcal{Y}$ as $\mathcal{P}_{\bullet|\mathcal{Y}}$ or its complement $\mathcal{P}^{\perp}_{\bullet|\mathcal{Y}}$, while the *oblique projection operator* "projects" $\bullet$ onto $\mathcal{Y}$ "along" $\mathcal{Z}$ as $\mathcal{P}^{||}_{\bullet|\mathcal{Y}\circ\mathcal{Z}}$. For instance, orthogonally projecting the vector $\mathbf{x}$, we have $\mathcal{P}_{\mathbf{x}|\mathcal{Y}}$ or $\mathcal{P}^{\perp}_{\mathbf{x}|\mathcal{Y}}$, while obliquely projecting $\mathbf{x}$ along $\mathcal{Z}$ is $\mathcal{P}_{\mathbf{x}|\mathcal{Y}\circ\mathcal{Z}}$ (see Appendix B). The matrix projection operator evolves from decompositions such as the LQ-decomposition (see Appendix C), where the operator is given in terms of its orthonormal bases as $\mathcal{P} = Q \times Q'$.

1.7 Summary

In this chapter, we introduced the concept of model-based signal processing (MBSP) based on the idea of incorporating more and more available a priori information into the processing scheme. Various signals were classified as deterministic or random. When the signal is random, the resulting processors are called estimation filters or estimators. A procedure was defined (estimation procedure) leading to a formal decomposition, which will be used throughout this text. After a couple of examples motivating the concept of model-based signal processing, a more formal discussion followed with a concrete RC-circuit example completing the chapter. Next we introduced the idea of model-based identification (MBID), which is equivalent to a variety of P-E methods, where a model developed from the underlying problem phenomenology is incorporated directly into the processor. After discussing the identification of the RC-circuit parameters, the subspace identification (SID) approach we introduced conceptually and illustrated through a sinusoidal identification example. Finally, the notation utilized throughout the text was discussed completing this introductory chapter.

MATLAB Notes

MATLAB is a command oriented vector–matrix package with a simple yet effective command language featuring a wide variety of embedded C language constructs making it ideal for signal processing and graphics. All of the algorithms we have applied to the examples and problems in this text are MATLAB-based in solution ranging from simple simulations to complex applications. We will develop these notes primarily as a summary to point out to the reader many of the existing commands that already perform the signal processing operations discussed in the presented chapter and throughout the text.

More specifically, MATLAB consists of a variety of "toolboxes" enabling the users to access a large number of numerically stable and accurate algorithms rapidly along with a corresponding suite of demonstration examples. For this text, the most important toolboxes are the signal processing (**signal**), system identification (**ident**), control system (**control**), and the statistics (**stats**), which can be used to solve the majority of problems introduced in this text. Throughout we will refer to specific commands that can be applied to a variety of problems and provide meaningful solutions.

References

1 Rossing, T., Dunn, F., Hartmann, W. et al. (2007). *Handbook of Acoustics*. New York: Springer.
2 Hayes, M. (1996). *Statistical Digital Signal Processing and Modeling*. New York: Wiley.
3 Candy, J. (2006). *Model-Based Signal Processing*. Hoboken, NJ: Wiley/IEEE Press.
4 Bucker, H. (1976). Use of calculated sound fields and matched-field detection to locate sound in shallow water. *J. Acoust. Soc. Am.* 59: 329–337.
5 Tolstoy, A. (1993). *Matched Field Processing for Ocean Acoustics*. Hackensack, NJ: World Scientific.
6 Candy, J. (2016). *Bayesian Signal Processing, Classical, Modern and Particle Filtering*, 2e. Hoboken, NJ: Wiley/IEEE Press.
7 Ljung, L. (1999, 2e). *System Identification: Theory for the User*. NJ: Prentice-Hall.
8 Ljung, L. and Soderstrom, T. (1983). *Theory and Practice of Recursive Identification*. Cambridge, MA: M.I.T. Press.
9 Juang, J. (1994). *Applied System Identification*. Upper Saddle River, NJ: Prentice-Hall PTR.
10 Aoki, M. (1990). *State Space Modeling of Time Series*, 2e. London: Springer.
11 Sorenson, H. (1986). *Parameter Estimation*. New York: IEEE Press.

12 Norton, J. (1986). *An Introduction to Identification*. New York: Academic Press.
13 Kalman, R. (1960). A new approach to linear filtering and prediction problems. *Trans. ASME J. Basic Eng.* 82: 34–45.
14 Van Der Veen, A., Deprettere, E.F., and Swindlehurst, A.L. (1993). Subspace-based methods for the identification of linear time-invariant systems. *Proc. IEEE* 81 (9): 1277–1308.
15 Viberg, M. (1995). Subspace based signal analysis using singular value decomposition. *Automatica* 31 (12): 1835–1851.
16 van Overschee, P. and De Moor, B. (1996). *Subspace Identification for Linear Systems: Theory, Implementation, Applications*. Boston, MA: Kluwer Academic Publishers.
17 Verhaegen, M. and Verdult, V. (2007). *Filtering and System Identification: A Least-Squares Approach*. Cambridge: Cambridge University Press.
18 Katayama, T. (2005). *Subspace Methods for System Identification*. London: Springer.

Problems

1.1 We are asked to estimate the displacement of large vehicles (semitrailers) when parked on the shoulder of a freeway and subjected to wind gusts created by passing vehicles. We measure the displacement of the vehicle by placing an accelerometer on the trailer. The accelerometer has inherent inaccuracies, which is modeled as

$$y = K_a x + n$$

with y, x, n the measured and actual displacement and white measurement noise of variance R_{nn} and K_a the instrument gain. The dynamics of the vehicle can be modeled by a simple mass–spring–damper. (a) Construct and identify the measurement model of this system. (b) Construct and identify the process model and model-based estimator for this problem.

1.2 Think of measuring the temperature of a liquid in a breaker heated by a burner. Suppose we use a thermometer immersed in the liquid and periodically observe the temperature and record it. (a) Construct a measurement model assuming that the thermometer is linearly related to the temperature, that is, $y(t) = k\Delta T(t)$. Also model the uncertainty of the visual measurement as a random sequence $v(t)$ with variance R_{vv}. (b) Suppose we model the heat transferred to the liquid from the burner as

$$Q(t) = C\, A\Delta T(t),$$

where C is the coefficient of thermal conductivity, A is the cross-sectional area, and $\triangle T(t)$ is the temperature gradient with assumed random uncertainty $w(t)$ and variance R_{ww}. Using this process model and the models developed above, identify the model-based processor representation.

1.3 We are given an RLC series circuit driven by a noisy voltage source $V_{\text{in}}(t)$ and we use a measurement instrument that linearly amplifies by K and measures the corresponding output voltage. We know that the input voltage is contaminated by and additive noise source, $w(t)$ with covariance, R_{ww} and the measured output voltage is similarly contaminated with noise source, $v(t)$ with R_{vv}.
(a) Determine the model for the measured output voltage, $V_{\text{out}}(t)$ (measurement model).
(b) Determine a model for the circuit (process model).
(c) Identify the general model-based processor structures. In each scheme, specify the models for the process, measurement, and noise.

1.4 A communications satellite is placed into orbit and must be maneuvered using thrusters to orientate its antennas. Restricting the problem to the single axis perpendicular to the page, the equations of motion are

$$J\frac{d^2\theta}{dt^2} = T_{\text{c}} + T_{\text{d}}$$

where J is the moment of inertia of the satellite about its center of mass, T_{c} is the thruster control torque, T_{d} is the disturbance torque, and θ is the angle of the satellite axis with respect to the inertial reference (no angular acceleration) A. Develop signal and noise models for this problem and identify each model-based processor component.

1.5 Consider a process described by a set of linear differential equations

$$\frac{d^2c}{dt^2} + \frac{dc}{dt} + c = Km.$$

The process is to be controlled by a proportional–integral–derivative (PID) control law governed by the equation

$$m = K_p\left(e + \frac{1}{T_i}\int e\,dt + T_{\text{d}}\frac{de}{dt}\right)$$

and the controller reference signal r is given by

$$r = e + c$$

Suppose the reference is subjected to a disturbance signal, and the measurement sensor, which is contaminated with additive noise, measures the "square" of the output. Develop the model-based signal and noise models for this problem.

1.6 The elevation of a tracking telescope is controlled by a DC motor. It has a moment of inertia J and damping B due to friction, and the equation of motion is given by

$$J\frac{d^2\theta}{dt^2} + B\frac{d\theta}{dt} = T_{\mathrm{m}} + T_{\mathrm{d}},$$

where T_{m} and T_{d} are the motor and disturbance torques and θ is the elevation angle. Assume a sensor transforms the telescope elevation into a proportional voltage that is contaminated with noise. Develop the signal and noise models for the telescope and identify all of the model-based processor components.

1.7 Suppose we have a two-measurement system given by

$$y = \begin{bmatrix} 3 \\ 4 \end{bmatrix} + v$$

where $R_{vv} = \mathrm{diag}[1,\ \ 0.1]$.
(a) What is the batch least-squares estimate ($W = I$) of the parameter x, if $y = [7\ \ 21]'$?
(b) What is the batch weighted least-squares estimate of the parameter x with W selected for minimum variance estimation?

1.8 Calculate the batch and sequential least-squares estimate of the parameter vector x based on two measurements $y(1)$ and $y(2)$ where

$$y(1) = C(1)x + v(1) = \begin{bmatrix} 2 \\ 1 \end{bmatrix}$$

$$y(2) = c'x + v(2) = 4$$

$$C = \begin{bmatrix} 1 & 1 \\ 0 & 1 \end{bmatrix}, \quad c'(1) = [1\ \ 2], \quad W = I$$