2

Linear System Representation: Models and Equivalence

In this chapter, models used for system simulation and model-based control design are presented. The treatment is focused on linear systems and the linearised approximation of non-linear systems due to the necessary limitation in space. As disturbance rejection is a key objective in many control applications, disturbance models are also introduced.

2.1 Introduction: Objectives of Modelling

In the previous chapter, it was shown that the "ideal" control requires the inversion of the plant model. Thus, any control structure will take advantage of a good process model to compute the control action, even if the model is not perfect.

In this book, process models are tools for designing the control system, for simulating the behaviour of the controlled system, and for analysing its properties and evaluating the goals' achievement. Thus, their level of detail, range of validity and presentation will be determined by their use.

For each application, control goal or design methodology, a given model will be more or less suitable. Given a process, different models can be attached to it, some of them being equivalent, but, in any case, all them should be "coherent" [84].

For instance, for regulatory and tracking purposes, a CT/DT dynamic model would be required, but for production optimisation or management a simplified and aggregated model, or even a steady-state model, would be more appropriate. For alarm treatment, a discrete-event model will represent the evolution from one operating condition to the next, probably combined with some regulatory actions.

State-based models will be extensively used in this book, due to their relationship with first-principle models and their ease of computer implementation, as well as the availability of computer aided control system design packages for them.

Other than processes, signals should be also modelled. They can be considered as the output of processes (generators) with some particular properties. In particular, deterministic disturbances (such as steps, ramps or sinusoidal variations) can be modelled as the output of uncontrollable generators, and stochastic disturbances will be mainly modelled by their mean and variance properties.

2.2 Types of Models

As previously mentioned, based on the type of signals involved in the model we can find models of different natures: CT, DT, discrete-event, hybrid or stochastic models.

The usual CT and DT signals are functions of time, as defined in Appendix A.1. Multivariable signals are composed by stacking a set of individual signals in column vector form.

A *binary* or *logical* signal only takes two possible values, being synchronous if changes are only allowed at predefined time instants or asynchronous if the changes may happen at any moment.

Random variables and stochastic processes, being characterised by their statistical properties, will be considered in a later section and in Appendix E.

Although different kinds of models can be defined, unless otherwise stated, the hypothesis of Section 1.4, namely linearity, time-invariance and lumped parameters (finite-dimensional system), will be assumed to hold.

A non-linear system is a broader (and more common) representation of actual processes. The diversity of options and their specific and usually more difficult mathematical treatment puts the study of non-linear systems out of the scope of this book. Some simple cases will be outlined in Section 9.5. non-linearity, time-variation and spatial variation will be accommodated by control systems that are tolerant enough regarding modelling error.

Locality. The models usually only represent the relationship between increments of the variables around a given *operating point*. This is quite usual in modelling non-linear systems if we are interested in their approximate linearised behaviour around an equilibrium point. In general, non-linear models are better suited to modelling the global behaviour of a process, relating absolute values of the variables.

Variables. Based on the kind of variables involved, we can define: input/output or *external* models, and state variables or *internal* models. In the case of external models, we can also consider the so-called *black-box* models, where only the input and output variables are involved, or *white-box* models, where the internal structure of the process is somehow represented.

Methodology. The last distinction can be also related to the approach followed to obtain the model. If the basis of the process operation is known,

its dynamical behaviour can be expressed by balance and fundamental equations, leading to a *first-principle model*. If, on the other hand, this fundamental behaviour is unknown or the resulting equations would be too detailed and complicated, and it is possible to experiment with the process, its response to some excitation can be used to get an *experimental model* representing its input/output behaviour, without any reference to what happens internally.

2.3 First-principle Models: Components

Let us consider the following illustrative example.

Example 2.1 (First-principle modelling). Let us consider a continuous-flow stirred tank reactor (CSTR)[1], where a first-order exothermic reaction $A \rightarrow B$ happens, with a cooling jacket [89]. We have a rough model of the CSTR, knowing that due to the entrance of a flow rate input stream, F_o, with C_{ao} concentration of component A, at a temperature T_o, there is an internal level, h, temperature, T, and component A concentration, C_a, and there is an outlet flow rate, F, at temperature T and concentration C_a. This can be represented by the block shown in Figure 2.1, where the cooling jacket water flow, F_j, enters at temperature T_{jo}, leaving at temperature T_j, and the total jacket volume, V_j, is fixed.

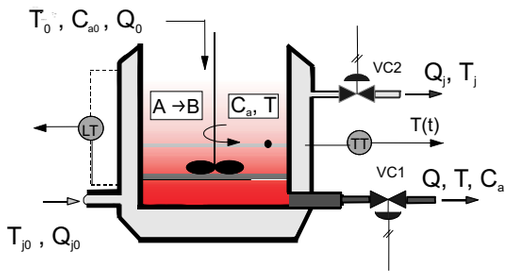

Figure 2.1. CSTR reactor

If all the processes inside the reactor are known, the following set of equations can be written:

1. Total mass balance:
$$\frac{dV}{dt} = F_o - F \tag{2.1}$$
2. Mass balance on component A:
$$\frac{d(VC_a)}{dt} = F_o C_{ao} - FC_a - \alpha V C_a e^{-\frac{E}{RT}} \tag{2.2}$$
where R is the perfect gas constant and α is the pre-exponential factor from the Arrhenius law, and E is an activation energy.

[1] The work about this process has been carried out in collaboration with M. Perez [101].

3. Energy balance in the reactor:

$$\frac{dT}{dt} = \frac{F_o T_o - FT}{V} + \frac{H\alpha}{\rho c_p} C_a e^{-\frac{E}{RT}} - \frac{UA}{\rho c_p V}(T - T_j) \tag{2.3}$$

   where H is the reaction heat, ρ and c_p are the density and heat capacity respectively of the inlet and outlet streams, U is the overall heat transfer coefficient through the jacket, A is the transfer area.
4. Energy balance in the jacket:

$$\frac{dT_j}{dt} = \frac{F_j}{V_j}(T_{jo} - T_j) + \frac{UA}{\rho_j c_{pj} V_j}(T - T_j) \tag{2.4}$$

   where ρ_j and c_{pj} are the density and heat capacity respectively of the cooling stream.

In this way, a set of non-linear differential equations represents the CSTR dynamics.

Simplifications. If we were only interested in the reactor components evolution, or the reaction were isothermal, the energy-balance equations would be missing:

$$\frac{dV}{dt} = F_o - F \qquad \frac{d(VC_a)}{dt} = (F_o C_{ao} - FC_a) - \alpha V C_a e^{-\frac{E}{RT}} \tag{2.5}$$

Temperature variations, if any, would amount to having time-variance in the parameters on the reduced model.

If interest were focused on long-term production, only the static relation among the variables would be relevant. Then, for given input constant values, if the operation is stable, an equilibrium point will be reached and a set of (algebraic) equations will model the steady-state behaviour. Steady-state equations are obtained by setting to zero all derivatives (as magnitudes are constant):

$$\begin{gathered} F = F_o; \qquad F_o(C_{ao} - Ca) - \alpha V C_a e^{-\frac{E}{RT}} = 0 \\ \frac{F_o}{V}(T_o - T) + \frac{H\alpha}{\rho c_p} C_a e^{-\frac{E}{RT}} - \frac{UA}{\rho c_p V}(T - T_j) = 0 \\ \frac{F_j}{V_j}(T_{jo} - T_j) + \frac{UA}{\rho_j c_{pj} V_j}(T - T_j) = 0 \end{gathered} \tag{2.6}$$

Similar to the reactor equations, basic equations describing the elements' phenomena can be used. We distinguish between static elements, leading to algebraic equations, or dynamic elements.

Static elements. Examples of static behaviour are, for instance:

- resistors: $V = IR$,
- wall heat transmission: $Q_{12} = k(T_1 - T_2)$,
- springs: $f = K(l - l_0)$,
- outlet flow: $F_{12} = k\sqrt{h}$,
- pipes: $F = k(P_1 - P_2)$,
- viscous friction: $f = kv$,
- balances, such as:
  - $\sum f_i = 0$, the total force applied to a body,

- $\sum F_i = 0$, the net flow in a pipe junction if there is no storage, or
- $\sum V_i = 0$ the total voltage drop in a loop, and so on.

The meaning of the different constants and parameters is clear for those introduced in the respective field. Some of these expressions are approximations of non-linear relationships (friction, spring, resistor) but some others, like the tank outlet flow, are explicitly non-linear. Under some circumstances, a linear approximation will be possible.

Dynamic elements. Dynamic elements are those where the involved variables are not related instantaneously but in different times or by time increments. For instance, accummulative or delay components, such as:

- capacitors ($\frac{dV}{dt} = \frac{1}{C}I$) and coils ($\frac{dI}{dt} = \frac{1}{L}V$),
- heat storage: $\frac{dT}{dt} = \frac{1}{MC_e}(Q_{in} - Q_{out})$,
- mass storage: $\frac{dV}{dt} = \frac{1}{M}(F_{in} - F_{out})$,
- motion equations: acceleration $\frac{dv}{dt} = \frac{1}{M}F_{tot}$, $\frac{d\omega}{dt} = \frac{1}{I}T_{res}$, or velocity $\frac{dx}{dt} = v$, $\frac{d\phi}{dt} = \omega$ for linear or angular motions,
- chemical reactions: $\frac{dx}{dt} = f(x_i, T)$, as previously used, where the product composition evolves with time,
- transportation belt: $m_{out}(t) = m_{in}(t - \tau)$,
- stack or queue systems: $n(k+1) = n(k) + \sum u_i(k)$.

We must notice that these relationships (and many others) are similar, leading to a component behaviour that is common to some of them. These analogies allow for a unified treatment of any dynamical system without much relevance of the supporting technology. It should be pointed out that the last dynamic equation is slightly different, as the time is discrete and the variables are assumed to be integers. We will see more of that later on.

Basic equations. The equation

$$\frac{dy(t)}{dt} = \alpha u(t) \tag{2.7}$$

represents the *storage* of u, α being a scaling constant to deal with the appropriate measurement units of y and u. In fact, the same relationship can be written as:

$$y(t) = \alpha \int_0^t u(\tau)\, d\tau + y(0)$$

The equivalent DT equations would be, respectively:

$$y(k+1) - y(k) = \alpha u(k) \qquad y(k) = \alpha \sum_{i=0}^{k-1} u_i + y(0) \tag{2.8}$$

In DT it is also very easy to represent some delays such as:

$$y(k) = u(k - d) \tag{2.9}$$

where d is the number of time delay intervals in this relationship.

These storage and delay equations are the basic dynamic equations and will be the kernel in defining the state space model in the next section.

2.4 Internal Representation: State Variables

A process is composed of a number of interconnected elements as described above. In order to completely define a process model, we must get the same number of independent equations as internal variables, allowing theoretically the computation of the internal variables given the external input, $u(t)$.

Some of the equations in the model will be dynamic and some of them will be algebraic. By substitution, some equations can be removed. A so-called *normalised representation* can be obtained by removing the algebraic equations and manipulating the rest of them to be expressed as storage equations (2.7) or (2.8).

State equation. If all the dynamic equations are first-order differential equations, they can be arranged in a normalised way. Denoting by *state variables* the storage variables previously defined, the process model could be summarised by the so-called state equation:

$$\frac{dx(t)}{dt} = f(x(t), u(t), t) \tag{2.10}$$

where $x \in \mathbb{R}^n$ is the state vector, $u \in \mathbb{R}^m$ is the input vector and f is an n-dimensional vector of non-linear functions. The argument t explicitly indicates the possibility of being time-varying functions. This system representation is denoted as being an n-order system.

Output equation. If the output variables, $y \in \mathbb{R}^p$, are not the state variables, they will be related to them, and possibly also the inputs, by the (algebraic) output equation:

$$y(t) = g(x(t), u(t), t) \tag{2.11}$$

where g is a p-dimensional vector of non-linear functions. The set of state and output equations is denoted as a *system realisation* or normalised *state space representation*.

Example 2.2. Looking at the equations in Example 2.1, let us rewrite the model as state equations. Only Equation (2.2) should be transformed into:

$$\frac{dC_a}{dt} = \frac{F_o(C_{ao} - C_a)}{V} - \alpha C_a e^{-\frac{E}{RT}}$$

the rest of the equations, (2.1), (2.3) and (2.4), constituting a fourth-order system with states (V, C_a, T, T_j).

For DT processes, the state and output equations will be, respectively:

$$x(k+1) = f(x(k), u(k), k) \tag{2.12}$$
$$y(k) = g(x(k), u(k), k) \tag{2.13}$$

where the argument k stands for the integer instant of time.

The delay element, (2.9), in CT models requires an infinity of state variables to be represented (all the past input values), leading to infinite-dimensional systems. Some approximations are usually done, (2.42).

However, in a DT process, if the time delay is a multiple of the period, it will admit a simplified treatment introducing as many delayed variables as delay intervals. For instance, for $d = 2$, $y(k) = u(k-2)$ is equivalent to:

$$x_1(k+1) = x_2(k); \quad x_2(k+1) = u(k); \qquad y(k) = x_1(k) \tag{2.14}$$

The Concept of State

The previously defined state variables have a number of interesting properties:

- **memory**. They summarise the past history of the process,
- **state as internal variables**. They are not directly connected to the input, but their derivatives (in DT, future values) as well as any other process variable can be expressed as a function of these and the inputs,
- **minimality**. There is a minimum number of state variables so that the process internal model cannot be further reduced by removing any internal variable; otherwise, the dynamic equations will be of an order higher than one,
- **non-uniqueness**. Any set of n-independent internal variables, not directly connected to the input, represents the system. By independent we mean each one representing a different storage process.
  If the set vector $x(t)$ is a state vector, any variable vector $\bar{x}(t)$ such that

$$x(t) = T\bar{x}(t) \tag{2.15}$$

  T being an n-square regular matrix, is also a state vector.
  Indeed, knowing $x(t)$, the new state can be computed as $\bar{x}(t) = T^{-1}x(t)$ and *viceversa* (2.15). The T-matrix represents a linear state transformation or a *similarity transformation*. The transformation of Equation 2.10 yields:

$$\frac{d\bar{x}}{dt} = T^{-1} f(T\bar{x}(t), u(t), t)$$

  so it is also a normalised representation.

2.5 Linear Models and Linearisation

As previously mentioned, most of our study is referred to linear, time-invariant (LTI) systems. For this kind of systems, the state equation (2.10) is simplified: the time variable is no more an argument (time-invariance) and the functions f express *linear* combinations of the state and input variables. Thus:

$$\begin{aligned} \tfrac{dx_1}{dt} &= a_{11}x_1 + a_{12}x_2 + \cdots + a_{1n}x_n + b_{11}u_1 + \cdots + b_{1m}u_m \\ &\vdots \\ \tfrac{dx_n}{dt} &= a_{n1}x_1 + a_{n2}x_2 + \cdots + a_{nn}x_n + b_{n1}u_1 + \cdots + b_{nm}u_m \end{aligned} \tag{2.16}$$

or, in matrix form

$$\dot{x}(t) = Ax(t) + Bu(t) \tag{2.17}$$

Similarly, for the output equation (2.11), it would be:

$$\begin{aligned} y_1 &= c_{11}x_1 + c_{12}x_2 + \cdots + c_{1n}x_n + d_{11}u_1 + \cdots + d_{1m}u_m \\ &\cdots \\ y_p &= c_{p1}x_1 + c_{p2}x_2 + \cdots + c_{pn}x_n + d_{p1}u_1 + \cdots + d_{pm}u_m \end{aligned} \tag{2.18}$$

or, in matrix form:

$$y(t) = Cx(t) + Du(t) \tag{2.19}$$

The state space model involves the four matrices (A, B, C, D) having the following dimensions and meaning:

- **A**, the $n \times n$ system matrix, represents the internal interconnection among state variables,
- **B**, the $n \times m$ input matrix, represents the input-to-state direct connection,
- **C**, the $p \times n$ output matrix, represents the state-to-output direct connection, and
- **D**, the $p \times m$ coupling matrix, represents the input-to-output direct connection or input/output coupling.

So, in total, the number of parameters is $n^2 + n \times m + p \times n + p \times m$. A system will be denoted by the 4-tuple shorthand notation $\Sigma := (A, B, C, D)$.

It is easy to show that under a similarity transformation such as (2.15), the new equivalent state space model will be $(\bar{A}, \bar{B}, \bar{C}, \bar{D})$, being:

$$\bar{A} = T^{-1}AT; \bar{B} = T^{-1}B; \bar{C} = TC; \bar{D} = D \tag{2.20}$$

and the new state and output equations:

$$\begin{aligned} \dot{\bar{x}} &= \bar{A}\bar{x} + \bar{B}u \\ y &= \bar{C}\bar{x} + Du \end{aligned}$$

Some transformations are of special interest for deriving or emphasising some properties of the representation, denoted as *canonical representations*, usually

reducing the number of significant parameters. For instance, if the eigenvalues of A are distinct, it is possible to find a matrix T such that $\bar{A} = \Lambda$ is diagonal (or, in a general case, a Jordan canonical form). Appendix B details some essential definitions and properties regarding matrices.

If we want to make a distinction between manipulated and disturbance inputs, a generalised state space model can be expressed by:

$$\dot{x}(t) = Ax(t) + Bu(t) + B_d d(t) \tag{2.21}$$

$$y(t) = Cx(t) + Du(t) + D_d d(t) \tag{2.22}$$

where $d \in \mathbb{R}^q$ is a q-dimensional vector of disturbances.

Equilibrium points. Under some conditions, the non-linear model (2.10 and 2.11) can be approximated by a linear one, in particular, around an equilibrium point. For a constant value of the input vector, u^0, an *equilibrium point* is defined as the state vector (set of variables), x^0, solution of:

$$0 = f(x, u^0) \tag{2.23}$$

It is worth noting that for the same input u^0 there may be one, none or many solutions of (2.23), and thus, there will be the corresponding number of equilibrium points.

Any other process variable will reach a steady-state value. For instance, the output vector will be:

$$y^0 = g(x^0, u^0) \tag{2.24}$$

Note that the equilibrium point can be stable or unstable. Further detail will be given later.

Linearisation. If the vector functions in the non-linear model, f and g, are continuous and derivable at an equilibrium point (x^0, u^0), a *linearised* model can be attached to the process by means of a truncated Taylor series expansion of these functions.

For small input amplitudes, the second and higher-order terms are negligible, and an approximate linearised process model in the form (2.17)–(2.19) can be derived:

$$\dot{\bar{x}} = A\bar{x} + B\bar{u}; \qquad \bar{y} = C\bar{x} + D\bar{u}$$

where $\bar{x} = x - x^0$ and $\bar{u} = u - u^0$ represent the variable increments around the equilibrium point, and the elements of the corresponding matrices are the Jacobian components of the non-linear functions:

$$a_{ij} = \frac{\partial f_i}{\partial x_j}(x^o, u^o) \quad i, j = 1, \ldots, n \tag{2.25}$$

$$b_{ij} = \frac{\partial f_i}{\partial u_j}(x^o, u^o) \quad i = 1, \ldots, n \quad j = 1, \ldots, m \tag{2.26}$$

$$c_{ij} = \frac{\partial g_i}{\partial x_j}(x^o, u^o) \quad i = 1, \ldots, p \quad j = 1, \ldots, n \tag{2.27}$$

$$d_{ij} = \frac{\partial g_i}{\partial u_j}(x^o, u^o) \quad i = 1, \ldots, p \quad j = 1, \ldots, m \tag{2.28}$$

Remark 2.3. If the process evolves to another equilibrium point, the linearised model parameters change. If the variation in the variables is large enough to invalidate the higher-order terms' truncation, the linear model is not valid anymore.

Example 2.4. Let us consider the CSTR of Example 2.1. Assume a constant volume inside the reactor ($F_o = F$), as well as the nominal conditions and parameters in Table 2.1.

Table 2.1. Reactor notation and steady-state variables.

Variable	*Description*	*Value*
C_{a0}	Input concentration (kmol A/m^3)	8
V	Reactor volume (m^3)	1.36
T_o	Inlet flow temperature (K)	294.7
V_j	Jacket volume (m^3)	0.085
α	Arrhenius exponential factor (h^{-1})	7.08×10^{10}
E	Activation energy (kJ/kmol)	69815
U	Heat transmission coeff. (kJ/hm^{2}K)	3065
A	Heat transmission surface (m^2)	23.22
T_{jo}	Cooling water input temperature (K)	294.7
R	Perfect gas constant (kJ/kmolK)	8.314
H	Reaction heat (kJ/kmol)	69815
cp	Thermal capacity (kJ/kg·K)	3.13
cp_j	Water thermal capacity (kJ/kg·K)	4.18
r	Reactive and product density (kg/m^3)	800
r_j	Water density (kg/m^3)	1000

Substituting the data in the table into (2.2)–(2.4), the particular model is:

$$\frac{d(C_a)}{dt} = \frac{F_o}{1.36}(8 - C_a) - 7.08 \times 10^{10} C_a e^{-8397.3/T}$$

$$\frac{dT}{dt} = \frac{F_o}{1.36}(294.7 - T) + 197.4 \times 10^{10} Ca.e^{-8397.3/T} - 20.8987(T - T_j)$$

$$\frac{dT_j}{dt} = \frac{F_j}{0.085}(294.7 - T_j) + 200.3076(T - T_j)$$

Thus, the state equation with $x = [C_a \quad T \quad T_j]'$, and $u = [F \quad F_j]']$ would be:

$$\dot{x}_1 = f_1(x,u) = 0.7353u_1(8 - x_1) - 7.08 \times 10^{10} x_1 e^{-8397.3/x_2}$$

$$\dot{x}_2 = f_2(x,u) = 0.7353u_1(294.7 - x_2) + 197.4 \times 10^{10} e^{-8397.3/x_2} - 20.90(x_2 - x_3)$$

$$\dot{x}_3 = f_3(x,u) = 11.76u_2(294.7 - x_3) + 200.31(x_2 - x_3)$$

The operating point is defined assigning, for instance, the inlet flow $F_o = 1.13$ m^3/h and cooling flow in the jacket $F_j = 1.41$ m^3/h. Thus, the operating point, denoted by x_s, can be computed by solving the above system equation, making the

left-hand derivative terms null as in (2.6). Due to the non-linearity, in this case there are three equilibrium points. One of the options is:

$$x_s = [C_{as} \quad T_s \quad T_{js}]^T = [4.031 \quad 333.6 \quad 331.4]^T$$

The Jacobian elements, $a_{ij} = \frac{\partial f_i}{\partial x_j}(x_s, u_s)$, of the linearised model can be easily computed. For instance, the first linearised equation is:

$$\frac{d\bar{x}_1}{dt} = -(0.7353u_{1s} + 7.08 \times 10^{10} e^{-8397.3/x_{2s}})\bar{x}_1$$
$$+(-7.08 \times 10^{10} x_{1s} e^{-8397.3/x_{2s}} \frac{8397.3}{x_{2s}^2})\bar{x}_2 + 0.7353(8 - x_{1s})\bar{u}_1$$

Proceeding with all the elements, the following linear model is obtained:

$$A = \begin{pmatrix} -1.705 & -0.2519 & 0 \\ 23.088 & -28.71 & 20.9 \\ 0 & -200.3 & -216.89 \end{pmatrix}; \qquad B = \begin{pmatrix} 2.918 & 0 \\ -28.6 & 0 \\ 0 & -415.29 \end{pmatrix}$$

Using the MATLAB® command[2] `ss`, assuming $C = I$, that is, all the state components are measurable, and putting $D = 0$, the state space representation is created:

```
sys=ss(A,B,C,D)
```

The individual matrices can be retrieved as `sys.a`, `sys.b`, *etc.*, if needed.

MATLAB®: Some commands implementing algorithms related to the contents of this section are: `gradient,diff,ss`.

Remark 2.5. A linearised model can also be obtained around a nominal trajectory. If the system input is defined in a time interval, $u^0(t), \forall t \in [t_i, t_f]$, and the nominal state trajectory is given by $x^0(t)$, a linearised model can be attached to this nominal trajectory by relating the variations of the trajectory with respect to variations of the nominal input. Nevertheless, the linearised model would be, generally, time-varying, as the linear model will change with the state[3].

A good example of this situation is a web treatment station, quite common in the plastic, paper, metallurgical and many other industrial sectors. The web is transferred from a coil to another one through the treatment area. The basic goal is to control the rotational speed of the leading shaft as well as the position of the tensional arm to keep a constant web velocity and tension. Under nominal operating conditions, the diameter and inertia of the leading coil are increasing with time, thus the nominal values of the parameters change. The nominal shaft speed is time-varying to keep the web velocity constant, and a linearised model can be obtained around this reference.

[2] All the commands in this book refer to MATLAB® version 5.3.

[3] In fact, the Jacobian coefficients in (2.25) would be like: $a_{ij}(t) = \frac{\partial f_i}{\partial x_j}(x^o(t), u^o(t))$.

Advantages and drawbacks. The state space representation has the following advantages:

- it may be obtained by the process behaviour description, as realised with the reactor example, with an immediate physical interpretation, if so derived,
- it presents separately the input effect and the measurement system,
- it is valid for linear and most non-linear systems,
- it can also represent the action of external disturbances (they are just treated as non manipulated inputs),
- it provides the full (internal) description of the process, allowing the linking of the model with the internal structure,
- there is a bunch of control system design approaches based on this representation,
- it is valid for SISO and MIMO systems, with many common properties. One key point in the MIMO case is that a complex system of interactions is described with the minimal set of independent variables.

But also, some drawbacks should be taken into account:

- the model may be over-parameterised with respect to other representations,
- it is not unique and it may be unnecessarily complicated (*i.e.*, over-dimensioned), if irrelevant physical phenomena are modelled or a non-minimal representation is used,
- the *frequency response* is not easily connected with the parameters of the model,
- delays cannot be modelled in the normalised representation, only approximately, by (2.43).

Anyway, it will be shown that it is possible to get this representation from any other, and to transform it to any other as well, at least in the linear case.

Discrete Models

If instead of differential equations the process is defined by difference equations, a discrete state space model can be used. In this case, the state and output equations are, respectively:

$$x(k+1) = Ax(k) + Bu(k) \tag{2.29}$$

$$y(k) = Cx(k) + Du(k) \tag{2.30}$$

Linearisation. Equilibrium points in non-linear discrete systems are calculated by replacing all instances of a particular variable, x (with different delays), with an equilibrium value, x^0. For time-invariant DT systems, the equilibrium points are given by the solutions to:

$$x^0 = f(x^0, u^0); \qquad y^0 = g(x^0, u^0) \tag{2.31}$$

Once the (non-linear) equilibrium points are calculated, linearisation is carried out obtaining the Jacobian, as in the CT case.

2.6 Input/Output Representations

In the previous section, we have built up a model by connecting the components according to the system structure, but it is also possible to look at the process as an m-input/p-output information processor. If only these variables are used, the system representation will be composed of a set of p-differential equations of order 1 or greater, as an extension of the differential equation model of a SISO system. An example of this is:

$$\begin{aligned} \ddot{y_1} + 2\dot{y_1} - y_2 &= 0.3\dot{u_1} + u_1 \\ \dot{y_1} + \ddot{y_2} &= 2.5u_2 \end{aligned} \tag{2.32}$$

This is a linear time-invariant CT two-input-two-output (TITO) system. Again, these differential equations could be non-linear and time-variant (even partial differential equations can be considered to represent distributed parameter systems or ∞-dimensional systems). For simplicity, we will assume the above equation system is formed by linear differential equations with constant coefficients.

2.6.1 Polynomial Representation

The Laplace transform (Appendix A) can be applied to the equation system, (2.32). If the terms involving the initial condition vanish or cancel, that is, if the initial variable values correspond to an equilibrium point of the system, an algebraic equation system on the Laplace variable, s, is obtained:

$$\begin{aligned} s^2y_1(s) + 2sy_1(s) - y_2 &= 0.3su_1(s) + u_1(s) \\ sy_1(s) + s^2y_2(s) &= 2.5u_2(s) \end{aligned}$$

If the terms related to the same variable are put together, the model can be expressed by the following matrix equation:

$$\begin{pmatrix} (s^2+2s) & -1 \\ s & s^2 \end{pmatrix} \begin{pmatrix} y_1(s) \\ y_2(s) \end{pmatrix} = \begin{pmatrix} (0.3s+1) & 0 \\ 0 & 2.5 \end{pmatrix} \begin{pmatrix} u_1(s) \\ u_2(s) \end{pmatrix} \tag{2.33}$$

In a generic MIMO system, the model would be:

$$D(s)y(s) = N(s)u(s) \tag{2.34}$$

where $D(s)$ and $N(s)$ are *polynomial matrices*, with dimensions $p \times p$ and $p \times m$ respectively, whose elements are polynomials on the s variable. This

new model, (2.34), is called the *differential operator* model, as a result of the use of the s-variable, or *polynomial representation*. Instead of the four real matrices (A, B, C, D) needed in the state space representation, only two matrices (D, N) represent the system, although their elements are polynomials (see Appendix B.6).

For a given system, the polynomial representation is not unique. In fact, if $P(s)$ is a square, $p \times p$, regular polynomial matrix, the process may be modelled by the new pair $(\bar{D}, \bar{N})$, being:

$$P(s)D(s)\,y(s) = P(s)N(s)\,u(s);\ \Rightarrow\ \bar{D}(s)\,y(s) = \bar{N}(s)\,u(s)$$

Example 2.6. A trivial example would be, in (2.33), to multiply by:

$$P(s) = \begin{pmatrix} 1 & 0 \\ -1 & (s+2) \end{pmatrix}$$

leading to an alternative representation of the same system with:

$$\bar{D}(s) = \begin{pmatrix} (s^2+2s) & -1 \\ 0 & s^2(s+2)+1 \end{pmatrix}; \bar{N}(s) = \begin{pmatrix} (0.3s+1) & 0 \\ -(0.3s+1) & 2.5(s+2) \end{pmatrix}$$

The polynomial representation of DT models is similar, using the z-variable.

Advantages and drawbacks. The polynomial representation also presents some properties. Among the advantages:

- it relates the input and output variables,
- it may be used to describe subsystems,
- it may be obtained by linearisation of a set of non-linear differential equations,
- there are some *ad hoc* approaches to designing control systems based on polynomial operators,
- the number of parameters is lower than in the state space model.

However:

- it is not unique,
- it is not easy to handle (requiring symbolic computation and matrix inversions),
- it does not allow for easy autonomous system analysis (free response),
- it is only valid for linear systems.

2.6.2 Transfer Matrix

From (2.34), assuming $D(s)$ is invertible (*i.e.*, the p-differential equations are independent), the output can be expressed as a function of the input:

$$y(s) = D^{-1}(s)N(s)u(s) = G(s)u(s) \tag{2.35}$$

The rational matrix, $G(s) = D^{-1}(s)N(s)$, dimension $p \times m$, is denoted as the process *transfer matrix*, and its elements are quotients of polynomials (see Section B.6). It is a function of the Laplace variable, s, and can be considered as an operator analogous to transfer functions in SISO systems. This is one of the most attractive properties of this representation.

Example 2.7. In the example, (2.32), it is easy to obtain:

$$\begin{pmatrix} y_1 \\ y_2 \end{pmatrix} = \begin{pmatrix} (s^2+2s) & -1 \\ s & s^2 \end{pmatrix}^{-1} \begin{pmatrix} (0.3s+1) & 0 \\ 0 & 2.5 \end{pmatrix} \begin{pmatrix} u_1 \\ u_2 \end{pmatrix}$$

and, for instance, using the *Mathematica*® language:

```
GC=Inverse[{{s^2+2s,-1},{s,s^2}}].{{0.3s+1,0},{0,2.5}}
```

it yields:

$$\begin{pmatrix} y_1 \\ y_2 \end{pmatrix} = \begin{pmatrix} \frac{s(0.3s+1)}{s^3+2s^2+1} & \frac{2.5}{s^4+2s^3+s} \\ \frac{-(0.3s+1)}{s^3+2s^2+1} & \frac{2.5s+5}{s^3+2s^2+1} \end{pmatrix} \begin{pmatrix} u_1 \\ u_2 \end{pmatrix} \tag{2.36}$$

In this case, the system is represented by one $p \times m$ matrix, $G(s)$, whose elements are rational functions. For instance:

$$G_{i,j}(s) = \left. \frac{y_i(s)}{u_j(s)} \right|_{u(s)_k=0, \forall k \neq j} \tag{2.37}$$

is the SISO transfer function between the input, u_j, and the output, y_i. That is, the Laplace transform of the i-output over that of the j-input, assuming that the rest of the inputs as well as the initial condition terms are null.

In general, $p \leq m$. That is, the number of output (controlled) variables is lower than the number of input (manipulated) variables. If $p = m$, $G(s)$ is a square matrix and some matrix operations will be allowed.

Again, the internal structure of the process is lost and this representation only provides information about the effect of the inputs in the outputs. If disturbances are considered, the model can be extended to become:

$$y(s) = G_p(s)u(s) + G_d(s)d(s) \tag{2.38}$$

where $d(s)$ is the Laplace transform of the vector of disturbances. Similarly, the element $G_{di,j}(s)$ is the transfer function between the j-disturbance and the i-output. Equation (2.38) is equivalent to (2.21) and (2.22).

Discrete case. The discrete transfer matrix representation of a general DT process, equivalent to (2.38), is expressed by:

$$y(z) = G_p(z)u(z) + G_d(z)d(z) \tag{2.39}$$

Advantages and drawbacks. The transfer matrix representation presents a number of advantages:

- it relates the input and output variables,
- it may be used as an operator to combine subsystems,
- it is unique (except for common factors in the rational elements, to be reduced),
- there are some *ad hoc* approaches to designing control systems based on this model structure,
- the number of parameters is minimised,
- its elements represent one-input to one-output connections, allowing experimental identification and partial model validation. Although it is out of the main scope of this book, its importance should be stressed. Some options for these tasks are outlined in Appendix A.4),
- in the same way, it shows up the *interactions* among different inputs and outputs,
- delays can be considered.

However:

- the elements are rational functions (including exponential terms if delays are considered),
- some global system properties, easily derived from the state matrix A, do not appear so clearly,
- it does not allow for the autonomous system analysis,
- it is only valid for linear systems.

Example 2.8. The internal representation of the CSTR, Example 2.4, is converted into a transfer matrix by the MATLAB® command `tf` (or `zpk` to get it in factorised form), as shown in the following example:

```
sys=ss(A,B,C,D); G=tf(sys); Gp=zpk(sys)
     TF from input 1 to output...          From input 2 to output...
      2.918 (s+57.43)(s+190.6)                    2186.3814
#1:  ---------------------------    #1:  ---------------------------
     (s+1.83) (s+54.36) (s+191.1)          (s+1.83)(s+54.36)(s+191.1)

      -28.6 (s+216.9)(s-0.6506)              -8679.561 (s+1.705)
#2:  ---------------------------    #2:  ---------------------------
     (s+1.83)(s+54.36)(s+191.1)          (s+1.83)(s+54.36)(s+191.1)

          5728.58 (s-0.6506)              -415.29(s+28.49)(s+1.922)
#3:  ---------------------------    #3:  ---------------------------
     (s+1.83)(s+54.36)(s+191.1)          (s+1.83)(s+54.36)(s+191.1)
```

MATLAB®: Some commands implementing algorithms related to the contents of this section are: `ss,tf,zpk,ss2tf,tf2ss`.

Systems with Time Delay

Transportation phenomena, among others, introduce pure time delays in the relation between variables. If:

$$y(t) = u(t - \tau) \tag{2.40}$$

$y(t)$ is a delayed version of signal $u(t)$. In fact, if $y(t)$ is the output and $u(t)$ is the input of a system, this is a *distributed* process and the general mathematical treatment of this kind of system is rather complicated.

On the other hand, the Laplace transform is easy to apply because:

$$\mathcal{L}[y(t)] = e^{-s\tau}u(s)$$

In a delayed system, if the delays appear at either the input or output signals, it is quite common to model it by a transfer matrix.

Example 2.9 (MIMO experimental identification). A mixing process (see Section 5.8.2 on page 162 for description and analysis of steady-state behaviour) has an intermediate buffer tank, a solvent valve and a pure-product valve. Outputs of interest are concentration (C, %) and flow (F, l/s). The experimental response of both outputs to a 10% valve opening on each valve is depicted in Figure 2.2.

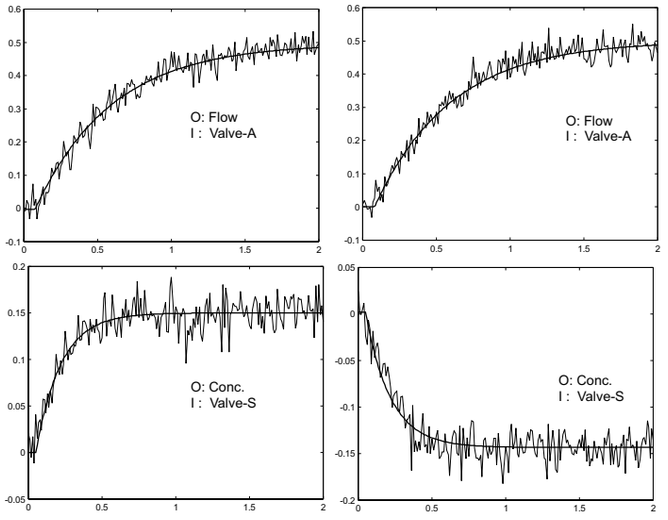

Figure 2.2. Mixing process: step response

Applying experimental identification (Section A.4), the following transfer functions can be approximately determined, targeting a widely-used first-order plus delay type of model:

$$\frac{F}{V_A} = \frac{5e^{-0.08s}}{0.52s+1}; \quad \frac{F}{V_S} = \frac{5e^{-0.08s}}{0.52s+1}; \quad \frac{C}{V_A} = \frac{1.5e^{-0.06s}}{0.17s+1}; \quad \frac{C}{V_S} = \frac{-1.45e^{-0.06s}}{0.17s+1}$$

also depicted in the above figure in thicker line. Hence, the experimental transfer function matrix is:

$$G(s) = \begin{pmatrix} \frac{5e^{-0.08s}}{0.52s+1} & \frac{5e^{-0.08s}}{0.52s+1} \\ \frac{1.5e^{-0.06s}}{0.17s+1} & \frac{-1.45e^{-0.06s}}{0.17s+1} \end{pmatrix} \tag{2.41}$$

It is difficult to know if the initial response, heavily masked by noise, includes a delay or it is caused by high-order dynamics. So, there are fundamental issues regarding quality of the model, signal-to-noise ratio, *etc.* that may determine the success of multivariable control strategies based on this crudely approximate model: Chapter 8 is devoted to them.

Unfortunately, the delays do not always appear in such a neat form as in the above example because there are internal and crossed delays and the elements of the transfer matrix can also be complicated[4].

A general approach to deal with time delays is to use the Padé approximation, converting the exponential into an approximated rational form. Based on the exponential approximation:

$$e^{-\tau s} = \frac{e^{-\frac{\tau}{2}s}}{e^{\frac{\tau}{2}s}} = \frac{1 - \frac{\tau}{2}s + \frac{\tau^2}{8}s^2 + \dots}{1 + \frac{\tau}{2}s + \frac{\tau^2}{8}s^2 + \dots} \tag{2.42}$$

the first-order approximation is:

$$e^{-\tau s} \approx \frac{1 - \frac{\tau}{2}s}{1 + \frac{\tau}{2}s} \tag{2.43}$$

Higher-order approximations are more accurate but introduce additional complexity into the model.

For DT representations, (2.9), if the time delay, τ, is a multiple of the time interval in the elements of the sequence, $\tau = dT$, it can be transformed into a shift, or, in $\mathcal{Z}$-transform, to multiply by the term z^{-d}, see Appendix A.3.

Transfer Matrix Poles and Zeros

The elements of the transfer matrix are rational functions of the Laplace variable, s. For a transfer function, $g(s) = \frac{n(s)}{d(s)}$, poles and zeros are the roots of $n(s)$ and $d(s)$ respectively (Appendix A.2.1).

The concept of pole and zero can also be defined for MIMO systems.

Given a transfer matrix, extract the common denominator

$$G(s) = \frac{N(s)}{d(s)}$$

[4] For instance, the closed-loop sensitivity function (4.5) of a SISO delayed system $G = e^{-s}(s+1)$ with proportional control $k = 4$ is $(1+kG)^{-1} = \frac{s+1}{(s+1)+4e^{-s}}$, so e^{-s} cannot be extracted as a factor.

where $N(s)$ is a polynomial matrix and $d(s)$ is a polynomial. Assume for simplicity that it has only single roots, $d(s) = \prod_{i=1}^{n}(s - p_i)$. These are the *poles* of the system. In order to determine the pole's multiplicity, decompose $G(s)$ into partial fractions by decomposing each element, $\frac{n_{ij}}{d(s)} = \sum_{l=1}^{n} \frac{\alpha_{ij,l}}{s-p_l}$. Thus, by suitably arranging α_{ij}, l in matrix form, we can express:

$$G(s) = \sum_{l=1}^{n} \frac{N_l}{s - p_l}; \qquad rank(N_l) = n_l$$

Then, the matrix rank n_l is the multiplicity of the pole p_l.

Assume a square system, $m = p$ for simplicity. In most cases, $G(s)$ is regular, *i.e.*, $rank(G(s)) = m$ for almost all s. The MIMO system transfer matrix *zeros* are the values z_i such that $rank(G(z_i)) < m$.

Let us consider the following trivial examples:

Example 2.10.

$$G(s) = \begin{pmatrix} \frac{-1}{s(s+1)} & \frac{1}{s} \\ \frac{-1}{s} & \frac{1}{s(s+1)} \end{pmatrix} = \begin{pmatrix} \frac{-1}{s} + \frac{1}{s+1} & \frac{1}{s} \\ \frac{-1}{s} & \frac{1}{s} + \frac{1}{s+1} \end{pmatrix} \tag{2.44}$$

$$G(s) = \frac{1}{s(s+1)} \begin{pmatrix} -1 & s+1 \\ -(s+1) & 1 \end{pmatrix} = \frac{1}{s} \begin{pmatrix} -1 & 1 \\ -1 & 1 \end{pmatrix} + \frac{1}{s+1} \begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix} \tag{2.45}$$

Thus, $s = -1$ is a double pole and $s = 0$ is a single one. There are three poles at $p = 0, -1, -1$. For $s \to 0$, the transfer matrix

$$\lim_{s \to 0} G(s) = \lim_{s \to 0} \begin{pmatrix} \frac{-1}{s} & \frac{1}{s} \\ \frac{-1}{s} & \frac{1}{s} \end{pmatrix}$$

loses the rank in one order. Thus, there is a zero at $s = 0$.

Example 2.11.

$$G(s) = \begin{pmatrix} \frac{s+1}{s} & \frac{-2}{s} \\ \frac{-1}{s} & \frac{1}{s} \end{pmatrix} = \begin{pmatrix} 1 & 0 \\ 0 & 0 \end{pmatrix} + \frac{\begin{pmatrix} 1 & 2 \\ -1 & 1 \end{pmatrix}}{s}$$

Thus, $s = 0$ is a double pole; $z = -1$ is a local zero for g_{11}, but it is not a multivariable zero, because $rank(G(-1)) = 2$. On the other hand, it is easy to check that there is a multivariable zero at $z = 1$.

In the next chapter, a more complete treatment of poles and zeros in MIMO systems, for any representation, is presented.

2.7 Systems and Subsystems: Interconnection

Between a detailed internal representation and a pure black-box one, a system may be represented by the interconnection of a number of subsystems. If the model of each element is known, a global model can be obtained.

For example, in a control system, at least, two subsystems will always be considered: the plant to be controlled and the controller. The structure of each one may be also decomposed into a number of subsystems.

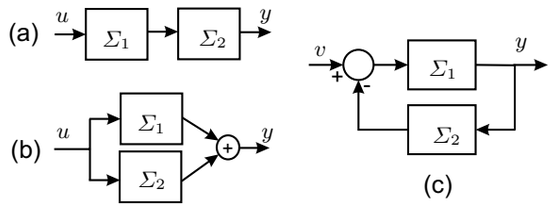

Figure 2.3. Series, parallel and feedback interconnection

2.7.1 Series, Parallel and Feedback Connection

Let us consider the basic subsystem interconnections. Let us define two subsystems, in state space and transfer matrix form:

$$\Sigma_i : \begin{array}{l} \dot{x}_i = A_i x_i + B_i u_i \\ y_i = C_i x_i + D_i u_i \end{array}; \qquad y_i(s) = G_i(s) u_i(s) \quad i = 1, 2 \tag{2.46}$$

Assuming appropriate dimensions in the input and output vectors, u_i, y_i, a global system may be arranged with joint state vector, $x = [x_1^T \; x_2^T]^T$, and its equation for each interconnection will be discussed.

Series connection. Also denoted by cascade connection. The input/output vectors to each subsystem in Figure 2.3(a) are:

$$y = y_2; \qquad u = u_1; \qquad u_2 = y_1$$

This results in the global state space system:

$$\dot{x} = \begin{pmatrix} A_1 & 0 \\ B_2 C_1 & A_2 \end{pmatrix} x + \begin{pmatrix} B_1 \\ B_2 D_1 \end{pmatrix} u \tag{2.47}$$

$$y = [D_2 C_1 \; C_2] x + D_2 D_1 u \tag{2.48}$$

and, in transfer matrix form:

$$y(s) = G_2(s) G_1(s) \, u(s) \tag{2.49}$$

Parallel connection. The interconnection in Figure 2.3(b) is represented by:

$$y = y_1 + y_2; \qquad u_1 = u_2 = u$$

This results in the global system:

$$\begin{array}{l} \dot{x} = \begin{pmatrix} A_1 & 0 \\ 0 & A_2 \end{pmatrix} + \begin{pmatrix} B_1 \\ B_2 \end{pmatrix} u \\ y = [C_1 \; C_2] x + (D_1 + D_2) u \end{array}; \qquad y(s) = (G_1(s) + G_2(s)) u(s) \tag{2.50}$$

Feedback connection. Figure 2.3(c). In this case, there is a loop and a summation point. Assuming an external input, v, null input/output coupling in the forward path ($D_1 = 0$) and negative feedback, the input/output vectors are:

$$y = y_1; \qquad u_1 = v - y_2; \qquad u_2 = y_1$$

This results in $y(s) = G_1(s)e(s) = G_1(s)(r - G_2(s)y)$, so the global system can be written as:

$$\dot{x} = \begin{pmatrix} A_1 - B_1 D_2 C_1 & -B_1 C_2 \\ B_2 C_1 & A_2 \end{pmatrix} + \begin{pmatrix} B_1 \\ 0 \end{pmatrix} v \tag{2.51}$$
$$y = (C_1 \ 0)x$$

$$y(s) = (I + G_1(s)G_2(s))^{-1} G_1(s) v(s) \tag{2.52}$$

In the last transfer matrix equation, the summation sign would be negative for positive feedback.

A system may be composed of a number of such interconnections, leading to a complex model, where also disturbances can be present. In these examples of interconnection, the ease of use of the transfer matrix as an operator is clearly illustrated.

Remark 2.12. In this and any other MIMO case, special care must be taken with block-diagram operations, as the matrix product is *not* commutative, contrary to the SISO case.

2.7.2 Generalised Interconnection

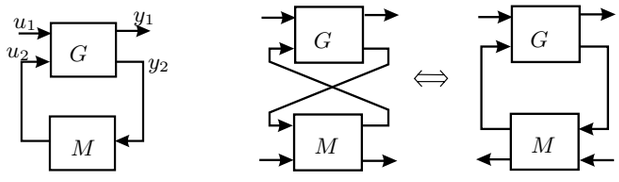

Figure 2.4. General interconnections

The interconnection between subsystems can be expressed in a general framework by just using two subsystems in a unique loop as in Figure 2.4 (left), where the subsystem in the feedback has a transfer matrix M, and the one in the upper subsystem is suitably partitioned into four blocks, G_{11}, G_{12}, G_{21} and G_{22}. In this framework, the equations are:

$$u_2 = My_2 = M(G_{21}u_1 + G_{22}u_2) \ \Rightarrow \ u_2 = (I - MG_{22})^{-1} MG_{21} u_1 \tag{2.53}$$
$$y_1 = G_{11}u_1 + G_{12}u_2 = \left(G_{11} + G_{12}(I - MG_{22})^{-1} MG_{21}\right) u_1 \tag{2.54}$$

Example 2.13. If $G = \begin{pmatrix} 0 & I \\ \Sigma_1 & 0 \end{pmatrix}$ the result is: $y_1 = M\Sigma_1 u_1$, *i.e.*, the series connection.

If $G = \begin{pmatrix} \Sigma_1 & I \\ I & 0 \end{pmatrix}$ the result is: $y_1 = (\Sigma_1 + M)u_1$, *i.e.*, the parallel connection.

If $G = \begin{pmatrix} I & -I \\ I & -I \end{pmatrix} \Sigma_1$ the result is the feedback connection. Indeed, in this case, (2.54) has as TF matrix $\Sigma_1 - \Sigma_1(I + M\Sigma_1)^{-1}M\Sigma_1$. As:

$$(I + M\Sigma_1)^{-1}M\Sigma_1 + (I + M\Sigma_1)^{-1} = (I + M\Sigma_1)^{-1}(M\Sigma_1 + I) = I \qquad (2.55)$$

the output transfer matrix can be expressed as $\Sigma_1 - \Sigma_1(I - (I + M\Sigma_1)^{-1}) = \Sigma_1(I + M\Sigma_1)^{-1}$. Applying the push-through rule (B.5), Equation (2.52) is obtained.

In this way, by including in G not only the system equations but also the interconnection structure, many control problems of engineering significance can be cast as the block-diagram in Figure 2.4 (left). G is then called the *generalised plant*, one of its subsystems being the actual plant. Indeed, this formulation allows us to deal with open-loop, closed-loop and other set-ups in a unified way, including performance specifications, and also allowed a significant theoretical breakthrough in the 1980s (see Sections 4.5.3 and 7.4). This block-diagram and Equation (2.54), where the block-diagram has been simplified by ellimination of u_2 and y_2, are denoted as a *lower linear fractional transformation* (LFT), and the transfer matrix in (2.54) is usually represented in shorthand form as $\mathcal{F}_L(G, M)$.

The diagrams at the right of the referred figure present an even more general form of interconnection denoted as Redheffer star product, where each block has two input and two output vectors. It is easy to verify the correspondence to the previous models if appropriate elements of their transfer matrices are chosen. For instance, for

$$G = \begin{pmatrix} 0 & G_2 \\ I & 0 \end{pmatrix}; \quad M = \begin{pmatrix} G_1 & 0 \\ 0 & 0 \end{pmatrix}$$

the serial connection, (2.47), is also obtained. And similarly, the parallel and feedback connections can be represented. The LFT diagram is a particular case of the star product, if only $M_{11} \neq 0$, such as the M above.

The MATLAB® command `lft` allows us to compute transfer matrices of systems described in an LFT or star-product structure. In particular, the expression `sys=lft(P,K,nu,ny)` connects the first "nu" outputs of K to the last "nu" inputs of P, and the last "ny" outputs of P to the first "ny" inputs of K. The resulting system model, `sys`, maps the remaining inputs to the remaining outputs. If `nu` and `ny` are omitted, the system with lower input and output vector dimensions is assumed to be M in the LFT block-diagram.

MATLAB®: Some commands implementing algorithms related to the contents of this section are: `lft,feedback,connect,starp,series,parallel`.

2.8 Discretised Models

Discretised (sampled) signals. Although the physical variables involved in a process are mainly CT, in order to process them on a computer, to simulate the process behaviour or to design a digital controller, approximate DT models are required. In general, attached to a CT signal, a DT sequence can be defined by just taking the sampled values at some given time instants. For a regular sampling period, T, the elements of the sequence are $y_k = y(kT), \quad k = 0, 1, 2, \ldots$. If the unit time delay is expressed by the delay operator, z^{-1}, these *sampled-data* (SD) signals are represented by using the $\mathcal{Z}$-transform (Appendix A).

These sampled-data signals, under a fast enough sampling pattern, keep most of the information carried by the CT signal. Shannon sampling theorem [95] states that the sampling frequency must be twice the frequency up to which significant content is present in the CT signal.

DT process models. Similarly, a DT model of the CT process (or a *sampled-data* model) can be derived. It will approximate the process behaviour relating the involved variables. However, some warnings should be needed in this case:

- the attached DT model is an approximation of the CT process behaviour. Depending on the approximation criteria, the following situations may appear:
  - the CT model and the SD model have a similar time-response to a given input signal (impulse, step, ramp, ...),
  - the CT model and the SD model have a similar frequency-response in a range of frequencies,
  - the CT model and the SD model have a similar mathematical appearance (by substituting the time derivative by increments ratio, for instance).
- the intersampling behaviour may be rather different. Caution should be taken about the information lost from the input signal,
- the DT model parameters will change if the sampling period is changed.

The most usual and common discretisation is based on the use of a digital computer, leading to regular sampling characterised by a period T, together with a constant holding of the input during the same period. To get an SD-equivalent model of the process, the simplest discretisation approaches are based on the *approximation of the derivative operator*. There are many options to implement this approximation:

- **Euler.** For each derivative term, the following approximation is implemented:
$$\dot{x} \approx \frac{\Delta x}{\Delta t} \approx \frac{x_{k+1} - x_k}{T} \tag{2.56}$$
provided T is small enough. This approach can be also applied to non-linear systems. The equation (2.10) is transformed into:

$$\frac{dx(t)}{dt} = f(x(t), u(t), t) \rightarrow x_{k+1} = x_k + Tf(x_k, u_k, k) \tag{2.57}$$

The DT model will relate the DT signals. For this purpose, dealing with linear systems, the sequences are represented by their $\mathcal{Z}$-transform. To get the Euler-discretisation of a CT transfer matrix, the Laplace variable, s, is replaced by:

$$s = \frac{1 - z^{-1}}{T}$$

- **forward operator.** In a similar way, a CT transfer matrix can be approximated by replacing the Laplace variable, s, by:

$$s = \frac{z - 1}{T}$$

- **bilinear transformation.** Among the possible variations of the previous approximations, if the Laplace variable is replaced by:

$$s = \frac{2}{T}\frac{1 - z^{-1}}{1 + z^{-1}} \tag{2.58}$$

the transformed DT model will be stable if and only if the source CT model is so, which is indeed of interest. This is a useful discretisation approach, although it is only valid for linear systems[5].

In the next chapter, for LTI systems and based on the solution of the state equation, exact (step response) SD models will be also derived. Non-regular sampling may be considered, leading to more complex DT representations (Section 9.4).

2.9 Equivalence of Representations

All the representations of the same process should be coherent. If they relate to the same variables, they should be equivalent. Thus, given the state space model or the polynomial operator, the transfer matrix should be easily obtained. In fact, by applying the Laplace transform to Equations (2.17) and (2.19), assuming that the initial condition terms are null, the following equivalence is obtained:

$$sx(s) = Ax(s) + Bu(s)$$
$$y(s) = Cx(s) + Du(s)$$

[5] For computer simulation of non-linear systems, an approximate equivalent is the mid-point discretisation formula:

$$x_{k+1} = x_k + Tf(x_k + T/2f(x_k, u_k), \frac{u_k + u_{k+1}}{2})$$

$$y(s) = [C(sI - A)^{-1}B + D]u(s) = G(s)u(s)$$

Thus:

$$G(s) = [C(sI - A)^{-1}B + D] \tag{2.59}$$

Based on this equivalence, as

$$(sI - A)^{-1} = \frac{adj(sI - A)}{\det(sI - A)}$$

the poles of $G(s)$ are the eigenvalues of A (solution of the characteristic equation $\det(sI - A) = 0$).

An interesting representation for $(sI - A)^{-1}$ can be derived if this matrix is expanded in series $I + As + As^2 + \ldots$:

$$G(s) = D + CBs^{-1} + CABs^{-2} + CA^2Bs^{-3} + \ldots = \sum_{i=0}^{\infty} H_i s^{-i} \tag{2.60}$$

Also, for DT systems:

$$G(z) = D + C(zI - A)^{-1}B = D + CBz^{-1} + CABz^{-2} + \ldots = \sum_{i=0}^{\infty} H_i z^{-i} \tag{2.61}$$

H_i are denoted as *Haenkel parameters* or coefficients. In this DT representation, H_i is the impulse response at time i, because $y(z) = G(z)$ if $u(z) = 1$.

The reverse transformation is not so easy. First, the state representation is not unique, thus there will be many internal representations for the same input/output model. Second, some of these representations may be overdimensioned and so useless, spurious, internal variables will be included. One option would be to use the concept of memory or accumulation attached to the state variables and try to define them in this way. Another alternative is to attach a state variable to each process pole.

To properly deal with this issue, some knowledge about the process structure and properties should be available. This is the subject of the next chapter (Section 3.7.5), and thus the reverse transformation is postponed, with only a very simple example being presented now.

Example 2.14. Let us consider the 2×2 transfer matrix of Example 2.10. Based on the partial fraction decomposition, let us assign the state variable, x_1, to the pole at the origin and the variables x_2 and x_3 to those related to the pole at -1. A possible state space representation would be:

$$\dot{x}_1 = -u_1 + u_2; \quad \dot{x}_2 = -x_2 + u_1; \quad \dot{x}_3 = -x_3 - u_2$$

$$y_1 = x_1 + x_2; \quad y_2 = x_1 + x_3$$

That is:

$$A = \begin{pmatrix} 0 & 0 & 0 \\ 0 & -1 & 0 \\ 0 & 0 & -1 \end{pmatrix}; \quad B = \begin{pmatrix} -1 & 1 \\ 1 & 0 \\ 0 & 1 \end{pmatrix}; \quad C = \begin{pmatrix} 1 & 1 & 0 \\ 1 & 0 & 1 \end{pmatrix}$$

Also, the transfer matrix may be extended to become:

$$G(s) = \begin{pmatrix} 0 & 1 \\ -1 & 2 \end{pmatrix} s^{-1} + \begin{pmatrix} -1 & 0 \\ 0 & -1 \end{pmatrix} s^{-2} + \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix} s^{-3} + \dots$$

From (2.35), the relationship between the transfer matrix and the polynomial representations is clear:

$$G(s) = D^{-1}(s)N(s)$$

Again, the reverse is not unique and many options are possible.

2.10 Disturbance Models

Disturbances are non-manipulated inputs to the process. They are generated elsewhere but affect the process behaviour and are signals coming from another processes.

2.10.1 Deterministic Signals

If the structure of these processes, denoted as *generators*, is known, the disturbances are said to be *deterministic*. For instance, polynomial disturbances can be considered as being generated by a chain of integrators. The initial condition of each integrator will determine one of the parameters of the polynomial signal. For instance:

- constant signals, $y = a_1$, are generated by:

$$\dot{x}_1 = 0; \quad x_1(0) = a_1; \quad y(t) = x_1(t)$$

- ramp signals, such as $y(t) = a_1 + a_2 t$, by:

$$\dot{x}_1 = x_2; \quad \dot{x}_2 = 0; \quad x_1(0) = a_1; \quad x_2(0) = a_2; \quad y(t) = x_1(t)$$

and, in general, *polynomial* signals, such as

$$y(t) = \sum_{i=1}^{i=n} \frac{a_i}{(i-1)!} t^{i-1}$$

are generated by:

$$\begin{aligned} \dot{x}_1 = x_2;\ \dot{x}_2 = x_3;\ \dots;\ \dot{x}_n = 0 \\ y = x_1 \end{aligned}$$

where $a_i = x_i(0)$, and $n!$ denotes the factorial of n.

Discrete disturbances. If the generators are built up with accumulators instead of integrators, DT signals can be generated. For instance, with:

$$x_1(k+1) = x_1(k) + x_2(k); \quad x_2(k+1) = x_2(k) + x_3(k); \quad \ldots$$
$$x_n(k+1) = x_n(k); \qquad y(k) = x_1(k)$$

the output is given by:

$$y(k) = x_1(0) + kx_2(0) + \frac{k(k-1)}{2}x_3(0) + \ldots = \sum_{i=1}^{n} \begin{pmatrix} k \\ i-1 \end{pmatrix} x_i(0)$$

Similar expressions can be obtained for exponential or sinusoidal signals.

The most common deterministic signals are summarised in Table 2.2. Disturbance randomness can be inserted if random impulse inputs, ψ, are assumed at time t.

Table 2.2. Generation of deterministic disturbances

	CT	DT
Steps	$d(t) \stackrel{\text{def}}{=} A$ $\dot{d} = \psi$	$d_k \stackrel{\text{def}}{=} A$ $d_{k+1} = d_k + \psi_k$
Ramps	$d \stackrel{\text{def}}{=} at + b$ $\dot{x} = \begin{pmatrix} 0 & 1 \\ 0 & 0 \end{pmatrix} x + \psi$ $d = [1 \quad 0]x$	$d_k \stackrel{\text{def}}{=} ak + b$ $x_{k+1} = \begin{pmatrix} 1 & 1 \\ 0 & 1 \end{pmatrix} x_k + \psi_k$ $d = [1 \quad 0]x$
Sinusoidal	$d \stackrel{\text{def}}{=} A\sin(\omega t + B)$ $\dot{x} = \begin{pmatrix} 0 & 1 \\ -\omega^2 & 0 \end{pmatrix} x + \psi$ $d = [1 \quad 0]x$	$d_k \stackrel{\text{def}}{=} A\sin(\omega Tk + B)$ $x_{k+1} = \begin{pmatrix} \cos\omega T & \sin\omega T \\ -\sin\omega T & \cos\omega T \end{pmatrix} x_k + \psi_k$ $d = [1 \quad 0]x$

Example 2.15. It is easy to show that, with the system defined by:

$$\dot{x} = a.x; \; x(0) = x_0; \quad y = x$$

the signal $y(t) = x_0 e^{at}$ is generated. Similarly, by:

$$\dot{x}_1 = x_2; \; \dot{x}_2 = w^2 x_1; \; x_1(0) = A\sin(\varphi); \; x_2(0) = A\cos(\varphi)$$
$$y = x_1$$

the signal $y(t) = A\sin(wt + \varphi)$ is generated.

2.10.2 Randomness in the Signals

It frequently happens that the model of the process generating the external signals is very complex or unknown. In this case, it is easier to characterise

the signals by their stochastic properties. They can also be considered as the output of stochastic generators or as the filtered response of an original stochastic signal. These disturbances may appear at the output (as measurement noise), at the input or at the state level. Let us briefly describe these types of variables:

- **measurement noise**. The output of the sensor system is affected by a number of random variables from which no information is available, such as device drifts or interference from other devices. The main goal would be to *filter*, reduce or cancel the noise, getting a *clean* measurement to feed the monitoring or control system,
- **stochastic inputs**. They affect the process behaviour as external variables and they are unavoidable. If we consider the positioning of an antenna, the force of the wind will present such a characteristic,
- **internal noise**. These are the most complicated disturbances and, in some cases, they can be modelled by assuming stochastic parameters in the process models. A way of dealing with this situation is by considering uncertainties in the model or in the variables. An introduction to the available tools is presented in Chapter 8.

In a general sense, the control goals under stochastic disturbances will be: to filter the measurement noise or output disturbances, to attenuate or cancel the input disturbances, probably by measuring them if it is possible, and to assure some performances under process noise, due to uncertainties in models or disturbances.

In order to characterise the stochastic signals, some basic statistic concepts and measurements about random variables should be remembered. Some of them are random variables, distribution and density functions, mean, variance, covariance, correlation, statistical independence and linear regression, among others. A short review of these concepts is included in Appendix E, where special attention is paid to the multivariable case.

2.10.3 Discrete Stochastic Processes

A signal, of which the value at any time instant is a random variable, is called a *stochastic process*. A discrete stochastic process is a *sequence* of random variables and can also be obtained as a result of sampling a CT stochastic process.

In the following, only discrete stochastic processes are considered. Their main properties are related to those of the random variables in the sequence, as well as their interaction. Let us review the most common (simplified) models of stochastic processes.

White noise. White noise (WN) is the simplest stochastic process, $\{\varepsilon_k\}$. The discrete random variables in the sequence are zero-mean ($E(\varepsilon_k) = 0$), and independent of each other. If their variance is $E(\varepsilon_k^2) = \sigma_k^2$, the *covariance* between samples j and k is:

$$\sigma_{jk} = \begin{cases} \sigma_k^2 & j = k \\ 0 & j \neq k \end{cases} \tag{2.62}$$

Due to the independence among variables, the knowledge of the current or past values in a sequence is useless for determining the future values. In this sense, WN cannot be compensated for or predicted.

If there is some kind of relationship among the variables, that is, if the random variables in the sequence are not independent, some additional information is available and the "noise" treatment may be easier. The rest of the stochastic processes are denoted as *coloured noise*. Some of them are easily modelled.

Random walk (drift). This can be considered as an accumulated white noise:

$$v_{k+1} = v_k + \varepsilon_k; \qquad \varepsilon_k \text{ is WN} \tag{2.63}$$

Thus, the difference between two consecutive variables is WN. Their covariance can be expressed by:

$$\sigma_{k(k+1)}^2 = E(v_k v_{k+1}) = E(v_k(v_k + \varepsilon_k)) = E(v_k^2) + E(v_k \varepsilon_k) = \sigma_k^2$$

Coloured noise (general model). A general model of coloured noise can be represented as the output of a dynamic (DT) process:

$$\begin{aligned} x_{k+1} &= Ax_k + \varepsilon_k \\ v_k &= Cx_k \end{aligned} \tag{2.64}$$

where ε_k is a WN. Thus, based on the past and current values of the $\{v(k)\}$ sequence, some information about the future values can be estimated.

Similarly, if the noise is disturbing a DT process which has, in addition, manipulated inputs, the equations equivalent to (2.21) and (2.22) would be:

$$\begin{aligned} x_{k+1} &= Ax_k + Bu_k + B_d v_k \\ y_k &= Cx_k + Du_k + D_d w_k \end{aligned} \tag{2.65}$$

where v_k and w_k are white noises with covariance matrices V and W respectively. Usually, $D = 0$ and $D_d = I$, w_k being the measurement noise and v_k the process noise.

Input/output models

Other than the internal representation above, a monovariable stochastic process may be represented by a difference equation or, alternatively, using the general concept of the $\mathcal{Z}$-transform of a sequence, it can be also expressed by an input/output model such as:

$$v(z) = G_n(z)\varepsilon(z) \qquad \varepsilon_k \text{ is WN} \tag{2.66}$$

It is worth mentioning some typical representations that are rather common in literature [30, 84]:

- Moving-Average (MA). The signal is obtained as a weighted sum of current and past inputs:
$$y(k) = \sum_{i=0}^{n} b_i \varepsilon(k-i)$$
- Auto-Regressive (AR). The weighted sum of the signal is a random noise:
$$y(k) + \sum_{i=1}^{n} a_i y(k-i) = \varepsilon(k)$$
- Auto-Regressive-Moving-Average (ARMA). This is a combination of the two above,
$$y(k) + \sum_{i=1}^{n} a_i y(k-i) = \sum_{i=0}^{n} b_i \varepsilon(k-i)$$
- External-Auto-Regressive-Moving-Average (ARMAX). In this case, there is also a manipulated input, $u(k)$, leading to:
$$y(k) + \sum_{i=1}^{n} a_i y(k-i) = \sum_{i=0}^{n} b_i u(k-i) + \sum_{i=0}^{n} c_i \varepsilon(k-i) \tag{2.67}$$

However, the state space representation will be pursued in this book for disturbance rejection control designs, as all the above models can be expressed in a form similar to (2.64) or (2.65).

2.11 Key Issues in Modelling

As has been emphasised in this chapter, the model of a process is a partial representation of its behaviour. In this sense, the model we are interested in is a model suitable for designing the control system. Thus, the selection of the model and the modelling approach depends on the final goal:

- the details of the model are determined by the control goals. Static, low-frequency range, autonomous, DT or hybrid models are examples of options,
- the variables and relationships selected to build the model will depend on the effects we are interested in in our study. Some variables can be deleted or treated as disturbances if they are not so relevant, or they can become crucial if their particular effect has to be modelled,
- the control design methodology will recommend, or even determine, the type of representation,
- the knowledge and/or availability of the process to get actual data will allow for theoretical or experimental modelling techniques,
- the accuracy and complexity of the model will depend on the control requirements.

How do we obtain a model? As a summary, two main approaches can be followed:

- **first-principle (internal) model.** Leading to (detailed) non-linear models, with physical insight into the variables and equations, and a number of parameters to be determined. Techniques of model reduction may need to be applied,
- **experimental modelling** (Appendix A.4). By comparing the behaviour of the plant to that of some predefined models and structures, the models' parameters can be estimated, leading to (simple) approximate linearised models, being usually input/output representations. This approach is suitable for modelling disturbances if historical records of them are available.

According to the model purpose and the analysis of its properties, a change in the representation frame, its complexity, its range of validity or the number of involved variables may be suggested. Thus, the modelling phase should be revisited in the control design process, to enhance the available model to better fulfill the control requirements, the final goal in our study.

2.12 Case Study: The Paper Machine Headbox

The purpose of the headbox is to deliver a uniform and stable jet velocity profile in both cross and machine directions to form the paper sheet, Figure 2.5. The stock flows into the headbox chamber controlled by a valve. The top

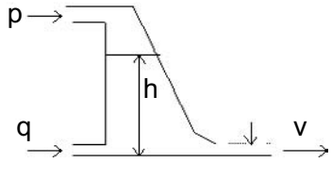

Figure 2.5. Headbox simple schema

of this chamber is filled with compressed air to dampen pressure pulsations. The airflow is also controlled by an input valve. The stock is homogenised in this chamber and it flows through the slice channel to the wire.

2.12.1 Simplified Models

First-principle. A very simple first-principle model can be derived by mass balances.

Stock balance in the headbox:

$$A(h)\frac{dh(t)}{dt} = q(t) - S(t)v(t)$$

where h is the stock level, in m, A is the headbox section, in m^2, q is the stock inflow, in kg/s, S is the slice lip section, in m^2, and v is the stock exit speed, in m/s.

The stock exit speed due to the total "head pressure" H, in m, is: $v(t) = \sqrt{2gH(t)}; \quad H(t) = h(t) + p(t)$

The air compression is assumed to be isothermal, thus $p(t)V_a(t) = const$, where p is the air pressure, in equivalent stock height (m) and V_a is the volume of air in m^3.

Air pressure variations are due to air net inlet flow and volume reduction, thus:

$$\frac{dp(t)}{dt} = \lambda(P_a(t) - p(t)) - \kappa\frac{dh}{dt}$$

where λ is the chamber dynamics inverse time constant, in s^{-1}, P_a is the source air pressure, in equivalent stock height, in m, and κ is a compression adimensional constant.

The slice lip exit area is manipulated by an external motor. In this simplified model, it is assumed constant or slowly-varying.

In this simple model, the head and pressure can be chosen as state variables, whereas the manipulated variables could be the stock inlet flow and the source pressure, leading to the state model:

$$\begin{aligned}\dot{h} &= -\frac{S}{A}\sqrt{2g(h+p)} + \frac{1}{A}q \\ \dot{p} &= \kappa\frac{S}{A}\sqrt{2g(h+p)} - \lambda p - \kappa\frac{1}{A}q + \lambda P_a\end{aligned}$$

Experimental. By applying steps at the inputs at an operating point, the elements of an approximate transfer matrix may be estimated. Additional transportation flow delays between the stock valve and the headbox chamber input may be realised. This can be done on the real plant or on a simulated model. This could result in a transfer matrix model obtained as in Example 2.9:

$$\begin{bmatrix} h \\ v \end{bmatrix} = \begin{bmatrix} \frac{e^{-s}}{1+s} & \frac{-1}{1+0.5s} \\ \frac{e^{-s}}{(1+s)(1+2s)} & \frac{0.2+s}{(1+0.5s)(1+2s)} \end{bmatrix} \begin{bmatrix} q \\ P_a \end{bmatrix}$$

Discrete model

From the first-principle non-linear state space representation, a simple DT model may be obtained by the Euler approximation of the two state variable derivatives.

Alternatively, from the experimental model above, the following discretisation approximations can be obtained:

1) Discrete transfer matrix. To get a discrete model, it could be easy to discretise each element of the transfer matrix, assuming a zero-order hold (ZOH) device in the input and a regular sampling period, T, at the output. If this period is assumed to be $T = 0.5$ s, the time delay of 1 s, will be equivalent to two delay units, z^{-1}. That is, $\mathcal{Z}[e^{-s}] = z^{-2}$.

Taking care of the delays, the MATLAB® command `Gd=c2d(G,.5,'zoh')` returns the DT transfer matrix:

$$G(z) = \begin{bmatrix} \dfrac{0.3935}{z^2(z-0.6065)} & \dfrac{-0.6321}{z-0.3679} \\ \dfrac{0.0489z+0.0381}{z^2(z-0.6065)(z-0.7788)} & \dfrac{0.2908z-0.2628}{(z-0.3679)(z-0.7788)} \end{bmatrix} \tag{2.68}$$

2) State representation based on physical variables. From this DT transfer matrix, grouping the common poles by rows or columns, the block-diagram in Figure 2.6 can be drawn.

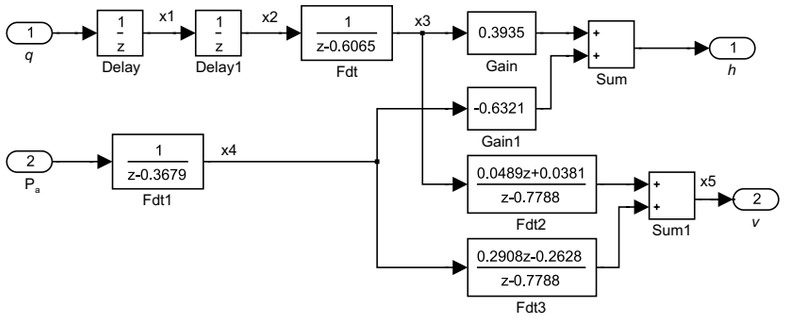

Figure 2.6. DT Block-diagram of the headbox (where $F \equiv q$)

Then, a state variable can be directly assigned to each first-order block (`Fdt2` and `Fdt3` share the same one), leading to the internal representation:

$$x_1(z) = \frac{1}{z}q(z) \quad \rightarrow \quad x_1(k+1) = q(k)$$

$$x_2(z) = \frac{1}{z}x_1(z) \quad \rightarrow \quad x_2(k+1) = x_1(k)$$

$$x_3(z) = \frac{1}{z-0.6065}x_2(z) \quad \rightarrow \quad x_3(k+1) = x_2(k) + 0.6065x_3(k)$$

$$x_4(z) = \frac{1}{z-0.3679}P_a(z) \quad \rightarrow \quad x_4(k+1) = 0.3679x_4(k) + P_a(k)$$

$$x_5(z) = \frac{0.0489z+0.0381}{z-0.7788}x_3(z) + \frac{0.2908z-0.2628}{z-0.7788}x_4(z)$$

$$(z-0.7788)\,x_5(z) = (0.0489z+0.0381)\,x_3(z) + (0.2908z-0.2628)\,x_4(z)$$

$$x_5(k+1) = 0.7788x_5(k) + 0.0489x_3(k+1) + 0.0381x_3(k)$$

$$+0.2908x_4(k+1) - 0.2628x_4(k)$$

and rearranging the last equation:

$$\begin{aligned} x_5(k+1) &= 0.7788x_5(k) + 0.0489\left[x_2(k) + 0.6065x_3(k)\right] + 0.0381x_3(k) \\ &\quad +0.2908\left[0.3679x_4(k) + P_a(k)\right] - 0.2628x_4(k) \\ x_5(k+1) &= 0.7788x_5(k) + 0.0489x_2(k) + 0.0678x_3(k) - 0.1559x_4(k) + 0.2908P_a(k) \end{aligned}$$

The output equation would be:

$$\begin{aligned} h(k) &= 0.3935x_3(k) - 0.6321x_4(k) \\ v(k) &= x_5(k) \end{aligned}$$

Altogether, the state representation is:

$$\begin{cases} x(k+1) = A_f x(k) + B_f u(k) \\ y(k) = C_f x(k) + D_f u(k) \end{cases}$$

$$A_f = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0.6065 & 0 & 0 \\ 0 & 0 & 0 & 0.3679 & 0 \\ 0 & 0.0489 & 0.0678 & -0.1559 & 0.7788 \end{bmatrix}; \quad B_f = \begin{bmatrix} 1 & 0 \\ 0 & 0 \\ 0 & 0 \\ 0 & 1 \\ 0 & 0.2908 \end{bmatrix}; \tag{2.69}$$

$$C_f = \begin{bmatrix} 0 & 0 & 0.3935 & -0.6321 & 0 \\ 0 & 0 & 0 & 0 & 1 \end{bmatrix}; \quad D_f = \begin{bmatrix} 0 & 0 \\ 0 & 0 \end{bmatrix}$$

2.12.2 Elaborated Models

An in-depth model has been developed at the Pulp and Paper Centre (University of British Columbia, Vancouver, Canada), reported in [127]. To express the dynamics of the upper chamber, the interconnected liquid and gas-flow systems are considered. A mass-balance equation for the stock and the stock flow out of the headbox can be obtained by applying:

$$\frac{dh}{dt} = q - C_s S\sqrt{2gh_j} - 2S_o\sqrt{2\frac{(p-p_0)}{\rho_w}}A \tag{2.70}$$

where h_j is the head at the slice lip, in m, C_s is the valve-sizing coefficient, S_o is the area of overflow valve opening, in m^2, p_0 is the atmospheric pressure, in m, and ρ_w is the density of stock, in $\mathrm{Kg/m}^{-3}$.

A mass-balance equation for the air using the pressure density relationship gives the following equation in terms of pressure:

$$\frac{dp}{dt} = \frac{kp_0}{\rho_0}\left(\frac{p}{p_0}\right)^{\frac{(k-1)}{k}}\left[\frac{q_i - q_e}{V_a} + \frac{\rho_0 A(\frac{p}{p_0})^{\frac{1}{k}}\frac{dh}{dt}}{V_a}\right] \tag{2.71}$$

where ρ_0 is the air density, in kg/m^{-3}, κ is the specific heat ratio, q_i and q_e are the air inflow into the upper chamber and outflow through the bleed valve, in kg/s.

To write the equation which governs the rate of change of the head at the slice lip, h_j, the headbox inlet head, h_i, and the friction head loss, h_f, are considered. The following equation is obtained:

$$\frac{dh_j}{dt} = \frac{\sqrt{2gh_j}}{L}\left[H - h_j - h_f\right] \tag{2.72}$$

where L is the constant of the slice channel, in meters, and h_f is the friction head loss.

The stock level, h, in the headbox, the airpad pressure, p, and the head at the slice lip, h_j, are chosen as the state variables.

The stock inflow, q, the gas flow, q_e, defining the net air inflow and the slice lip area, S, are treated as inputs.

The headbox dynamics can thus be represented by a third-order non-linear dynamical system. The model is characterised by a number of construction-dependent parameters (A, V_a, S_o, h_f and C_{ds}) as well as some fundamental physical parameters (k, p_0, ρ_0, ρ_w and g). The values of q, $q_i - q_e$ or S_j depend upon the operating point. Knowing the design details of the headbox and the operating point, all these variables could be calculated quite easily.

The three equations are highly coupled. Equation (2.70) shows that if the input stock flowrate increases, the liquid level increases. This in turn increases the airpad pressure and the total head. This increases the outgoing stock flow. There is thus a self-regulating effect in the system.

Equation (2.71) shows that the airpad pressure will return to its original value when the equilibrium point is reached. Any deviation in the level will thus be contributing to the total head.

Changing the air output area changes the air outflow, affecting both the level and the pressure inside the airpad. An increase in the slice opening will produce a decrease in the level, airpad pressure and the total head.

The most important control problem for a headbox is to maintain constant jet velocity and to have a good dynamic behaviour when changing the grades. We must consider as disturbances the external pressure of air and flow inlet, as well as the consistency of the stock or physical characteristics of the flow resistance.

In spite of the model complexity due to non-linearities, the model order is reduced to 3, which is even lower than the experimental one obtained in the last chapter.

Linearisation. The previous non-linear model is useful for simulating and studying the dynamic behaviour of the headbox. To design control strategies, it is always convenient to have linearised models which approximate the non-linear dynamics for small disturbances around a steady-state point. In [127], the following linearised model is obtained for some given parameters and operating conditions, using the MATLAB® command `linmod`.

$$A = \begin{bmatrix} 0 & -0.00083 & -0.03 \\ 0 & -0.08849 & -1.041610 \\ 9.1371 & 9.1371 & -8.883063 \end{bmatrix}; \quad C = \begin{bmatrix} 1\ 0\ 0 \\ 0\ 1\ 0 \\ 0\ 0\ 1 \end{bmatrix} \tag{2.73}$$

$$B = \begin{bmatrix} 0.47597 & 0 & -0.048587 \\ 14.03999 & -0.00162 & -1.433194 \\ 0 & 0 & -57.9134 \end{bmatrix}; \quad D = \begin{bmatrix} 0\ 0\ 0 \\ 0\ 0\ 0 \\ 0\ 0\ 0 \end{bmatrix}$$

The eigenvalues of the system matrix A are $(-0.0005, -10.3254, -0.9950)$. Thus, the system has a very slow mode with a time constant of about 512 s and two fast modes with time constants of less than a second.