Chapter 7

SYSTEM IDENTIFICATION

Identification of process dynamics is perhaps the most time consuming step in implementing an MPC and one that requires relatively high expertise from the user. In this section, we give introduction to various identification methods and touch upon some key issues that should help an engineer obtain models on a reliable basis. Since system identification is a very broad subject that can easily take up an entire book, we will limit our objective to giving just an overview and providing a starting point for further exploration of the field. Because of this, our treatment of various methods and issues will necessarily be brief and informal. References will be given at the end for more complete, detailed treatments of the various topics presented in this chapter.

7.1 PROBLEM OVERVIEW

The goal of identification is to obtain a mathematical relation that reliably predicts the behavior of outputs, using input output data gathered from the process. For conveninence, the mathematical relation searched for is often limited to linear ones. As we saw in the previous chapter, both known and

unknown inputs affect the outputs. Since many inputs change in a random, but correlated manner, it is often desirable to identify a model that has both deterministic and stochastic components.

In terms of how input output data are translated into a mathematical relation, the field of identification can be divided broadly into two branches: parametric identification and nonparametric identification. In parameteric identification, the structure of the mathematical relation is fixed *a priori* and parameters of the structure are fitted to the data. In nonparametric identification, no (or very little) assumption is made with respect to the model structure. Frequency response identification is nonparametric. Impulse response identification is also nonparametric, but it can also be viewed as parametric identification since a impulse response of a finite length is often identified.

As a final note, it is important not to forget the end-use of the model, which is to analyze and design a feedback control system in our case. Accuracy of a model must ultimately be judged in view of how well the model predicts the output behavior with the intended feedback control system in place. This consideration must be reflected in all phases of identification including test input design, data filtering, model structure selection, and parameter estimation.

7.2 PARAMETRIC IDENTIFICATION METHODS

In parametric identification, the model structure is set in prior to the model fitting. The key objective is then to identify the model parameters, based on given input output data. Although a particular model structure is assumed for parameter estimation, one often adjusts the model structure iteratively based on the result of fitting (for example, through residual analysis).

7.2.1 MODEL STRUCTURES

A general structure for parametric identification is:

$$y(k) = G(q,\theta)u(k) + H(q,\theta)\varepsilon(k) \tag{7.1}$$

where y is the output and u is the input (most of times, this will be a manipulated input, but it can also be a measured disturbance variable). For systems with stationary disturbances, $\varepsilon(k)$ can be assumed to be white noise and $H(q,\theta)$ a stable, stably invertible and normalized (*i.e.*, $H(\infty,\theta) = 1$) transfer function, without loss of generality. In the case that the disturbance is better described by a stochastic process driven by *integrated* white noise, we can replace $y(k)$ and $u(k)$ with $\Delta y(k)$ and $\Delta u(k)$.

Within the general structure, different parametrizations exist. Let us discuss some popular ones, first in the single input, single output context.

- *ARX Model* If we represent G as a rational function and express it as a linear equation with an additive error term, we obtain

$$y(k)+a_1y(k-1)+\cdots+a_ny(k-n) = b_1u(k-1)+\cdots+b_mu(k-m)+\varepsilon(k) \tag{7.2}$$

  When the equation error $\varepsilon(k)$ is taken as a white noise sequence, the resulting model is called an ARX model (AR for Auto-Regressive and X for eXtra input u). Hence, the ARX model corresponds to the following parametrization of the transfer functions:

$$\begin{aligned} G(q,\theta) &= \frac{B(q)}{A(q)} \triangleq \frac{b_1q^{-1}+\cdots+b_mq^{-m}}{1+a_1q^{-1}+\cdots+a_nq^{-n}} \\ H(q,\theta) &= \frac{1}{A(q)} \triangleq \frac{1}{1+a_1q^{-1}+\cdots+a_nq^{-n}} \end{aligned} \tag{7.3}$$

  A high-order ARX model is a good choice when the system order is

unknown. To see this, note that (7.1) can be written as

$$H^{-1}(q,\theta)y(k) = H^{-1}(q,\theta)G(q,\theta)u(k) + \varepsilon(k) \tag{7.4}$$

Since H^{-1} is assumed stable, if G is stable,

$$\begin{aligned} H^{-1}(q,\theta) &\approx 1 + a_1 q^{-1} + \cdots + a_n q^{-n} \\ H^{-1}(q,\theta)G(q,\theta) &\approx 1 + b_1 q^{-1} + \cdots + b_m q^{-m} \end{aligned} \tag{7.5}$$

for sufficiently large n and m.

- *ARMAX Model* A natural extension to the ARX parametrization is the ARMAX model, which expresses the equation error term as a moving average of white noise:

$$\begin{aligned} & y(k) + a_1 y(k-1) + \cdots + a_n y(k-n) \\ = \; & b_1 u(k-1) + \cdots + b_m u(k-m) \\ & + \varepsilon(k) + c_1 \varepsilon(k-1) + \cdots + c_\ell \varepsilon(k-\ell) \end{aligned} \tag{7.6}$$

  For the ARMAX model, the parametrization of the noise transfer function changes to

$$H(q,\theta) = \frac{C(q)}{A(q)} \triangleq \frac{1 + c_1 q^{-1} + \cdots + c_\ell q^{-\ell}}{1 + a_1 q^{-1} + \cdots + a_n q^{-n}} \tag{7.7}$$

  Because of the moving average term, an ARMAX model can potentially represent a system with much fewer parameters when compared to an ARX model. In fact, a state-space system of order n always have an input output representation given by an n_{th} order ARMAX model. However, parameter estimation is more complicated and over-parametrization can cause loss of identifiability (i.e., the parameter values can become nonunique).

- *Output Error Model* Both ARX and ARMAX model puts common poles on G and H. In some cases, it may be more natural to model

them separately. One such parametrization is the Output Error (OE) model given below:

$$\begin{aligned} \tilde{y}(k) + a_1\tilde{y}(k-1) + \cdots + a_n\tilde{y}(k-n) &= b_1u(k-1) + \cdots + b_mu(k-m) \\ y(k) &= \tilde{y}(k) + \varepsilon(k) \end{aligned} \tag{7.8}$$

In the above $\tilde{y}(k)$ represents the disturbance-free output. Customarily, $\varepsilon(k)$ is assumed to be white noise. This means the OE structure gives

$$G(q,\theta) = \frac{A(q)}{B(q)} \quad \text{and} \quad H(q) = 1 \tag{7.9}$$

A slightly more general case is when $H(q)$ is not 1, but completely known (i.e., disturbance is a colored noise with known spectrum). In this case, we can write

$$\underbrace{H^{-1}(q)y(k)}_{y_f(k)} = G(q,\theta)\underbrace{H^{-1}u(k)}_{u_f(k)} + \varepsilon(k) \tag{7.10}$$

Note that the above is in the form of (7.8). Simple prefiltering of input and output decorrelates the noise and gives the standard OE structure. Parameter estimtion is complicated by the fact that $\tilde{y}$'s are not known, and depend on the parameters.

- *FIR and Orthogonal Expansion Model* A special kind of output error structure is obtained when $G(q,\theta)$ is parametrized linearly. For instance, when $G(q)$ is stable, it can be expanded as a power series of q^{-1}. One obtains

$$G(q) = \sum_{i=1}^{\infty} b_i q^{-i} \tag{7.11}$$

Truncating the power series after n-terms, one obtains the model

$$y(k) = \left(b_1q^{-1} + b_2q^{-2} + \cdots + b_nq^{-n}\right)u(k) + H(q)\varepsilon(k) \tag{7.12}$$

This is the Finite Impulse Response model that we used in the basic

part of this book.

A general form of an orthogonal expansion model is

$$G(q) = \sum_{i=1}^{\infty} b_i B_i(q) \tag{7.13}$$

One of the popular choices for $\{B_i(q)\}$ is the so called Laguere functions defined as

$$B_i(q) = \frac{\sqrt{1-\alpha^2}}{q-\alpha}\left(\frac{1-\alpha q}{q-\alpha}\right)^{i-1} \tag{7.14}$$

An advantage of using this function is that the knowledge of process's dominant time constant can be incorporated into the choice of α to speed up the convergence (since it helps curtail the number of parameters).

- *Box-Jenkins Model* A natural generalization of the output error model is to let the disturbance transfer function be a rational function of unknown parameters. This leads to the Box-Jenkins model which has the structure of

$$y(k) = \frac{B(q)}{A(q)}u(k) + \frac{C(q)}{D(q)}\varepsilon(k) \tag{7.15}$$

This model structure is quite general, but the parameter estimation is nonlinear and loss of identifiability can occur.

All of the above models can be generalized to the case where $H(q,\theta)$ contains an integrator. For instance, we can extend the ARMAX model to

$$y(k) = \frac{B(q)}{A(q)}u(k) + \frac{1}{1-q^{-1}}\frac{C(q)}{A(q)}\varepsilon(k) \tag{7.16}$$

The above is called ARIMAX model (I for integration). In terms of parameter estimation, the resulting problem is the same since we can

transform the above to

$$\Delta y(k) = \frac{B(q)}{A(q)}\Delta u(k) + \frac{C(q)}{A(q)}\varepsilon(k) \tag{7.17}$$

The same holds for all the other model structues.

Extensions to the multivariable case are mostly straightforward, but can involve some complications. One may choose to fit each output independently using one of the above structures (This is called "MISO identification"). In this case, the only modification to the above is that $B(q)$ is now a row vector containing n_u polynomials, where n_u is the number of inputs. The parameter estimation problem remains e the same except in the number of parameters. On the other hand, some applications require a model that capture disturbance correlations among different outputs. This requires MIMO identification where all the outputs are fitted to a single multivariable model on a simultaneous basis. In this case, $A(q), B(q)$, etc. are matrix polynomials of appropriate dimension. For instance, the ARX model becomes

$$y(k) = A^{-1}(q)B(q)u(k) + A^{-1}(q)\varepsilon(k) \tag{7.18}$$

where $A(q)$ and $B(q)$ are $n_y \times n_y$ and $n_y \times n_u$ matrix polynomials respectively. The parameterization of these matrices can be a subtle issue. For instance, if all matrix entries are assumed to be unknown, one can easily lose identifiability. In general, significant prior knowledge is needed to obtain a correct parameterization. In addition, parameter estimation can be numerically challenging due to the large number of parameters, especially when the model structure leads to a nonlinear estimation problem.