PART ONE

Basic Principles

CHAPTER ONE

The Operating Regime Approach to Nonlinear Modelling and Control

TOR ARNE JOHANSEN and RODERICK MURRAY-SMITH

Multiple model approaches have appeared more or less independently in several branches of science and engineering, disguised in different terminology. The purpose of this chapter is to introduce the basic ideas of multiple model approaches, focusing on their similarities rather than their differences. Existing paradigms for the application of multiple model and operating regime approaches to nonlinear modelling, identification and control are explored. A survey and overview of the state of the art in terms of procedures, algorithms and tools for visualisation and interpretation are also provided.

1.1 INTRODUCTION

Technological development is steadily increasing the complexity of process plants, vehicles and other engineered systems, and economical and environmental constraints are raising the awareness of the need for practical approaches which can be used to aid engineers to better understand and perform such complex modelling and control tasks.

These highly complex systems are characterised by a large number of components which are strongly coupled and have a wide operating range, among other factors. However, despite the increasing mathematical sophistication of research in modelling and control over the last decades, and the greater use of powerful computing facilities, many of the advanced methods have not been regularly applied to real problems under normal working conditions. This is often because of the theoretical sophistication required to understand the methods. Hence, if we want to deal with complex high-dimensional, coupled, nonlinear and non-stationary systems, tomorrow's automatic control systems must be more autonomous, robust, intelligent and user-friendly. However, dealing with complexity is obviously an inherently difficult problem, by the very definition of the word. The principle of incompatibility (Zadeh 1973) tries to make the implications of complexity more precise: *As the complexity of a system increases, our ability to make precise and yet significant statements about its behaviour diminishes until a threshold is reached beyond which precision and significance (or relevance) become almost mutually exclusive characteristics.* A consequence of this principle is obviously that models and analysis of complex systems will be less precise than for simple systems. A further consequence is that one should perhaps look for other

model representations and tools that can make use of less precise system knowledge than the traditional approaches which worked well for low and medium complexity modelling and control problems in the past. This is indeed the trend in the area of intelligent control where fuzzy logic, qualitative modelling, neural networks, expert systems and probabilistic reasoning are being explored, e.g. (Åström and McAvoy 1992, Antsaklis *et al*. 1991, Åström *et al*. 1986).

In everyday life, as well as in solving engineering problems, the standard approach to complex problem solving is the divide-and-conquer strategy: *A complex problem is somehow partitioned into a number of simpler subproblems that can be solved independently, and whose individual solutions yield the solution of the original complex problem.* The key to successful problem solving with this approach is to find suitable axes along which the problem can be partitioned. In this work we will focus on one approach to the decomposition of modelling and control problems that has recently attracted significant attention, namely operating regime decomposition.

The core of the operating regime approach is to make use of a partitioning of the operating range of the system in order to solve modelling and control problems. The operating regime approach thus leads to multiple model or multiple controller approaches, where different local models or controllers are applied under different operating conditions, see Figure 1.1. The supervisor (or scheduler) will coordinate the local models or controllers. This coordination may include selection of a single one, or combining the actions or parameters of a number of local models or controllers.

The operating regimes can often be characterised by different sets of phenomena or behaviours of the system. The rationale behind this approach is basically that the development of local models (or controllers) is simpler because the interactions between the relevant phenomena in each operating regime are simpler locally than globally. For instance, if the system phenomena or behaviour change smoothly with the operating point, then a linear model (or controller) will always be sufficient locally by making each operating regime sufficiently small, even though the system may contain complex nonlinearities when viewed globally. Other motivations for operating regime approaches include:

- The model/controller structure is easy to understand and interpret, both qualitatively and quantitatively, and the approach has its roots in traditional engineering methods.
- Various types of knowledge can be incorporated and integrated within the framework, including qualitative knowledge, empiricism, measured data and available models. Operating regime based modelling is closely related to grey-box modelling.
- Reduced computational complexity compared to other nonlinear methods.

Chapter overview

In the remainder, we will discuss this approach in some detail. In section 1.2 the fundamentals are introduced and numerous simple examples are given. Next, in section 1.3, a broader perspective is taken, and the operating regime approach is seen in relation to other approaches. Details on the available algorithms, procedures and tools for operating regime based modelling and control can be found in sections 1.4 and 1.5, respectively. Further details on analysis, validation and interpretation can be found in section 1.6. Finally, some concluding remarks are given in section 1.7.

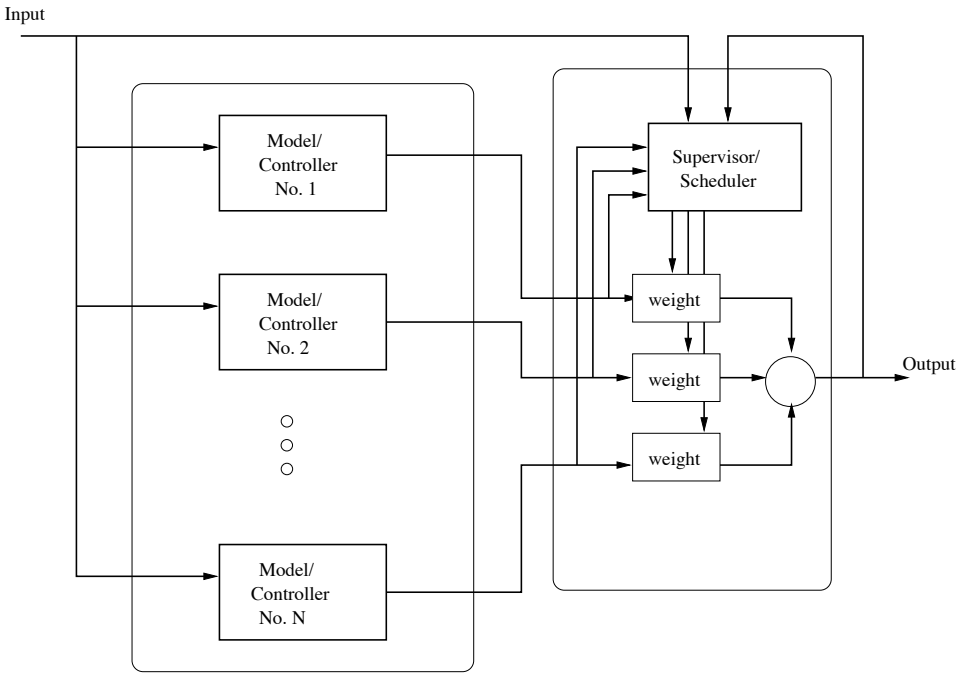


Figure 1.1 The multiple model/controller approach. This is a general visualisation of the methods described in this book, where multiple sub-models or -controllers are organised in some way by a scheduling or gating block. Each method varies in the style of scheduling done, the variables used for the scheduling, whether parameters or states are scheduled and in the structures of the individual local models or controllers.

1.2 OPERATING REGIME APPROACHES

1.2.1 The basics of the framework

Any model or controller will have a limited range of operating conditions in which it is sufficiently accurate or performs sufficiently well in order to serve its purpose. This range may be restricted by several factors, such as validity of linearisation, modelling assumptions, stability properties, or experimental conditions. A model or controller that is useful in a region less than the full range of operating conditions is called a local model or controller, as opposed to a global model or controller which is useful over the full range of operating conditions. Of course, the ultimate goal is a global model or controller. However, we will argue that in some cases it may be beneficial to achieve this goal by developing a number of local models or controllers.

The basis of the framework is a decomposition of the system's full range of operation into a number of possibly overlapping operating regimes, as illustrated in Figure 1.2. In each operating regime, a simple local model or controller is applied. These local models or controllers are then combined in some way to yield a global model or controller. Hence, model or controller development within this framework typically consists of the following tasks:

- Decompose the system's full range of operation into operating regimes. This task includes a definition of the full operating range, as well as the identification of variables that can be used to characterise the operating regimes.
- Select simple local model or controller structures within each operating regime. These structures will often be determined by the relevant system knowledge that is available under different operating conditions, as well as the intended purpose of the model or controller.
- The local model or controller structures are usually parameterised by certain variables that must be determined.
- A method for combining the local models or controllers into a global one must be applied. Numerous approaches exist, and can be characterised according to deterministic vs. stochastic assumptions, soft or hard partitions etc.

For practical problems it will not always be easy to find a natural sequence in which these tasks should be approached. Several iterations of the same tasks are usually needed before a satisfactory model or controller is found.

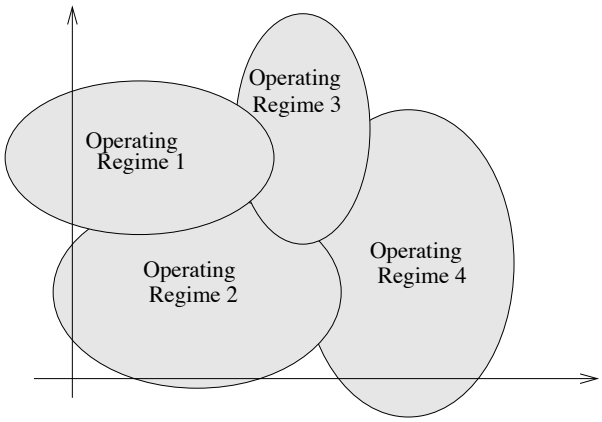


Figure 1.2 The operating range of a complex system is decomposed into a number of operating regimes. These can then be used to represent the system using associated simple subsystems. Much of this book is concerned with how to determine these regions, how to bring subsystems together, and how to interpret the resulting systems.

1.2.2 Some introductory examples

Before we proceed, this section will let us give some simple examples of how operating regime based approaches can be used to approach some nonlinear modelling and control design problems. The more fundamental aspects of partitioning approaches will then be discussed in section 1.2.3.

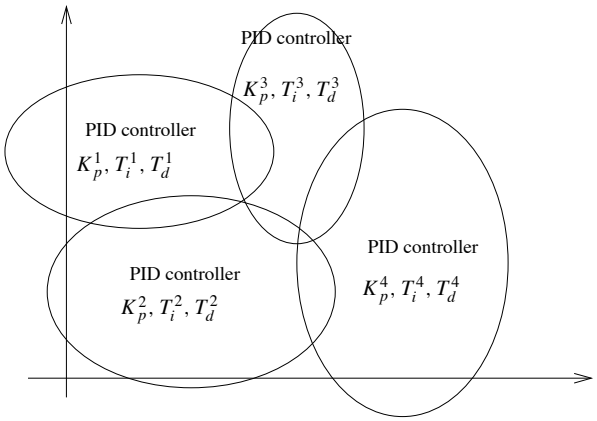


Figure 1.3 A nonlinear PID-like controller is designed by patching together local linear PID controllers.

A nonlinear PID-like controller

Suppose we want to design a nonlinear PID-like controller of the form

$$u(t) \quad = \quad K_p(z(t))e(t) + \frac{1}{T_i(z(t))}\int_0^t e(t)dt + T_d(z(t))\frac{d}{dt}e(t),$$

where the gain K_p, integral time T_i and derivative time T_d are functions of the system's operating point $z(t)$. In standard PID controllers these functions are all constant, and there exist numerous design procedures which guarantee that various stability, performance and robustness specifications are met, e.g. (Åström and Hägglund 1988). Of course, with the nonlinear PID-like controller above, the design problem is much more difficult. However, the operating regime based approach offers an engineering-friendly solution to this design problem. One can for instance design a number of standard linear PID controllers to meet the desired stability, performance and robustness criteria locally when the system is operating in neighbourhoods of some selected operating points $z_1, z_2, ..., z_{n_\mathcal{M}}$:

$$u(t) \quad = \quad K_p^i e(t) + \frac{1}{T_i^i}\int_0^t e(t)dt + T_d^i\frac{d}{dt}e(t), \qquad \text{when } z(t) \text{ is close to } z_i,$$

for $i = 1, 2, ..., n_\mathcal{M}$, see Figure 1.3. Designing such local PID-controllers is often simpler than approaching the nonlinear PID-control design problem directly, even if a nonlinear dynamic model of the system exists. In addition, we need an algorithm for combining or switching between the local PID-controllers. For instance, a weighting of the local PID-parameters as a function of the operating point $\rho_i(z(t))$:

$$u(t) \quad = \quad \sum_{i=1}^{n_\mathcal{M}} \left(K_p^i e(t) + \frac{1}{T_i^i}\int_0^t e(t)dt + T_d^i\frac{d}{dt}e(t) \right) \rho_i(z(t)),$$

i.e.

$$K_p(z(t)) \quad = \quad \sum_{i=1}^{n_\mathcal{M}} K_p^i \rho_i(z(t)),$$

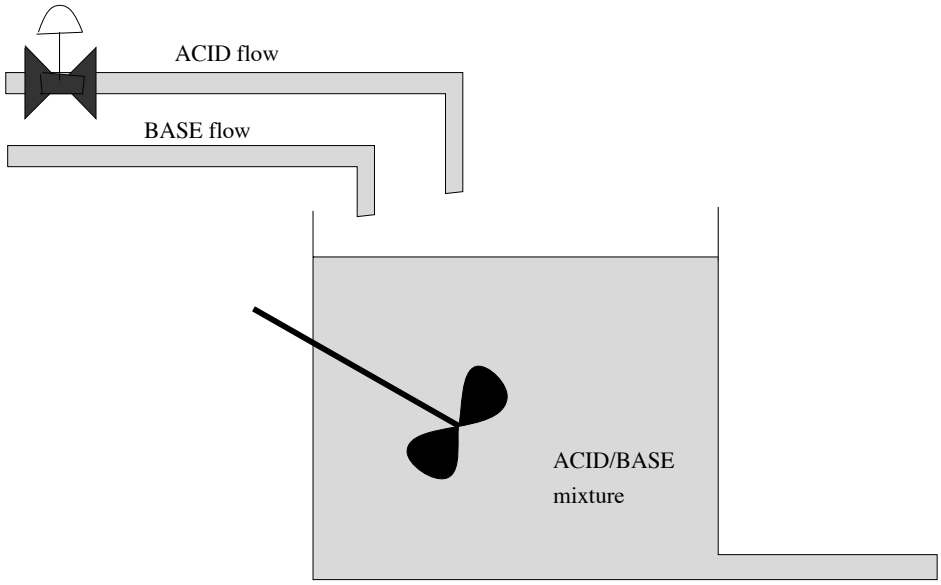


Figure 1.4 A pH neutralisation tank.

$$\begin{aligned} \frac{1}{T_i(z(t))} &= \sum_{i=1}^{n_{\mathcal{M}}} \frac{1}{T_i^i}\rho_i(z(t)), \\ T_d(z(t)) &= \sum_{i=1}^{n_{\mathcal{M}}} T_d^i \rho_i(z(t)). \end{aligned}$$

It should be noted that even though all the locally designed PID controllers satisfy the design criteria locally, extensive analysis or simulation must usually be applied to validate and verify the global properties of the control system.

A nonlinear input–output model of a neutralisation tank

Consider, as a second example, a pH neutralisation tank where acid and base flows into a stirred tank through pipes with valves, see Figure 1.4. The mixture exits through another pipe. Suppose we select a nominal tank level, pH and flow-rates corresponding to an equilibrium point for the tank. Exciting the system about this equilibrium point and collecting experimental data, we can identify a linear transfer function model

$$\mathrm{pH}(s) = \frac{Ke^{-\tau s}}{1+Ts} q(s),$$

where $q(t)$ is the flow-rate through the valve, $\mathrm{pH}(t)$ is the pH value in the tank, and s is the complex variable in the Laplace transform. Suppose we repeat this experiment for a number of different nominal tank levels, pH and flow-rates corresponding to different equilibrium points. Then we would get different identified values of K, τ and T, because the gain would depend on the pH value, and the time-delay would depend on the flow-rate, and the time-constant would depend on both the flow-rate and the tank level. Hence, we have different

linear transfer function models which are reasonable descriptions of the system only within some small operating regimes. Again, these local models can be combined using some weighting function into a globally accurate model.

Modelling the longitudinal dynamics of a car

The speed (v) of a car can be described by a dynamic model that takes into account variables such as gear position (g) and throttle angle (α):

$$v(t) \quad = \quad f(v(t-1), \alpha(t-1), g(t-1)).$$

The dynamics of a linearised model

$$\Delta v(t) \quad = \quad a_1 \Delta v(t-1) + b_1 \Delta \alpha(t-1) \tag{1.1}$$

will obviously depend on the gear position, but also on the throttle angle (due to e.g. the nonlinear engine characteristics) and the speed (due to e.g. rolling and drag forces). A model could be based on the operating regime decomposition in Figure 1.5 and local linear models of the form (1.1). A complete example can be found in (Hunt *et al.* 1996*a*).

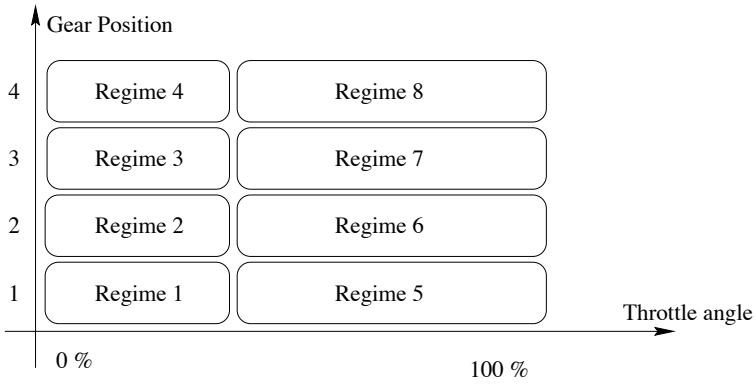


Figure 1.5 Operating regimes for a vehicle.

A nonlinear semi-mechanistic model

Consider a chemical reactor where a set of chemical reactions take place. A mechanistic model typically consists of mass and energy balances. The structure of the balance equations can typically be developed quite easily. The main problem of modelling such systems is to specify how the reaction-rates, mass-transfer coefficients, enthalpies and other model variables depend on the system's state. This typically requires a good understanding of the reaction kinetics, thermodynamics, fluid dynamics, mass- and heat-transfer phenomena present in the reactor. If such knowledge is unavailable or only qualitative knowledge about these phenomena is present, the use of operating regime based approaches can be used to formulate a semi-mechanistic model of the reactor. To illustrate this, let us consider a specific reactor, namely a semi-batch fermentation reactor where the fermentation of glucose to gluconic acid takes place, Figure 1.6.

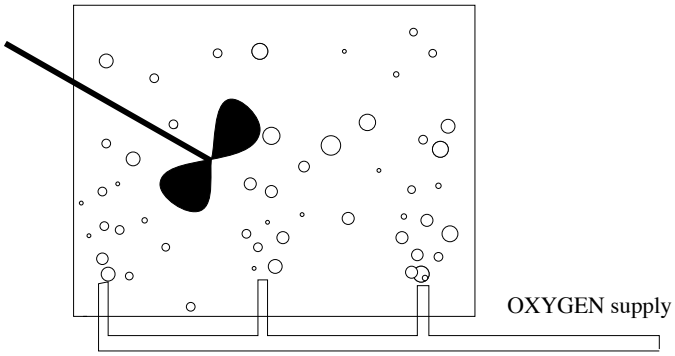


Figure 1.6 A fermentation reactor.

The main overall reaction mechanism can be described by

$$\begin{aligned} \text{Cells} + \text{Glucose} + O_2 &\rightarrow \text{More Cells} \\ \text{Glucose} + O_2 &\xrightarrow{\text{Cells}} \text{Gluconolactone} \\ \text{Gluconolactone} + H_2O &\rightarrow \text{Gluconic Acid} \end{aligned}$$

The first reaction is the reproduction of cells, consuming the substrate glucose and oxygen. The second reaction is the production of gluconolactone, again consuming glucose and oxygen. This reaction is enzyme-catalyzed by the cells, while the final product, gluconic acid, is formed by the last reaction.

Let x be the state vector that contains the compositions of cells, glucose, gluconolactone, gluconic acid, and oxygen. A mass balance can be written in the form

$$\dot{x} = r(x, T, pH) + q(x), \tag{1.2}$$

where $r(x, T, pH)$ is a vector of reaction rates, $q(x)$ is a vector of flow-rates that describes the uptake of oxygen, and T is temperature. An examination of the reaction mechanism (Johansen and Foss 1993*b*) reveals that the operation of the reactor can naturally be decomposed into four operating regimes:

1 **Initial regime** when it is the number of cells that limits the chemical reactions. Only the first reaction will be significant.
2 **Growth regime** when all reactions proceed at a high rate, limited only by the oxygen supply. All reactions will be significant.
3 **Growth termination regime**, characterised by shortage of glucose which limits the reaction rates.
4 **Termination regime** when all glucose-consuming reactions have terminated, and it is only the production of gluconic acid that is significant.

These operating regimes can for instance be characterised in terms of glucose and oxygen concentrations, see Figure 1.7. Within each operating regime the dynamics of the reactor can be described by linear differential equations with high accuracy, if the dependence on temperature and pH are neglected. Otherwise, one must decompose the regimes further on the basis of these variables, or apply nonlinear local models which take these dependences into account.

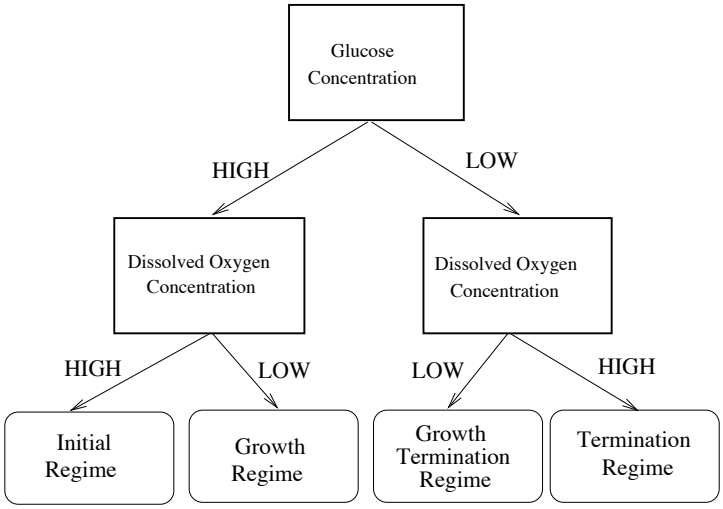


Figure 1.7 Operating regimes for batch fermentation reactor.

Similarly, the flow-rate term q in (1.2) can also be decomposed. The uptake of oxygen will depend on the oxygen concentration, stirring and other fluid dynamics phenomena that in turn may depend on the concentration of cells, since it affects the viscosity and other key properties.

Process operator heuristics

Consider a polymerisation reactor where the operator determines the feed (monomer) flow-rate (Sugeno and Yasukawa 1993). The operator's control procedures can be represented as a set of heuristic rules such as

```
IF (monomer concentration is high)
  AND (monomer concentration is increasing)
  AND (monomer feed-rate is low)
    THEN monomer flow-rate set-point is small.
```

The premise of this rule clearly determines a set of operating conditions that can be viewed as an operating regime. In (Sugeno and Yasukawa 1993), six control rules are identified on the basis of the observed behaviour of the operator.

1.2.3 Fundamental properties

The local versus global dilemma

No matter what underlying model or controller representation one chooses, e.g. state-space or input–output, lumped or distributed, discrete-time or continuous time, the nonlinear model or controller development will typically involve the specification of one or more nonlinear functions. Examples of such functions can be the dependence of the gain on certain process variables, or relations that can be linked to physical phenomena like chemical reactions. The core of the operating regime based approach is to decompose the domain

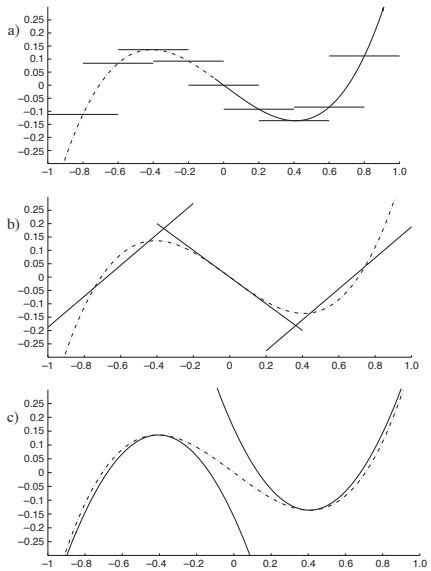


Figure 1.8 Function approximation using local models. The dashed-dotted curve is the function to be approximated, while the solid lines are local approximations. Part a) applies local constant functions, part b) applies local linear functions, while part c) applies local quadratic functions.

of functions into operating regimes and the design of simpler functions that are adequate approximations to the relationships we want to model within their respective operating regimes, see Figure 1.8. Obviously, there is a trade-off between the number and size of the operating regimes on the one hand, and the complexities of the local models on the other. For instance, at one extreme one can have only one large operating regime that covers the full range of operation. The corresponding local model must typically be complex, since it is actually the global model. In general, a decomposition into a few 'large' operating regimes will require more complex local models than a decomposition into numerous 'small' operating regimes, see Figure 1.8b) and c). On the other extreme, one can have a very fine partitioning into operating regimes based on a large number of characterising variables. In this case, the functions we are representing can be approximated by constant values locally, see Figure 1.8a).

Provided the functional relationship we want to approximate is smooth, it should be intuitively clear that approximations based on operating regimes and local models can be made arbitrarily accurate either by making the decomposition into operating regimes sufficiently fine, or by making the local models sufficiently complex. This is studied in detail by (Johansen and Foss 1993*a*), see also (Johansen 1994*c*) for some slight improvement. Explicit bounds on the approximation error as a function of the regularity (smoothness) of the underlying function, the granularity of the decomposition, and the complexity of the local models are given. Moreover, it is shown that any continuous function can be approximated to arbitrary uniform accuracy this way using polynomial local models of arbitrarily low order (like linear or constant local approximation) on a compact domain. Similar bounds and approximation results are given by (Omohundro 1987, Kosko 1994, Zeng and Singh 1994). As mentioned above, these results are quite intuitive and can almost be seen without any mathematical proof.

The curse of dimensionality

An inherent problem with all function approximation approaches that are based on partitioning of the function's domain is the curse of dimensionality: *With an increasing number of variables on which the function depends, the number of partitions required in a uniform partitioning will increase exponentially.* Consequently, a uniform partitioning is undesirable and unrealistic for anything else than low complexity problems (Friedman 1991). Fortunately, a uniform partitioning is usually not necessary. The reasons for this are diverse:

- First, the required accuracy of the model or controller may be significantly higher in some operating regions than in others. For instance, in some regions it may be necessary to have an accurate model or controller in order to optimise the control performance, while in other regions there are other criteria which are important. Fulfilment of these criteria need not always require an accurate model or controller.
- Second, in a dynamical system a large fraction of the state-space will typically be infeasible, in the sense that the system can never be in these states because they are not compatible with the physics of the system.
- Thirdly, if the local models or controllers are sufficiently complex, they will typically describe the system adequately along certain axes, while inadequately along other axes. It can be seen that it is sufficient to further decompose into operating regimes only along those axes which are inadequately described by the local model or controller (Johansen and Foss 1993*a*). This is a major advantage over simpler partitioning approaches that apply very simple local models (such as radial basis function and wavelet series expansions).
- Finally, the complexity of the system is typically not uniform. Hence, sometimes a simple local model or controller will be sufficient in a large operating regime, while in other cases a more complex local model or controller may be necessary in a smaller operating regime.

Development of a simple non-uniform decomposition of the operating range is often difficult. Prior knowledge about the system, or careful examination of large amounts of empirical data is the key to achieving this goal. Hence, the curse of dimensionality can often be 'warded off', at least to some extent.

Reducing the dimension of the scheduling variable

Consider the development of a model of the form

$$\dot{x} = f(x, u).$$

It is easy to see that such a model can always be written

$$\dot{x} = a(x, u) + A(x, u)x + B(x, u)u, \tag{1.3}$$

which is a form that emphasises the close relationship to linear models of the form

$$\dot{x} = a_i + A_i x + B_i u \tag{1.4}$$

and quasi-linear models of the form

$$\dot{x} = a(z) + A(z)x + B(z)u. \tag{1.5}$$

By comparing (1.3) and (1.5) we observe that the variable z clearly should depend on x and u. Suppose we decompose the operating range into a number of operating regimes where

the system is described by linear models of the form (1.4). From (1.3) it is evident that the linearised model's parameters depend on the operating point. However, by a possible change of variables it is clear that under some conditions, one can find a characterising vector z satisfying

$$a(x,u) = a(z), \quad A(x,u) = A(z), \quad B(x,u) = B(z),$$

which contains fewer elements than the total number of elements in x and u (Johansen and Foss 1993*a*). In particular, if only an approximate model is sought, then this reduction in dimension can be significant, and the curse of dimensionality can be reduced directly. The use of prior knowledge about the system to reduce the effects of dimensionality is a recurrent theme throughout this work. How can we define modular or hierarchical structures which produce the right level of local complexity needed to model or control the real system?

Example of dimension reduction

As an example of the above considerations, consider the fermentation reactor described in section 1.2.2. A mass balance is

$$\begin{aligned}
\dot{x}_1 &= \mu(x_4, x_5)x_1, \\
\dot{x}_2 &= v(x_4)x_1 - k_1 x_2, \\
\dot{x}_3 &= k_2 x_2, \\
\dot{x}_4 &= -k_3\mu(x_4, x_5)x_1 - k_4 v(x_4)x_1, \\
\dot{x}_5 &= k_5(x_5^* - x_5) - k_6 v(x_4)x_1 - k_7\mu(x_4, x_5)x_1,
\end{aligned}$$

where x_1 is the cell concentration, x_2 is gluconolactone concentration, x_3 is gluconic acid concentration, x_4 is glucose concentration and x_5 is dissolved oxygen concentration. The functions (parameters) μ, v, and $k_1, ..., k_7$ depend on temperature and pH in addition to the states as written above. This fact follows directly from an examination of the reaction mechanism and basic principles of chemical reaction kinetics (Bailey and Ollis 1986). We clearly see that with local linear state-space models, then the operating regimes can be characterised by $z = (x_4, x_5)$ in addition to pH and temperature. Hence, the number of variables needed to characterise the operating regimes is reduced from seven to four – a significant improvement.

Such knowledge-based reduction of model complexity is fundamental to the approaches proposed in this book, and will appear in a number of guises, from the analysis of statistical distributions, hierarchical decomposition, graphical networks of behaviours and on to fuzzy logic rule bases.

1.2.4 Combining local models and controllers

Having partitioned the problem and developed a number of operating regimes and local models or controllers within each operating regime, the natural question is how to recombine the submodels – i.e. when and how to "switch" between the local models or controllers. In the discussion above we have not paid any attention to possible problems related to overlap between the operating regimes: for some operating conditions there may exist several local models that are partially relevant. Also, we have not argued whether there should be a sudden switching between the local models, or if there should be a smooth transition. The purpose of this section is to address these questions.

Hard partitions and discrete logic

The basic idea of discrete partitions is that at each operating point, exactly one local model or controller is chosen as a deterministic function of the operating point. This is often referred to as mode switching (Hilhorst *et al*. 1991, Hilhorst 1992, Söderman *et al*. 1993), or piecewise models and controllers (Opoitsev 1970, Dorofeyuk *et al*. 1970, Kasavin 1972, Rajbman *et al*. 1981, Bellman 1961, Haber *et al*. 1982, Omohundro 1987, Billings and Voon 1987, Farmer and Sidorowich 1987).

Related to the fermentation example decomposition in Figure 1.7, this means that the partition based on glucose concentration splits the set of operating points into two parts, depending on whether the glucose concentration is below or above the threshold that defines the boundary between low and high glucose concentration, see Figure 1.9.

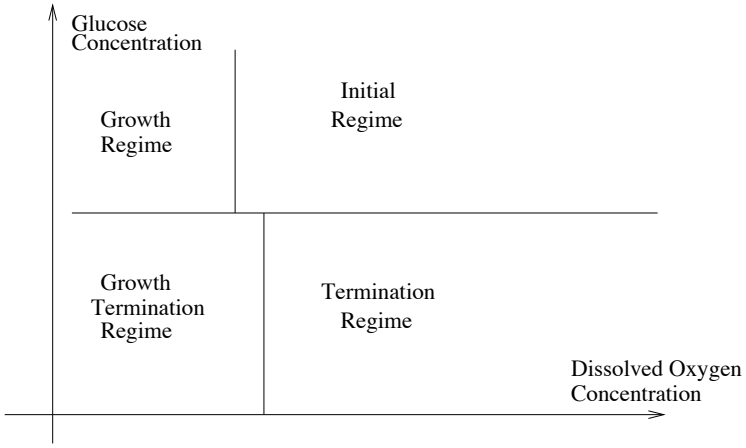


Figure 1.9 Rough decomposition of the operation of a fermentation reactor on the basis of two variables.

Convenient frameworks and representations that can be applied to describe such model or controller behaviour include decision trees (Breiman *et al*. 1984, Strömberg *et al*. 1991*b*, Sanger 1991*b*), discrete logic, expert systems, and hybrid systems (Barton and Pantelides 1994, Pettit and Wellstead 1993, Bencze and Franklin 1995, Simonyi *et al*. 1989, Konstantinov and Yoshida 1991, Konstantinov and Yoshida 1989) as well as variable structure systems (Utkin 1977, Badr *et al*. 1991). The representation in Figure 1.7 is essentially a decision tree. A discrete logic or expert system equivalent would be a number of logical statements of the form

```
IF glucose concentration IS HIGH,                          (1.6)
  AND dissolved oxygen concentration is low,
   THEN the system is operating in the Growth Regime.
```

In practice, such operating regime based models and controllers can be implemented using anything from simple programmable logic systems to sophisticated expert systems. A supervisory system based on discrete logic that lies on top of the traditional control system to handle various situations such as startup, shutdown, exceptions, safety and product changes, is the standard approach to dealing with wide operating range plants in industry.

Assume the modelling problem is a static function approximation problem, where local approximations $f_1, f_2, ..., f_{n_\mathcal{M}}$ are known for each operating regime. The operating regimes must form a complete partition of the operating range, and they must not overlap. The global approximation is then the piecewise approximation

$$f(u) = \sum_{i=1}^{n_\mathcal{M}} f_i(u)\mu_i(u),$$

where μ_i is the characteristic function for the set of points that defines the operating regime with index i.

Finite state automata

Switching between the operating regimes can also be described using a finite state automaton. A finite state automaton consists of a finite number of discrete states, each corresponding to an operating regime, or a functional state (Halme 1989, Branicky 1994, Zhang *et al*. 1994), see also Chapters 4 and 5. Transition between discrete states is described by a discrete state transition function. Suppose that within each discrete state the dynamics are described by difference equations. This leads to the hybrid model

$$q(t+1) = \mathcal{A}(q(t), x(t), u(t)), \tag{1.7}$$
$$x(t+1) = f_{q(t)}(x(t), u(t)), \tag{1.8}$$

where x is the continuous state and q is the discrete state. The deterministic function $\mathcal{A}$ represents the discrete state transition function.

Soft partitions and fuzzy logic

In some cases it may not be natural to have a sudden change between operating regimes. This may for example be the case when the operating regimes are characterised by different behaviours or mechanisms which change gradually as the operating point moves between different operating regimes. Most physical phenomena have this property. In such cases one can describe the operating regimes as overlapping sets and implement a smooth deterministic transition between them.

Consider again the decomposition described in Figure 1.9. We would like to represent that the boundary between the operating regimes characterised by low and high glucose concentration is soft. Hence, when the glucose concentration is near the overlapping boundary between low and high glucose concentration, then a blend of both local models or controllers is applied.

Frameworks that can be applied to describe such soft boundaries between operating regimes include fuzzy sets and fuzzy logic (Zadeh 1973, Takagi and Sugeno 1985) and interpolation methods (Stokbro *et al*. 1990*b*, Jones and co-workers 1991, Johansen and Foss 1993*a*). Fuzzy sets are characterised by gradual membership and are a very natural way of describing an operating regime. In fuzzy logic, the logical statement (1.6) is interpreted in terms of fuzzy sets definitions of the terms HIGH and LOW, see Figure 1.10. The theory of fuzzy logic also defines natural ways of making inference on the basis of such a rule-base. The resulting inference mechanism (Zadeh 1973, Takagi and Sugeno 1985) can be viewed as an interpolation algorithm that gives more or less weight on the local models or controllers in the different operating regimes, depending on the operating point.

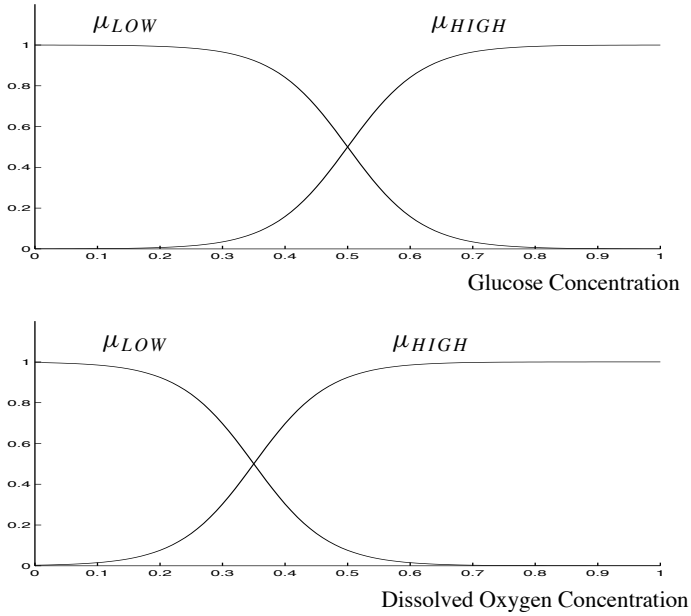


Figure 1.10 Fuzzy membership functions for representation of the terms HIGH and LOW.

Consider again the function approximation problem described above. The global approximation is now an interpolation of the local approximations:

$$f(u) = \sum_{i=1}^{n_{\mathcal{M}}} f_i(u)\rho_i(u),$$

where the smooth weighting functions ρ_1, ρ_2, ..., $\rho_{n_{\mathcal{M}}}$ provide soft transitions between the operating regimes. Typically, the weighting functions satisfy

$$\sum_{i=1}^{n_{\mathcal{M}}} \rho_i(u) = 1$$

for all u. In the case of a fuzzy logic inference (Takagi and Sugeno 1985), then

$$\rho_i(u) = \frac{\mu_i(u)}{\sum_{j=1}^{n_{\mathcal{M}}} \mu_j(u)}, \tag{1.9}$$

which clearly satisfies the equation above. The function μ_i is now the membership function for the fuzzy set that represents the operating regime with index i. The inference mechanism (1.9) is studied in (Johansen 1995).

Probabilistic approaches to partitioning

Both with the hard and soft partitioning approaches mentioned above, the characterisation of operating regimes was deterministic. An alternative is to use statistical methods to infer

which operating regime is most appropriate at each time instant.[1]

The basis for mixture models (Titterington *et al*. 1985, McLachlan and Basford 1988) is that each local model or controller has an associated probability density that indicates how correct or appropriate it is (Jacobs *et al*. 1991, Jordan and Jacobs 1994, Petridis and Kehagias 1996). The basis functions can be regarded as the components of a mixture density model. For example, the function approximation problem has the following solution

$$f(u) \quad = \quad \sum_{i=1}^{n_{\mathcal{M}}} f_i(u) P(i|u), \tag{1.10}$$

where $P(i|u)$ is the posterior probability for the local model or controller with index i being the correct one, given the data. Depending on the probabilistic assumptions, the posterior $P(i|u)$ can be computed in a number of different ways. For example, using just priors $\rho_1, \rho_2, ..., \rho_{n_{\mathcal{M}}}$

$$P(i|u) \quad = \quad \frac{\rho_i(u)}{\sum_{j=1}^{n_{\mathcal{M}}} \rho_j(u)} \tag{1.11}$$

(cf. (1.9)), or priors modified by the data according to Bayes' law

$$P(i|u) \quad = \quad \frac{p(u|i)\rho_i(u)}{\sum_{j=1}^{n_{\mathcal{M}}} p(u|j)\rho_j(u)},$$

where $p(u|i)$ is the probability density function for the input u given that local model or controller with index i is the correct one. For dynamic modelling, an approach based on statistical pattern recognition and decision theory is described in (Skeppstedt *et al*. 1992). Moreover, there is a long tradition within the control community with multiple model estimation based on Kalman filter banks and Markov models, e.g. (Lainiotis 1976*a*, Athans *et al*. 1977, Greene and Willsky 1980, Lund *et al*. 1991, Lund *et al*. 1992, Blom and Bar-Shalom 1988) and Chapter 11. The Markov model is a probabilistic relative of the finite state automaton. Transition between the discrete states is described in terms of probabilities or probability densities. Suppose that a discrete time process is modelled by a Markov chain with discrete state $q(t)$ taking values in a finite set $\{1, ..., n_{\mathcal{M}}\}$. At each time step, the probability of transition between j and i is

$$p_{ij} = P[q(t) = i|q(t-1) = j]$$

with

$$\sum_{i=1}^{n_{\mathcal{M}}} p_{ij} = 1, \quad \text{for all } j = 1, ..., n_{\mathcal{M}}. \tag{1.12}$$

In general, the transition probabilities may be densities depending on the system state, input and some parameters. In the static function approximation case, the density may depend only on the input u and the parameters W:

$$p_{ij} \quad = \quad p_{ij}(u(t),\ W).$$

[1]The relationship between probabilistic and fuzzy methods is interesting, as in some ways, fuzzy logic is deemed *'orthogonal to probability theory as it focuses on ambiguities in describing events, rather than uncertainty of occurrence or non-occurrence'* (Pearl 1988), but it is possible to describe fuzzy logic in terms of uncertainty of which label to use. In the multiple model case a fuzzy representation of the model validity functions would amount to uncertainty about which local model was the 'correct' one for a given non-fuzzy operating point.

In the Markov Mixture of Experts model (MME) described in Chapter 5 the same input u is also processed in parallel by a number of static modules called *experts*. Each expert is a local model which outputs a value $f(u(t), \theta_i)$, $i = 1, 2, \ldots, n_{\mathcal{M}}$.

The output y of the system is chosen from among the outputs of the experts, using the discrete state of the Markov chain as an indicator variable. Thus, if the discrete state is $q(t)$, then the output is $f_{q(t)}(u(t), \theta_i)$, cf. (1.8). Only the input u and the output y are usually measured; the discrete state q of the Markov chain is hidden to the exterior observer.

The model is schematically represented in Figure 5.2 on page 149. Taking the functions p_{ij} and f to be constant functions (independent of the input), it is easy to see that the MME model contains the Hidden Markov Model (HMM) as a special case; in fact, the former can be viewed as a time variant, continuous output distribution HMM. It is also a generalisation of the mixture of experts architecture (Jordan and Jacobs 1994), to which it reduces when the columns of the transition probability matrix are all equal.

Heterogeneous local models and controllers

As discussed above, there will be a transition between local models or controllers when the system moves between operating regimes. When the local models or controllers are homogeneous, in the sense that they have the same state-space, then information can be transferred between the local models and controllers quite easily. This is highly desirable, since it gives smooth behaviour and undesirable transients are avoided. A very simple example of such information transfer is "bumpless transfer" mechanisms which are applied when switching between controllers in process control systems. This means that the local controller is reinitialised to avoid transients when re-entering an operating regime.

When the local models or controllers have different structures, only in special cases is it possible to find one-to-one information transformation mappings between the different operating regimes. One solution is to let the local models or controllers have local states and simply accept possible transients (Kuipers and Åström 1994, Gawthrop 1995), see also Chapter 9. Alternatively, one can use state-estimators or observers to reduce the transients (Johansen and Foss 1993*b*, Gawthrop 1995, Gawthrop 1996, Shorten 1996).

1.2.5 Hierarchical modelling and control

We have discussed the use of locality in operating regime based models and controllers for the limitation of complexity, but how can we decide on the suitable level of locality for any given part of the operating range? For complex systems, the operating conditions of the system must often be described by a large number of variables.[2] How can we reduce the effect of high dimensionality by only concentrating on the operating conditions and variables of interest?

A common technique for the control of complexity and high-dimensionality, found in nature, society and technical systems, is *hierarchy*, and although touched on in almost every area of science, there is still little systematic analysis of the concepts involved.[3] Hierarchical methods offer, due to their structure, the ability to more effectively hide local complexity through a multi-resolution representation of the system.

[2] By a large number of variables we understand a number that is too large to allow simple visualisation of the operating regimes, i.e. larger than two or three.

[3] See (Mesarovic *et al*. 1970) for a thorough technical discussion of hierarchical systems, and (Simon 1976) for an interesting analysis of hierarchical structure in human organisations. Modern research in mathematical organisation theory can be found in the journal *Computational and Mathematical Organization Theory*.

The idea is first to roughly decompose the operating range according to the most important characterising variable. This gives rise to a number of operating regimes. Now, each of these operating regimes can be further decomposed according to another characterising variable, and so forth, see Figure 1.11. Of course, characteristic variables can reappear at different levels in the hierarchy, and the same questions of partitioning style are relevant, as discussed in the previous section.

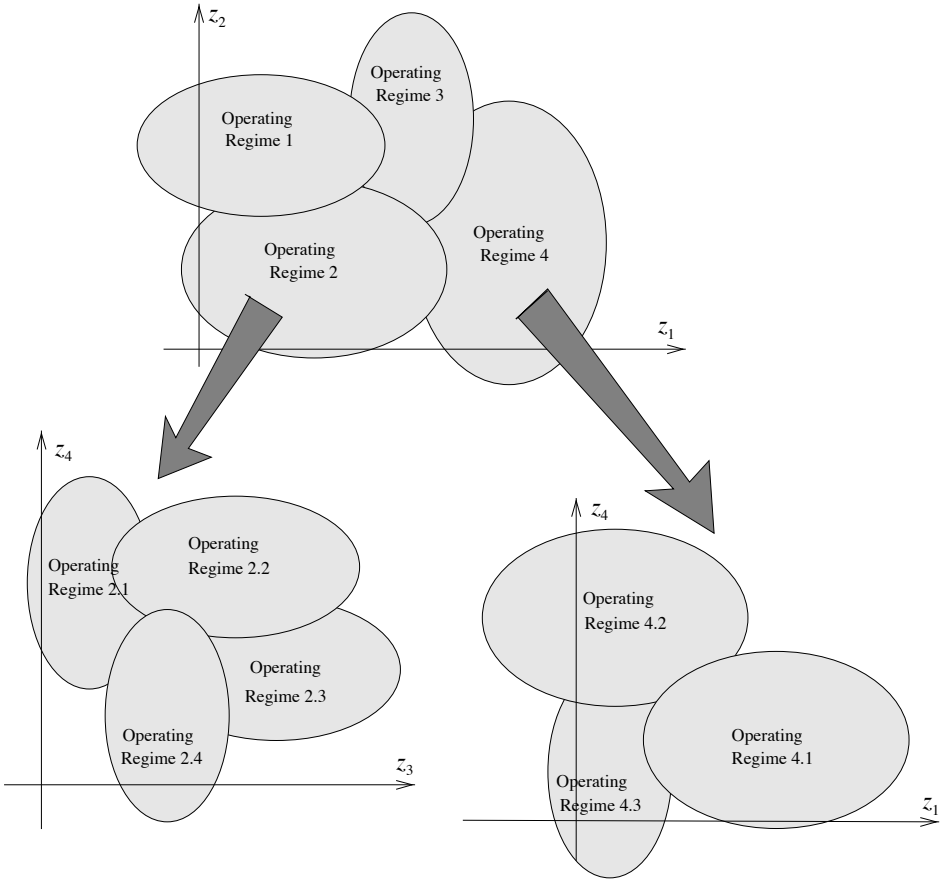


Figure 1.11 Hierarchical decomposition into operating regimes. On the top level, the four operating regimes are characterised using the variables z_1 and z_2. The second and fourth operating regimes are further decomposed using the variables z_3 and z_4, and z_1 and z_4, respectively.

Examples of hierarchical decomposition

As an example of hierarchical operating regime based modelling, consider the fermentation reactor discussed above, see (Foss *et al*. 1995) for more details. We have concluded that the operating range should be decomposed into four regimes on the basis of two variables such as dissolved oxygen concentration and glucose concentration. However, to also take into account the strongly nonlinear dependence of the reaction kinetics on temperature and

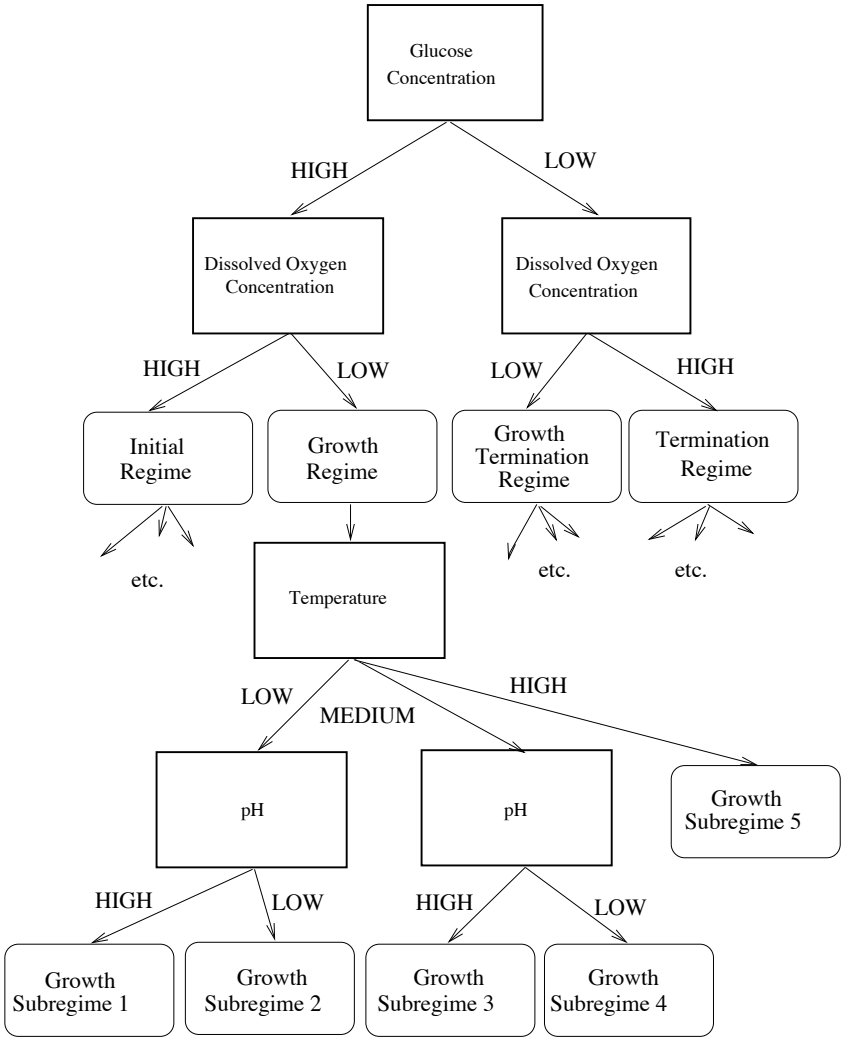


Figure 1.12 Hierarchical decomposition of the batch fermentation reactor into operating regimes.

pH, one can decompose further each of the four operating regimes into five new operating regimes on the basis of temperature and pH. This gives a total of twenty operating regimes, where each operating regime is characterised by four variables. It is clear that this is quite a complex decomposition, but as Figure 1.12 shows, it can be easily visualised and interpreted due to its hierarchical nature.

Likewise consider a hierarchical control system, Figure 1.13. On a high level in the hierarchy, the operating condition of the system can be characterised as startup, production or shutdown. The production regime can be further decomposed into normal operation, product change, and operating regimes corresponding to various failures, maintenance, and significant disturbances. Furthermore, the normal operation regime can be broken down

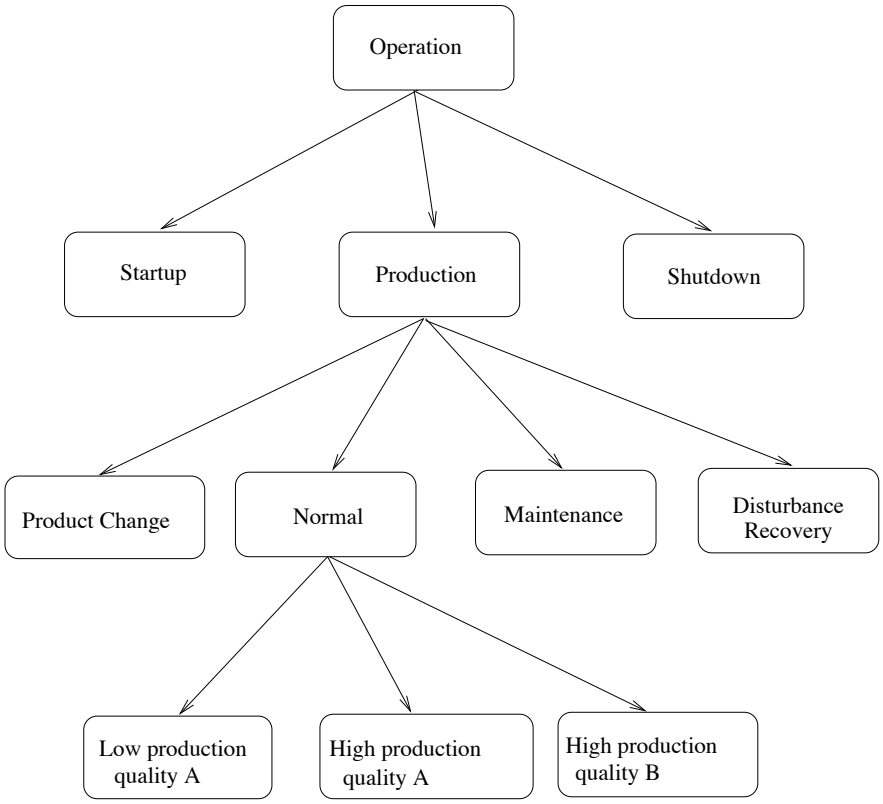


Figure 1.13 Hierarchical decomposition of the control system for a process plant.

into regimes characterised by various production levels and qualities, or raw material types. Within each operating regime there are typically different control objectives that are handled by different controllers or operating procedures. For example, during recovery after a major disturbance it is usually of primary interest to bring the plant safely back into normal operation as quickly as possible, while during normal production it is usually of primary interest to fine-tune the performance to optimise quality and production rate within the constraints given by the equipment, raw material and other factors. The number of operating regimes can be quite large, and we see that a hierarchical structure is very useful for efficiently visualising the model structure and handling the complexity. We review the literature of learning algorithms for hierarchical structures in section 1.4.5 on page 42.

1.3 OPERATING REGIME APPROACHES IN PERSPECTIVE

We will now attempt to present the operating regime approach in a broader perspective, focusing on relationships, advantages and disadvantages compared to other approaches to modelling and control of complex systems.

As discussed above, the advantage of the operating regime approach is that the modelling problem can be solved by developing simple local models that are assumed to be valid only

when the system operates within their respective operating regimes. Likewise, the control problem can be solved by developing simple local controllers for each operating regime. The local models (or controllers) can then be combined by different means in order to give a global model (or controller) for the system, as described in earlier sections.

The drawback of this approach is that the number of operating regimes may increase rapidly as the complexity of the system increases. Also, it may be difficult to determine the operating regimes, and how they should be characterised, which is the key to successful modelling and control within this framework. One advantage is that qualitative, vague and imprecise system knowledge can more easily be incorporated into the model or controller than with the other decomposition approaches mentioned above.

1.3.1 Decomposition of complex modelling and control problems

In everyday life, as well as in solving engineering problems, the standard approach to complex problem solving is the divide-and-conquer strategy: *A complex problem is somehow partitioned into a number of simpler subproblems that can be solved independently, and whose individual solutions yield the solution of the original complex problem.* The key to successful problem solving with this approach is to find suitable axes along which the problem can be partitioned.

In the following, consider the problem of developing a dynamic model or a controller of a complex system. This problem can be decomposed along at least five axes:

1 decomposition into physical components,
2 decomposition based on phenomena,
3 decomposition in terms of mathematical series expansions,
4 decomposition into goals, and
5 decomposition into operating regimes.

Decomposition into physical components

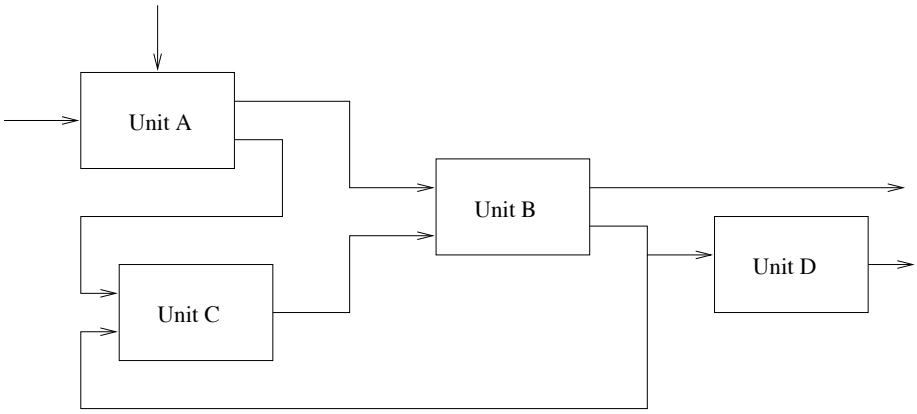
The first approach we consider is to decompose the complex system into system components or unit processes that correspond to physical components like heat exchangers, chemical reactors, buffer tanks, compressors, valves etc.

The modelling problem can now be solved by assuming that simple sub-models can be built for each component. These can be combined into a total model by establishing relations between the components in terms of mass, energy, momentum and information flow, see Figure 1.14(a). It is clearly simpler to develop sub-models for each component, and such sub-models can be recycled for similar components.

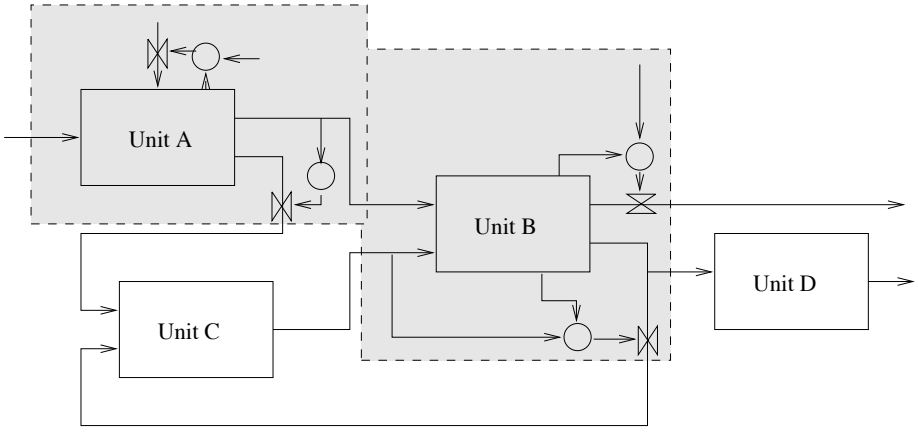
The control problem can be solved by designing decentralised controllers for each unit process, see Figure 1.14(b). This is a great simplification since a number of SISO (Single Input Single Output) controllers can be designed, rather than a single MIMO (Multiple Input Multiple Output) controller. Decentralised controllers may work well together when the interactions between the units are not too strong.

The main difficulty with this approach is to identify a set of components that are loosely coupled. It does not make much sense to decompose a system into components that are strongly coupled, because it does not untie the complexity. A major problem may be that some of the components may still be quite complex because they cannot naturally be broken down into loosely coupled sub-components.

Consider the operating range of a system which consists of a number of interconnected



(a) A complex system is decomposed into physical components that are related by flows.



(b) A complex system is decomposed into physical unit processes that are controlled by decentralised controllers.

Figure 1.14 Complex systems can be decomposed in several different ways.

components. Clearly, when the system becomes very complex, it may be difficult to characterise the operation of this complex system. It makes more sense to consider the operating ranges of each of the components individually, which is the natural way of combining operating regime based and component based decompositions.

Decomposition into phenomena

A second approach is to decompose into the set of relevant phenomena that governs the system behaviour, see Figure 1.15. For instance, in a chemical reactor, relevant phenomena may be various chemical reactions, heat transfer and thermodynamical phenomena like heat

conduction, compression and phase transition, or mass transfer phenomena like diffusion and convection. Models of each phenomenon can be developed and combined by taking into account their interactions. This approach is the basis for mathematical modelling using first principles.

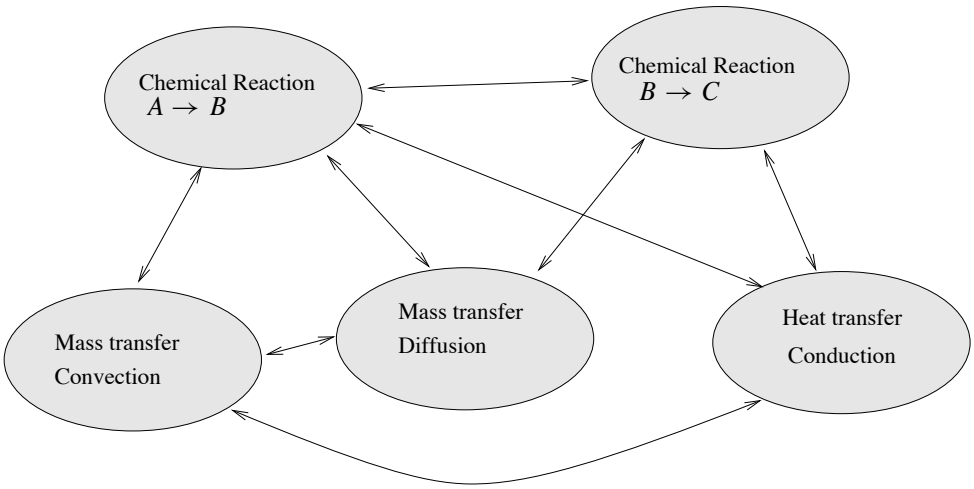


Figure 1.15 A complex system is decomposed into interacting phenomena.

The key to successful model development with this approach is to be able to identify the relevant phenomena, and describe their interactions correctly. Unfortunately, this may be difficult for a complex system that has strongly interacting phenomena. These interactions will appear in the balance equations as terms that contain a large number of variables. As the dimensionality and complexity of the system increase, the number and complexity of these terms increases rapidly and may make this approach unmanageable.

As we have illustrated above, operating regimes may at least sometimes be characterised by phenomena. The local models within operating regimes characterised by phenomena may be first principles models based on the assumed set of phenomena that is relevant within each operating regime. Likewise, local controllers with different structures may be designed to control the most important phenomena within each operating regime. Hence, we have established a very close similarity between decomposition on the basis of phenomena and operating regimes. As an example, consider the four operating regimes of the batch fermentation reactor discussed above. The first operating regime was characterised by the phenomena growth of cells and supply of oxygen, while none of the other possible phenomena were relevant. A mass-balance for this operating regime can therefore be simplified, since only one of the three chemical reactions need be taken into account. Likewise, there are different sets of chemical reactions (phenomena) that are relevant in the other operating regimes as well.

Decomposition based on mathematical series expansion

A third decomposition approach is the basis for black-box modelling and control. General classes of complex operators f can be mathematically built up from simpler basis-operators

$f_1, f_2, ..., f_{n_\mathcal{M}}$ in a series expansion

$$y = f(u) = \sum_{i=1}^{n_\mathcal{M}} f_i(u)$$

see also Figure 1.16. Clearly, such operators can be useful as input/output models or controllers for a complex system. In order to decouple the development of each basis-operator, the set of basis-operators is often chosen to be orthogonal, or at least close to orthogonal.

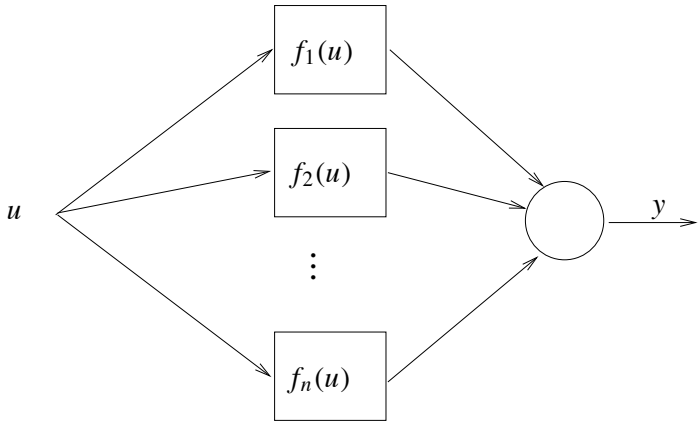


Figure 1.16 A complex system is decomposed mathematically by basis-operators that together describe the input/output behaviour.

For representing models, the basis-operators $f_1, f_2, ..., f_{n_\mathcal{M}}$ are typically chosen on the basis of empirical data and prior knowledge about certain system characteristics. For representing controllers, the basis-operators are either chosen on the basis of some system model and control design procedure, or they are chosen adaptively.

The key to successful modelling or control with this approach is to select the set of basis-operators carefully, and make sure that enough information is available to fit the parameters of the basis-operators.

The advantage of decomposition based on mathematical series expansions and black-box modelling is that it is a generic approach in the sense that the same set of basis-operators are suitable for defining the model or control structure for a wide range of systems. Only certain model or controller parameters must be tuned in order to adapt the model or controller to the particular application. However, this approach has serious disadvantages that may make it unsuitable for modelling highly complex systems: a sufficient amount of empirical data for every operating condition is required in order to "guarantee" (statistically) that the model is accurate. Various economic, safety and operational constraints, may make this assumption unrealistic. Moreover, system knowledge, over and above empirical data, may be difficult to incorporate directly into the series expansion.

Orthogonal basis-functions like Laguerre polynomials have been applied both for decomposing linear and nonlinear models (Wahlberg 1991, Wiener 1958) in series expansions. Non-orthogonal polynomial operators are used in the well known Volterra expansion of nonlinear system, e.g. (Rugh 1981). More recently, neural networks similar to series expansions and wavelet basis-functions have been suggested for nonlinear system modelling,

e.g. (Sjöberg *et al*. 1994, Benaim 1994, Cannon and Slotine 1995) and nonlinear control, e.g. (Hunt *et al*. 1992, Narendra and Parthasarathy 1990).

Quite interestingly, there exists a very close relationship between operating regime based models and certain series expansions. This relationship is particularly evident for series expansions based on basis-functions which are local, in the sense that they are zero or close to zero over most of their input domain. Hence, the local basis-functions define a subset of the input domain that can be viewed as an operating regime. Such basis-functions act in exactly the same way as weighting functions, interpolation functions and fuzzy membership functions as described in section 1.2.4:

$$f(u) \quad = \quad \sum_{i=1}^{n_{\mathcal{M}}} \theta_i \rho_i(u),$$

where $\theta_1, \theta_2, ..., \theta_{n_{\mathcal{M}}}$ are real parameters and $\rho_1, \rho_2, ..., \rho_{n_{\mathcal{M}}}$ is the set of basis-functions.[4]

Furthermore, within the framework of series expansions, it has been suggested that the constant parameter θ_i should be replaced by a simple parameterised function of u, e.g. linear (Stokbro *et al*. 1990*b*, Jones and co-workers 1991)

$$f(u) \quad = \quad \sum_{i=1}^{n_{\mathcal{M}}} f_i(u, \theta_i) \rho_i(u),$$

which is clearly functionally equivalent to an operating regime approach. Relations between fuzzy models, certain series expansions, and operating regime based models are further discussed in (Jang and Sun 1993*a*, Foss and Johansen 1993, Hunt *et al*. 1995, Hunt *et al*. 1996*b*).

It is widely accepted that series-expansions based on orthogonal basis-functions have some useful properties compared to non-orthogonal basis function sets. It is clear that if the operating regimes are non-overlapping, the equivalent series expansion will be orthogonal since the weighting functions will be orthogonal

$$\int_u \rho_i(u) \rho_j(u) du \quad = \quad 0$$

for $i \neq j$. If the operating regimes do not overlap significantly, we can informally state that the basis-functions are almost orthogonal, in the sense that the inner product above is close to zero, see Figure 1.17. Hence, operating regime based models and controllers have much the same advantages as orthogonal series expansions: the local model or controller within each operating regime can be modified without significantly affecting the model accuracy or controller performance outside that operating regime. Hence, single local models or controllers can be developed, added, analysed and modified more or less independently.

[4]Numerous basis-function sets satisfy the constraints

$$\sum_{i=1}^{n_{\mathcal{M}}} \rho_i(u) \quad = \quad 1$$

for all u, i.e. at any point in the input domain, the sum of all weighting functions should be 1. These include B-splines (Lane *et al*. 1992) and normalised basis-functions (Moody and Darken 1989):

$$\rho_i(u) \quad = \quad \frac{\Phi_i(u)}{\sum_{j=1}^{n_{\mathcal{M}}} \Phi_j(u)},$$

where $\Phi(u)$ is the general *unnormalised* basis-function. Normalisation can be important for such basis-functions, often making the model less sensitive to the parameterisation of the basis-functions, but it also has a number of side-effects which are discussed in detail in Chapter 8.

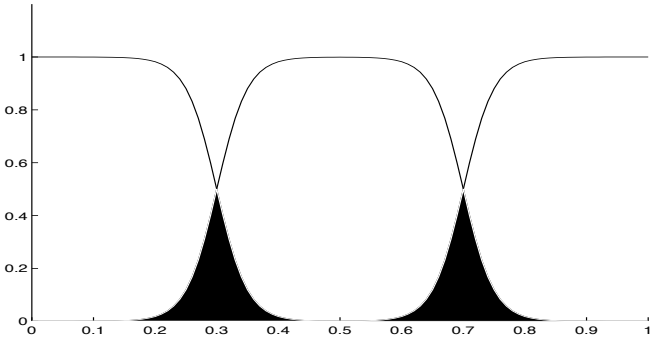


Figure 1.17 Basis-functions (or weighting functions) without too much overlap are close to orthogonal since the inner product (related to the shaded area) is small.

Decomposition based on goals

Most engineering control systems can be viewed as attempting to satisfy certain goals.[5] Complex systems are typically designed to satisfy multiple goals. Sometimes, several goals should be fulfilled simultaneously, but often the system must give priority to different goals under different conditions. Consequently, the behaviour and internal mechanisms of the system change with the changing priorities. Other approaches to coping with conflict when combining behaviours are discussed in (Yavnai 1993).

This attitude to problem decomposition is shown in the *subsumption architectures* applied to robotics problems in (Brooks 1986). These are layered architectures where each level is capable of control, independent of the higher levels, but where the higher levels are capable of increasingly complex behaviours. When a higher level behaviour is switched on it *subsumes* the lower level one, altering its internal states, while utilising its functionality. Different goal-oriented models are suggested in (Lind 1988, Lind 1991) as a basis for fault detection and diagnosis in process plants and by (Rasmussen 1986) in models of human problem solving.

Goal-oriented models and controllers are characterised by changes in goal under different conditions. These conditions may in many cases be naturally viewed as operating regimes, and it may be possible to describe the system's behaviour when attempting to satisfy different goals using local models or controllers.

For example, during normal operation, the goal of a control system is typically to control quality or production rate. On the other hand, during faulty operation, startup or shutdown, the goal of the control system is typically to bring the system quickly and safely to normal operation. In this case, the relations between the goal and the operating regimes are easily seen. Hence, we argue that the operating regime approach and goal-oriented modelling and control have very much in common and can benefit from each other.

[5] Whether or not *natural* systems should be viewed as teleological, or goal-directed, is a long-disputed topic, but in some cases such an analogy can, at a given level of abstraction, still be useful for the analysis of systems exhibiting a range of complex behaviours.

Which decomposition approach to apply?

Of course, all the above decomposition approaches are useful for solving practical problems. However, they have distinct characteristics that may clearly favour one over the other for certain systems and applications, and they have overlapping characteristics such that the choice between two of them may be more or less arbitrary in some cases.

It should be noted that the different approaches can (and must!) be combined in a number of natural ways. In particular, both component-based decomposition and operating regime-based decompositions must necessarily be combined with either series-expansion based or phenomena-based models or controllers, since the local models or controllers, or component sub-models at some point must be represented in terms of equations based on series-expansions or phenomena. In other words, goal-oriented, operating regime and component approaches are mainly useful as an approach for breaking down the problem by introducing a high level representation that brings the modelling or control problem to a higher and more user-friendly level than equations. However, as discussed above, at some point one cannot decompose further, but must deal with the sub-model or sub-controller in terms of low level representations based on equations and numerical parameters.

1.4 SOLVING MODELLING PROBLEMS WITH OPERATING REGIME METHODS

The use of operating regime approaches is aimed at improving techniques for the design of models of nonlinear dynamic systems with the aid of computationally intensive data-driven techniques. This involves integrating knowledge about the system with data from experiments. This allows the developer to produce more transparent model structures and identify their parameters, as well as being able to validate the accuracy of the final model. Modelling from data and knowledge is often viewed as an art form, mixing 'expert' insight with the information in observed data, while using *ad hoc* simplifications to make the problem solvable. The typical sources of information could be:

- Experimental data, such as responses for perturbations about a number of operating points.
- A possibly incomplete nonlinear model or controller that may be too simple or too complex (computationally or conceptually) to be directly applicable.
- Qualitative knowledge, i.e. phenomena, behaviours, or operators' and engineers' heuristics.

A model will only be able to represent certain aspects of a system, so obviously, in order to decide which aspects the model should capture, we need to know the purpose of the model. Hence, modelling without a particular purpose in mind does not make sense. Typical applications of models include system analysis and optimisation, system design, control, automatic supervision and fault detection, prediction etc. The operating regime based approach is not restricted to a particular purpose, but is a generic framework. So once the phase of *formulation of modelling goal and purpose of the model* is complete we can view the typical abstract modelling cycle to be as shown in Figure 1.18. This identifies a number of further tasks that are essential in modelling:

- Experiment design and data acquisition.
- Raw data processing and analysis.
- Analysis of *a priori* knowledge, physical laws and available models.
- Structure and parameter identification – machine modelling.

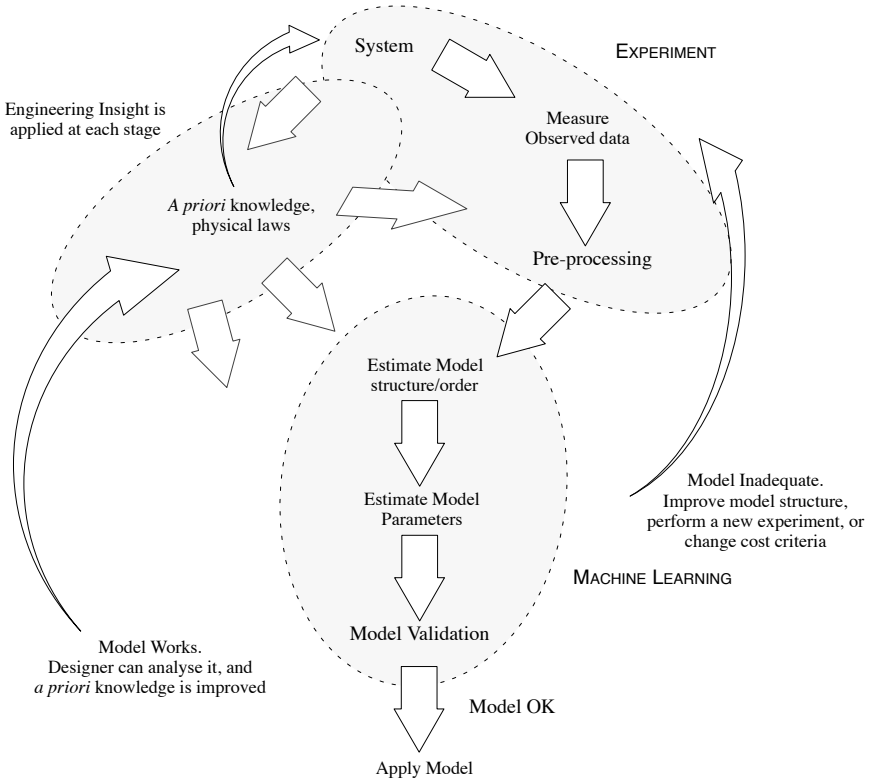


Figure 1.18 The Engineering Cycle for developing a model. Note the interaction of *a priori engineering insight, experiment design, data acquisition* and *pre-processing,* followed by *machine modelling* and *validation.*

- Model reduction and simplification.
- Model validation, analysis and interpretation.

The purpose of this section is to discuss the advantages of approaching these tasks with operating regime methods.

1.4.1 Experiment design

Experimental data is used for identifying the parameters and structure of the model, and for model validation. The general process of experimentation is shown in Figure 1.19.

It is obviously important that the data set has captured the important aspects of the system that we want to represent in the model. In particular, the data should cover the relevant areas of the operating space, but it is also important to consider the relative importance of the various areas of the operating space. The basic considerations being:

- Information about the relative significance of individual data samples is important. This could be in the form of probability distribution functions, showing the relative frequency of particular situations, noise levels, or the definition of regions which are particularly important.

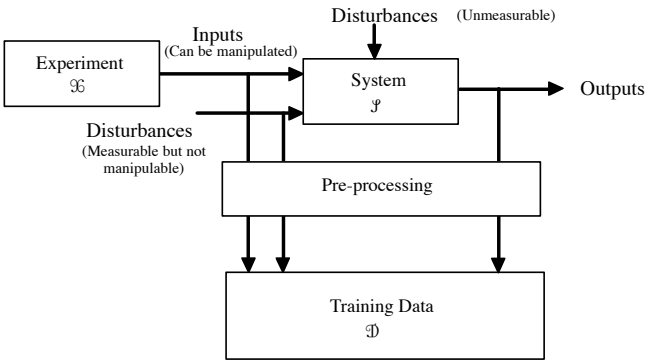


Figure 1.19 Acquiring and preparing experimental data.

- In many processes it is important to have very accurate models in relatively stable areas, rather than less accurate models throughout the operating range, as the stable areas are often where most of the operating time is spent.
- It is not only important to have identification data covering the operating range, but to have larger amounts of data where the system is most complex or its behaviour or control is most critical.
- Certain operating conditions of the system may be associated with danger – how should these be treated in the model and in the acquisition of experimental data?

From this we can see that constructing a representative set of experimental data may often be more difficult, and require more consideration than the subsequent task of identifying the model! *Experiment Design* is the science of optimal data collection (Fedorov 1972, Goodwin and Payne 1977), and has been recently applied by workers in machine learning within the *active learning* framework. See Chapter 6 for a review of recent progress in this area.

When local operating regions are used, the experiment design phase is often simplified, because properties of the data set can often be directly linked to particular operating regimes and local models. For instance, lack of data from a certain operating condition can explain why the related local model's parameters are not accurately identified. If operating regimes are further decomposed, there is also a clearer link to the requirements of future experiments.

1.4.2 Pre-structuring operating regime based models

Operating regime based models can be viewed as general structures which are well suited for use in modelling dynamic systems, and especially well suited for use with learning algorithms. Determining the structure of the model is one of the most challenging tasks in model development. For operating regime models we describe structure identification as the task of determining a decomposition into operating regimes as well as determining the structure of the local models. Powerful algorithms for identifying a model structure from experimental data are emerging, and will be discussed in section 1.4.3, but as already mentioned, the analysis and use of *a priori* knowledge is essential in the search for a good model structure.

Analysis of a priori *information in structuring the model*

Empirical modelling from observed data is supposed to reduce the need to understand the detailed physical relationships within the system under investigation, but as can be seen in Figure 1.18 on page 30, a major feature of the cycle is the important role of *a priori* knowledge at each stage of the modelling process. *A priori* information is initial knowledge about the system, or problem in question. This includes aspects such as the goals of the problem, the characteristics of the process, its parameters, the effect of the environment (expected noise, disturbances), and the robustness requirements for different situations. Learning or adaptive systems are sometimes used because of the insufficiency of the *a priori* information, however, in practical applications it is usually necessary to constrain the learning task by introducing *a priori* knowledge into the model.

Depending on the available knowledge for a given problem, the approach for practical operating regime based modelling or controller development will differ considerably. We discuss two main classes of approach: *data-driven* and *model-driven*, but there will always be an element of both in any real problem. *A priori* knowledge can be used both *implicitly* and *explicitly*. It is used *implicitly* when framing the problem, in the act of creating a representative set of data, deciding which identification algorithms and structures are best suited to the problem and which inputs and pre-processing algorithms are likely to make the identification task easiest. The *explicit* use of *a priori* knowledge involves the direct integration of sub-models or rule-bases into the model. Model structure, dynamic order and sampling rate are dependent on *a priori* knowledge. Knowledge about physical constraints can be used to restrict the parameter set. Existing models or controllers (e.g. human operators) can also be used, where valid. Operating regime based models are characterised by their operating regimes, local model structures and local model parameters.

One of the major advantages of operating regime based modelling and control is that qualitative knowledge in various forms can be utilised directly when developing the model or controller structure. This includes the number of local models, the type and complexity of local model used, constraints on their parameters, and in particular, the decomposition of the operating range into operating regimes. This leads to more interpretable models which can be more reliably identified from a limited amount of experimental data.

Local models – number and type

The most general form of information is the expected dynamic order of the system, and the form of model to be identified (e.g. simple linear ARX models, state-space models etc). If more knowledge is available, the local models could be physically oriented models, possibly with only a subset of their variables to be identified, thus allowing the engineer to easily create *grey-box* models.[6]

Certain areas of the state-space may well be already accurately modelled using existing physically motivated models. These may be directly included as local models, where the associated weighting functions can be estimated by comparing the model outputs with the observed data. Another reason for including physical models is that there will not be sufficient data to identify the model from data throughout the state-space. This is especially true in areas outside normal desired operation, where the model may have to be very robust, and well understood, but where there are few data. These situations can be covered by fixing

[6] A generalised form would allow the designer to specify a pool of feasible local models, which could be locally tested for suitability in the various operating regimes defined by the basis functions. See for example (Johansen 1994*c*).

a priori models in the given areas, and applying learning techniques only where the data is available and reliable. Examples of this approach include (Kramer *et al*. 1992, Thompson and Kramer 1994, Foss and Johansen 1993, Su *et al*. 1992).[7]

A further option, for cases where a complex nonlinear model already exists and is valid for certain operating regimes, is to pre-set fixed local models to be locally accurate linearisations of this nonlinear function at operating points throughout these regimes. This is useful in cases where good models already exist, but they are too computationally expensive (or financially expensive in the case of proprietary software!) to be used in interactive control law development, or on-line in real-time systems. The use of a consistent framework also allows the comparison of parameters identified from data with the linearisations taken from existing models. This can then be added to the developer's toolkit as a validation method.[8]

For a linearisation to have any meaning in the classical sense, however, it must be around an equilibrium point of the dynamic system. Care must therefore be taken when choosing the scheduling inputs. This is discussed in section 1.6.

The problem of choosing the local model dynamic order parameters is certainly not a trivial problem. The general problem of order selection is widely discussed in the system identification literature, and several statistical criteria and general guidelines exist, e.g. (Box and Jenkins 1970, Ljung 1987). In the operating regime based modelling framework, there may be a need for a varying dynamic order of the different local models in different areas of the state space. Sbarbaro has suggested representing continuous-time local linear models with Laguerre networks to reduce the sensitivity of the model with respect to the choice of dynamic order (see Chapter 10).

A further possibility is to include physical constraints on the relationship between multiple parameters. Examples of *a priori* knowledge of physical constraints being used to improve generalisation in models can found in (Kramer *et al*. 1992, Thompson and Kramer 1994), and in (Röscheisen *et al*. 1992) who demonstrate the use of *a priori* models in identifying an RBF model of a rolling mill. A general method for constraining the parameters of physically motivated models is to select a prior distribution for parameters based on knowledge of, or beliefs about, the physical system. A theoretically elegant approach is to define a prior probability distribution for each parameter, where the developer must choose a distribution for the parameter. This helps prevent spurious combinations of variables, given sparse and noisy data, but is also a time-consuming method and may often be difficult to apply due to the lack of prior knowledge. Supporters of this approach, such as MacKay (1991), also show how priors can be parameterised with hyperparameters which can be derived from the training data during learning. These concepts are important and it seems likely that many ideas from Bayesian statistics (Bernardo and Smith 1994) are likely to become more widely used in data-driven modelling work in the near future.

Using transformed inputs to characterise the operating regimes

It is often possible to characterise the operating regimes using variables transformed in such a way that they still 'capture the nonlinearity', while having a reduced dimension (see

[7] In (Psichogios and Ungar 1992, Aoyama and Venkatasubramanian 1993, Brockett 1993), combinations of neural network structures and mechanistic model structures are suggested. Prior knowledge has been used for structuring neural nets (Mavrovouniotis and Chang 1992), initialisation of neural network parameters and interpretation of the resulting model through linearisations (Scott and Ray 1993).

[8] Note that the linearised models produced by such an approach will often have different parameters to those created by fitting a linear model from data using a least squares approach.

also the discussion of design of gain scheduling systems in section 1.5.1). This can range from the use of a subspace of the state and input space, for example some sort of principal components transformation, such that the operating regimes are characterised using as few variables as possible, to some form of knowledge based transformation.

For example, when modelling dynamic systems, we can often produce a dramatic reduction in the operating point vector by assuming that the operating point can be approximated using a lower order dynamic state (i.e. fewer delays in an ARX representation), while the local models use the full state vector.

Using prior knowledge to place the operating regimes

The partition of the problem is achieved by weighting functions associated with each local model which define its operating range, and the style of transition made from one behaviour/model to another. Locality of representation provides advantages for learning efficiency, and transparency. It is, however, very difficult to automatically find the 'correct' level of locality for a given subspace of an arbitrary problem, significantly more difficult than optimising the parameters of the local models. The resulting weighting functions have a major effect on the performance and interpretability of the trained model or controller. It is therefore important to constrain wherever possible the freedom of the weighting functions with prior knowledge.

The probabilistic interpretation of the weighting functions used in (Jordan and Jacobs 1994) (see also Chapter 6) views the weighting function ρ_i as the conditional prior probability of the local model f_i being the correct model. We therefore see the initialisation of the weighting functions as providing an initial set of priors for the likelihoods of the individual local models being correct (it is also possible to provide meta-priors for these weighting function priors, which constrain their parameters during learning (Waterhouse *et al*. 1996, McMichael 1995). As mentioned earlier, the probabilistic interpretation allows us to incorporate techniques from the areas of Belief networks, and uncertain reasoning (Pearl 1988, Shafer and Pearl 1990) to provide more sophisticated representation.

Using fuzzy rules for the model structure

It is possible to use other knowledge-based methods to position the weighting functions, or initialise their parameters. Because of the strong links between fuzzy membership functions and basis functions,[9] the *a priori* knowledge of how best to decompose the problem could be expressed as linguistic rules with accompanying membership functions. A key point here is that each rule will typically only be conditioned on a subspace of the state space, reducing the dimensionality problems of the learning algorithm.

Vague human intuition about the relative complexity of the operating range can often be expressed in `IF .. THEN` rules of the form

$$R_i \ : \ \texttt{IF}\ z_1\ \texttt{is}\ Z_{i1}\ \texttt{AND}\ \ldots\ \texttt{AND}\ z_d\ \texttt{is}\ Z_{id}\ \texttt{THEN}\ y = \hat{f}_i(u)$$

which can either be used to provide initial parameters, or constraints on possible model structures. The use for initialisation of a model structure is fairly clear, the premise forming the basis function and the consequent forming the local model or controller, but their use as constraints is maybe less so: the membership functions which determine the distance

[9] See section 1.4.4 for a review of the relevant literature, and section 1.5.3 for a review of fuzzy control. Chapters 2 and 13 also deal with fuzzy controllers.

or similarity of a variable z_j from the fuzzy set Z_{ij} can still be optimised using learning algorithms, but the variables they are conditioned on must remain limited to the subset defined in the rule. This not only constrains the solution to one which should be better (assuming useful constraints were chosen), but to one which is also more interpretable, because of the 'conjunctive' or axis-orthogonal partition of the operating range (see Figure 1.21, and also the discussion in section 2.3.3 on page 85).

Using graphical structures to structure models

Graphical network models are being increasingly applied to represent human knowledge of causal links or of independence or irrelevance relationships between variables (e.g. (Whittaker 1990, Buntine 1994)). When used to structure Bayesian networks (Jensen 1996) they can be viewed as alternatives to the fuzzy rule-based approach to basis function representation just described.

Graphical structures can also be used to describe possible transitions between model states. This can be a very useful way of constraining operating regime models and controllers with prior knowledge. Many applications can usefully be abstracted in terms of a transition graph, where each node of the graph corresponds to a given submodel or behaviour. Figure 1.20 shows a general graph of behaviours. The actual transition functions can be purely random, state or input dependent, and can also be learned from data, but the very act of eliminating some transition possibilities will simplify the learning of the other transition functions. The transition functions can use the same sort of transformations described in the previous section. These methods are used in Chapters 4 and 5 to constrain the possible transitions between operating modes. This approach has the advantage that many engineers are already used to thinking in terms of finite state automata, as well as using them for documentation. This should make it easier to include useful prior knowledge, and to maintain it in future projects.

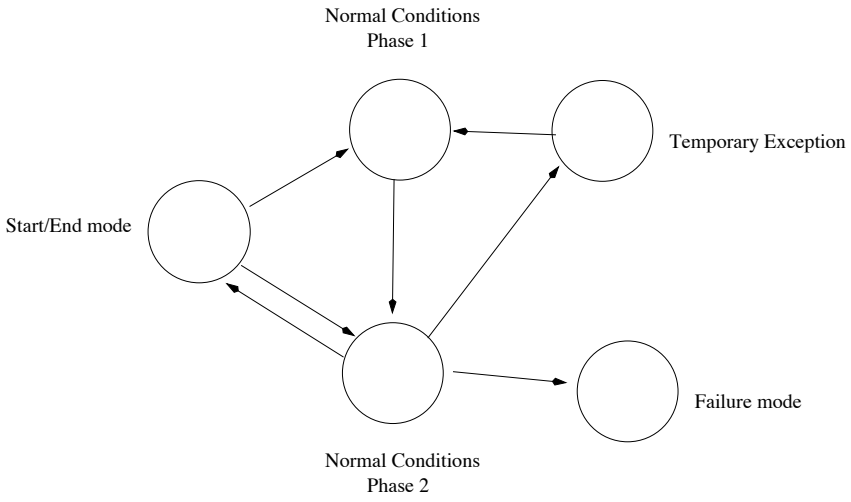


Figure 1.20 Example of a general graph method for constraining behaviour transitions. Only transitions marked by arrows are permitted.

1.4.3 Learning operating regimes from data

As many problems are not well enough understood for the model structure to be fully specified in advance, it will often be necessary to adapt the structure for a given problem, based on information in the data. The optimisation of the model structure is, however, a difficult non-convex optimisation problem, and is probably one of the most important areas of research for data-driven modelling approaches.

The goal of the structure identification procedure is to provide a problem-adaptive learning scheme which automatically relates the density of operating regimes and their size to the local complexity and importance of the system being modelled. The desirable features of a structure identification algorithm are:

- *Convergence* – as the number of data points increases the algorithm should produce models which approximate the real process more accurately.
- *Parsimony* – the model structure produced by the algorithm should be the simplest possible which can represent the process to the required accuracy.
- *Robustness* – the model structures produced should be as robust as possible with regards to noisy data or missing data.
- *Interpretability* – the model structure produced should ideally be as interpretable as possible, given the available data, local models and basis functions.

The model set is typically large, and parameterised by elements of both function spaces, discrete spaces, and Euclidean spaces. *Model structure identification* deals with search and optimisation in function spaces and discrete spaces. It is often an iterative process, where the model structure from the previous step in the iteration is used as a basis for the selection of a more promising model structure, either refined or simplified. Often, the structure identification problem is a combinatorial one, so good heuristics are important to minimise computational effort. Structure identification algorithms for operating regime based models must necessarily rely on some particular representation of the operating regimes, see Figure 1.21, and the options discussed in section 1.2.4 provide a range of options.

Robustness is an important aspect, as nonlinear models can obviously be very powerful, possibly representing the identification data very accurately by using a large number of parameters, but usually then leading to a high variance. The choice of model structure plays a major role in the *bias-variance trade-off* (see (Geman *et al*. 1992) for details about the trade-off), and this should be reflected in the cost functions.

Algorithms for structure identification from data should take into account the complexity of the system, the representational ability of the local models and their associated parameter estimation algorithms and the local availability of data. As shown in Figure 1.22, this can often lead to a number of confusing interacting effects.

Model reduction and simplification

Usually, one wants a model structure which is as simple as possible, but still able to capture the main aspects of the system that are relevant for the given application. A simple structure is desired because it is usually more easy to interpret and understand. Moreover, experience has shown that simpler models are typically more robust and reliable and the task of parameter identification and validation may be simpler.

Certain aspects of model reduction are simplified with operating regime based models. For example, a simplified model can be generated by merging two or more neighbouring

operating regimes into a larger one, provided the local models within the regimes can be merged in a natural way (see Figure 2.6 on page 87). Another example is the reduction of single local models. For example, a high-order local model may be needed in some operating regimes, while a simpler low-order model may be sufficient in other operating regimes.

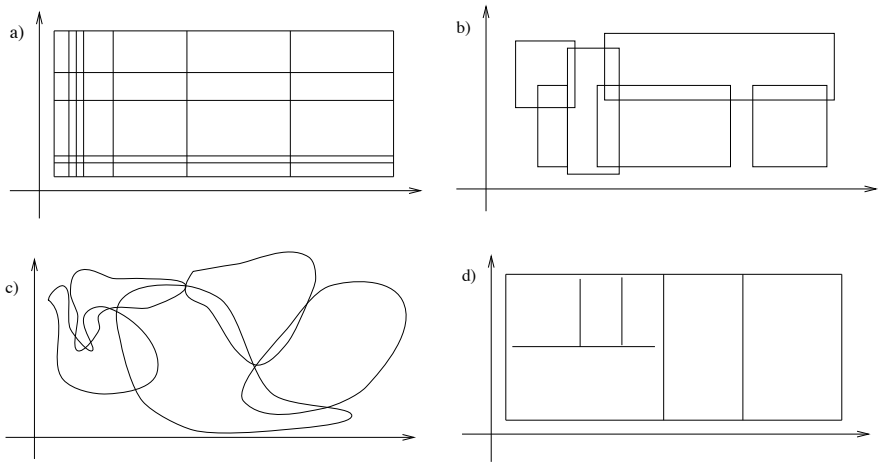


Figure 1.21 Examples of different operating regime representations. In a) we represent operating regimes with a simple regular grid, in b) we represent the operating regimes as rectangles, in c) the operating regime boundaries are parameterised by complicated functions, and d) shows a hierarchical partition of the operating range (as would be produced by a decision tree).

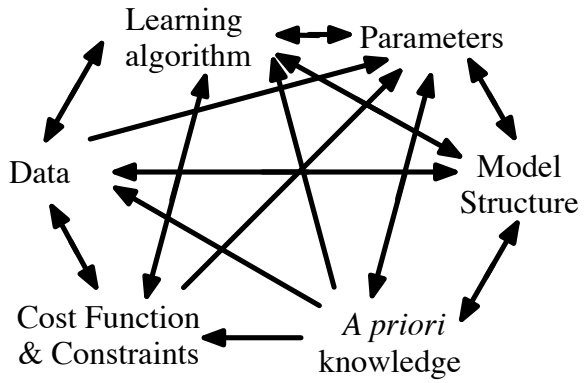


Figure 1.22 The structure learning process involves a number of complex interactions. In practice, many problems are successfully solved only by the intensive use of knowledge of the system structure. Automated approaches can help in this process, but should not be relied on alone.

1.4.4 Literature of local methods in learning and modelling

The earliest model we have found directly based on local models, is from the field of mathematical biology, and dates back to Kolmogoroff (1936). The model is an alternative to the classical Lotka–Volterra model, and based on a decomposition of the system into different regimes (called zones) with qualitatively different behaviour, see (Johansen 1994*c*) for a detailed discussion.

Piecewise models and splines

Piecewise linear models clearly fall into this framework. Early contributions describing modelling procedures based on piecewise linear models and an optimisation formulation of the regime decomposition problem are found in (Bellman 1961, Opoitsev 1970, Dorofeyuk *et al*. 1970, Kasavin 1972, Rajbman *et al*. 1981). Various other procedures are described in (Breiman and Meisel 1976, Cyrot-Normand and Mien 1980, Haber *et al*. 1982, Omohundro 1987, Farmer and Sidorowich 1987, Strömberg *et al*. 1991*a*, Hilhorst 1992, Söderman *et al*. 1993, Štecha and Havlena 1994).

Pao's *Functional Link Network* (Pao 1992), and Billing's closely related *Extended Model Set* ideas (Billings and Chen 1989), use links with a fixed non-linear function built in to expand the input vector, resulting in the production of extra 'higher-order' inputs. These are linearly weighted by the network's parameters.

(Skeppstedt *et al*. 1992) describes the use of local dynamic models for modelling and control purposes, but with hard transfers from one model regime to the next. (Pottmann *et al*. 1993) describes a multi-model approach where although the local models overlap there is still a sharp transition from one model to the next.[10]

A particularly interesting piecewise modelling approach from the biochemical engineering literature is described in Chapter 4, where the application of *a priori* knowledge for the decomposition into regimes (called functional states) is highlighted. The Markov Mixtures of Experts architecture described in Chapter 5, is another example of such switching methods, where the transition functions are identified from data.

The use of more complex local models than linear, for example a piecewise combination of several local neural network models (Sørheim 1990) and local polynomial models (Pottmann *et al*. 1993) are also described in the literature. Clearly, piecewise polynomials are closely related to the spline-based modelling approach, e.g. (Wahba 1990), with the main difference lying in the way splines handle multi-dimensionality. While piecewise polynomials are multi-dimensional by nature, splines are not, and multi-dimensionality is typically introduced by the means of tensor products (this is often also true of fuzzy logic when a conjunction of univariate fuzzy sets is used). The *Gaussian Bar* nets (Hartman and Keeler 1991, Kurcova 1992) are also closely related. These are nets with 'semilocal' basis functions formed by taking one-dimensional local basis functions and forming their tensor product to approximate a multi-variable function, as with tensor product spline models such as the ASMOD system (Kavli 1993). The advantage of these basis functions is that they can better cope with some classes of high dimensional problems, as they do not need to partition 'uninteresting' dimensions. A disadvantage is that the representation does not cope well with processes where the nonlinearity varies with several variables.

[10] To minimise model switching a heuristic criterion is introduced which only allows switching after three consecutive steps in the direction of the new model.

Basis function nets

Basis Function Networks and their equivalents have been used for function approximation and modelling in various forms for many years. The original Radial Basis Function methods came from *interpolation theory* and are described in (Powell 1987), where a basis function is associated with each data point, as in (Specht 1991). *Potential Functions* (Aizermann *et al*. 1964), *Kernels* (Wahba 1992) and *Spline Models* (Wahba 1990) are all similar structures. Albus' CMAC ideas have a great deal of overlap with basis function nets with uniformly distributed local basis functions (Albus 1975, Lane *et al*. 1991, Brown and Harris 1994).

Basis function nets have also received attention from the neural network community, starting with the early papers (Moody and Darken 1989, Broomhead and Lowe 1988). (Hlavácková and Neruda 1993) gives a brief review of the use of RBF nets. (Poggio and Girosi 1990), (Girosi *et al*. 1993) and (Mason and Parks 1992) describe the networks within the mathematical framework of *Regularisation Theory* for function approximation. (Hutchinson 1994) describes the use of RBF nets for financial time series modelling. (Park and Sandberg 1991) and (Park and Sandberg 1993) proved the universal approximation abilities of RBF nets.

The well known *polynomial methods* such as Volterra models and Group Method of Data Handling (GMDH) (Ivakhnenko 1970, Ivakhnenko 1971) can also be viewed as Basis Function systems, as there is again a single layer of nonlinear functions, and the parameter optimisation is a linear process. The problem here is to find the suitable model structure – i.e. which set of basis functions can approximate the system adequately. Some off-line structure identification algorithms are described in (Chen and Billings 1994, Ivakhnenko 1971). Holden describes a general framework for basis function nets, calling them *Phi-nets* in (Holden 1994).

Statistics

The idea of using locally accurate models is also described in the statistical literature in (Cleveland *et al*. 1988) (see (Atkeson 1990) for a review of the literature of *local learning* methods in statistics), where local linear or quadratic models are weighted by smoothing functions. These methods store the entire data set and, for a given input point, form a locally weighted nonparametric representation of the system from the neighbouring data points. *Smoothing methods* such as Gaussian Kernel methods, as described in (Hastie and Tibshirani 1990) like other local averaging methods, suffer from the curse of dimensionality (Friedman 1991), and are computationally expensive. Such methods are applied to control problems in (Schaal and Atkeson 1994). The advantages of local representations are discussed in (Bottou and Vapnik 1992), where they suggest that a proper compromise between local and global methods will usually prove most effective as varying levels of complexity are required throughout the state space, although they claim that the 'local capacity' should match the data density, which is not necessarily true, as this would not relate model complexity to system complexity, but rather to data density. The more general goal of allocating local capacity is that the learning system should match the *local complexity* of the system.

Mixture Models used in statistics (Titterington *et al*. 1985, McLachlan and Basford 1988, Hastie and Tibshirani 1994) are created by mixing a number of probability distributions, and have many similarities to RBF nets (Bishop 1994, Nowlan 1991).

Operating regime based models could be viewed as a finite parameterisation of the state-dependent model. (Priestley 1988) describes *State Dependent Models* for non-linear time

series which are basically linear models where the parameters depend in some way on the state. Tong's *Smooth Threshold Autoregressive* (STAR) models (Tong 1990) are also structurally equivalent to local model nets. In neither Priestley's or Tong's case, however, is much detail given about how to find the smooth weighting functions. (Billings and Voon 1987) also use a number of linear models to approximate a nonlinear system, but do not have smooth interpolation between basis functions.

Fuzzy models

The close relation of basis function nets to classes of *fuzzy logic systems* has also been discussed in (Jang and Sun 1993*b*), and (Brown and Harris 1994), where the similarity between membership functions and basis functions is pointed out. Some *fuzzy logic* systems can also be viewed as operating regime based models, e.g. the methods used in (Takagi and Sugeno 1985), where each operating regime is represented as a fuzzy set. This representation is appealing, since many systems change behaviour smoothly as a function of the operating point, and the soft transition between the regimes introduced by the fuzzy sets representations captures this feature in an elegant fashion. Algorithms for the identification of a decomposition into regimes on the basis of data are described by (Sugeno and Kang 1988) and several experimental and simulated examples are given (Takagi and Sugeno 1985, Sugeno and Kang 1986, Sugeno and Kang 1988, Nakamori *et al*. 1992). Similar applications are reported in (Sugeno and Kang 1988, Foss and Johansen 1993, Wang 1994) and (Harris *et al*. 1993). Related identification algorithms based on cluster analysis are described in (Bezdek *et al*. 1981*a*, Bezdek *et al*. 1981*b*, Hathaway and Bezdek 1993, Yoshinari *et al*. 1993, Nakamori and Ryoke 1994) and Chapters 2 and 13. Similar ideas are applied in chemometrics by (Næs and Isaksson 1991) and (Næs 1991). (Hunt *et al*. 1996*b*) discuss the similarity between fuzzy and basis function systems in more detail. Section 1.5.3 discusses the use of fuzzy methods in control.

Neural network models

Neural networks provide black-box nonlinear models and associated learning algorithms which have some analogies with biological information processing systems, and which have been increasingly applied to engineering problems in recent years. In terms of local model style architectures, the neural network community suggested the RBF network with parameterised local functions instead of constant coefficients in (Jones *et al*. 1989), followed up by (Stokbro *et al*. 1990*a*) and (Barnes *et al*. 1991), which describe approaches to modelling based on local linear models as generalisations of neural networks with localised receptive fields and radial basis-functions. The Adaptive Expert networks in (Jacobs *et al*. 1991) are also essentially local model systems, where the local models are called *expert networks* and the integration of the various *experts* is made by *gating networks*. These were developed into hierarchical models in (Jordan and Jacobs 1991, Jordan and Jacobs 1994). See also the similar approach in (Murray-Smith 1992). This probabilistic approach is found in Chapters 5 and 6. (Back and Tsoi 1991) describe the use of dynamic models as nodes in Multi-Layer Perceptrons – this could be seen as an MLP implementation of operating regime methods (with less locality).

Committee networks have received attention recently in the neural nets field, and have some relevance to multiple model methods. Here sub-models are combined in a committee (Perrone and Cooper 1993, Jacobs 1995) to improve performance. In the simplest case a uniform average of committee members is used, which is equivalent in structure to a

local model net with uniform global basis functions – the difference in output of committee members is due to either differing types of committee members, or different local minima being found, if the optimisation methods used are nonconvex. More complex variations include differing weightings of committee members. Local model nets can be viewed as committee networks with 'more intelligent' weighting functions which demarcate the areas of validity of the 'committee members' – the local models/classifiers/controllers.[11]

Many other approaches to producing artificial systems with intelligent behaviour have developed sophisticated approaches to switching between 'behaviours' or 'skills', and much of this could prove relevant for multiple model approaches to modelling and control. We already mentioned the subsumption architectures (Brooks 1986), and arbitration networks (Yavnai 1993). Albus, in his wide-ranging paper (Albus 1991) states that *intelligence is the helmsman of behaviour*, a statement which fits in well with the idea of switching between or blending submodels or controllers.

1.4.5 Literature of structure identification methods

Parametric optimisation of weighting functions

A gradient descent optimisation technique for weighting function parameter optimisation is described in (Poggio and Girosi 1990). This method was applied in (Röscheisen *et al*. 1992) and also in (Hutchinson 1994), where some practical guidelines for clustering are given. Other RBF researchers used methods where a fixed number of basis functions was assumed, and the structure identification task was seen as the optimisation of the centres and widths. In the papers (Moody and Darken 1989, Sbarbaro 1992) clustering algorithms such as self-organising maps or k-means clustering were used to place the centres. A disadvantage of clustering algorithms is that they do not relate the location of the weighting functions to the complexity of the system being modelled, only to the location of data in the state space. Clustering methods such as k-means can be viewed as special cases of the EM (Expectation Maximisation) method (Dempster *et al*. 1977), which has been gaining increasing recognition in the neural net community (Nowlan 1991). In (Jordan and Jacobs 1994) the use of EM allowed the local model mismatch to be included in the optimisation of weighting functions.

The disadvantage of the techniques described above is that the user must still define the number of local models before identification starts, but as the complexity of the system is usually not fully understood, the optimal model structure is also unknown.

Constructive techniques

A hierarchical clustering technique based on a binary tree approach is used to recursively partition the operating range in (Stokbro *et al*. 1990*a*). The resulting model is 'flat', only the partitioning process is hierarchical. The partitioning is, however, not related to the local complexity of the system, but simply to the presence of data – a drawback of other similar algorithms.

Another option is to start off with a simple model, to estimate its parameters, determine where the representation is still unsatisfactory and to dynamically add new submodels to the model. This leads to a sequence of model structures followed by a parameter identification

[11] A Bayesian interpretation of such model averaging is discussed in section 10.7 of (Bishop 1995). Here, instead of just picking the most accurate model, all models are summed, weighted by the posterior probabilities of each model.

and confidence estimation stage. Constructive techniques which gradually enhance the model in this manner have a number of advantages: They automate the learning process by letting the model grow to fit the complexity of the system, but they do this robustly, by forcing growth to be guided by the availability of data and the complexity of the local models. This automatically determines the size of the model needed to approximate the system adequately, while preventing overfitting, e.g. (Murray-Smith and Gollee 1994, Murray-Smith and Hunt 1995) and (Murray-Smith 1992). (Waterhouse and Robinson 1996) use constructive methods together with EM to train hierarchical mixtures of experts models. See also page 87 for constructive approaches to fuzzy systems.

(Chen *et al*. 1991) used orthogonal least squares for the clustering task in developing radial basis function models. Their algorithm is a constructive one, which uses every data point as a candidate centre. Each time a basis function is added to the model, it chooses the best candidate centre by attempting to minimise the variance in the prediction error due to the model parameters. The *Resource Allocation Net* described in (Platt 1991) is a simple constructive algorithm, where when a data point is presented which causes an error larger than a given threshold a new basis function would be added at that point. Wynne-Jones suggests a constructive method where existing basis functions are split into two. The *Hierarchical Self-Organising Learning* (HSOL) algorithm, a hierarchical strategy for the construction of basis function models for classification, is described in (Lee and Kil 1991). Basis functions are added to the model in a coarse to fine strategy. (Carlin 1992) applied HSOL to modelling problems and similar coarse-to-fine ideas have been used for spline-based modelling applications (e.g. ASMOD in (Kavli 1993)). (Fritzke 1994) describes a constructive method based on a self-organising map framework. The heuristic local search algorithm (LSA) for operating regime based models splits the state space orthogonally to the axes of the state space (Johansen and Foss 1995). The spline-based MARS algorithm (Friedman 1991) mentioned earlier is also a constructive method.

The *Model Merging algorithm* described in (Omohundro 1991) attacks the problem in a different way, using a fine-to-coarse learning algorithm, where each data point is initially viewed as a model, and increasingly global models are created by merging the existing models. The bottom-up or top-down approaches both have advantages, and the choice will depend strongly on the application domain and the style of interpretation desired. See chapter 2 for a detailed example of such a procedure.

Hierarchical modelling and identification algorithms

The use of hierarchy in learning algorithms breaks down into two general camps. The decision tree methods started in the 1970s with *k-d trees* (Bently 1975), *Classification And Regression Trees* (CART) (Breiman *et al*. 1984), and *ID3* and *C4.5* (Quinlan 1993), and involve hierarchies of sharp partitions, dividing the input space into ever smaller areas. (Isaksson *et al*. 1991) and (Strömberg *et al*. 1991*a*) describe the use of such trees with dynamic models in the leaf nodes to model nonlinear dynamic systems. An alternative approach is the use of *soft partitions*, where the input space is no longer sharply partitioned, but there is a gradual transition between local models, allowing smooth interpolation between the local models. This also means that a point in the state space can activate local models in several leaves of the tree, with a differing level of membership to each.

Examples of this type of structure include *Basis-Function Trees* (Sanger 1991*b*, Sanger 1991*a*). The *Hierarchical Mixtures of Experts* (HME) structure, is a hierarchical structure (Jordan and Jacobs 1994) identified using Expectation Maximisation techniques (EM).

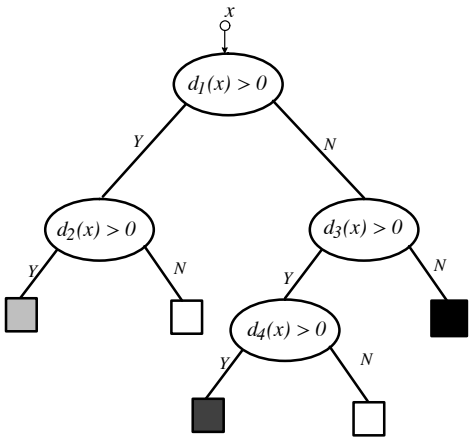


Figure 1.23 Decision tree structure. The conventional approach requires a number of discrete decisions which select leaves of the tree. These leaves can also be complex models or controllers, and the discrete nature of the node decisions can be altered to provide a smooth hierarchical blending structure.

(Miller and Rose 1996) presents more recent work which includes 'annealing' like characteristics in the learning process. Friedman used spline-based techniques to extend the CART ideas to soft splits for his *Multiple Adaptive Regression Splines* (MARS) algorithm (Friedman 1991). Quinlan also started to work with continuous systems modelling using *Model Trees* (Quinlan 1992). Links between wavelets and hierarchical networks (Bakshi and Stephanopoulos 1993) have also been investigated. Banan describes a constructive hierarchical method, with local linear models, which repeatedly partitions the operating range *randomly* in the areas producing errors. The 'average' network produced by the random splits is then the result of identification (Banan and Hjelmstad 1992). (Omohundro 1991) describes *Bump-* and *Balltrees* which have linear classifiers in the leaves of the tree. Constructive methods for hierarchical local model networks are described in (Murray-Smith 1992, Murray-Smith 1994).

The restriction of splits to hyperplanes orthogonal to the axes makes the method simple to implement and interpret, but it scales up poorly to higher dimensions and more complex models, especially when look-ahead search is used. Splits may be axis-oblique, i.e. the split depends on several variables, as shown in Figure 1.24, and they can also be crisp or soft (overlapping regimes). The use of smooth and oblique splits allows a variety of gradient-based optimisation algorithms to be applied to find the optimal partition, which is not possible in the classical decision tree methodology due to their crisp, axis-orthogonal partitions.

The longer term goal for hierarchical learning in multiple model systems is to make real use of hierarchy to make identification more efficient and tractable. A model can develop hierarchically to fit the data, and as the hierarchy develops and the representation of the system improves, decisions made earlier in the learning process (i.e. at higher levels in the decomposition process) can be re-evaluated, leading to gradual changes over a number of levels.

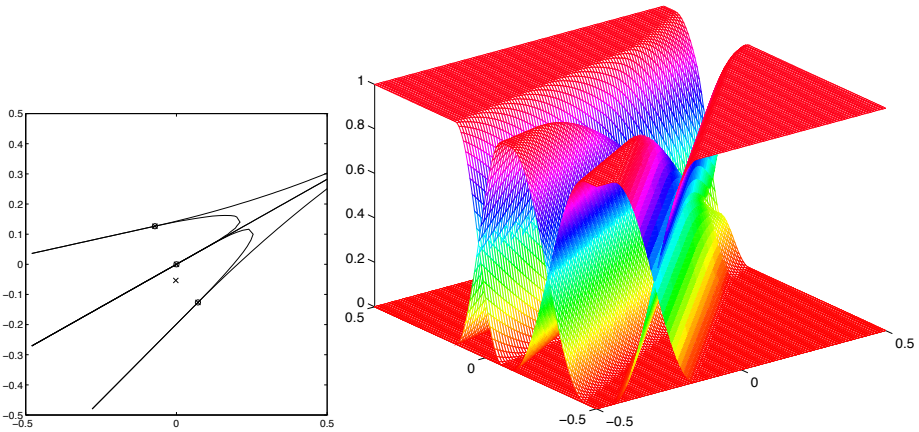


Figure 1.24 Contour and 3D representations of the basis functions of the leaves of the model tree for an axis-oblique partition. This shows graphically that the hierarchical structure can also be profitably viewed as a single layer partition.

1.4.6 Outlook for learning operating regime models

User-friendly methods for the easy integration of *a priori* knowledge into the model will become more and more important as the demands for accuracy, robustness and transparency of simulation models and control systems increase. There is a need for easy-to-use tools which allow the developer to creatively build engineering knowledge directly into the modelling process, leaving the learning algorithms to cope with the uncertainty in the process, and to warn the user when more information is needed, and support the user in obtaining this information.

Despite the improvements in tools, it is worth remembering that *'Thinking, intuition and insight cannot be made obsolete by automated model construction' (Ljung 1987).*

An immediately practical view of the methods described in this section is to see them as computationally- and data-intensive ways of supporting more traditional modelling methods, allowing the engineer to gain insight into the physical nature of the process by reproducing the behaviour with a learned model, understanding the behaviour and then creating a 'hand-built' simplified model which exhibits the essential behaviour of the structurally more complex empirically learned model. This is then more easily understood and validated, and is therefore more likely to be used in practical applications.

1.5 OPERATING REGIME BASED CONTROL – ALGORITHMS AND PROCEDURES

1.5.1 Gain scheduling like approaches

Traditionally, gain scheduling has been perhaps the most common systematic approach to control of strongly nonlinear systems in practice (Åström and Wittenmark 1989, Rugh 1991, Shamma and Athans 1992, Stein 1980, Whatley and Pott 1984, Gangsaas *et al.* 1986, Hyde and Glover 1993, Reichert 1992, Nichols *et al.* 1993). Even with the introduction of powerful nonlinear control strategies such as model predictive control (Garcia *et al.* 1989)

and feedback linearisation (Isidori 1989), gain scheduling remains an attractive control strategy because it has some favourable engineering advantages such as simple design and tuning, and low computational complexity. It should be noted that by gain scheduling we mean a quite general class of control structures characterised by multiple local controllers, including Takagi–Sugeno fuzzy controllers (Takagi and Sugeno 1985).

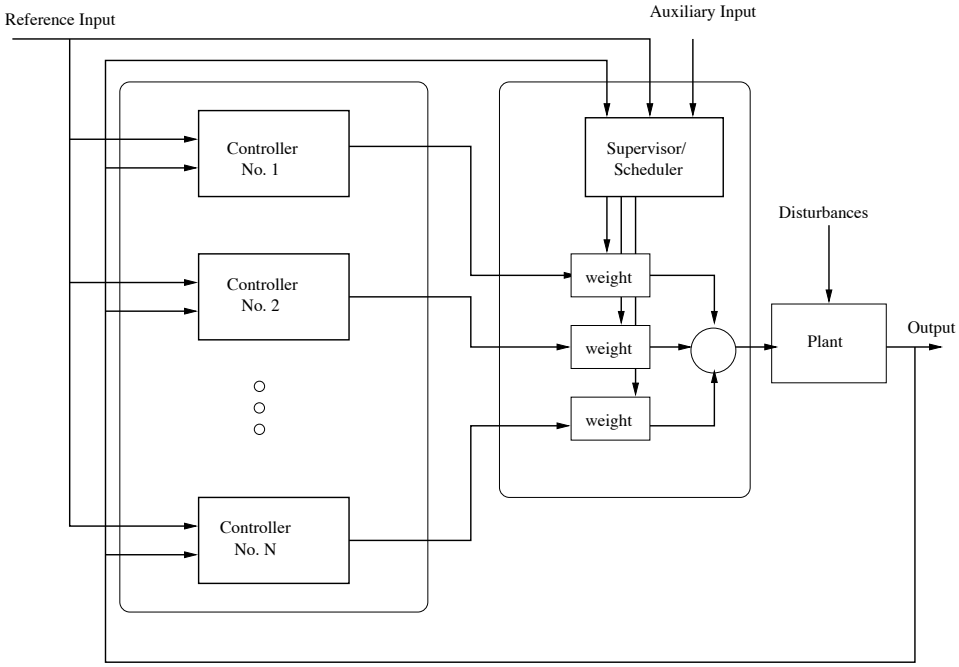


Figure 1.25 Typical structure of gain scheduled control system. Note that most of the literature of gain scheduled systems describes interpolation of parameters, but outputs can also be interpolated. See (Shorten 1996) for a discussion of the merits of parameter versus output interpolation.

A gain scheduling control system consists of two major components, see Figure 1.25: A family of controllers, and a scheduler that at each time instant assigns a controller (or a mixture of controllers) to be applied to the plant. One can think of the control system as having multiple local controllers that are applied in different operating regimes. In practice, gain scheduling can appear in different forms:

- A single controller from the family of controllers is selected. Alternatively, two or more controllers can be interpolated.
- One can interpolate local controllers' parameters (if they are parameterised in the same way), or the controllers' outputs.
- The family of controllers can be finite or infinite. In the infinite case the family of controllers is typically designed on the basis of a linearisation family of local models, e.g. (Wang and Rugh 1987).
- The scheduler can make its decision on the basis of a number of different variables and criteria. The variables can be states, inputs, measured disturbances, auxiliary variables, time etc.

- Various control design methods can be applied and mixed to develop the family of controllers.

Local controller design

In virtually all design cases when explicit models are involved, the local controllers are designed on the basis of linear (or linearised) models about a family of equilibria. This is related to the fact that linear models and controllers are well understood and supported by software tools for identification, design and analysis. The gain scheduling framework in itself does not impose any restrictions on the local controller design algorithms, unless global stability and performance properties are taken directly into account. Hence, one can find examples of application of all the well known linear control design algorithms and heuristics.

Standard linear control design methods such as root locus (Tanaka and Sugeno 1992), pole placement (Shahruz and Behtash 1992) and frequency domain tools are widely applied. So are optimisation based methods such as LQR (Palm *et al*. 1995), generalised predictive control (GPC) (Chow *et al*. 1995), and optimal PID tuning (Jiang 1994). Recently, there has been a significant number of reports on the use of so-called robust and H_∞ control design methods in gain scheduling (Kellet 1991, Hyde and Glover 1993, Nichols *et al*. 1993). In particular, the use of Linear Matrix Inequalities for local control design with guaranteed global stability, robustness and performance are investigated in (Packard 1994, Apkarian *et al*. 1994, Apkarian and Ghainet 1995, Banerjee *et al*. 1995, Zhao *et al*. 1995*a*, Zhao *et al*. 1995*b*), see Chapters 12 and 13.

Scheduler design

Unlike local controller design, the scheduler design problem is not so well understood and there are fewer tools in the control engineer's toolbox to approach this design problem. The scheduler design problem consists of two major tasks: first to define the scheduling variables, and second to select a scheduling algorithm for selecting the local controller(s) to be applied.

Scheduling variables are usually selected on the basis of the following two heuristics (Shamma and Athans 1990, Shamma and Athans 1992):

- The scheduling variables should capture the nonlinearities. After all, dealing with nonlinearities due to a wide operating range is usually the main reason for applying gain scheduled controllers.
- The scheduling variables should be slowly time-varying compared to the desired bandwidth of the closed loop. This is motivated by the beneficial effect of separating timescales in complex and hierarchical control loops. It should, however, be noted that this heuristic is not always relevant. In particular, when the system is time-invariant and first order, this heuristic may not be necessary (Shamma and Athans 1990, Shamma and Athans 1992, Rugh 1991).

The simplest form of scheduling is hard switching between controllers or control parameters. Examples of this can be found in (Jiang 1994, Jacobs and Jordan 1993, Marcos and Artaza 1995, Morse 1995). Alternatively, the use of linear interpolation is very common (Hyde and Glover 1993, Nichols *et al*. 1993, Chow *et al*. 1995). As we have discussed in previous sections, operating regimes can conveniently be described as fuzzy sets and fuzzy inference can be applied to form a kind of interpolation (Takagi and Sugeno 1985, Tanaka

and Sugeno 1992, Palm *et al*. 1995, Zhao *et al*. 1993, Zhao *et al*. 1995*b*, Cao *et al*. 1995). An advantage of interpolation in contrast to switching is that the control action will be smoother. If switching is applied, one should take explicit action to provide 'bumpless transfer' between the local controllers. Usually, this can be implemented by initialising the state of the controllers correctly after switching.

In general, one can interpolate the local controller outputs, or one can interpolate the local controller parameters (Kuipers and Åström 1994), see also Chapter 9. Of course, interpolation of controller parameters assumes that the controllers are homogeneous in structure, whereas interpolation of the local controller outputs does not impose any such constraints. Interpolation of outputs will increase controller order, whereas interpolation of parameters will not (Shorten 1996). This can be easily seen by considering the transfer function representation of each controller in the SISO case; in the presence of constant external scheduling vectors the transfer function of the control structure at intermediate equilibrium conditions is a convex sum of the individual controller transfer functions. The parameterisation of the controller may be of some importance if parameters are interpolated. Usually, the interpolation of zeros, poles and gains provides smoother and more robust control than the interpolation of polynomial coefficients in rational transfer functions (Chow *et al*. 1995, Shorten 1996).

Gain scheduled control system analysis

Analysis of the properties of the closed loop is of great importance. Somewhat surprisingly, gain scheduling control loop has been a neglected topic in the control theory literature until recently. In this section we will describe some results on stability and robustness of the closed loop, and discuss their relevance. Finally, we will briefly discuss performance.

Consider the problem of controlling a nonlinear system

$$x(t+1) \quad = \quad f(x(t), u(t)) \tag{1.13}$$

that can approximately be described as a weighted combination of a number of linear systems:

$$x(t+1) \quad = \quad \sum_{i=1}^{N} (x_i + A_i(x(t) - x_i) + B_i(u(t) - u_i))w_i(z(t)), \tag{1.14}$$

where the point (x_i, u_i) corresponds to an equilibrium, $z(t)$ is the scheduling variable, (A_i, B_i) is the controllable local linear model

$$\begin{aligned} A_i &= \frac{\partial f}{\partial x}(x_i, u_i), \\ B_i &= \frac{\partial f}{\partial u}(x_i, u_i). \end{aligned}$$

Furthermore, consider a gain scheduled local linear state feedback

$$u(t) \quad = \quad u_i + \sum_{i=1}^{N} K_i(x(t) - x_i)w_i(z(t)). \tag{1.15}$$

The feedback matrix K_i can be designed using any design method, and the scheduler can be defined by arbitrary positive definite functions that satisfy

$$\sum_{i=1}^{N} w_i(z) \quad = \quad 1$$

for all z. It is clear that if there exists a matrix $P > 0$ such that $\tilde{A}_i^T P \tilde{A}_i - P = Q < 0$, then the linear control system

$$\begin{aligned} x(t+1) &= x_i + A_i(x(t) - x_i) + B_i(u(t) - u_i) \\ u(t) &= u_i + K_i(x(t) - x_i) \end{aligned}$$

with the closed-loop behaviour

$$\begin{aligned} x(t+1) - x_i &= \tilde{A}_i(x(t) - x_i) \\ \tilde{A}_i &= A_i + B_i K_i \end{aligned}$$

is asymptotically stable. This can easily be seen since the Lyapunov function $V(x) = x^T P x$ is decreasing along the system trajectory, e.g. (Khalil 1992). In (Tanaka and Sugeno 1992), this argument is extended to the nonlinear system (1.14) controlled by the gain scheduled controller (1.15). The stability condition is now that there must exist a common matrix $P > 0$ such that

$$\tilde{A}_{ij}^T P \tilde{A}_{ij} - P < 0, \qquad \text{for all } i, j = 1, 2, ..., N \text{ where } w_i w_j \neq 0, \tag{1.16}$$

and $\tilde{A}_{ij} = (\tilde{A}_i + \tilde{A}_j)/2$, see also (Tanaka and Sano 1994). Again, this is seen quite easily from a Lyapunov argument, see also (Packard 1994, Apkarian *et al*. 1994, Apkarian and Ghainet 1995). In special cases, when for instance the system is first-order or in some canonical form, this is a quite strong result. However, one should be aware that these are only sufficient conditions for stability, that may be very conservative, in general. Hence, even if there does not exist a P matrix that satisfies the inequalities (1.16), then the closed loop may still be stable. Also notice that this result depends on how well the model (1.14) approximates the system (1.13). This is a topic that has been discussed in (Perev *et al*. 1995, Johansen and Foss 1993*a*), and in section 1.2.3.

In order to test the existence of the above-mentioned P matrix, it has been observed that (1.16) can be reformulated into a set of linear matrix inequalities (LMI) (Zhao *et al*. 1995*a*)

$$\begin{pmatrix} P & P\tilde{A}_{ij} \\ \tilde{A}_{ij}^T P & P \end{pmatrix} > 0, \qquad \text{for all } i, j = 1, 2, ..., N \text{ where } w_i w_j \neq 0.$$

LMIs can be studied and solved using the tools described in (Boyd *et al*. 1993, Boyd *et al*. 1994). If a P matrix is found, then Q can be computed and from the Lyapunov function it is possible to determine an upper bound on the convergence rate of the norm of the state, see Chapter 13. Related algorithms that guarantee H_∞ optimal performance are found in (Becker and Packard 1994, Apkarian *et al*. 1994, Apkarian and Ghainet 1995) and Chapter 12.

Robustness issues are studied in (Zhao *et al*. 1995*b*) and Chapter 13, who consider an uncertain model of the form

$$x(t+1) = x_i + (A_i + \Delta A_i)(x(t) - x_i) + (B_i + \Delta B_i)(u(t) - u_i),$$

where ΔA_i and ΔB_i represent the uncertainties. Related results can be found in (Tanaka and Sano 1994, Packard 1994, Becker and Packard 1994, Apkarian *et al*. 1994, Apkarian and Ghainet 1995).

With the formulation above, the closed loop can be written

$$x(t+1) = \tilde{A}(z(t))x(t) \tag{1.17}$$

for some matrix $\tilde{A}$. If the local controllers are designed such that the poles of $A_i + B_i K_i$ are within the unit disc, then the system (1.17) will be asymptotically stable for any fixed value of the scheduling variable $z(t)$. However, it is well known that this is not a sufficient condition for uniform asymptotic stability of the system (1.17). If we consider $\tilde{A}(z(t)) = \tilde{A}(t)$ as an explicit function of time, then it is well known that a sufficient condition for uniform asymptotic stability of the system (1.17) is that the scheduling variables are sufficiently slowly time-varying (Desoer 1970), (Shamma and Athans 1990) and (Lawrence and Rugh 1990), in the sense that there exists a sufficiently small δ such that

$$||\tilde{A}(t) - \tilde{A}(t-1)|| \quad \leq \quad \delta. \tag{1.18}$$

Hence, the heuristic to schedule on slow variables can be justified. However, it is important to bear in mind that when $\tilde{A}$ is viewed as a function of time, important structural information is lost when for instance it is partially a function of the state. Hence, the stability condition (1.18) may be conservative, and tight performance bounds and robustness margins may be hard to compute.

1.5.2 Multiple model adaptive control

In gain scheduling control systems, the scheduling algorithm is a deterministic function of the scheduling variables. The closely related Multiple Model Adaptive Control approach differs mainly from gain scheduling by the use of an estimator based scheduling algorithm used to weight the local controllers, see Chapter 11 or (Lainiotis 1976*a*, Lainiotis 1976*b*, Athans *et al*. 1977, Greene and Willsky 1980). A typical structure contains a model bank with Kalman filters with the twofold purpose of estimating the local model states as well as estimating local measures of accuracy. Sometimes a Markov model of the transition between the local models is also included (Sworder 1969, Blom and Bar-Shalom 1988). The local controllers are often LQG controllers (Chizeck *et al*. 1986, Griffiths and Loparo 1985), but other approaches exist. Closely related is also the approach in (Gendron *et al*. 1993, Narendra *et al*. 1995), where local model performance indices are used to select the local controller. One potential benefit of these approaches compared to gain scheduling is improved transient performance. The potential drawback is lack of robustness because there is a risk of selecting a poor controller when the system is not persistently excited. This is a general and well known problem with adaptive control approaches, e.g. (Narendra and Annaswamy 1989).

1.5.3 Fuzzy logic control

Within the area of fuzzy control there exists a wide variety of control structures and approaches. Somewhere in virtually all of these approaches one can find the idea of making a fuzzy partition of some region. Sometimes, this region can be viewed as the system's operating range, and the fuzzy partition can be interpreted as operating regimes.

The Takagi–Sugeno–Kang model has been discussed in section 1.4.4, and is used in Chapters 2 and 13, and the corresponding control structure is completely equivalent to gain scheduling based on local linear models and a fuzzy logic based scheduler. Therefore, this approach to fuzzy control was discussed in section 1.5.1 on gain scheduling, and we will not dwell on it here.

Fuzzy linguistic control is a control strategy based on linguistic (or rule-based) knowledge, see e.g. (Langari and Berenji 1992) or (Driankov *et al*. 1993) for a comprehensive introduction. The typical structure of a fuzzy linguistic controller is as shown in Figure 1.26.

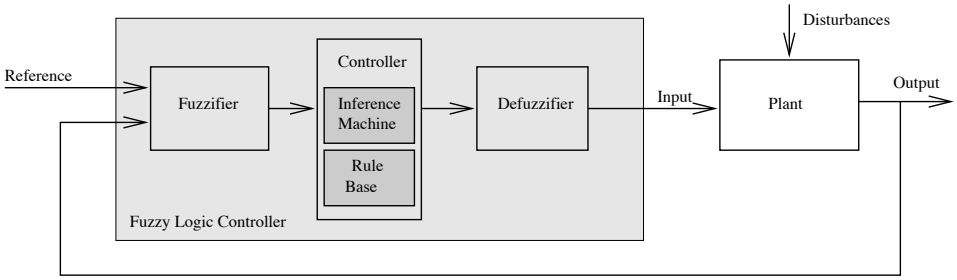


Figure 1.26 Typical structure of fuzzy logic control system.

Table 1.1 Example of a nonlinear PI-like fuzzy control rule table. The table shows the rate of change in control action as a function of the error and its derivative. The linguistic values are NL, negative large, NS, negative small, ZE, zero, PS, positive small, PL, positive large, which are being represented by fuzzy sets.

$\dot{e}/e$	NL	NS	ZE	PS	PL
NL	PL	PL	PS	ZE	ZE
NS	PL	PL	ZE	ZE	NS
ZE	PS	ZE	ZE	NS	NS
PS	ZE	ZE	ZE	NL	NL
PL	ZE	NS	NS	NL	NL

The effect of the components of the linguistic fuzzy logic controller is illustrated in Figure 1.27. In the first step the fuzzifier converts numerical information (like a measured output value, see Figure 1.27a) into a fuzzy set that the fuzzy inference mechanism can take as input, see Figure 1.27b. Uncertainty in the numerical information (such as measurement noise) can naturally be represented. The control rules in the rule base, Figure 1.27c, relate operating conditions (or the state) to control actions, such as

`IF` e `is positive small AND` $\dot{e}$ `is negative large THEN` $\dot{u}$ `is zero.`

Clearly, the premise part of such control rules corresponds to a set of states that can be viewed as an operating regime, and the knowledge base can often be visualised and interpreted as being based on a (fuzzy) partitioning of the system's operating range, e.g. (Qin and Borders 1994). Often, the rule-base can be naturally represented as a table, see Table 1.1. The inference mechanism combines the fuzzy set representation of the measurements with the rule-base, and infers a fuzzy set representation of the control action, see Figure 1.27d. Finally, the defuzzification converts this fuzzy set into numerical information such as a command to the actuator, Figure 1.27e.

Hence, the design of such controllers consists of three main tasks:

1 Design of the rule-base on the basis of e.g. operators' or engineers' experience, knowledge and heuristics.
2 Design of membership functions that represents the fuzzy sets involved in the rule-base. This is often a considerable task due to the large number of numerical tuning parameters involved.

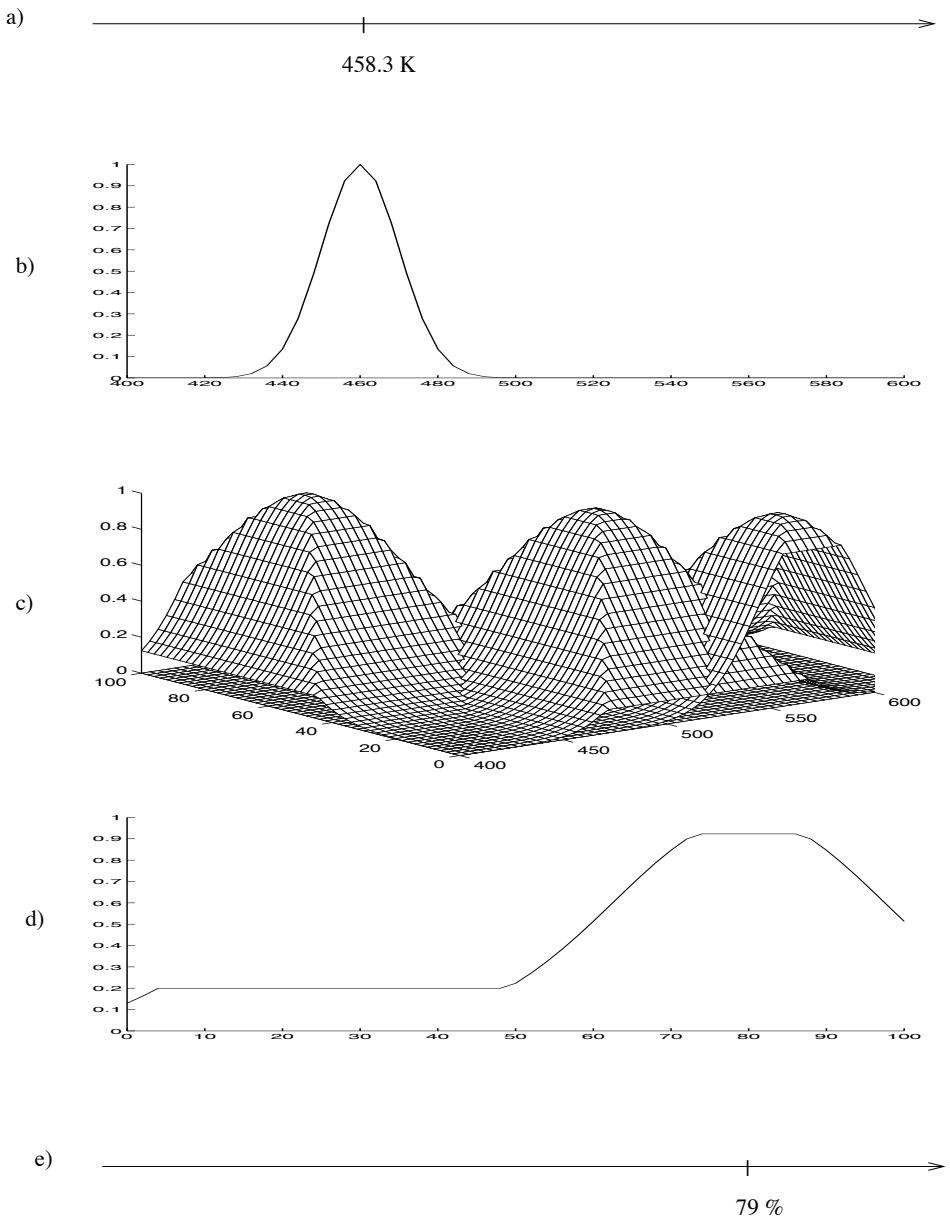


Figure 1.27 Example of fuzzy control inference.

3 Choice of inference mechanism, fuzzification and defuzzification algorithms. We will not discuss these issues here, but refer instead to (Driankov *et al*. 1993, Lee 1990*a*, Lee 1990*b*).

The analysis of stability, robustness and performance of such control systems is still very much an open problem, although some contributions exist. In (Langari and Tomizuka 1990)

it is assumed that the fuzzy logic controller is simply a static nonlinear function relating control error to control input. Applying a small gain type of argument, stability conditions related to the fuzzy logic controller parameters are then derived.

Another approach is taken by (Marin and Titli 1995) who assume the system is linear. It is suggested to compute the describing function of the nonlinear fuzzy logic controller numerically and use the Nyquist stability criterion to check stability of the closed loop. This technique is applied in (Olafsson 1995). Other approaches to stability analysis involve the circle criterion or conicity criterion (Driankov *et al*. 1993), and Interval Matrix method (Jamshidi and Titli 1995).

1.5.4 Global model-based control using local models

Instead of explicitly exploiting operating regimes directly in the control structure, one can apply operating regime approaches to develop an operating regime based model that can be applied in a model-based controller. This indirect approach has several advantages over the direct approaches described in previous sections. The main advantage is that a global controller is designed, which is expected to give better plant-wide control performance simply because global information can be applied to determine the control input at each time-step. This approach has been extensively studied using quite general model-based control design methods such as nonlinear model predictive control (Nakamori *et al*. 1992, Suzuki *et al*. 1992, Foss *et al*. 1995), feedback linearisation (including the adaptive control case) (Wang 1993, Johansen 1994*b*, Johansen 1994*a*), and sliding mode control (Ishigame *et al*. 1993).

With such model-based approaches it is evident that it is the quality of the particular model combined with the general properties of the model-based controllers that determines the stability, robustness and performance of the closed loop. Hence, the reader is referred to the general literature on such approaches (Rawlings *et al*. 1994, Isidori 1989, Sastry and Isidori 1989, Utkin 1992).

1.6 MODEL ANALYSIS, VALIDATION AND INTERPRETATION

Once a model structure and parameters have been identified from data it is necessary to validate the model's accuracy and properties. This is obviously a very important stage in a process which by its very nature has insufficient available intuition about the behaviour of the target system. Model validation is an issue which has often been given too little emphasis in the literature on nonlinear modelling and identification, but one which is very important in industrial situations. In 'local' methods, as with others, there will usually be a trade-off between flexibility and interpretability, the outcome of which will depend on their relative importance for a given application. It is therefore important to combine data-driven validation – is the model adequately accurate and robust for its purpose? – with more subjective validation, i.e. does the model behave in a way which seems physically plausible? Can the model be interpreted to give the engineer a better understanding of the system in question?

This can be rephrased as 'How do we extract *a posteriori* knowledge from the identified model?'. We have already discussed how the local representation, with its decomposition into operating regions, limited the interaction between model components and eased the introduction of *a priori* knowledge. It also, by the same logic, makes the extraction of knowledge *a posteriori* more simple.

Each operating regime can be handled more or less independently of the other operating regimes allowing one to solve complex nonlinear modelling or control problems by designing simple (e.g. linear) local models or controllers in each operating regime independently. This is important for validation, as the framework supports incremental modelling and simple model maintenance, because the modification of a single local model or operating regime has a more predictable effect on the global model.

Not surprisingly, this approach will in some cases *lead to spurious global behaviour of the model or controller even though the local behaviours and designs are well-behaved*, for various reasons. Hence it *is* important to analyse the global properties of the model or controller (see Chapter 9 for qualitative methods for simulating global behaviour). As we have discussed in section 1.5.1 and will discuss later, there exist some theoretical assumptions which relate the local properties to the global properties. However, in most practical situations one must rely on a less stringent analysis, based on extensive simulation of the global behaviour and transients, for example. This is partially due to conservativeness and lack of verifiability of the theoretical results.

One important aspect mentioned above, is that during modelling and identification the engineer is also gaining a better understanding of the system. This means that the *a priori* knowledge about the system in question will be added to, and can be used in a subsequent modelling iteration. The developer is still a vital part of the process, and the goal should be to enhance the power of the interactive software by automating the identification process wherever possible, but by always giving the engineer the freedom to intervene in any decisions. As knowledge about the system increases, more possibilities for better structuring and simplifying the identification task should become clearer. This then allows the automatic modelling and identification techniques to be used more successfully, and closes the loop in the modelling cycle seen in Figure 1.18, where the *a priori* knowledge is improved by the availability of the validated model.

Documentation properties of different representations

The final product, a working model, should therefore be seen as a contribution to the more general pool of engineering and scientific knowledge. The classical laws, rules or models we take as *a priori* knowledge today were also once poorly understood observed behaviour, which was then measured, analysed and turned into some simpler law or model. Kepler's laws of planetary motion were found only after painstaking acquisition of observed data, and the application of a variety of model structures to the data, estimation of the model parameters and validation of the models on new data!

Development of complex mathematical models is an expensive and time-consuming procedure. To lower costs and improve the maintainability of models, it is important that the model structures can be well documented for the use of future engineers. This means that the choice of local model structure, the variables used to schedule on, and the interactions between variables should all be easily documented. Local approaches are well suited to this, as the description of any complex system requires some sort of partitioning into understandable subsystems. They are certainly better suited to this than nonlinear black-box approaches such as neural networks. Also, if local models specific to a certain engineering discipline are used, it may often be possible to use many of the existing specialist tools for documenting the behaviour of such models.

By their very nature, graphical model structures such as Finite State Automata, Belief Networks or Markov Graphs have obvious advantages here. The interconnections between operating regimes are clear, and the scheduling/state transition variables can be clearly

labelled, and if nonlinear transition functions are used, the same sort of visualisation techniques can be applied as for the basis functions in the next section. Decision trees are also graphical in their nature, and can be clarifying for some applications (e.g. diagnosis), but may not be ideal for intuitively describing dynamic systems.

Visualising the weighting functions

An advantage of operating regime approaches is that the model or controller structure can be visualised in terms of operating regimes, as we have seen in numerous figures above. This transfers the modelling or control design problem to a different level than thinking in terms of complex nonlinear equations. This framework is also engineering-friendly, since the operating regime concept is widely applied among engineers. This section discusses the interpretability of operating regimes and local models.

For one-dimensional problems it is relatively easy to get a clear picture of the model, as can be seen in Figure 1.28. Another interesting technique for one dimension is the effective kernel approach used in section 7.6.2, which gives insight into the data set and the learning algorithm, but this is unlikely to be of much use in other than toy examples.

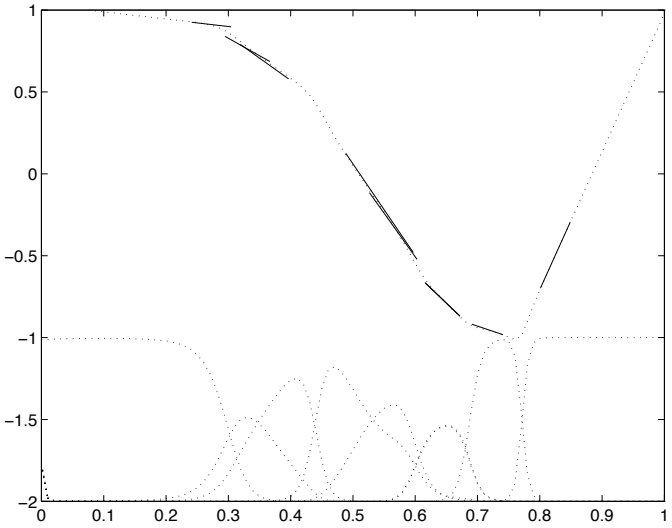
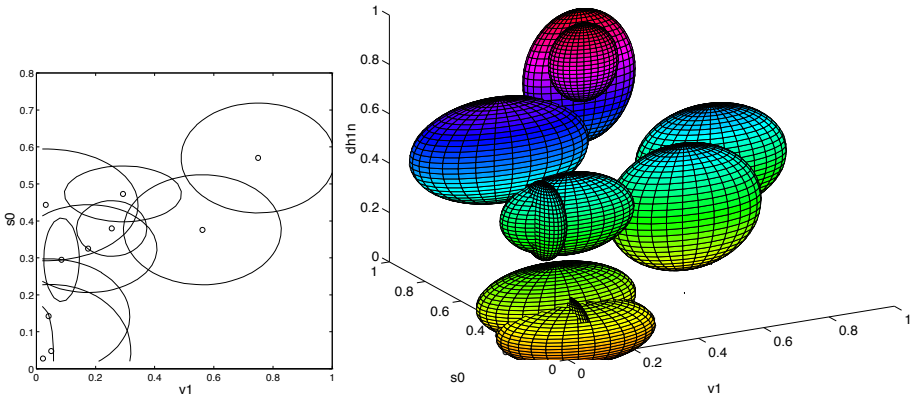


Figure 1.28 Local models, global model output and weighting functions.

In two and three dimensions it becomes harder, but it is still feasible to plot the weighting functions, as in Figure 1.29, from (Murray-Smith 1994). For higher dimensions it becomes difficult to get meaningful visualisation, other than 'slices' through the state space, unless a hierarchical representation is applied.

Note, however, that the weighting functions drawn above were unnormalised. When weighting functions are normalised their shape changes significantly (see, e.g. Figure 1.30). Chapter 8 deals more thoroughly with the effect of normalisation on the behaviour of the model.

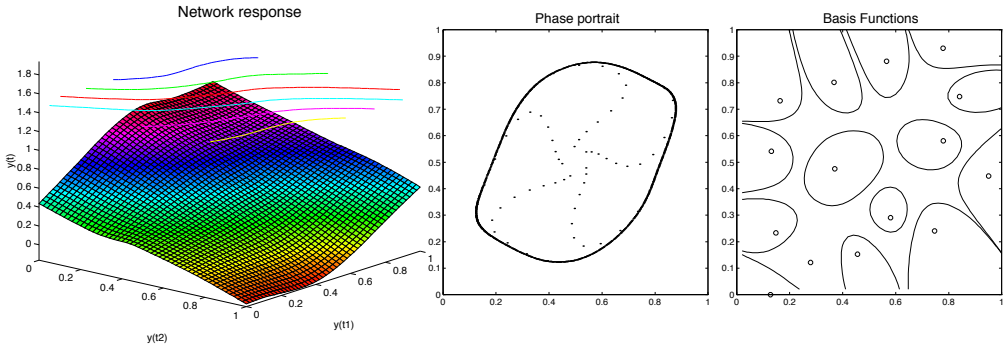
A further aid to understanding the model is to go through the validation runs comparing the model output and actual output, while viewing the variables characterising the operating



(a) 2-D basis function contour plot (contours drawn at 0.5).

(b) 3-D plot of weighting functions for roll mill model. The weighting functions are shown here in the unnormalised form, basically as ellipsoids representing the volume equivalent to that of the contour plot at 0.5. The hyper-ellipsoids in the figure correspond to the scales of the distance metrics of the weighting functions.

Figure 1.29 Visualisation of model operating regimes in two and three dimensions. A typical training run for the system (an aluminium mill) which is modelled by these operating regimes is shown in Figure 1.31.



(a) Model response (b) Test run (c) Basis functions

Figure 1.30 Time series model with local linear models. This figure shows the nonlinear model response surface, time series data from a test run, and the normalised weighting functions.

regimes. The top plot in Figure 1.31 shows the model and actual output, while the plot below shows the characteristic variables for the run for the entire data set, with the portion being examined above marked as the darker area.

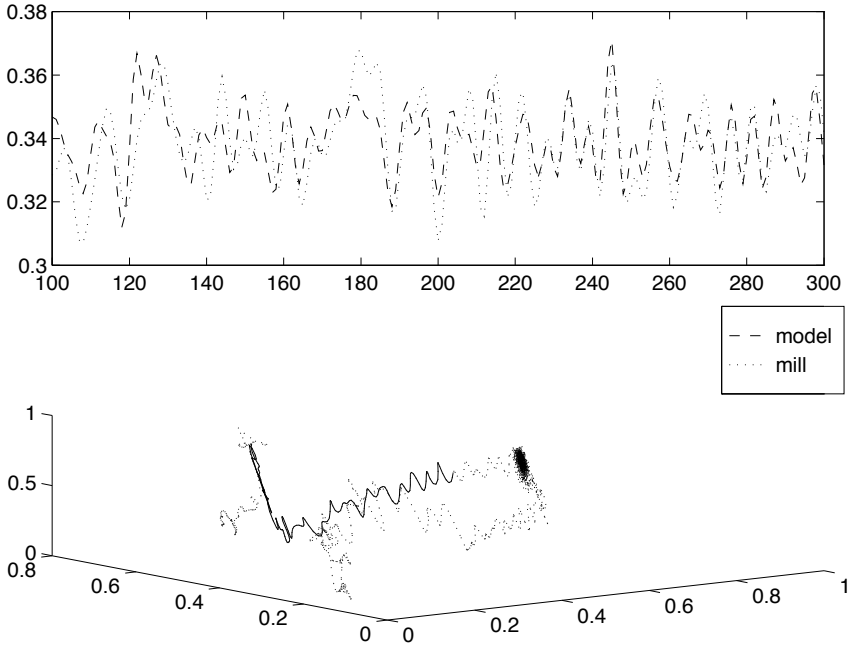


Figure 1.31 Model and system output, with the related area of the operating range. The visualisation tool allows you to examine the model residuals, while viewing the current position in the operating space below (the subset of the data corresponding to the model plot above is highlighted). More powerful tools would allow the use of such plots in conjunction with images such as Figure 1.29, where the local model closest to the data shown could be 'clicked' on and the structure and parameters viewed.

The use of such visualisation tools lets the developer examine the areas of the operating space responsible for large model deviations, to try and determine possible inadequacies in the model structure (e.g. too few local models in a particular region, ill-suited local model structures, etc.) with the aid of graphs such as shown in Figure 1.29. These tools are obviously limited to any three given dimensions, but can still provide useful insight in many cases. Further development of such tools would allow the user to select certain areas of the input space and find the nearest local models, so that their structures and parameters can be investigated, or gain more insight about the model workings. Axis-orthogonal partitions are produced by methods such as tensor product splines and fuzzy systems. These are easier to interpret, as in Figure 1.21. As described earlier, the hierarchical form of operating regime model can be viewed as a decision tree, a representation which, for binary trees with axis-orthogonal partitions (i.e. splitting on a single variable at a time), is often claimed to be especially suited to human interpretation.

For dynamic systems it is also helpful to plot the weighting functions against time, as in Figure 1.32.

The basis functions in a local model net are usually visualised as the individual ρ_k, but there are other ways of interpreting estimated models. Nonlinear data sets are often

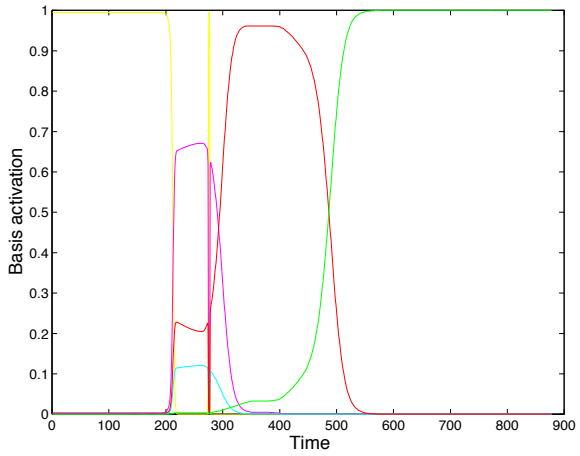


Figure 1.32 Plotting functions against time for a dynamic system.

modelled by simply *smoothing* the training data (Hastie and Tibshirani 1990) around the point of interest. This smoothing interpretation is described in section 7.6.2 on page 199, with graphical illustrations of such smooths.

Interpreting the local models

The local models in the operating regime approach are supposed to bring a greater transparency because they are *meaningful in their operating regimes*.[12] How far can we take micro-analysis of the component models, compared to macro-analysis of the combination of sub-models? The validity of the different methods of interpretation will depend on the training methods used. For example, linear local models estimated locally (see Chapter 7) around an equilibrium point (see Figure 1.33[13]) can be meaningfully treated as classical linearisations (although there will be differences from results produced by perturbation methods). If their parameters have been optimised globally however, it is no longer easy to do this if there is significant overlap with neighbouring local models.

A different method is to view the parameters of the current 'local model' of a homogeneous local model network as the weighted average of the local models' parameters . This can then be used to produce plots of important variables changing with the characterising variables. Examples of this include local models' poles and zeros (see Figure 3.7(b) on page 117 for an example of this), the eigenvalues of local state-space models, and parameter values themselves, if physically meaningful, with their local variance.[14]

The question of local interpretability becomes more complex when we deal with dynamic systems whose state trajectories pass through a number of local regions rapidly. Assume we have dynamic models that are supposed to be valid in some local regime i, i.e. $\dot{x} = f_i(x)$, where each local regime is parameterised by some weighting function ρ_i, leading to the

[12] We would like to thank Roger Bjørgan for contributing this section.

[13] We would like to thank K. Hunt and R. Haas for this diagram.

[14] Local estimates of accuracy are relatively easy to produce because of the partitioning of the state space inherent to the model structure, see (Murray-Smith 1994) for details.

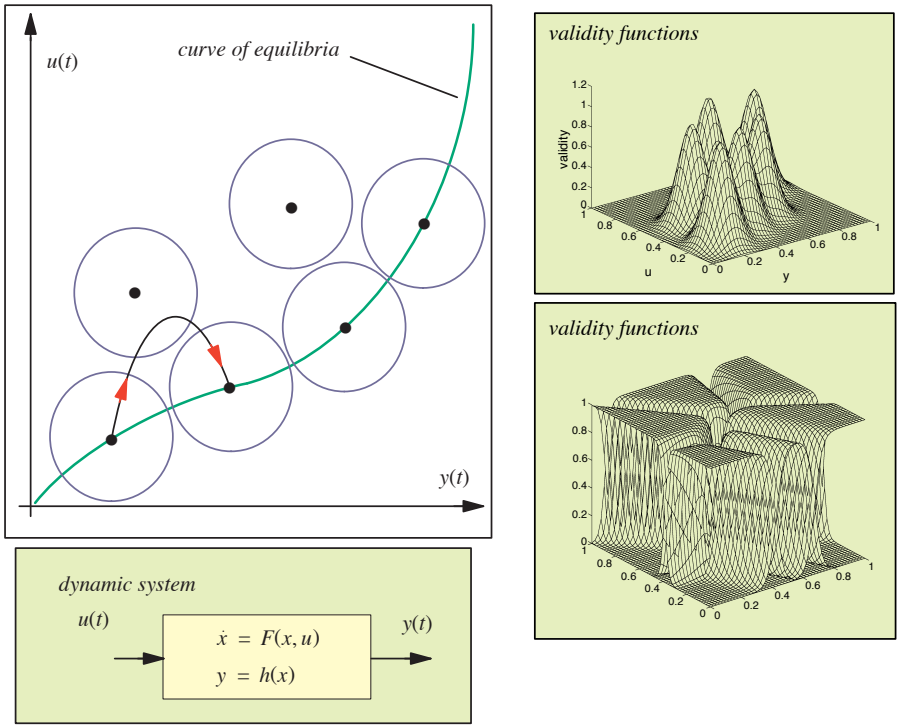


Figure 1.33 Interpreting local models. On-equilibrium local models can be interpreted classically, while off-equilibrium models can only be viewed as representing the dynamics on a part of a trajectory through the state space.

following global, dynamic model:

$$\dot{x} = \sum_{i=1}^{n_{\mathcal{M}}} \rho_i(x) f_i(x) = g(x). \tag{1.19}$$

If we introduce a small change to our notation:

$$\dot{\vec{x}} = \sum_{i=1}^{n_{\mathcal{M}}} \rho_i(x) \vec{f}_i(x) = \vec{g}(x), \tag{1.20}$$

for any trajectory of x, the vector $\vec{g}(x(t))$ is tangent to $\vec{x}(t)$. In Figure 1.34 we see a simple example of a vector field together with one trajectory produced by this vector field.

Having defined the local models and vector fields, we can return to the question of the interpretability of these models. The usual justification for local, linear models in dynamic systems theory is that if the operating trajectory is approaching an equilibrium point, then there is a regime around that equilibrium point where the linearised system describes the trajectory. This is the classical interpretation of local models, with a solid theoretical backing, producing valid approximations around the equilibrium point. If the system is adequately characterised by a linear (or affine) system in some region, even if it is not around

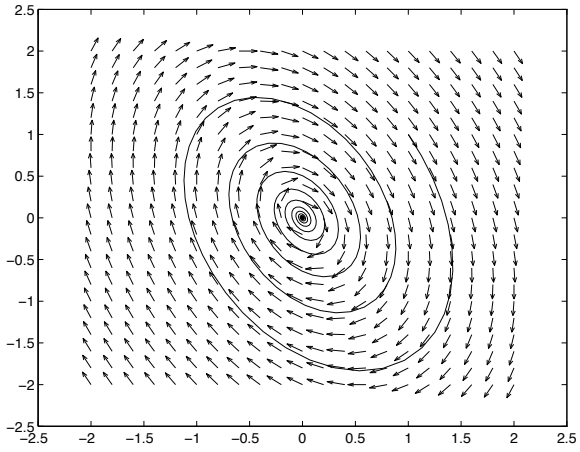


Figure 1.34 The vector field of a stable linear system and a sample trajectory.

an equilibrium point, we can still interpret the behaviour in this regime by using the local, linear model, *provided that the trajectory stays in this regime*.

If the trajectory doesn't stay in a single regime, a new vector field will be patched together by the local models. The linearity of each of the individual local models does not *necessarily* imply that the overall behaviour of the full system will be easy to interpret. To illustrate this we use two of the linear systems shown in Figure 1.34, and give them different offset terms (see equation (1.21)), and smooth validity functions to switch between the regimes

$$\dot{x} = \rho_1(x)\,[Ax + b] + \rho_2(x)\,[Ax - b]\,, \tag{1.21}$$

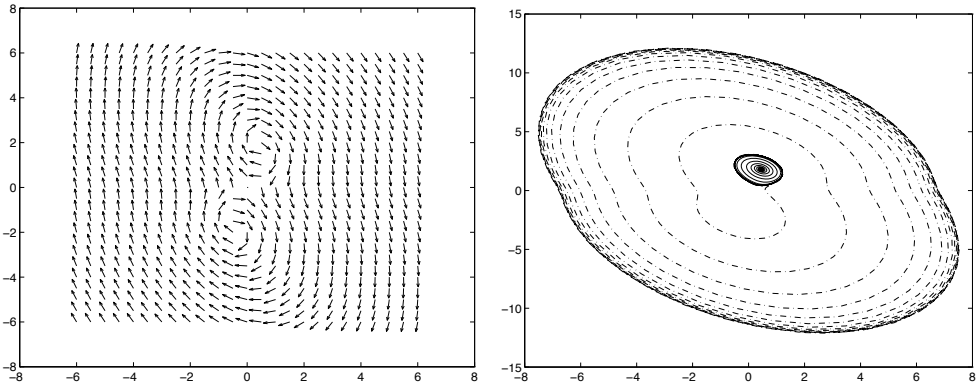
$$A = \begin{bmatrix} 2 & 5 \\ -10 & -3 \end{bmatrix} \qquad b = \begin{bmatrix} -10 \\ 10 \end{bmatrix}. \tag{1.22}$$

Two trajectories created by this system from the two following, slightly different initial conditions,

$$x_0 = \begin{bmatrix} 0 \\ 3.047427004 \end{bmatrix} \qquad x_0 = \begin{bmatrix} 0 \\ 3.047427005 \end{bmatrix} \tag{1.23}$$

are visualised in Figure 1.35(b): One trajectory is attracted to the upper equilibrium point, the other is pulled away into a large limit cycle. Naturally, this can be described as the usual sensitivity to initial conditions that often occur in nonlinear, dynamic systems. The main point is that even though the different regimes may be simple linear systems, if the trajectory doesn't stay in one of the regimes, the overall response of the interpolated system will be hard to interpret in terms of individual regimes. The new dynamic system can only be understood as a whole, hence making the complete system less interpretable than we may initially have hoped for.

Also note that in our simple example we can visualise the vector field for the full model, but even for this simple case it is difficult to predict which trajectory an initial condition would produce. A driving input to the system or a higher-order system would further complicate matters.



(a) The vector field of two stable linear systems.

(b) The response of two slightly different initial conditions. Dashed line goes to limit cycle, solid to an equilibrium point.

Figure 1.35 Vector field, and actual trajectories. Although the vector field appears smooth, the actual trajectories followed can quickly diverge.

Local behaviour around an equilibrium point

Treating the model as a linear parameter varying system,

$$\dot{x} = \sum_{i=1}^{n_{\mathcal{M}}} \rho_i(x)(A_i x + d_i) = A(x)x + d(x) = g(x), \tag{1.24}$$

we can linearise in the usual way to find the local behaviour around an equilibrium point x_0 of this system, $A(x_0)x_0 = -d(x_0)$:

$$\Delta\dot{x} = \frac{dg(x_0)}{dx}\Delta x = \left(A(x_0) + \frac{dA(x_0)}{dx}x_0 + \frac{dd(x_0)}{dx} \right) \Delta x. \tag{1.25}$$

Finding the partial derivatives for our particular structure and substituting we get

$$\Delta\dot{x} = \sum_{i=1}^{n_{\mathcal{M}}} \left((A_i x_0 + d_i)\frac{d\rho_i(x_0)}{dx}^T + \rho_i(x_0)A_i \right) \Delta x. \tag{1.26}$$

It follows immediately that in order to interpret the local model parameters,

$$A(x_0) \quad = \quad \sum_{i=1}^{n_{\mathcal{M}}} A_i \rho_i(x_0), \tag{1.27}$$

as the linearisation of the system (in the classical sense), it is required that

$$\sum_{i=1}^{n_{\mathcal{M}}} (A_i x_0 + d_i)\frac{d\rho_i(x_0)}{dx}^T \quad = \quad 0. \tag{1.28}$$

Notice that

$$\sum_{i=1}^{n_{\mathcal{M}}} \frac{d\rho_i(x_0)}{dx} = 0 \tag{1.29}$$

for any x_0, and by the nature of the weighting functions, $\frac{d\rho_i(x_0)}{dx} \neq 0$ only for regions i that belong to the neighbourhood of x_0:

$$\mathcal{N}(x_0) = \{i \mid \rho_i(x_0) > 0\}.$$

Hence, (1.28) is equivalent to

$$\sum_{i \in \mathcal{N}(x_0)} \left((A_i - A_j)x_0 + (d_i - d_j)\right) \frac{d\rho_i(x_0)}{dx}^T = 0, \tag{1.30}$$

where j is any element in $\mathcal{N}(x_0)$. Hence, it is clear that interpreting $A(x_0)$ as a linearisation in areas with non-negligible basis function gradients requires that neighbouring local model parameters are similar, i.e. the system changes behaviour smoothly, and the granularity of the decomposition is sufficiently fine. Moreover, it is desirable to have smooth weighting functions.

The problems of partitioning dynamic systems outlined in this section highlight the limits of sensible application of partitioning approaches to modelling and control. Local approaches can represent complex nonlinear systems, but if this results in local models which cannot be seen as classical approximations around an equilibrium point, then some of the hoped for interpretation advantages are lost.

1.7 CONCLUDING REMARKS

The purpose of this chapter is twofold. First, as the contributed chapters in the book represent a number of different approaches, each with its own language, insight and interpretations, this chapter should serve as an introduction to the basic concepts and ideas in operating regime and multiple model approaches to nonlinear modelling and control. This is basically about the motivation and the methods for partitioning a nonlinear dynamic system such that it can be represented by simpler subsystems, and can be more easily understood by humans.

The second half of the chapter gave a survey and overview of the current state of the art in terms of available procedures, algorithms, tools and applications. Here we have seen that there is a wide range of competing approaches. The main differences lie in the ease with which the various methods can incorporate external knowledge, and their suitability for use with learning algorithms.

There was also some discussion of methods for the interpretation of multiple model systems. This extends from simple visualisation of the operating regions, to results about model stability, and as in any approach to modelling or control of complex systems, the development of user-friendly tools will be a decisive factor in determining its popularity and success. There is already a great deal of applicable theory in this area, and practising engineers have always instinctively applied similar techniques. We believe that the theory and experience now needs to be translated into intuitively understandable design techniques and associated software tools which can integrate heterogeneous model structures, provide

graphical interpretations and automatically diagnose potential problems with the given design – all the features needed for it to be regularly applied by nn-researchers to real world problems.

REFERENCES

AIZERMANN, M., E. BRAVERMAN AND L. RONZONOER (1964) 'Theoretical foundations of the potential function method in pattern recognition learning'. *Automatika i Telemekhanika* **25**, 147–169.

ALBUS, J. S. (1975) 'A new approach to manipulator control: The cerebellar model articulation controller (CMAC)'. *Trans. ASME. Jnl. Dyn. Sys. Meas. and Control* **63**(3), 220–227.

ALBUS, J. S. (1991) 'Outline of a theory of intelligence?'. *IEEE Transactions on Systems, Man, and Cybernetics* **21**(3), 473–509.

ANTSAKLIS, P. J., K. M. PASSINO AND S. J. WANG (1991) 'An introduction to autonomous control systems'. *IEEE Control Systems Magazine* **11**, 5–13.

AOYAMA, A. AND V. VENKATASUBRAMANIAN (1993) Integrating neural networks with first-principles knowledge for bioreactor modeling and control. Paper 147i, 1993 Annual AIChE Meeting, November, St. Louis.

APKARIAN, P. AND P. GHAINET (1995) 'A convex characterization of gain scheduled H_∞ controllers'. *IEEE Trans. Automatic Control* **40**, 853–864.

APKARIAN, P., P. GHAINET AND G. BECKER (1994) Self-scheduled H_∞ control of linear parameter-varying systems. In 'Proc. American Control Conference'. pp. 856–860.

ÅSTRÖM, K. J. AND B. WITTENMARK (1989) *Adaptive Control*. Addison-Wesley.

ÅSTRÖM, K. J. AND T. HÄGGLUND (1988) *Automatic tuning of PID controllers*. Instrument Society of America.

ÅSTRÖM, K. J. AND T. J. MCAVOY (1992) 'Intelligent control'. *J. Process Control* **2**, 115–125.

ÅSTRÖM, K. J., J. J. ANTON AND K. E. ÅRZÈN (1986) 'Expert control'. *Automatica* **22**, 277–286.

ATHANS, M., D. CASTANOU, K.-P. DUNN, C. S. GREENE, W. H. LEE, N. R. SANDELL AND A. S. WILLSKY (1977) 'The stochastic control of the F-8C aircraft using a multiple model adaptive control (MMAC) method – Part 1: Equilibrium flight'. *IEEE Trans. Automatic Control* **22**, 768–780.

ATKESON, C. G. (1990) Memory-based approaches to approximating continuous functions. In 'Workshop on Nonlinear Modelling and Forecasting, Santa Fe Institute'. Addison-Wesley.

BACK, A. D. AND A. C. TSOI (1991) 'FIR and IIR synapses, a new neural network architecture for time series modeling'. *Neural Computation* **3**(3), 375–385.

BADR, A., W. GHARIEB AND Z. BINDER (1991) Multimodel variable structure control systems. In 'Proc. IEEE Conf. Decision and Control, Brighton, UK'. pp. 2068–2073.

BAILEY, J. E. AND D. F. OLLIS (1986) *Biochemical Engineering Fundamentals*. McGraw-Hill, Singapore.

BAKSHI, B. R. AND G. STEPHANOPOULOS (1993) 'Wave-Net: a multiresolution, hierarchical neural neural network with localized learning'. *AIChE Journal* **39**(1), 57.

BANAN, M. R. AND K. D. HJELMSTAD (1992) Self-organization of architecture by simulated hierarchical adaptive random partitioning. In 'IJCNN, Baltimore'. Vol. III. pp. 823–828.

BANERJEE, A., Y. ARKUN, R. PEARSON AND B. OGUNNAIKE (1995) H_∞ control of nonlinear processes using multiple linear models. In 'Proc. European Control Conference, Rome'. pp. 2671–2676.

BARNES, C., S. BROWN, G. FLAKE, R. JONES, M. O'ROURKE AND Y. C. LEE (1991) Applications of neural networks to process control and modelling. In 'Artificial Neural Networks, Proceedings of 1991 Internat. Conf. Artif. Neur. Nets'. Vol. 1. pp. 321–326.

BARTON, P. I. AND C. C. PANTELIDES (1994) 'Modeling of combined discrete/continuous processes'. *AIChE Journal* **40**, 966–979.

BECKER, G. AND A. PACKARD (1994) 'Robust performance of linear parametrically varying systems using parametrically-dependent linear feedback'. *Systems and Control Letters* **23**, 205–215.

BELLMAN, R. E. (1961) *Adaptive Control Processes*. Princeton University Press. Princeton, NJ.

BENAIM, M. (1994) 'On functional approximation with normalised gaussian units'. *Neural Computation* **6**(1), 319–333.

BENCZE, W. J. AND G. F. FRANKLIN (1995) 'A separation principle for hybrid control system design'. *IEEE Control Systems Magazine* **15**(2), 80–85.

BENTLY, J. L. (1975) 'Multidimensional binary search trees used for associative searching'. *Communications of the ACM* **18**(9), 509–517.

BERNARDO, J. M. AND A. F. M. SMITH (1994) *Bayesian Theory*. John Wiley. New York.

BEZDEK, J. C., C. CORAY, R. GUNDERSON AND J. WATSON (1981a) 'Detection and characterization of cluster substructure. II. Fuzzy c-varieties and complex combinations thereof'. *SIAM J. Applied Mathematics* **40**, 352–372.

BEZDEK, J. C., C. CORAY, R. GUNDERSON AND J. WATSON (1981b) 'Detection and characterization of cluster substructure. I. Linear structure: Fuzzy c-lines'. *SIAM J. Applied Mathematics* **40**, 339–357.

BILLINGS, S. A. AND S. CHEN (1989) 'Extended model set, global data and threshold model identification of severely non-linear systems'. *Int. J. Control* **50**, 1897–1923.

BILLINGS, S. A. AND W. S. G. VOON (1987) 'Piecewise linear identification of non-linear systems'. *Int. J. Control* **46**, 215–235.

BISHOP, C. (1995) *Neural Networks for Pattern Recognition*. Clarendon Press, Oxford.

BISHOP, C. M. (1994) Training with noise is equivalent to Tikhonov Regularization. Submitted for publication.

BLOM, H. A. P. AND Y. BAR-SHALOM (1988) 'The interacting multiple model algorithm for systems with Markovian switching coefficients'. *IEEE Trans. Automatic Control* **33**, 780–783.

BOTTOU, L. AND V. VAPNIK (1992) 'Local learning algorithms'. *Neural Computation* **4**(6), 888–900.

BOX, G. E. P. AND G. M. JENKINS (1970) *Time Series Analysis: Forecasting and Control*. Holden-Day, San Francisco, CA.

BOYD, S., L. EL GHAOUI, E. FERON AND V. BALAKRISHNAN (1994) *Linear Matrix Inequalities in System and Control Theory*. SIAM, Philadelphia.

BOYD, S., V. BALAKRISHNAN, E. FERON AND L. EL GHAOUI (1993) Control system analysis and synthesis via linear matrix inequalities. In 'Proc. American Control Conference'. pp. 2147–2154.

BRANICKY, M. (1994) Stability of switched and hybrid systems. In 'Proceedings of Conference on Decision and Control, pp. 3498-3503'.

BREIMAN, L. AND W. S. MEISEL (1976) 'General estimates of the intrinsic variability of data in nonlinear regression models'. *J. American Statistical Association* **71**, 301–307.

BREIMAN, L., J. H. FRIEDMAN, R. A. OLSHEN AND C. J. STONE (1984) *Classification and Regression Trees*. Wadsworths & Brooks, Monterey, CA.

BROCKETT, R. (1993) 'Hybrid models for motion control systems'. In Hybrid Systems, Lecture Notes in Computer Science.

BROOKS, R. A. (1986) 'A robust layered control system for a mobile robot'. *IEEE Journal of Robotics and Automation*.

BROOMHEAD, D. S. AND D. LOWE (1988) 'Multivariable functional interpolation and adaptive networks'. *Complex Systems* **2**, 321–355.

BROWN, M. AND C. HARRIS (1994) *Neurofuzzy Adaptive Modelling and Control*. Prentice Hall. Hemel Hempstead, UK.

BUNTINE, W. (1994) 'Operations for learning using graphical models'. *Journal of Artificial Intelligence Research*.

CANNON, M. AND J.-J. SLOTINE (1995) 'Space-frequency localized basis function networks for nonlinear system estimation and control'. *Neurocomputing* **9**(3), 293–342.

CAO, S. G., N. W. REES AND G. FENG (1995) Stability analysis and design for a class of continuous-time fuzzy control systems. Preprint.

CARLIN, M. (1992) Radial Basis Function networks and nonlinear data modeling. In 'Neuro–Nimes 92'. pp. 623–633.

CHEN, S. AND S. A. BILLINGS (1994) Neural networks for nonlinear dynamic system modelling and identification. In H. C. J. (Ed.). 'Advances in Intelligent Control'. Taylor & Francis. London. pp. 85–112.

CHEN, S., C. F. N. COWAN AND P. M. GRANT (1991) 'Orthogonal least squares learning algorithm for radial basis function networks'. *IEEE Transactions on Neural Networks*.

CHIZECK, H. J., A. S. WILLSKY AND D. CASTANON (1986) 'Discrete-time Markovian jump linear quadratic optimal control'. *Int. J. Control*, 213–231.

CHOW, C.-M., A. G. KUZNETSOV AND D. W. CLARKE (1995) Using multiple models in predictive control. In 'Proc. European Control Conference, Rome'. pp. 1732–1737.

CLEVELAND, W. S., S. J. DEVLIN AND E. GROSSE (1988) 'Regression by local fitting'. *Journal of Econometrics* **37**, 87–114.

CYROT-NORMAND, D. AND H. D. V. MIEN (1980) Non-linear state-affine identification methods: Application to electrical power plants. In 'Proc. IFAC Symposium on Automatic Control in Power Generation, Distribution and Protection'. pp. 449–462.

DEMPSTER, A. P., N. M. LAIRD AND D. B. RUBIN (1977) 'Maximum likelihood from incomplete data via the EM algorithm'. *J. Royal Statistical Society Series B* **39**, 1–38.

DESOER, C. A. (1970) 'Slowly varying discrete system $x_{t+1} = a_t x_t$'. *Electronics Letters* **6**, 339–340.

DOROFEYUK, A. A., A. D. KASAVIN AND I. S. TORGOVITSKY (1970) Application of automatic classification methods to process identification in industry. In 'Preprints 2nd IFAC Symposium on Identification and Process Parameter Identification, Prague, Czechoslovakia'. p. 4.1.

DRIANKOV, D., H. HELLENDOORN AND M. REINFRANK (1993) *An Introduction to Fuzzy Control*. Springer-Verlag, Berlin.

FARMER, J. D. AND J. J. SIDOROWICH (1987) 'Predicting chaotic time series'. *Physical Review Letters* **59**(8), 845–848.

FEDOROV, V. V. (1972) *Theory of Optimal Experiments*. Academic Press. New York.

FOSS, B. A. AND T. A. JOHANSEN (1993) On local and fuzzy modelling. In '3rd Int. Conf. on Industrial Fuzzy Control and Intelligent Systems'. Houston, Texas.

FOSS, B. A., T. A. JOHANSEN AND A. V. SØRENSEN (1995) 'Nonlinear predictive control using local models – applied to a batch fermentation process'. *Control Engineering Practice* **3**, 389–396.

FRIEDMAN, J. H. (1991) 'Multivariate Adaptive Regression Splines'. *Annals of Statistics* **19**, 1–141.

FRITZKE, B. (1994) Supervised learning with growing cell structures. In J. D. Cowan, G. Tesauro and J. Alspector (Eds.). 'Advances in Neural Information Processing Systems 6'. Morgan Kaufmann Publishers. San Franciso, CA. pp. 255–262.

GANGSAAS, D., K. V. BRUCE, J. D. BLIGHT AND U. L. LY (1986) 'Applications of modern synthesis of aircraft control. Three case studies'. *IEEE Trans. Automatic Control* **31**, 995–1014.

GARCIA, C. E., D. M. PRETT AND M. MORARI (1989) 'Model predictive control: Theory and practice – A survey'. *Automatica* **25**, 335–348.

GAWTHROP, P. J. (1995) Continuous-time local state local model networks. In 'Proc. IEEE Conf. Systems, Man and Cybernetics, Vancouver, Canada'.

GAWTHROP, P. J. (1996) Continuous-time local model networks. In R. W. Żbikowski and K. Kunt (Eds.). 'Neural Adaptive Control Technology'. Vol. 1. World Scientific.

GEMAN, S., E. BIENENSTOCK AND R. DOURSAT (1992) 'Neural networks and the bias/variance dilemma'. *Neural Computation* **4**(1), 1–58.

GENDRON, S., M. PERRIER, J. BARETTE, M. AMJAD, A. HOLKO AND N. LEGAULT (1993) 'Deterministic adaptive control of SISO processes using model weighting adaptation'. *Int. J. Control* **58**, 1105–1123.

GIROSI, F., M. JONES AND T. POGGIO (1993) Priors, stabilizers and basis functions: from regularization to radial, tensor and additive splines. MIT AI Memo 1430. MIT.

GOODWIN, G. C. AND R. L. PAYNE (1977) *Dynamic System Identification: Experiment Design and Data Analysis*. Academic Press, New York.

GREENE, C. S. AND A. S. WILLSKY (1980) An analysis of the multiple model adaptive control algorithm. In 'Proc. of the IEEE Conference on Decision and Control'. pp. 1142–1145.

GRIFFITHS, B. E. AND K. A. LOPARO (1985) 'Optimal control of jump linear quadratic Gaussian systems'. *Int. J. Control* **42**, 791–819.

HABER, R., I. VAJK AND L. KEVICZKY (1982) Nonlinear system identification by "linear" systems having signal-dependent parameters. In 'Preprints 6th IFAC Symp. on Identification and System Parameter Identification'. Washington D.C.. pp. 421–426.

HALME, A. (1989) Expert system approach to recognize the state of fermentation and to diagnose faults in bioreactors. In N. M. Fish, R. I. Fox and N. F. Thornhill (Eds.). 'Computer Applications in Fermentation Technology'. Elsevier.

HARRIS, C., C. G. MOORE AND M. BROWN (1993) *Intelligent Control: Aspects of Fuzzy Logic and Neural Nets*. World Scientific.

HARTMAN, E. AND J. D. KEELER (1991) 'Predicting the future: Advantages of semilocal units'. *Neural Computation* **3**(4), 566–578.

HASTIE, T. AND R. TIBSHIRANI (1994) Discriminant analysis by Gaussian mixtures. Technical report. AT&T Bell Laboratories. Murray Hill, NJ.

HASTIE, T. J. AND R. J. TIBSHIRANI (1990) *Generalized Additive Models*. Monographs on Statistics and Applied Probability 43. Chapman and Hall. London.

HATHAWAY, R. J. AND J. C. BEZDEK (1993) 'Switching regression models and fuzzy clustering'. *IEEE Trans. Fuzzy Systems* **1**, 195–204.

HILHORST, R. A. (1992) Supervisory Control of Mode-Switch Processes. PhD thesis. University of Twente – Electrical Engineering Department.

HILHORST, R. A., J. VAN AMERONGEN AND P. LÖHNBERG (1991) Intelligent adaptive control of mode-switch processes. In 'Proc. IFAC International Symposium on Intelligent Tuning and Adaptive Control, Singapore'.

HLAVÁCKOVÁ, K. AND R. NERUDA (1993) 'Radial Basis Function networks'. *Neural Network World* **1**, 93–101.

HOLDEN, S. B. (1994) On the Theory of Generalization and Self-Structuring in Linearly Weighted Connectionist Networks. PhD thesis. Cambridge University.

HUNT, K. J., D. SBARBARO, R. ŻBIKOWSKI AND P. J. GAWTHROP (1992) 'Neural networks for control systems – A survey'. *Automatica* **28**, 1083–1112.

HUNT, K. J., J. C. KALKKUHL, H. FRITZ AND T. A. JOHANSEN (1996a) 'Constructive empirical modelling of longitudinal vehicle dynamics using local model networks'. *Control Engineering Practice* **4**(2), 167–178.

HUNT, K. J., R. HAAS AND M. BROWN (1995) 'On the functional equivalence of fuzzy inference systems and spline-based networks'. *Int. J. Neural Systems* **6**, 171–184.

HUNT, K. J., R. HAAS AND R. MURRAY-SMITH (1996b) 'Extending the functional equivalence of radial basis function networks and fuzzy inference systems'. *Trans. IEEE on Neural Networks*.

HUTCHINSON, J. M. (1994) Radial Basis Function Approach to Financial Time Series Analysis. PhD thesis. Massachusetts Institute of Technology, Dept. EECS.

HYDE, R. A. AND K. GLOVER (1993) 'The application of scheduled H_∞ controllers to a VSTOL aircraft'. *IEEE Trans. Automatic Control* **38**, 1021–1039.

ISAKSSON, A. J., L. LJUNG AND J.-E. STRÖMBERG (1991) On recursive construction of trees as models of dynamical systems. In '30th Conf. on Decision & Control, Brighton'. pp. 1686–1687.

ISHIGAME, A., T. FURUKAWA, S. KAWAMOTO AND T. TANIGUCHI (1993) 'Sliding mode controller design based on fuzzy inference for nonlinear systems'. *IEEE Trans. Ind. Electronics* **40**, 64–70.

ISIDORI, A. (1989) *Nonlinear Control Systems*. Springer-Verlag.

IVAKHNENKO, A. G. (1970) 'Heuristic self-organization in problems of engineering cybernetics'. *Automatica* **6**, 207–219.

IVAKHNENKO, A. G. (1971) 'Polynomial theory of complex systems'. *IEEE Transactions on Systems, Man, and Cybernetics* **1**(4), 364–378.

JACOBS, R. A. (1995) 'Adaptive mixtures of local experts'. *Neural Computation* **7**, 867–888.

JACOBS, R. A. AND M. I. JORDAN (1993) 'Learning piecewise control strategies in a modular neural network architecture'. *IEEE Trans. Systems, Man, and Cybernetics* **23**, 337–345.

JACOBS, R. A., M. I. JORDAN, S. J. NOWLAN AND G. E. HINTON (1991) 'Adaptive mixtures of local experts'. *Neural Computation* **3**(1), 79–87.

JAMSHIDI, M. AND A. TITLI (1995) Stability of fuzzy control system via interval matrix method. In 'Proc. 3rd IEEE Mediterranean Symp. New Directions in Control and Automation, Cyprus'.

JANG, J.-S. R. AND C.-T. SUN (1993a) 'Functional equivalence between radial basis function networks and fuzzy inference systems'. *IEEE Trans. Neural Networks* **4**, 156–159.

JANG, J. S. R. AND C. T. SUN (1993b) 'Functional equivalence between radial basis function networks and fuzzy inference systems'. *IEEE Transactions on Neural Networks* **4**(1), 156–158.

JENSEN, F. V. (1996) *An introduction to Bayesian Networks*. UCL Press.

JIANG, J. (1994) 'Optimal gain scheduling controller for a diesel engine'. *IEEE Control Systems Magazine* **14**(4), 42–48.

JOHANSEN, T. A. (1994a) Adaptive control of MIMO non-linear systems using local ARX models and interpolation. In 'IFAC ADCHEM 94'. Kyoto, Japan.

JOHANSEN, T. A. (1994b) 'Fuzzy model based control: Stability, robustness, and performance issues'. *IEEE Trans. Fuzzy Systems* **2**, 221–234.

JOHANSEN, T. A. (1994c) Operating regime based process modeling and identification. Technical Report 94-109-W. Dr. Ing. thesis. Department of Engineering Cybernetics, Norwegian Institute of Technology, Trondheim, Norway. http://www.itk.unit.no/ansatte/Johansen_Tor.Arne/dring_taj.ps.gz.

JOHANSEN, T. A. (1995) On the optimality of the Takagi-Sugeno-Kang fuzzy inference mechanism. Accepted for the 4th IEEE Conf. on Fuzzy Systems, Yokohama, Japan.

JOHANSEN, T. A. AND B. A. FOSS (1993a) 'Constructing NARMAX models using ARMAX models'. *Int. J. Control* **58**, 1125–1153.

JOHANSEN, T. A. AND B. A. FOSS (1993b) State-space modeling using operating regime decomposition and local models. In 'Preprints 12th IFAC World Congress, Sydney, Australia'. Vol. 1. pp. 431–434.

JOHANSEN, T. A. AND B. A. FOSS (1995) 'Identification of non-linear system structure and parameters using regime decomposition'. *Automatica* **31**, 321–326.

JONES, R. D. AND CO-WORKERS (1991) Nonlinear adaptive networks: A little theory, a few applications. Technical Report 91-273. Los Alamos National Lab., NM.

JONES, R. D., Y. C. LEE, C. W. BARNES, G. W. FLAKE, K. LEE, P. S. LEWIS AND S. QIAN (1989) Function approximation and time series prediction with neural networks. Technical Report 90-21. Los Alamos National Lab., New Mexico.

JORDAN, M. AND R.A. JACOBS (1994) 'Hierarchical mixtures of experts and the EM algorithm'. *Neural Computation* **6**, 181–214.

JORDAN, M. I. AND R. A. JACOBS (1991) Hierarchies of adaptive experts. In J. E. Moody, S. J. Hanson and R. P. Lippmann (Eds.). 'Advances in Neural Information Processing Systems 4'. Morgan Kaufmann Publishers. San Mateo, CA.

KASAVIN, A. D. (1972) 'Adaptive piecewise approximation algorithms in the identification problem'. *Automation and Remote Control* **33**, 2001–2006.

KAVLI, T. (1993) 'ASMOD – an algorithm for adaptive spline modelling of observation data'. *International Journal of Control* **58**(4), 947–967.

KELLET, M. G. (1991) Continuous scheduling of H_∞ controllers for a MS760 Paris aircraft. In P. H. Hammond (Ed.). 'Robust Control System Design Using H_∞ and Related Methods'. Institute of Measurement and Control, London.

KHALIL, H. (1992) *Nonlinear Systems*. Macmillan.

KOLMOGOROFF, A. (1936) 'Sulla theoria di Volterra della lotta per l'esistenza'. *G. Istit. Ital. Degli Attuari* **7**, 74–80.

KONSTANTINOV, K. AND T. YOSHIDA (1989) 'Physiological state control of fermentation processes'. *Biotechnology and Bioengineering* **33**, 1145–1156.

KONSTANTINOV, K. B. AND T. YOSHIDA (1991) 'A knowledge-based pattern recognition approach for real-time diagnosis and control of fermentation processes as variable structure plants'. *IEEE Trans. Systems, Man, and Cybernetics* **21**(4), 908–914.

KOSKO, B. (1994) 'Fuzzy systems as universal approximators'. *IEEE Trans. Computers* **43**, 1329–1333.

KRAMER, M. A., M. L. THOMPSON AND P. M. PHAGAT (1992) Embedding theoretical models in neural networks. In 'Proc. American Control Conference, Chicago, Il.'. pp. 475–479.

KUIPERS, B. AND K. ÅSTRÖM (1994) 'The composition and validation of heterogeneous control laws'. *Automatica* **30**, 233–249.

KURCOVA, V. (1992) Universal approximation using feedforward neural networks with Gaussian Bar units. In 'Proc. ECAI'92, Vienna'. pp. 193–197.

LAINIOTIS, D. (1976a) 'Partitioning: A unifying framework for adaptive systems. I: Estimation'. *Proc. IEEE* **64**, 1126–1143.

LAINIOTIS, D. (1976b) 'Partitioning: A unifying framework for adaptive systems. II: Control'. *Proc. IEEE* **64**, 1144–1161.

LANE, S. H., D. A. HANDELMAN AND J. J. GELFAND (1991) Higher order CMAC neural networks - theory and practice. In 'Proc. American Control Conference, Boston, USA'. pp. 1579–1585.

LANE, S. H., D. A. HANDELMAN AND J. J. GELFAND (1992) 'Theory and development of higher-order CMAC neural networks'. *IEEE Control System Magazine* **12**(2), 23–30.

LANGARI, G. AND M. TOMIZUKA (1990) Stability of fuzzy linguistic control systems. In 'Proc. 29th IEEE Conf. Decision and Control, Honolulu, Hawaii'. pp. 2185–2190.

LANGARI, R. AND H. R. BERENJI (1992) Fuzzy logic in control engineering. In D. A. White and D. A. Sofge (Eds.). 'Handbook of Intelligent Control'. Van Nostrand Reinhold, New York.

LAWRENCE, D. A. AND W. J. RUGH (1990) 'On a stability theorem for nonlinear systems with slowly-varying input'. *IEEE Trans. Automatic Control* **35**, 860–864.

LEE, C. C. (1990a) 'Fuzzy logic in control systems: Fuzzy logic controller – Part I'. *IEEE Trans. Systems, Man, and Cybernetics* **20**, 404–418.

LEE, C. C. (1990b) 'Fuzzy logic in control systems: Fuzzy logic controller – Part II'. *IEEE Trans. Systems, Man, and Cybernetics* **20**, 419–435.

LEE, S. AND M. R. KIL (1991) 'A Gaussian potential function network with hierarchically self-organizing learning'. *Neural Networks* **4**, 207–224.

LIND, M. (1988) Representing goals and functions of complex systems: An introduction to multilevel flow modelling. Technical Report, Institute of Automatic Control Systems, Technical University of Denmark.

LIND, M. (1991) Representations and abstractions for interface design using multilevel flow modelling. In G. R. S. Weir and J. R. Alty (Eds.). 'Human-Computer Interaction and Complex Systems'. Academic Press, London. pp. 223–244.

LJUNG, L. (1987) *System Identification — Theory for the User*. Prentice Hall. Englewood Cliffs, New Jersey, USA.

LUND, E. J., J. G. BALCHEN AND B. A. FOSS (1991) Multiple model estimation with inter-residual distance feedback. In 'Preprints IFAC/IFORS Symposium on Identification and System Parameter Estimation, Budapest'. pp. 1512–1517.

LUND, E. J., J. G. BALCHEN AND B. A. FOSS (1992) Monitoring processes with structurally different operating regimes. In 'Preprints IFAC Symposium on On-line Fault Detection and Supervision in the Chemical Process Industries, Newark, Delaware'. pp. 126–131.

MACKAY, D. J. C. (1991) Bayesian Methods for Adaptive Models. PhD thesis. California Institute of Technology, Pasadena, CA.

MARCOS, M. AND F. ARTAZA (1995) A methodology for the design of rule-based supervisors. In 'Proc. European Control Conference'. pp. 1619–1624.

MARIN, J. P. AND A. TITLI (1995) On the use of describing function to analyse and design fuzzy controllers. In 'Proc. IFSA-95, Sao Paulo, Brazil'.

MASON, J. C. AND P. C. PARKS (1992) Selection of neural network architectures: Some approximation theory guidelines. In 'In: K. Warwick, G. W. Irwin, K. J. Hunt (Eds), Neural networks for control and systems'. Peter Peregrinus. pp. 151–180.

MAVROVOUNIOTIS, M. L. AND S. CHANG (1992) 'Hierarchical neural networks'. *Comp. Chem. Engr.* **16**, 347–369.

MCLACHLAN, G. AND K.E. BASFORD (1988) *Mixture Models: Inference and Applications to Clustering*. Marcel Dekker.

MCMICHAEL, D. W. (1995) Bayesian growing and pruning strategies for MAP-optimal estimation of Gaussian mixture models. In '4th IEE Intern. Conf. on Artificial Neural Networks'. pp. 364–368.

MESAROVIC, M. D., D. MACKO AND Y. TAKAHARA (1970) *Theory of Hierarchical, Multilevel Systems*. Academic Press.

MILLER, D. AND K. ROSE (1996) 'Hierarchical, unsupervised learning with growing via phase transitions'. *Neural Computation* **8**(2), 425–450.

MOODY, J. AND C. DARKEN (1989) 'Fast-learning in networks of locally-tuned processing units'. *Neural Computation* **1**, 281–294.

MORSE, A. S. (1995) Control using logic-based switching. In A. Isidori (Ed.). 'Trends in Control'. Springer-Verlag, London. pp. 69–114.

MURRAY-SMITH, R. (1992) A Fractal Radial Basis Function network for modelling. In 'Inter. Conf. on Automation, Robotics and Computer Vision, Singapore'. Vol. 1. pp. NW–2.6.1–NW–2.6.5.

MURRAY-SMITH, R. (1994) A Local Model Network Approach to Nonlinear Modelling. PhD Thesis. Department of Computer Science, University of Strathclyde, Glasgow, Scotland.

MURRAY-SMITH, R. AND H. GOLLEE (1994) A constructive learning algorithm for local model networks. In 'Proc. IEEE Workshop on Computer-intensive methods in control and signal processing, Prague, Czech Republic'. pp. 21–29.

MURRAY-SMITH, R. AND K. J. HUNT (1995) Local model architectures for nonlinear modelling and control. In K. J. Hunt, G. R. Irwin and K. Warwick (Eds.). 'Neural Network Engineering in Dynamic Control Systems'. Advances in Industrial Control. Springer-Verlag. pp. 61–82.

NÆS, T. (1991) 'Multivariate calibration when data are split into subsets'. *J. Chemometrics* **5**, 487–501.

NÆS, T. AND T. ISAKSSON (1991) 'Splitting of calibration data by clustering analysis'. *J. Chemometrics* **5**, 49–65.

NAKAMORI, Y. AND M. RYOKE (1994) 'Identification of fuzzy prediction models through hyperellipsoidal clustering'. *IEEE Trans. Systems, Man, and Cybernetics* **24**, 1153–1173.

NAKAMORI, Y., K. SUZUKI AND T. YAMANAKA (1992) A new design of a fuzzy model predictive control system for nonlinear processes. In T. Terano, M. Sugeno, M. Mukaidono and K. Shigemasu (Eds.). 'Fuzzy Engineering Toward Human Friendly Systems'. IOS Press, Amsterdam. pp. 788–799.

NARENDRA, K. S. AND A. M. ANNASWAMY (1989) *Stable Adaptive Systems*. Prentice-Hall, Englewood Cliffs, NJ.

NARENDRA, K. S. AND K. PARTHASARATHY (1990) 'Identification and control of dynamical systems using neural networks'. *IEEE Transaction on Neural Networks* **1**(1), 4–27.

NARENDRA, K. S., J. BALAKRISHNAN AND M. K. CILIZ (1995) 'Adaptation and learning using multiple models, switching and tuning'. *IEEE Control Systems Magazine* **15**(3), 37–51.

NICHOLS, R. A., R. T. REICHERT AND W. J. RUGH (1993) 'Gain scheduling for H_∞ controllers for a MS760 Paris aircraft'. *IEEE Trans. Control Systems Technology* **1**, 69–75.

NOWLAN, S. J. (1991) Soft Competitive Adaptation: Neural Network Learning Algorithms based on Fitting Statistical Mixtures. CMU-CS-91-126. School of Computer Science, Carnegie Mellon University. Pittsburgh, PA.

OLAFSSON, J. F. (1995) Fuzzy control of water desalination plants. Diploma thesis, Department of Engineering Cybernetics, Norwegian Institute of Technology, Norway.

OMOHUNDRO, S. M. (1987) 'Efficient algorithms with neural network behavior'. *J. Complex Systems* **1**, 273–347.

OMOHUNDRO, S. M. (1991) Bumptrees for efficient function, constraint and classification learning. In R. P. Lippmann, J. E. Moody and D. S. Touretzky (Eds.). 'Advances in Neural Information Processing Systems 3'. Morgan Kaufmann Publishers. San Francisco, CA. pp. 693–699.

OPOITSEV, V. I. (1970) 'Identification of static plants by means of piecewise linear functions'. *Automation and Remote Control* **31**, 809–815.

PACKARD, A. (1994) 'Gain scheduling via linear fractional transformations'. *Systems and Control Letters* **22**(1), 79–92.

PALM, R., D. DRIANKOV AND U. REHFUESS (1995) Lyapunov linearization based design of Takagi-Sugeno fuzzy controllers. Preprint.

PAO, Y.-H. (1992) 'Functional link net computing'. *Computer* pp. 76–79.

PARK, J. AND I. W. SANDBERG (1991) 'Universal approximation using radial-basis-function networks'. *Neural Computation* **3**(2), 246–257.

PARK, J. AND I. W. SANDBERG (1993) 'Approximation and Radial-Basis-Function networks'. *Neural Computation* **5**(3), 305–316.

PEARL, J. (1988) *Probabilistic Reasoning in Intelligent Systems*. Morgan Kaufmann.

PEREV, K., C. JACOBSON AND B. SHAFAI (1995) 'An algorithm for identification of smooth nonlinear systems based on parameterized linearization families'. *Int. J. Control* **61**, 1013–1043.

PERRONE, M. P. AND L. N. COOPER (1993) When networks disagree: ensemble methods for hybrid neural networks. In R. J. Mammone (Ed.). 'Artificial Neural Networks for Speech and Vision'. Chapman and Hall. London.

PETRIDIS, Y. AND A. KEHAGIAS (1996) 'Modular neural networks for map classification of time series and the partition algorithm'. *IEEE Trans. Neural Networks* **7**, 73–86.

PETTIT, N. AND P. E. WELLSTEAD (1993) Piecewise-linear systems with logic control: A state-space representation. In 'Proc. European Control Conference, Groningen'. pp. 1581–1586.

PLATT, J. (1991) 'A Resource-Allocating Network for function interpolation'. *Neural Computation* **3**(2), 213–225.

POGGIO, T. AND F. GIROSI (1990) Networks for approximation and learning. In 'Proceedings of the IEEE'. Vol. 78. pp. 1481–1497.

POTTMANN, M., H. UNBEHAUEN AND D. E. SEBORG (1993) 'Application of a general multimodel approach for identification of highly nonlinear processes – a case study'. *Int. J. Control* **57**(1), 97–120.

POWELL, M. J. D. (1987) Radial Basis Functions for multivariable interpolation: A review. In 'Algorithms for Approximation'. Clarendon Press. Oxford. pp. 143–167.

PRIESTLEY, M. B. (1988) *Non-linear and Non-stationary Time Series Analysis*. Academic Press.

PSICHOGIOS, D. C. AND L. H. UNGAR (1992) 'A hybrid neural network – first principles approach to process modeling'. *AIChE J.* **38**, 1499–1511.

QIN, S. J. AND G. BORDERS (1994) 'A multiregion fuzzy logic controller for nonlinear process control'. *IEEE Trans. Fuzzy Systems* **2**, 74–89.

QUINLAN, J. (1992) Learning with continuous classes. In 'Australian AI Conf'. pp. 343–348.

QUINLAN, J. (1993) *C4.5 Programs for Machine Learning*. Morgan Kaufmann.

RAJBMAN, N. S., A. A. DOROFEYUK AND A. D. KASAVIN (1981) Identification of nonlinear processes by piecewise approximation. In P. Eykhoff (Ed.). 'Trends and Progress in System Identification'. Pergamon Press, Oxford. pp. 185–238.

RASMUSSEN, J. (1986) *Information Processing and Human-Machine Interaction: An Approach to Cognitive Engineering*. North-Holland.

RAWLINGS, J. B., E. S. MEADOWS AND K. R. MUSKE (1994) Nonlinear model predictive control: A tutorial and survey. In 'Preprints IFAC Symposium ADCHEM, Kyoto, Japan'. pp. 203–214.

REICHERT, R. T. (1992) 'Dynamic scheduling of modern-robust-control autopilot designs for missiles'. *IEEE Control Systems Magazine* **12**(5), 35–42.

RÖSCHEISEN, M., R. HOFMANN AND V. TRESP (1992) Neural control for rolling mills: Incorporating domain theories to overcome data deficiency. In J. E. Moody, S. J. Hanson and R. P. Lippmann (Eds.). 'Advances in Neural Information Processing Systems 4'. Morgan Kaufmann. San Mateo, CA. pp. 659–666.

RUGH, W. (1991) 'Analytical framework for gain scheduling'. *IEEE Control Systems Magazine* **11**(1), 79–84.

RUGH, W. J. (1981) *Nonlinear System Theory: The Volterra-Wiener Approach*. Johns Hopkins Press.

SANGER, T. D. (1991a) 'A tree-structured adaptive network for function approximation in high-dimensional spaces'. *IEEE Trans. on Neural Networks* **2**(2), 285–293.

SANGER, T. D. (1991b) 'A tree-structured algorithm for reducing computation in networks with separable basis functions'. *Neural Computation* **3**(1), 67–78.

SASTRY, S. S. AND A. ISIDORI (1989) 'Adaptive control of linearizable systems'. *IEEE Trans. Automatic Control* **34**, 1123–1131.

SBARBARO, D. G. (1992) A comparative study of different learning algorithms for Gaussian networks. In 'IFAC symposium on Intelligent Components and Intruments for Control Applications. Malaga, Spain'. pp. 301–305.

SCHAAL, S. AND C. ATKESON (1994) 'Robot juggling: Implementation of memory based learning'. *IEEE Control Systems Magazine* **14**(1), 57–71.

SCOTT, G. M. AND W. H. RAY (1993) 'Creating efficient nonlinear neural network process models that allow model interpretation'. *J. Process Control* **3**, 163–178.

SHAFER, G. AND PEARL, J. (EDS.) (1990) *Readings in Uncertain Reasoning*. Morgan Kaufmann. San Mateo, CA.

SHAHRUZ, S. M. AND S. BEHTASH (1992) 'Design of controllers for linear parameter-varying systems by gain scheduling technique'. *J. Mathematical Analysis and Applications* **168**, 195–217.

SHAMMA, J. S. AND M. ATHANS (1990) 'Analysis of gain scheduled control for nonlinear plants'. *IEEE Trans. Automatic Control* **35**, 898–907.

SHAMMA, J. S. AND M. ATHANS (1992) 'Gain scheduling: Potential hazards and possible remedies'. *IEEE Control Systems Magazine* **12**(3), 101–107.

SHORTEN, R. (1996) A Study of Hybrid Dynamical Systems with Application to Automotive Control. PhD thesis. Department of Electrical Engineering, University College Dublin, Republic of Ireland, June 1996.

SIMON, H. A. (1976) *Administrative Behavior 3rd ed.*. Free Press.

SIMONYI, K. E., N. K. LOH AND R. E. HASKELL (1989) An application of expert hierarchical control to piecewise linear systems. In 'Proc. 28th IEEE Conf. Decison and Control, Tampa, FL.'. pp. 822–827.

SJÖBERG, J., H. HJALMARSSON AND L. LJUNG (1994) Neural networks in system identification. In 'Preprints 10th IFAC Symp. System Identification, Copenhagen'. Vol. 2. pp. 49–72.

SKEPPSTEDT, A., L. LJUNG AND M. MILLNERT (1992) 'Construction of composite models from observed data'. *Int. J. Control* **55**(1), 141–152.

SØRHEIM, E. (1990) 'A combined network architecture using ART2 and back propagation for adaptive estimation of dynamical processes'. *Modeling, Identification and Control* **11**, 191–199.

SÖDERMAN, U., J. TOP AND J.-E. STRÖMBERG (1993) The conceptual side of mode switching. In 'Proc. IEEE Conf. Systems, Man, and Cybernetics, Le Touquet, France'. pp. 245–250.

SPECHT, D. F. (1991) 'A general regression neural network'. *IEEE Transactions on Neural Networks*.

ŠTECHA, J. AND V. HAVLENA (1994) Parameter tracking with alternative noise models. In 'Proc. IEEE Workshop CMP'94'. pp. 171–175.

STEIN, G. (1980) Adaptive flight control — a pragmatic view. In K. S. Narandra and R. V. Monopoli (Eds.). 'Applications of Adaptive Control'. Academic Press, New York.

STOKBRO, K., D. K. UMBERGER AND J. A. HERTZ (1990a) 'Exploiting neurons with localized receptive fields to learn chaos'. *Complex Systems* **4**(3), 603–622.

STOKBRO, K., J. A. HERTZ AND D. K. UMBERGER (1990b) 'Exploiting neurons with localized receptive fields to learn chaos'. *J. Complex Systems* **4**, 603.

STRÖMBERG, J.-E., F. GUSTAFSSON AND L. LJUNG (1991a) Trees as black-box model structures for dynamical systems. In 'Proc. European Control Conference, Grenoble'. pp. 1175–1180.

STRÖMBERG, J.-E., F. GUSTAFSSON AND L. LJUNG (1991b) Trees as black-box model structures for dynamical systems. In 'European Control Conference, Grenoble'. pp. 1175–1180.

SU, H.-T., N. BHAT AND T. J. MCAVOY (1992) Integrated neural networks with first principles models for dynamic modeling. In Preprints IFAC DYCORD+ '92, College Park, Maryland.

SUGENO, M. AND G. T. KANG (1986) 'Fuzzy modelling and control of multilayer incinerator'. *Fuzzy Sets and Systems* **18**, 329–346.

SUGENO, M. AND G. T. KANG (1988) 'Structure identification of fuzzy model'. *Fuzzy Sets and Systems* **26**, 15–33.

SUGENO, M. AND T. YASUKAWA (1993) 'A fuzzy-logic-based approach to qualitative modeling'. *IEEE Trans. Fuzzy Systems* **1**, 7–31.

SUZUKI, K., Y. NAKA AND K. BITO (1992) Fuzzy multi-model control of a high-purity distillation system. In T. Terano, M. Sugeno, M. Mukaidono and K. Shigemasu (Eds.). 'Fuzzy Engineering Toward Human Friendly Systems'. IOS Press, Amsterdam. pp. 684–693.

SWORDER, D. D. (1969) 'Feedback control of a class of linear systems with jump parameters'. *IEEE Trans. Automatic Control* **14**, 9–14.

TAKAGI, T. AND M. SUGENO (1985) 'Fuzzy identification of systems and its applications for modeling and control'. *IEEE Trans. on Systems, Man and Cybernetics* **15**(1), 116–132.

TANAKA, K. AND M. SANO (1994) 'A robust stabilizing problem of fuzzy controller systems and its application to backing up control of a truck-trailer'. *IEEE Trans. Fuzzy Systems* **2**, 119–134.

TANAKA, K. AND M. SUGENO (1992) 'Stability analysis and design of fuzzy control systems'. *Fuzzy Sets and Systems* **45**, 135–156.

THOMPSON, M. L. AND M. A. KRAMER (1994) 'Modeling chemical processes using prior knowledge and neural networks'. *AIChE J.* **40**, 1328–1340.

TITTERINGTON, D., A.F.M. SMITH AND U.E. MAKOV (1985) *Statistical Analysis of Finite Mixture Distributions*. John Wiley & Sons. Chichester.

TONG, H. (1990) *Non-linear Time Series: A Dynamical System Approach*. Oxford University Press. Oxford Statistical Science Series 6.

UTKIN, V. I. (1977) 'Variable structure systems with sliding modes'. *IEEE Trans. Automatic Control* **22**, 212–222.

UTKIN, V. I. (1992) *Sliding Modes in Control and Optimization*. Springer-Verlag, Berlin.

WAHBA, G. (1990) Spline models for observation data. In 'Regional Conference Series in Applied Mathematics'. SIAM. Philadelphia, PA.

WAHBA, G. (1992) Multivariate function and operator estimation, based on smoothing splines and reproducing kernels. In M. Casdagli and S. Eubank (Eds.). 'Nonlinear Modeling and Forecasting, SFI Studies in the Sciences of Complexity'. Vol. XII. Addison-Wesley.

WAHLBERG, B. (1991) 'System identification using Laguerre models'. *IEEE Trans. Automatic Control* **36**, 551–562.

WANG, J. AND W. J. RUGH (1987) 'Feedback linearization families for nonlinear systems'. *IEEE Trans. Automatic Control* **32**, 935–940.

WANG, L.-X. (1993) 'Stable adaptive fuzzy control of nonlinear systems'. *IEEE Trans. Fuzzy Systems* **1**, 146–155.

WANG, L.-X. (1994) *Adaptive Fuzzy Systems and Control: Design and Stability Analysis*. Prentice Hall.

WATERHOUSE, S., D. MACKAY AND T. ROBINSON (1996) Bayesian methods for mixtures of experts. Neural Information Processing Systems, **8**, 351–357, MIT press. Cambridge, MA.

WATERHOUSE, S. R. AND A. J. ROBINSON (1996) Constructive algorithms for hierarchical mixtures of experts. Submitted to NIPS'95. Neural Information Processing Systems, **8**, 351–357, MIT press. Cambridge, MA.

WHATLEY, M. J. AND D. C. POTT (1984) 'Adaptive gain improves reactor control'. *Hydrocarbon Processing* pp. 75–78.

WHITTAKER, J. (1990) *Graphical Models in Applied Multivariate Statistics*. John Wiley & Sons.

WIENER, N. (1958) *Nonlinear problems in random theory.*. The Technology Press, MIT and John Wiley and Sons Inc. New York.

YAVNAI, A. (1993) Arbitration network: A new approach for combining reflexive behaviors and reasoning-driven behaviors in intelligent autonomous systems. In 'Intelligent Autonomous Systems 3'. IOS Press, Washington. pp. 428–438.

YOSHINARI, Y., W. PEDRYCZ AND K. HIROTA (1993) 'Construction of fuzzy models through clustering techniques'. *Fuzzy Sets and Systems* **54**, 157–165.

ZADEH, L. A. (1973) 'Outline of a new approach to the analysis of complex systems and decision processes'. *IEEE Trans. Systems, Man, and Cybernetics* **3**, 28–44.

ZENG, X.-J. AND M. G. SINGH (1994) 'Approximation theory of fuzzy systems – SISO case'. *IEEE Trans. Fuzzy Systems* **2**, 162–176.

ZHANG, X.-C., A. VISALA, A. HALME AND P. LINKO (1994) 'Functional state modelling approach for bioprocesses: Local models for aerobic yeast growth processes'. *J. Process Control* **4**, 127–134.

ZHAO, J., V. WERTZ AND R. GOREZ (1995a) Design a stabilizing fuzzy and/or non-fuzzy state feedback controller using LMI method. In 'Proc. European Control Conference, Rome'. pp. 1201–1206.

ZHAO, J., V. WERTZ AND R. GOREZ (1995b) Linear TS fuzzy model based robust stabilizing controller design. In 'IEEE Conf. Decision and Control, New Orleans'.

ZHAO, Z.-Y., M. TOMIZUKA AND S. ISAKA (1993) 'Fuzzy gain scheduling of PID controllers'. *IEEE Trans. Systems, Man, and Cybernetics* **23**, 1392–1398.