Part I
Problems

1
Problems – Introduction

Figure 1.1 maps process control development sequences as referred to in sections of *Applied Process Control – Essential Methods*. The problems in Chapters 2–9 are indexed with the same section references. One notes in Figure 1.1 that the main divisions are *state-based* systems versus *input–output systems*, with these two system types each again divided into *linear* and *non-linear*.

Recall that *state-based systems* are able to predict forward in time, knowing only the present state and future inputs, for example

$$\boldsymbol{x}(t + \Delta t) = \boldsymbol{A}\boldsymbol{x}(t) + \boldsymbol{B}\boldsymbol{u}(t) \tag{1.1}$$

$$\boldsymbol{x}(t + 2\Delta t) = \boldsymbol{A}[\boldsymbol{A}\boldsymbol{x}(t) + \boldsymbol{B}\boldsymbol{u}(t)] + \boldsymbol{B}\boldsymbol{u}(t + \Delta t) \tag{1.2}$$

and so on.

This idea is essential for many algorithms, including those using *dynamic programming* (Bellman, 1957). On the other hand, *input–output systems* need additional information about past inputs and outputs in order to make such a prediction, for example

$$\boldsymbol{x}(t + \Delta t) = \boldsymbol{A}_0\boldsymbol{x}(t) + \boldsymbol{A}_1\boldsymbol{x}(t - \Delta t) + \boldsymbol{B}_0\boldsymbol{u}(t) + \boldsymbol{B}_1\boldsymbol{u}(t - \Delta t) + \boldsymbol{B}_2\boldsymbol{u}(t - 2\Delta t) \tag{1.3}$$

$$\begin{aligned}\boldsymbol{x}(t + 2\Delta t) = \boldsymbol{A}_0[&\boldsymbol{A}_0\boldsymbol{x}(t) + \boldsymbol{A}_1\boldsymbol{x}(t - \Delta t) + \boldsymbol{B}_0\boldsymbol{u}(t) + \boldsymbol{B}_1\boldsymbol{u}(t - \Delta t) + \boldsymbol{B}_2\boldsymbol{u}(t - 2\Delta t)]\\ &+ \boldsymbol{A}_1\boldsymbol{x}(t) + \boldsymbol{B}_0\boldsymbol{u}(t + \Delta t) + \boldsymbol{B}_1\boldsymbol{u}(t) + \boldsymbol{B}_2\boldsymbol{u}(t - \Delta t)\end{aligned} \tag{1.4}$$

and so on.

If all of the accumulation variables (i.e. states) in a system have been identified, one might attempt to fit the state-based form Equation 1.1 to measurements. Alternatively, this form will arise naturally from theoretical modelling of the physics. Conversely, if one is not sure that one has all of the states, one would try to fit the input–output form of Equation 1.3, with a guessed number of lags, to measured data. This is also the form that arises from step-tests and use of a *dynamic matrix*. Another option, for a theoretical model where a state cannot be measured, is to insert an observer (e.g. Kalman filter) for the missing state(s) – thus allowing state-based methods to be used.

The second division mentioned for both of these system types is *linear versus non-linear*. A lot of fast, powerful and robust methods have been developed for linear systems. If one's system is anywhere near linear, one wants to get access to these methods. This is often done based on a linearisation at an average operating point, or even on a roving re-linearisation, such as in the *extended Kalman filter*. Nonetheless, there are some methods that will take care of non-linearities – such as models that run forward in predictive control. But one should beware of non-linear systems that require optimisation – *global optimisation* is not a well-developed art.

Applied Process Control: Efficient Problem Solving, First Edition. Michael Mulholland.

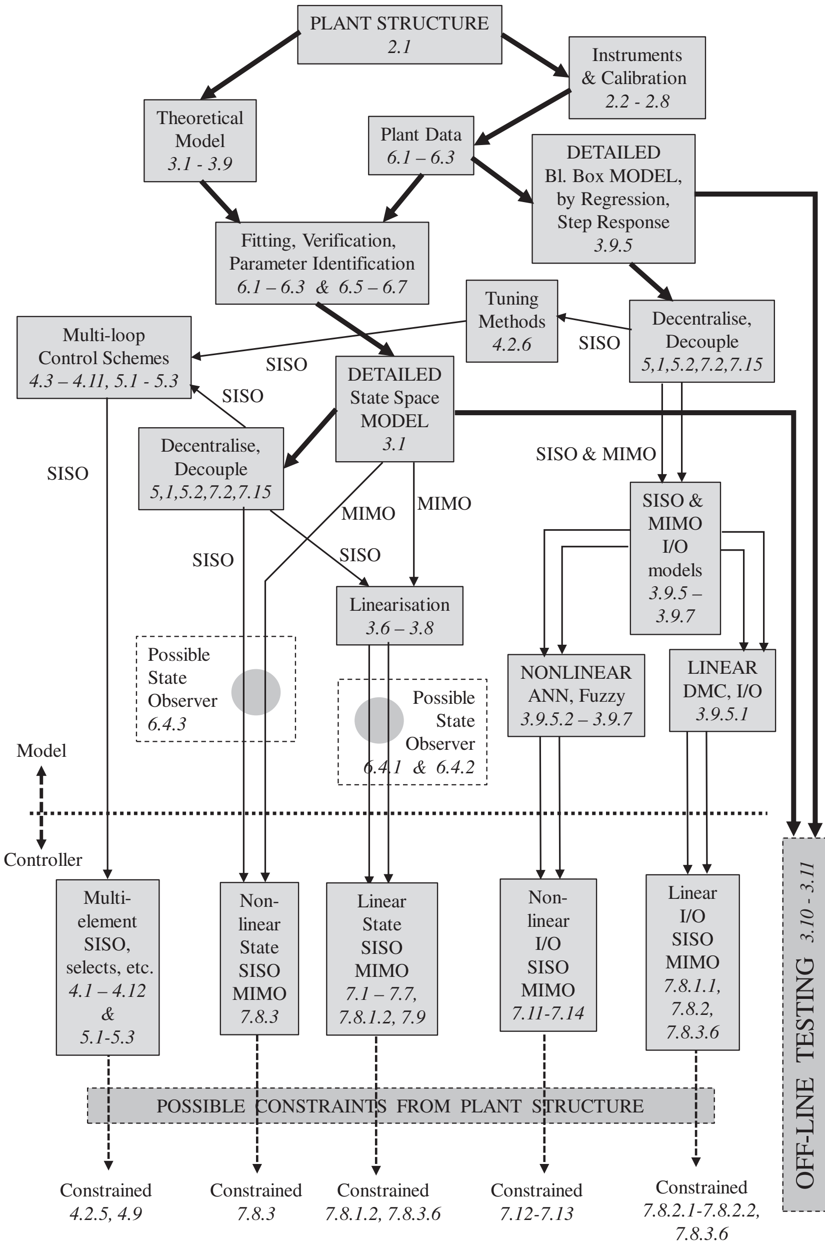

Figure 1.1 Aspects of process control related to book sections.

Figure 1.1 does not distinguish between systems presented in a *discrete* form versus *continuous*. Applications will generally be in discrete form, but if for some reason a continuous method is used, the equations can be discretised with a sufficiently small time step.

Some further points to note regarding the problems which are presented here:

a) The text consistently uses bold characters to signify matrices [$\boldsymbol{A}$], vectors [$\boldsymbol{x}$] and matrix transfer functions [$\boldsymbol{G}(s)$, $\boldsymbol{G}(z)$]. Non-bold characters are used for scalars.

b) Use of linear algebra allows multi-input, multi-output problems to be interpreted as simply as single-input, single-output problems, but beware of the multiplication order when working with MIMO symbols.

c) The control engineer needs to maintain a strong awareness of the *units* of all of the variables in an application, for example a gain as [°C $(\%\text{open})^{-1}$]. It is best to detail these at relevant lines in program code or in a calculation. One should be clear about pressures, for example [barg], [bara], or for a ΔP, [bar]. Old analogue systems used to work in fractions of range, but modern digital systems do indeed work in engineering units.